Learning with Attributes for Object Recognition: Parametric and Non-parametric Views

by

Viktoriia Sharmanska

THESIS
Presented to the Faculty of the Graduate School of the
Institute of Science and Technology Austria, Klosterneuburg, Austria
in Partial Fulfillment of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Supervisor:
Prof. Christoph Lampert, IST Austria, Klosterneuburg, Austria

Committee Member:
Prof. Chris Wojtan, IST Austria, Klosterneuburg, Austria

Committee Member:
Prof. Horst Bischof, TU Graz, Graz, Austria

Program Chair:
Prof. Herbert Edelsbrunner, IST Austria, Klosterneuburg, Austria

April, 2015

Declaration

I hereby declare that this dissertation is my own work, and it does not contain other people's work without this being stated; and this thesis does not contain my previous work without this being stated, and that the bibliography contains all the literature that I used in writing the dissertation, and that all references refer to this bibliography. I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Viktoriia Sharmanska

Brighton, UK

April, 2015

To Novi Quadrianto,
the most wonderful person I have ever met.

Acknowledgments

Completing the PhD would have been unthinkable without the help and support of many wonderful people, who I was delighted to meet during my studies. I would like to thank my supervisor, Christoph Lampert, for guidance throughout my studies and for patience in transforming me into a scientist, and my thesis committee, Chris Wojtan and Horst Bischof, for their help and advice. I would like to thank Elisabeth Hacker who perfectly assisted all my administrative needs and was always nice and friendly to me, and the campus team for making the IST Austria campus my second home. I was honored to collaborate with brilliant researchers and to learn from their experience. Undoubtedly, I learned most of all from Novi Quadrianto: brainstorming our projects and getting exciting results was the most enjoyable part of my work – thank you! I am also grateful to David Knowles, Zoubin Ghahramani, Daniel Hernández-Lobato, Kristian Kersting and Anastasia Pentina for the fantastic projects we worked on together, and to Kristen Grauman and Adriana Kovashka for the exceptional experience working with user studies. I would like to thank my colleagues at IST Austria and my office mates who shared their happy moods, scientific breakthroughs and thought-provoking conversations with me: Chao, Filip, Rustem, Asya, Sameh, Alex, Vlad, Mayu, Neel, Csaba, Thomas, Vladimir, Cristina, Alex Z., Avro, Amelie and Emilie, Andreas H. and Andreas E., Chris, Lena, Michael, Ali and Ipek, Vera, Igor, Katia. Special thanks to Morten for the countless games of table soccer we played together and the tournaments we teamed up for: we will definitely win next time:) A very warm hug to Asya for always being so inspiring and supportive to me, and for helping me to increase the proportion of female computer scientists in our group. Arigatou Ryoichi and Morten for the Japanese tea time we shared together, for inspiring meetings and enjoyable gatherings at Heuriger. I would like to express words of gratitude to Olga, little Maya and Chris for welcoming me and making me feel at home when living abroad. Thank you, Filip, Magdalena and Veronika, for coming to support me during my defense. Grazie Tatiana and

Michele for sharing the happiest days of our lives in Vienna and in Rome. I would like to thank Claudia for her support and understanding during the most demanding part of my PhD way, and for our dinner times that we shared together. Danke Marta for our Salzburg trip and Barcelona exploration together. I would like to thank my Texas friends at UT Austin who made my stay in the USA a wonderful experience: Adriana, Dinesh, Aron, Chao-Yeh, Suyog, Bo and my host Kristen Grauman. Thank you, my dear friend Aleksandra, for introducing me to the gym and for taking care of me during my stay in Austin! I would like to acknowledge the kind labmates with whom I shared my Cambridge experience: Sae, David, Tomo, Miguel, Sebastien, Christian and my host Zoubin Ghahramani. I am very grateful to Joe for helping me to proofread my thesis. Sonnchen, дякую за наші домашні зустрічі у Віденських музеях! Зірочко, дякую за те, що ти освітлювала і зігрівала мій довгий шлях своєю посмішкою. Ірчик, дякую за те, що ти була завжди поруч, відколи я себе пам'ятаю. Дякую дорогій Наталії Олександрівні за міцну і щиру дружбу. Дякую Мамі, Татові, Ігорчику, Баб Валічці, Дідові і моїм дорогим Львів'янам за постійну підтримку, віру і теплу любов! Terima kasih Pa, Kak, Pipi, Ko Aseng atas dukungan dan doa Anda. My last and most important words of love and gratitude are reserved for Novi: it is thanks to you I was able to find myself and to come to the finish line of the PhD. Thank You!

Kindly,
Viktoriia

Abstract

The human ability to recognize objects in complex scenes has driven research in the computer vision field over couple of decades. This thesis focuses on the *object recognition* task in images. That is, given the image, we want the computer system to be able to predict the class of the object that appears in the image. A recent successful attempt to bridge semantic understanding of the image perceived by humans and by computers uses attribute-based models. *Attributes* are semantic properties of the objects shared across different categories, which humans and computers can decide on. To explore the attribute-based models we take a statistical *machine learning* approach, and address two key learning challenges in view of object recognition task: learning augmented attributes as mid-level discriminative feature representation, and learning with attributes as privileged information. Our main contributions are *parametric and non-parametric* models and algorithms to solve these frameworks. In the parametric approach, we explore an autoencoder model combined with the large margin nearest neighbor principle for mid-level feature learning, and linear support vector machines for learning with privileged information. In the non-parametric approach, we propose a supervised Indian Buffet Process for automatic augmentation of semantic attributes, and explore the Gaussian Processes classification framework for learning with privileged information. A thorough experimental analysis shows the effectiveness of the proposed models in both parametric and non-parametric views.

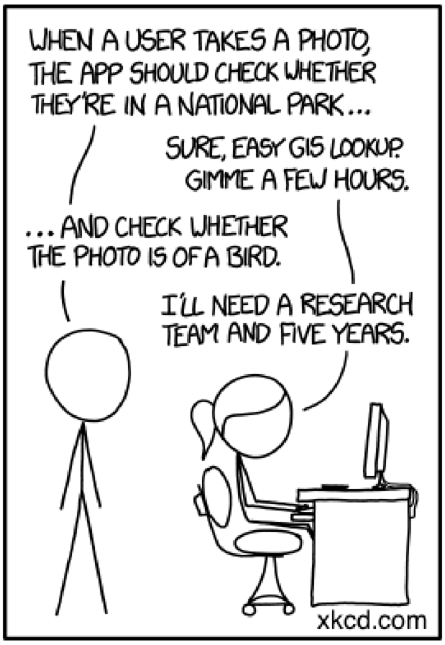
Keywords Attributes, Object Classification, Mid-level Feature Representation, Learning using Privileged Information, Autoencoder, Large Margin Nearest Neighbor, Indian Buffet Process, SVM, Gaussian Processes.

Contents

Chapter 1

Introduction

The starting point of computer vision research dates back to the 1960s with its utopian ideas of developing automated vision systems that can see and understand a complex environment the way humans do. However, it is only recently that progress in the field made it possible to build useful computer vision systems that many people value. The various applications – automatic face detection that is a built-in function in all camera devices, counting pedestrians on the city streets, anomaly detection in video surveillance records, human action recognition and activity forecasting, self-driving cars, etc. – made computer vision a closer reality. Tremendous progress in the field is driven by several factors: the availability of a large amount of data such as Internet images and users' collection of photographs, constantly growing computational resources and statistical learning models that introduce data-driven approach to computer vision problems. Despite the fact that computer vision is at the frontiers of research in our time, there are many difficult problems which remain unsolved, and object recognition is among the oldest. What appears a trivial task for humans, such as recognizing a bird in an image, remains challenging for a computer system to perform. This huge gap between the performance of the human visual system and computer vision is well captured in the following comic illustration:


WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.
xkcd.com

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Recent progress in computer vision has been made when introducing human semantics into the object recognition task via attribute-based modeling approach. Attributes are the semantic properties of objects like *natural*, *has stripes* and *round-shaped*. A computer program can learn attributes from image examples and then communicate about objects and their properties in the same way as humans. Attribute-based models open a whole new perspective on computer vision research with their natural applications to object recognition, object description and object comparison, among others. Attribute-based classification follows the idea of recognizing the objects based on their attribute properties and not visual appearance alone, as it is done in the traditional object recognition systems. Hence, it becomes possible for a computer vision system to recognize not only the object class that it has seen before, but also those classes it has not seen, an impossible task before.

1.1 Thesis Contribution

This thesis aims to address research challenges for attribute-based models in computer vision in the context of object classification task. Specifically, we introduce the models and algorithms for solving the following learning frameworks:

- Learning mid-level discriminative feature representation of images.

In recent years there has been much interest in this topic in the computer vision community. Mid-level feature representations go beyond low level vision feature descriptors and capture high-level semantic information about the object categories. We introduce the task of learning augmented attribute representation as a method for learning mid-level discriminative feature representation of images. It combines semantic and discriminative attributes into mid-level image representation, which can be used directly to perform object classification task. The main contributions of this part are parametric (Chapter 3) and non-parametric learning models (Chapter 4) for learning augmented attribute representations with application to object recognition task in images.

- Learning with privileged information for object recognition in images. Machine learning techniques have become prevalent for learning from image data. In the standard classification learning setting, we are given image-label data pairs, and the goal of learning is to infer a latent function that maps images to their labels. This function also called classifier will then be used to predict a label for any new test image. We can use low level visual features or attribute-based image representation for this task, as long as it is available at training and prediction time for evaluating the classifier. Learning with privileged information goes beyond this setting, and allows us to utilize more information during training than what will be available at test time. In this framework, we want to learn the image-label classifier, but in addition to image data we can use informative object representations such as attributes during training time. Additional information is called privileged as it is only given during training and is not available at test time. The main question is how to learn the image-label classifier that benefits from privileged information and shows better performance. The main contributions of this part are parametric (Chapter 5) and non-parametric learning models (Chapter 6) with privileged information for solving the object recognition task in images.

1.2 Thesis Structure

The rest of this thesis is organized into six chapters. The main content of each chapter is summarized below.

Chapter 2 Background. In this chapter, we cover general background knowledge needed for our later modeling and algorithm development. We start with defining what is an attribute of the object and how to build an attribute learning model. Specifically, we overview main principles for building the attribute vocabulary explored in the literature, and introduce two key learning models that are frequently used in practice. The first model uses the maximum margin principle and the second model uses the large margin nearest neighbor principle for learning a classifier model. These two principles play a key role in the subsequent approaches for the parametric view on learning augmented attribute representations in Chapter 3 and on learning with privileged information in Chapter 5. Then, we consider motivation for attribute-based approach using applications of attribute-based models that have been successfully applied to solve computer vision tasks, such as zero shot learning, describing usual and unusual appearance of objects in images, attribute-based image search, relative comparison of objects based on their attribute strength, etc. Next, we examine the background knowledge on visual feature representations extracted from image content. This includes a short overview of the SIFT descriptor as well as the bag-of-words histogram, Fisher vectors and deep convolutional neural network representations. We conclude the background chapter with general principles of parametric versus non-parametric learning systems. The latter plays a key role in the non-parametric view on learning augmented attribute representations in Chapter 4 and on learning with privileged information in Chapter 6.

Part I: Mid-level Discriminative Feature Learning with Attributes: Parametric and Non-parametric Views. The augmented attribute representation that combines semantic attributes and discriminative features learned from the data forms the so-called mid-level feature representation of images. We dedicate two chapters for learning mid-level feature representations, which cover parametric and non-parametric views.

Chapter 3 Augmented Attributes: Parametric View. In this chapter, we first consider the weak points of using semantic attributes alone as image representation for object recognition task. Despite the fact that semantic attribute representation is learned using large corpora of labeled images, it does not guarantee a perfect prediction in practice. The reliability of attribute predictors can be further improved with discriminative non-semantic attributes when pursuing directly the goal of attribute-based object classification. We propose a parametric model to augment the semantic attributes with a discriminative attributes part, such that the inferred augmented attribute representation can be directly used for nearest neighbor classification.

Chapter 4 Augmented Attributes: Non-parametric View. In this chapter, we solve the problem of learning discriminative attribute-based representation from the probabilistic modeling perspective. We take advantage of the non-parametric approach, and allow the augmented representation to grow with the data without pre-specifying the dimensionality of the attribute space as required in the parametric model. Moreover, we formalize the property of learning the discriminative representation in the form of a preference model that follows the folks wisdom principle: one prefers to stay close to its friends (samples from the same class) and far away from its enemies (samples from a different class).

Part II: Attributes in Learning with Privileged Information: Parametric and Non-parametric Views. In the second part of this thesis we envision the attributes as a source of rich and informative feature representation that can be used in addition to image data to train a better object recognition model than one would train from image data alone. We dedicate two chapters for solving this problem, which cover parametric and non-parametric views.

Chapter 5 Learning with Privileged Information: Parametric View. In this chapter, we study the case where we are given additional information about the training data such as attribute representation, which will not be available at test time. This situation with an asymmetric distribution of information between training and test time is called learning using privileged information (LUPI). We introduce the maximum margin approach to make use of this privileged source of information, when incorporating the margin information to be shared between image data and privileged features. Alternatively, this information can be viewed as an estimation of easy and hard data samples, which supervises the learning process of the classifier. Apart from attributes we consider textual description, object localization and human annotation of easy-hard scores as possible sources of privileged information for image data.

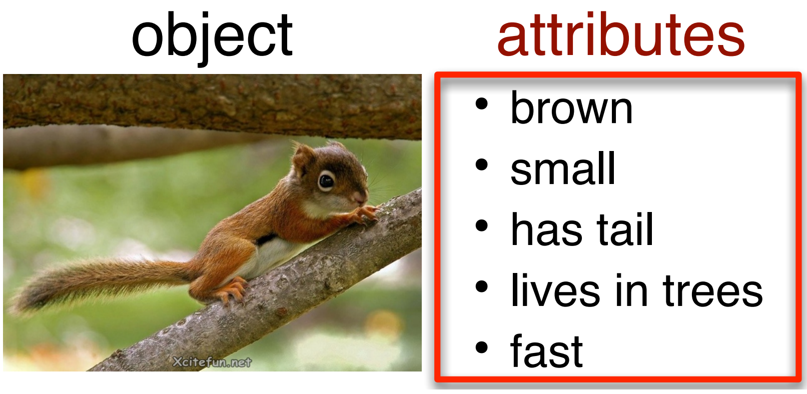
Chapter 6 Learning with Privileged Information: Non-parametric View. In this chapter, we solve the problem of learning with privileged information from the probabilistic non-parametric modeling perspective. In the probabilistic interpretation, the privileged information is used as a measure of certainty about the training samples and influences the likelihood model. A training example that is easy to classify based on its privileged representation causes a higher value of the likelihood term, which means we trust the training example and try to fit it well. Examples that are hard to classify result in decrease of the likelihood term, so we consider the training example less reliable and do not require the learning model to fit its label perfectly.

Chapter 7 Conclusions and Future Directions. In this chapter, we summarize the main results of this thesis and discuss attractive future directions for further exploration and exploitation of these results. Also, we address new perspectives on the attribute-based modeling in the world of 3D objects, where the attributes are learned using data-driven approach from the 3D data.

Chapter 2

Background

Imagine you are driving a car in a wildlife park and observing the animals walking around. If some look unfamiliar, you would take a snapshot and ask a computer program to identify these animals. You may also ask whether any of them can climb a tree, and which have claws. Or imagine doing shopping, and after looking the whole day for a particular pair of shoes to match your evening gown you end up searching online. Ideally, you would like to query a search engine with something like shiny shoes that match the color and texture of your gown. Or if looking for a new accommodation you would like to see the list of properties with a garden or ocean view rather than urban scenes, i.e. to have an application with an automatic scene analysis. All these applications and tasks are active research topics in the computer vision field in the area of attribute-based visual recognition. Attributes enable computers to understand and to communicate about objects and their properties in the same way as humans. In this chapter, we detail attributes and their role in a variety of computer vision tasks, and discuss state-of-the-art image features and data-driven methods as the background material for the thesis.



2.1 Attributes

In their simplest form, attributes are the semantic properties of objects, for example, *is brown*, *has tail*, which are shared across different objects categories, i.e. squirrel *has tail* as well as zebra and tiger, shoes are *brown* as well as dining table and chairs, and that a human and computer can decide on, i.e. a human can name objects and their attributes and a computer program can learn which objects have which attributes from the training examples. For the first time Ferrari & Zisserman (2008); Kumar et al. (2008) addressed the question whether it is possible to develop a computer program that can learn the attributes of different objects based on image examples.

From 2009, attribute-based models have been successfully used to solve many computer vision tasks. Farhadi et al. (2009) showed that computer programs can describe the objects in images by their attributes, including usual and unusual details in their appearance. Lampert et al. (2009) proposed to recognize objects in the images by their attribute descriptions, which enables recognition of previously unseen classes (zero-shot learning). A further development of the attribute-based models enabled relative comparisons of different objects by their attributes strength (Parikh & Grauman, 2011b), for example, distinguishing the celebrities *smiling more* than others based on their images. Kumar et al. (2008); Kovashka et al. (2012) proposed to enhance the human-machine communication via attributes; for example, if we search for images of *black* shoes that are *more formal* than sandals, we use binary and relative attributes to describe what properties we are looking for. Branson et al. (2010) explored fine-grained recognition of different bird species using attributes in the interactive 20 questions game. Patterson & Hays (2012) showed how to enable an automatic high-level description of scenes via scene attributes. The list of attribute-based models has grown rapidly over the last five years, and in this chapter we will detail a few of them.

First, we provide an overview of different ways to define the attributes, and then discuss how to learn the attributes using the data-driven approach. Finally, we describe various computer vision applications that become possible when utilizing the attribute-based models.

2.1.1 Defining the attributes

Attributes that are defined by humans inherit the property of interpretability. Therefore, they are also often called *semantic* attributes (Ferrari & Zisserman, 2008). One of the central questions in the attribute-based approach is how to define an attribute vocabulary. Several ways have been proposed in the literature to address this problem: (i) the vocabulary can be defined by experts, for example, field guides with animal species descriptions, (ii) the vocabulary can be defined by trusted amateurs, for example, colleagues and friends who provide common knowledge and unbiased expertise and (iii) the vocabulary can be automatically extracted from the web, as proposed by Berg et al. (2010); Rohrbach et al. (2010); Parikh & Grauman (2011a).

Depending on the application area and the available annotation, the attributes can be defined (i) per object class (per-class attributes) and (ii) per object image (per-image attributes). Per-class attributes occur when all objects in the class have a common attribute description, i.e. all images of the *zebra* class inherit its attributes: *striped*, *black*, *white*, *not a carnivore*. Per-image attributes capture only the attributes that are visible in the given image. For example, if a *squirrel* is hiding inside the tree and its tail is invisible in the image, then the attribute *has tail* is absent in the per-image attribute description.

2.1.2 Learning the Attributes: Data Driven Approach

The main principle of the statistical learning approach for attributes is to find the decision boundary between objects that share the attribute property (positive samples) and objects that do not (negative samples). Assume we want to learn the attribute a, for example, *has spots*, and we are given a set of N training images $\{x_1, \dots, x_N\} \subset \mathcal{X}$ together with their class label annotation, $\{y_1, \dots, y_N\} \subset \{+1, -1\}$, where $+1$ means the correspondent image has the object with attribute a, and -1 otherwise. In the context of computer vision, we consider the examples from $\mathcal{X}$ as image representations, and their features being extracted from the image content. For now, we do not make any specific assumption about the feature representation and keep a general notation of the d-dimensional feature vectors extracted from the visual information, $\mathcal{X} = \mathbb{R}^d$. The binary classification task is to learn a prediction function $f : \mathcal{X} \to \mathcal{Y}$, such that for a new given image $\bar{x}$ we are able to predict whether the object in this image has the attribute a or not using $f(\bar{x})$.

Currently, the two most popular learning principles are: learning the represen-

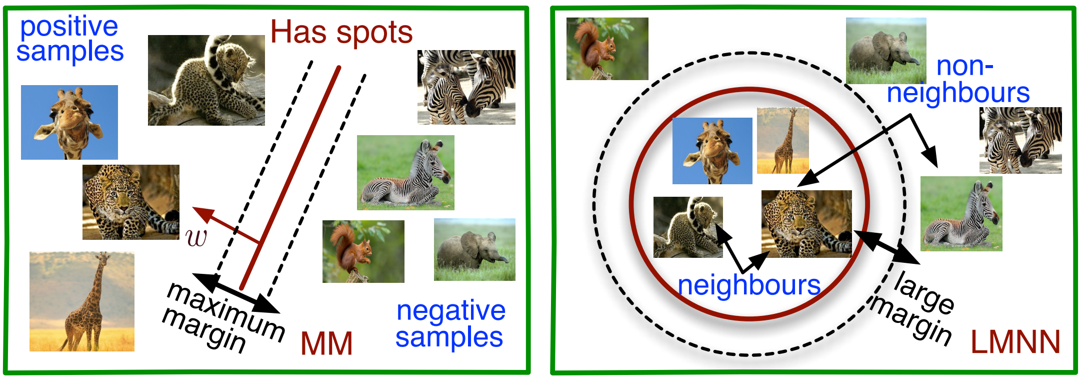


Figure 2.1: Illustration of two principles proposed for learning the attribute *has spots*. According to the *max margin principle* (left), the positive samples (animals with spots) must have maximum distance from the boundary that defines negative samples (animals without spots). According to the *large margin nearest neighbor principle* (right), any two samples that are neighbors (two samples with spots) have to be closer to each other than to any non-neighbor (a sample without spots).

tation that separates the positive and the negative samples with largest possible margin, the max margin principle (MM), and learning the object representation that groups similar samples (both positive or both negative) together and non-similar (positive and negative) far apart, the large margin nearest neighbor principle (LMNN). Figure 2.1 illustrates these principles.

To optimize the decision boundary between positive and negative examples, the main objective of learning is to minimize the mistakes in the given set of the training data images.

The first principle The maximum margin (MM) criterion is as follows:

$$\underset{w}{\text{minimize}} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \max\{0, 1 - y_i\langle w, x_i\rangle\}, \tag{2.1}$$

where w denotes a normal vector of the decision boundary between positive and negative samples, $w \in \mathbb{R}^d$. Here, y_i is $+1$ for positive sample x_i and -1 for negative x_i, and $\langle w, x_i\rangle$ is proportional to the distance from x_i to the decision boundary, where $\langle\cdot,\cdot\rangle$ denotes the dot product between two vectors. The first term in the objective, $\|w\|^2$, is a regularization term to control the complexity of the boundary, and the second term stands for a loss term of making mistakes on the training data. For simplicity of the notations, we omit the offset of the decision hyperplane from the origin (also known as bias term b), assuming it is implicitly included in the weight vector w, and all data points are augmented with a unit element. The maximum margin objective, also known as support vector machines (SVM), trains a classifier $f(x) = \langle w, x_i \rangle$ that is able to generalize to unseen data points. Hence, for a new given image of the object $\bar{x}$ we are able to predict whether this object has the attribute a or not using $f(\bar{x})$: if $f(\bar{x}) \geq 0$, then object $\bar{x}$ has the attribute a, otherwise it does not.

The second principle The large margin nearest neighbor (LMNN) criterion is formalized similarly:

$$\underset{M}{\text{minimize}} \quad \sum_{i \sim j} d^2_{\mathcal{M}}(x_i, x_j) + \sum_{\substack{i \sim j, \\ i \nsim k}} \max\left\{0, 1 - (d^2_{\mathcal{M}}(x_i, x_k) - d^2_{\mathcal{M}}(x_i, x_j))\right\}, \quad (2.2)$$

where $d_{\mathcal{M}}$ is the Mahalanobis distance metric between samples x_i and x_j parameterized by the matrix M of a linear transformation such that:

$$d^2_{\mathcal{M}}(x_i, x_j) = \|M(x_i - x_j)\|^2. \quad (2.3)$$

Here, samples x_i, x_j are considered neighbors, $i \sim j$, if $y_i = y_j$ and non-neighbors, $i \nsim j$, if $y_i \neq y_j$. The large margin nearest neighbor objective learns a metric space $\mathcal{M}$ with the goal that the neighbors always belong to the same class, while examples from different classes are separated by a large margin. Hence, it is most suitable for the nearest neighbor retrieval. In the $\mathcal{M}$ space, we can use a k-nearest neighbor (k-NN) classifier to predict whether a new object $\bar{x}$ has the attribute a or not based on its k neighbors, where k is the parameter of our choice (typically 1 or 3). First, we find its k nearest neighbors in the $\mathcal{M}$ space by the distance from $\bar{x}$ to all training examples using $d^2_{\mathcal{M}}(\bar{x}, x_i)$ as the distance function for $i = 1, \dots, N$. Then, we predict whether $\bar{x}$ has the attribute a or not depending on the majority of its k neighbors: if the majority has this attribute, then the object $\bar{x}$ has the attribute a, otherwise it does not.

In the following sections we describe a variety of computer vision applications that take attribute-based approach to solve the underlying tasks. Refer first to our visualization in Figure 2.2.

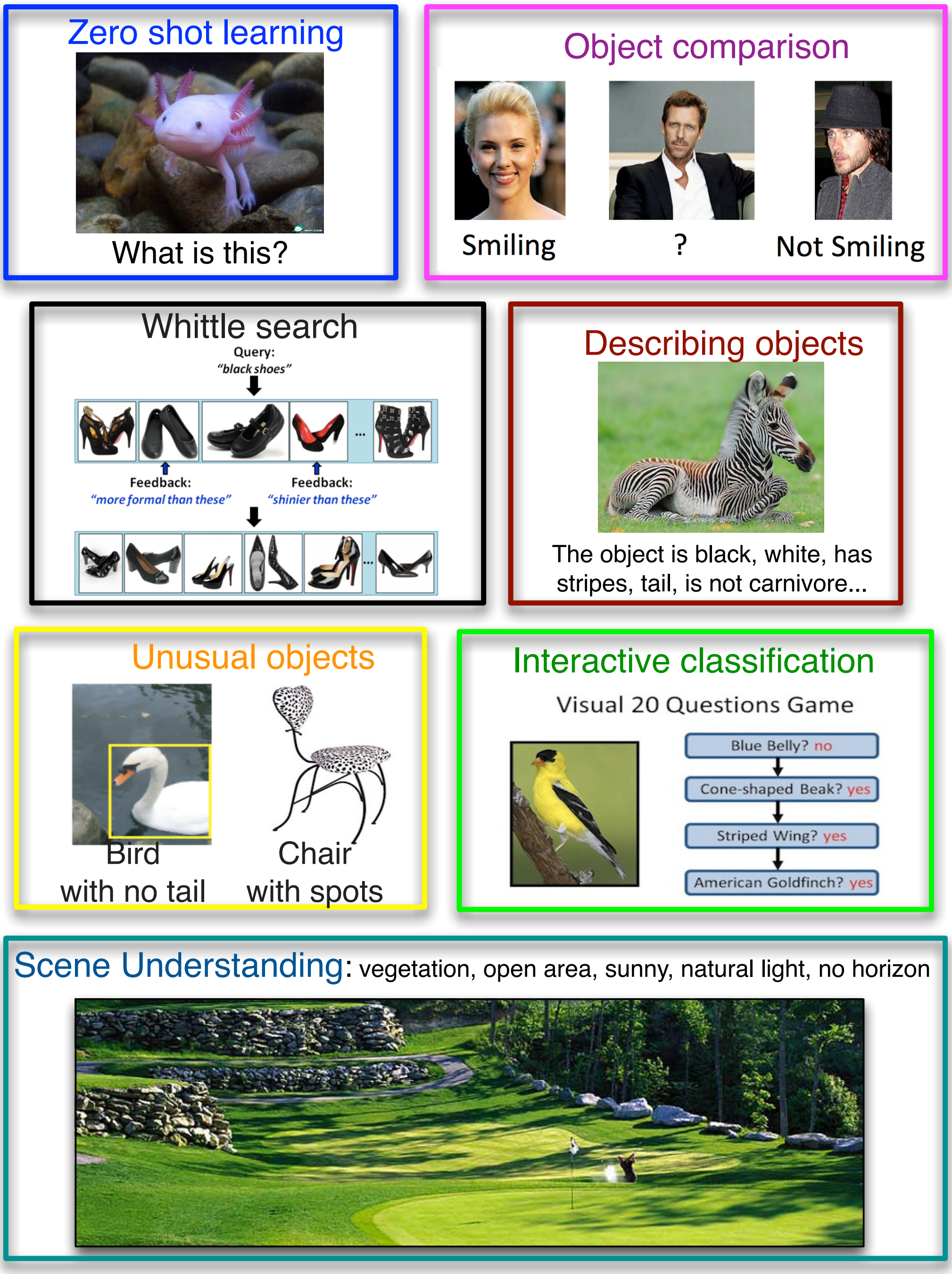


Figure 2.2: Learning with attributes is an active area of computer vision research inspired by their ability to communicate results in the same way as humans. Attributes have been shown to be useful in a variety of computer vision tasks and applications such as describing objects, zero shot learning, objects comparison, intelligent image search, to name a few, which are covered in this chapter.

2.1.3 Attributes for Zero Shot Learning

Zero shot learning as described in (Lampert et al., 2009; Palatucci et al., 2009) refers to a situation where we want to recognize unseen object classes, i.e. object classes with no training images available. In this setting, a direct object recognition model cannot be trained, and so far the only proposed solution is the attribute-based recognition. The main idea is to recognize objects by their attribute description. For example, if attribute description says, this is an image of an animal that is *black* and *white*, *has stripes*, and is *not a carnivore*, this matches the description of a *zebra*. Even if there was no training example of *zebra*, knowing the attributes of the image and attributes of the object classes is enough to fit the image to the best class description.

Defining the attributes Together with the zero shot scenario, the authors introduced the first dataset with the attribute annotation called *Animals with Attributes*(AwA) dataset[1]. It has images of animals from 50 classes together with 85 binary attributes of Osherson et al. (1991) that describe each animal class. Thus, in this case, the attributes are defined per animal class, and the class-attribute description is represented in the form of 50 × 85 binary matrix (refer here to Figure 2.3).

How does it work? From the class-attribute matrix, we can clearly see how the attributes are shared among all classes; for example, class *leopard* has *spots* and so do classes *giraffe* and *dalmatian*. This implies that attributes are not class-specific properties; hence, they can transfer knowledge between object classes: we can learn the attribute *swims* from the available training images of *dolphin*, *seal* and *blue whale* and predict that new images with *humpback whale* also have this attribute (with certain confidence). To sum up, we can learn visual attributes from the classes that have training examples, and predict the attributes on the unseen classes to perform zero shot learning.

Set-up During training, we are given annotated images of 40 animal classes and their class-attribute binary matrix (Figure 2.3). During testing, we have unlabeled images of the remaining 10 classes: *chimpanzee, giant panda, hippopotamus, humpback whale, leopard, pig, raccoon, rat, seal* and the class-attribute binary matrix. The task is to annotate the test images with class labels.

[1] http://attributes.kyb.tuebingen.mpg.de/

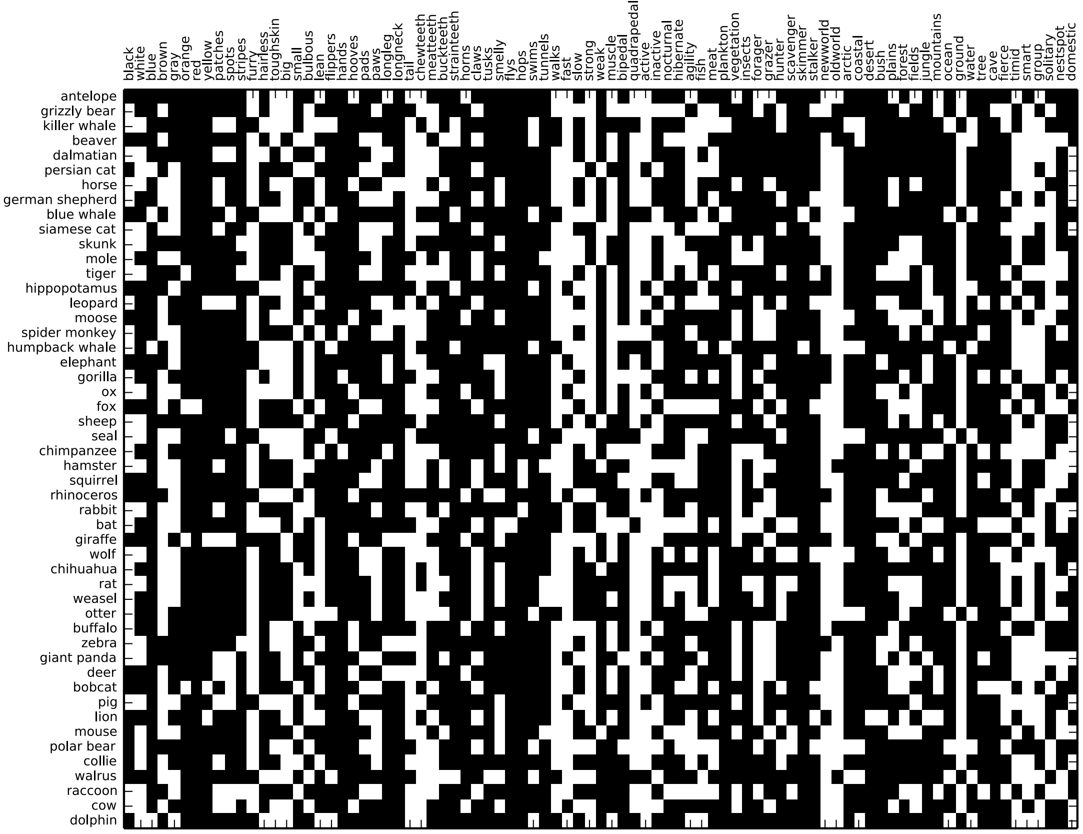


Figure 2.3: Class-attribute binary matrix from the *Animals with Attributes* (AwA) dataset. This dataset has images of animals from 50 classes (vertical axis) with 85 binary attributes that describe each animal class (horizontal axis). White color encodes 1 (the class has this attribute), and black color encodes 0 (the class does not have this attribute).

Learning model 85 attributes $a_1, a_2, \ldots, a_{85}$ are learned based on the annotated data from 40 classes. Each model is trained to classify between the images of those animal classes that have the attribute against images of classes which do not. In principle, SVM or any supervised learning method can be used to learn the attributes. Specifically, we are interested in exploring probabilistic classifiers that output the probability of the attribute being present in the image. To produce probabilistic outputs for the SVM model, a commonly used technique is to apply Platt scaling (Platt, 2000) that applies a logistic transformation to map the classifier scores into the $[0, 1]$ interval. Alternatively, we can learn probabilistic classifiers directly, for example, by the logistic regression model.

Logistic regression seeks class probability distribution parameterized by a vector w in the form of logistic sigmoid:

$$P(y|x,w) = \frac{1}{1+\exp(-y\langle w,x\rangle)}. \tag{2.4}$$

Given N training samples, logistic regression training maximizes the conditional data likelihood $p(y_1,\dots,y_N|x_1,\dots,x_N,w)$ with respect to the parameter w (or equivalently minimizes its negative log-likelihood):

$$w = \operatorname*{argmin}_{w} \sum_{i=1}^{N} \log(1+\exp(-y_i\langle w,x_i\rangle)). \tag{2.5}$$

The ℓ_2-regularized logistic regression model produces *maximum a posteriori* parameter estimate of w by incorporating the Gaussian prior with zero mean and unit covariance matrix $p(w) = \mathcal{N}(w|0,I)$. The multivariate Gaussian distribution $\mathcal{N}(w|\mu,\Sigma)$ is defined over a d-dimensional vector w as follows:

$$\mathcal{N}(w|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\{-\frac{1}{2}(w-\mu)^T\Sigma^{-1}(w-\mu)\},$$

where the d-dimensional vector μ is called the mean, the $d \times d$-dimensional matrix Σ is called the covariance, and $|\Sigma|$ denotes the determinant of Σ. The negative *log*-form of the Gaussian prior with zero mean and unit covariance matrix is:

$$-\log p(w) = \|w\|^2 + const, \tag{2.6}$$

where $\|w\|$ is the ℓ_2-norm of the parameter vector w. Finally, the regularized logistic regression training can be written in the following form:

$$w = \operatorname*{argmin}_{w} \ \|w\|^2 + C\sum_{i=1}^{N} \log(1+\exp(-y_i\langle w,x_i\rangle)), \tag{2.7}$$

where C is the regularization trade-off parameter that we choose based on the cross validation model selection procedure.

Once we have learned the 85 classifiers, given an image x, we can compute the probabilistic scores for each attribute to be present: $p(a_1|x),\dots,p(a_{85}|x)$. Here, $p(a_i|x)$ denotes $p(a_i = 1|x)$. Also, we can compute the joint probability of all attributes being present in the image:

$$p(a|x) = \prod_{i=1}^{85} p(a_i|x). \tag{2.8}$$

To predict a class label for a test image $\bar{x}$, Lampert et al. (2009) proposed the direct attribute prediction model (DAP) that uses attribute representation as an intermediate layer between image data and labels. After some simplifying assumptions on the prior distributions over attributes and over class labels, we obtain the following form of the label posterior:

$$p(y|\bar{x}) \propto \prod_{i=1}^{85} p(a_i^y|\bar{x}), \text{ and the classifier } f(\bar{x}) = \text{argmax}_{y \in \mathcal{Y}} p(y|\bar{x}). \tag{2.9}$$

Essentially, the label posterior $p(y|\bar{x})$ computes the probability of the class y under its class-attribute description $p(a_i^y|\bar{x})$ across 85 attributes $i = 1, \ldots, 85$. Here, $p(a_i^y|\bar{x}) = p(a_i|\bar{x})$ if animal class y has attribute a_i, and $p(a_i^y|\bar{x}) = 1 - p(a_i|\bar{x})$ if animal class y does not have it. For example, for a class *seal* we would compute the probability of the object $\bar{x}$ being *gray*, being able to *swim* and being *not furry* as $p(a_{gray}|\bar{x})$, $p(a_{swims}|\bar{x})$, $1 - p(a_{furry}|\bar{x})$. After computing the probabilities with respect to the class-attribute description of *seal* we multiply all of them to obtain a score for *seal* to be the class label of $\bar{x}$. Once we compute the scores for all the possible class labels, we take maximum to define the class of $\bar{x}$.

Remarks The performance of this model achieves 40.5% of object recognition accuracy over 10 unseen classes. Admittedly, these results directly rely on the quality of the attribute predictors, which requires much annotated training data (in this case, 40 animal classes, around 25000 images, have been used for learning the attributes). Recently, Jayaraman & Grauman (2014) showed that the zero-shot recognition results can be improved to 48.7% by taking into account the reliability of the attribute predictions during learning.

Being defined for the zero-shot scenario, the DAP model cannot benefit from the situation where we are given a few training samples of the test classes. In Chapter 3 and Chapter 4, we address this situation and propose several solutions to improve the attribute-based models and their recognition performance.

Another way of improving zero-shot learning scenario is to look for alternative sources of knowledge transfer rather than semantic attributes. Yu et al. (2013) proposed to design automatically discriminative category-level attributes that are not interpretable but aim directly at good object classification performance. Mensink et al. (2014) proposed to use co-occurrences of visual concepts in images that are easy to obtain from web-search hit counts. The classifier for the unseen class is then a weighted combination of related classes, where the co-occurrences statistics defines the weight.

2.1.4 Attributes for Comparing the Objects

Attributes like *live in water* and *carnivore* are naturally defined as binary attributes, because they are either present or not. Some attributes like *white*, *black* and *stripes* can also be defined for comparison of different objects with respect to how much of the attribute property the objects have. For example, a polar bear is *whiter* than a zebra, a panther is *blacker* than a dalmatian. The attributes that can predict a relative strength of the property in object images are called *relative* attributes and were introduced by Parikh & Grauman (2011b).

Defining the relative attributes The expression of the relative attribute strength can be of two types: *less than*, *more than* and *similar* amount, denoted as $\prec$, $\succ$ and $\sim$ correspondingly. The attribute relation is defined for object classes. For example, in the case of the attribute *natural* we have *highway* $\prec$ *coast* $\sim$ *mountain* $\sim$ *forest*, which means the category highway is *less natural* than coast, and the categories coast, mountain and forest are all *similarly natural.* Another type of object classes are celebrities; for example, Alex Rodriguez, Hugh Laurie, Scarlett Johansson. Their images are compared with different face properties, such as how much they are *smiling* and how *young* they look.

How does it work? Given the annotated training data that states how two object categories relate to each other with respect to an attribute, a ranking function $r(x)$ is learned to capture the relative attribute strength in images of these categories. At test time, the learned ranking function predicts the relative strength of the attribute in new images.

Learning model Learning the ranking function $r(x) = \langle w, x\rangle$ parameterized by the weight vector w, is similar to learning the SVM classifier with adjusted class information to encode the pairwise relationship between classes. The optimization task is formulated to capture orderings between pairs of samples with respect to the class-attribute relation table.

$$\underset{w,\xi_{ij},\gamma_{ij}}{\text{minimize}} \quad \frac{1}{2}\|w\|^2 + C(\sum_{y_i \succ y_j} \xi_{ij}^2 + \sum_{y_i \sim y_j} \gamma_{ij}^2) \tag{2.10a}$$

$$\text{s.t. } \langle w, x_i\rangle - \langle w, x_j\rangle \geq 1 - \xi_{ij} \quad \text{for all images with} \quad y_i \succ y_j \tag{2.10b}$$

$$|\langle w, x_i\rangle - \langle w, x_j\rangle| \leq \gamma_{ij} \quad \text{for all images with} \quad y_i \sim y_j \tag{2.10c}$$

$$\xi_{ij} \geq 0 \quad \text{and} \quad \gamma_{ij} \geq 0. \tag{2.10d}$$

For a new image $\bar{x}$, the rank model $r(\bar{x})$ predicts the relative attribute score of this image, which can be compared across images.

Remarks This model was the first to allow object comparison in relative terms. To quantify the effectiveness of such comparison, a user study was performed: image description using binary attributes was compared against image description with the relative attributes. This user study supported the claim that relative descriptions are more precise than the binary ones and placed a basement for a new type of human-computer communication using relative attributes.

2.1.5 Attributes for Intelligent Image Search

For a specific task like querying a large collection of face images or a shopping website, there is a strong need for intuitiveness of the search engine. Attributes suit well this need because they are interpretable and the attribute vocabulary can be defined as appropriate for the task. For example, when the user looks for *black* shoes, or a young woman *with glasses*, the binary attributes are capable of capturing this query (Kumar et al., 2008). To make the search results more specific, for example, in order to find shoes that are *more formal* than shoes on display, the relative attributes serve the purpose Kovashka et al. (2012). Refer here to Figure 2.4.

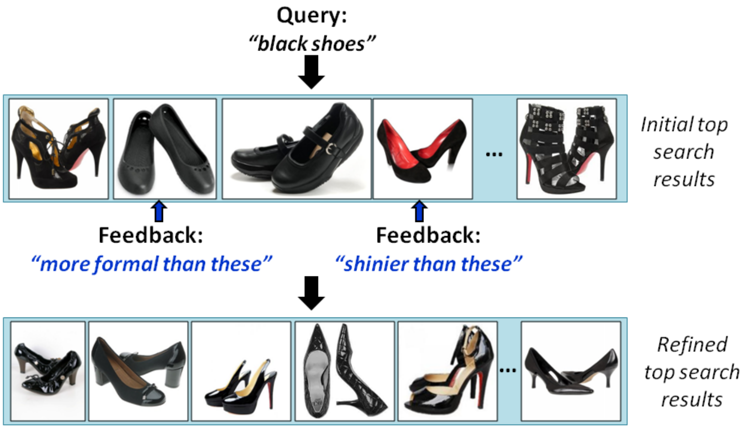


Figure 2.4: Whittle search allows a user to give the relative attribute feedback during querying, e.g. *more formal than these*, *shinier than these* (top). Such feedback refines the search results that are displayed to user for further interaction (bottom). This illustration is from the original work by Kovashka et al. (2012).

Defining the attributes The domain specific attribute vocabulary needs to be defined with respect to the properties of interest for image search. For example, for shoe shopping the suitable attribute vocabulary would be: *shiny*, *formal*, *heel height*, *sporty*, *open*, etc. Compared with previous work, where we trained relative attributes based on the category-level comparisons, here, we have one category *shoes* and the attributes strength is compared on the image level (shoes in image A are *more formal* than in image B).

How does it work? First, all relative attributes are learned in the form of ranking functions based on annotated data of image-level comparisons (shoes in image A have more of the attribute property than shoes in image B). The learning objective remains the same as in (2.10), which allows incorporating supervision via pairwise comparisons in images directly. Second, all relative attributes are evaluated on the pool of image data that will be used for human-computer interaction. At test time, the user observes the Top-k best matching images according to the query, and gives feedback in the form of relative attributes as illustrated in Figure 2.4 (top row). Each feedback imposes an additional constraint in the multi-dimensional attribute space: find all shoes x_i in the image pool that are *more formal than* the shoes $x_{feedback}$. The images are re-ranked based on how well their attributes satisfy all imposed constraints. The Top-k images that satisfy all (or almost all) user requests are displayed as most relevant for further interaction.

Learning model Relevance of the images is determined in the multi-dimensional space of relative attributes, and user feedback is seen as a constraint in this space. For example, the constraint *shoes that are more formal than these* determines a half space of all images with respect to the attribute *formal* that are ranked higher than the selected image. When we get two or more feedbacks from the user, the intersection of all subspaces gives us the images that satisfy all the constraints and are most relevant for search results. Images that satisfy all but one constraint are less relevant and so on. The approach is called *whittle*, since it allows users to "whittle away" irrelevant information via intuitive statements about their attribute preferences.

Remarks Combination of binary and relative attributes for image queries is also possible (Sadovnik et al., 2013), as well as queries with multiple attributes at the same time (Rastegari et al., 2013). Another interesting line of research is related to personalization of the image search queries, where domain adaptation can be effectively used to adapt the classifier to user preferences (Kovashka & Grauman, 2013), or learning user behavior in image search (Parikh & Grauman, 2013).

2.1.6 Attributes for Describing an Unusual Appearance

Attributes can be used to create automatically textual descriptions of images (Farhadi et al., 2009; Kulkarni et al., 2011). After all attribute classifiers have been trained, we can evaluate their values in the images and create automatic image captions. Most of the time, we would like to report either most prominent or most interesting attributes in the caption. In the first case, we can report top-k highest scores among the attribute predictions, where k is the parameter of our choice (typically less than 10). In the second case, we have to define what could be interesting to report, and this is indirectly related to the unusual appearance of the objects in images. Since people like to take pictures of something unusual in their surroundings, it makes sense to detect and describe this in the image caption. Discovering the notion of unusualness is a difficult task in computer vision, the main focus of which is to recognize the objects and not the unexpected aspects of the known objects. Farhadi et al. (2009); Saleh et al. (2013) showed that semantic attributes can be suitable to perform this task automatically.

How does it work? Semantic attributes are designed to capture object properties and serve as a measurement of what is "usual" in the object appearance. The ground truth annotation specifies which attributes are typical for each class. If the typical object attribute is absent or the object has atypical attributes, it is naturally treated as having "unusual" appearance. For example, let us look at the attributes of object parts that are visible, such as *has tail*, *has leg*, etc. Macro photo of the bird typically covers only the upper body part, and the attribute description would alert us that this is a photo of the bird with *no tail* visible. Another example is a chair with dalmatian textile; its description will indicate unusual attribute *has spots* for the object chair (refer to Figure 2.2).

Learning model In order to learn the semantic attributes, a logistic regression model or a linear SVM model can be trained for each attribute; the latter was shown to perform slightly better. The automatic description that reports unusual attributes in the image was evaluated manually and showed encouraging results. More recent work in this direction by Saleh et al. (2013) models the abnormality in images by relating unusualness to a surprise score of the objects in images based on their attribute description.

Remarks Farhadi et al. (2009) consider attributes of two types: semantic and discriminative. Semantic attributes describe the 2D/3D properties of the objects (*is 3D boxy*, *is cylindrical*), visible parts (*has head*, *has leg*, *has wheel*, *has wing*), material types (*has wood*, *is furry*, *has glass*, *is shiny*). The discriminative attributes do not necessarily have semantic meaning, but they serve as discriminative features for object recognition. This is important because semantic attributes are limited by the vocabulary size and are not always sufficient for differentiating between all object categories. For example, instances of both cats and dogs can share all semantic attributes in our vocabulary. In addition to 64 semantic attributes, the authors propose to learn 1000 discriminative attributes that correspond to certain splits in the visual feature space, and support class discrimination. The easiest principle is learning to distinguish between randomly chosen groups of classes: *cars* versus *zebra*, *chairs* and *dolphins* versus *airplanes*, for example. These splits capture class discrimination based on visual appearance or visual features of object classes and do not necessarily enforce semantic discrimination. The question of learning semantic and non-semantic attributes for better class discrimination addresses the need to learn class-discriminative visual elements called *mid-level features* (Boureau et al., 2010; Sharmanska et al., 2012; Rastegari et al., 2012; Singh et al., 2012; Quadrianto et al., 2013; Mittelman et al., 2013).

2.1.7 Attributes for Fine-grained Recognition and Scene Understanding

Bird classification with the 20 questions game Attributes have also been successfully applied also to the fine-grained classification setting, where the task is to distinguish finely between bird species, or car and motorcycle models, or architectural styles, to name a few. Branson et al. (2010) adopted the 20 questions game for interactive learning of bird species. In this game, the computer algorithm tries to guess the type of bird in the given image while asking the user about its attributes. The questions are mostly about the distinctive color and patterns of the body parts; for example, *Is the belly white?*, *Is the breast red?* The main principle of this game is to choose the next question such that it maximally reduces the uncertainty about bird species based on the image-attribute description. Uncertainty is measured with respect to the expected information gain of knowing the answer about the attribute, and is computed in each round of the game for each attribute. The most informative attribute is chosen for the next question and the game goes on.

Attributes for scene understanding Attributes have also been explored for scene understanding, mainly for the purpose of learning high-level representations of scenes, automatic image captioning and semantic image search. Patterson & Hays (2012) proposed a large scale dataset to discover, annotate and recognize scene attributes. This dataset has around 700 categories covering indoor and outdoor scenes, transport, natural and man-made scenes, scenes with water and snow, for example. More than 100 attributes were selected during user studies to discriminate between different scene categories. The experimental evaluations confirm that these attributes serve as efficient low-dimensional features to capture high-level context and semantics in scenes.

2.2 Image Representation

In this section, we will overview frequently used visual feature representations $x \in \mathcal{X}$ that can be extracted from the image content. These features can capture image properties on a global scale – for example, color histograms, shape contours, or edge histograms computed from the whole image – or they can stay concentrated locally on specific locations in the image, such as mountain peaks and building corners. The latter are often called keypoint features or interest points, and are described by the appearance of the patches of pixels that surround the interest point. Image descriptors have been originally introduced to match keypoint features for 3D reconstruction. As the field moved towards object category recognition with more realistic and more complex object appearances, the need for good quality image features that can generalize across object categories has arisen and remained until now.

In this thesis, we cover three types of features that have been proven best for object recognition challenge in the last 10 years of research. We start with the scale invariant feature descriptor called SIFT (Lowe, 2004) and the bag-of-words (BoW) representation of SIFT that is widely used in the community and among the best performing in the PASCAL VOC challenges during 2008 – 2012 (Everingham et al., 2014). Then, we proceed to the Fisher kernel representation that goes beyond count statistics of the bag-of-words features by using the probabilistic generative model of the local features (Perronnin & Dance, 2007; Perronnin et al., 2010). Finally, we conclude with the best performing visual features that are based on deep-convolutional neural networks and achieve state-of-the-art performance in the large-scale object recognition challenge ImageNet (Krizhevsky et al., 2012; Donahue et al., 2014).

2.2.1 SIFT and Bag-of-Words Representation

The scale invariant feature descriptor SIFT is formed from a selected image region, also called the interest point region or patch, that surrounds the interest point. First, the interest points in the image are identified by the SIFT detector and then their patch descriptors are computed by the SIFT descriptor. It is also common to use independently the SIFT detector, i.e. computing only the interest points, or the SIFT descriptor, i.e. computing the descriptors of given interest point regions.

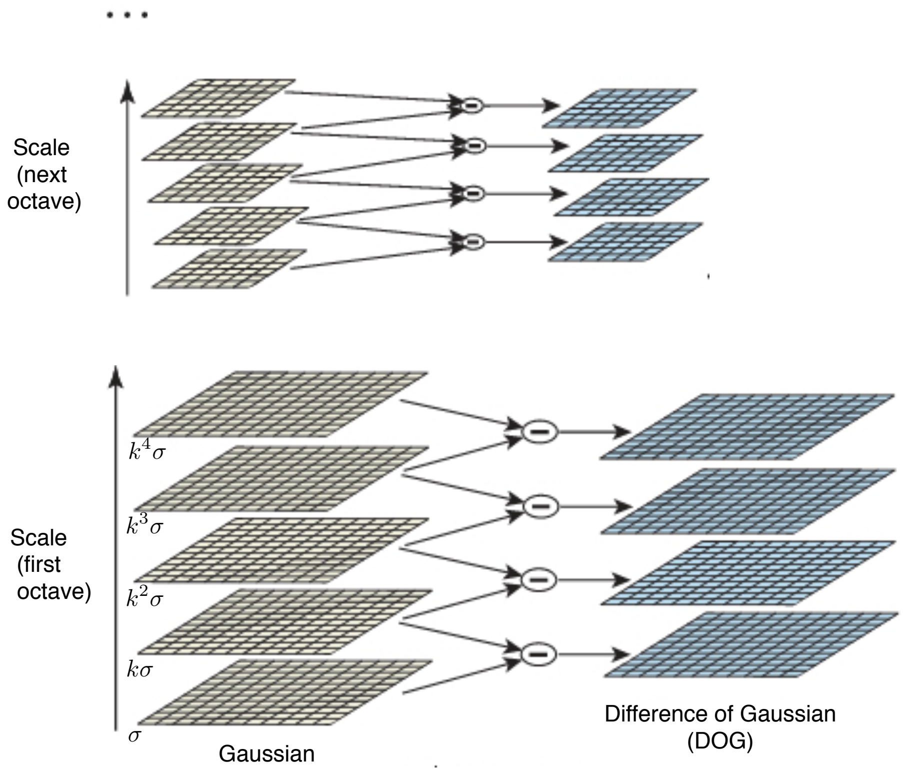


Figure 2.5: Formation of the scale space images in the SIFT detector. This illustration is from the original work by Lowe (2004).

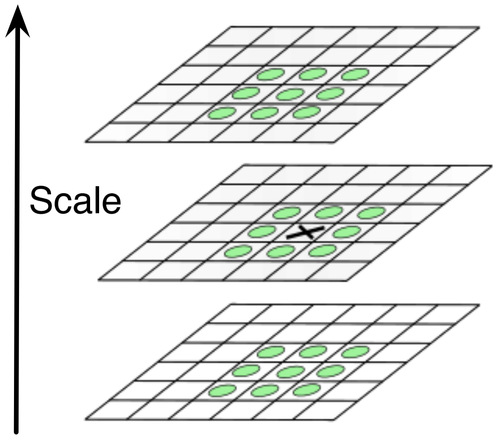


Figure 2.6: Visualization of 26 neighbors for computing local extrema in the scale space. This illustration is from the original work by Lowe (2004).

Stage 1: Interest point identification with SIFT detector stage. In this stage, image $I(x, y)$ is repeatedly convolved with Gaussians at different scales σ:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad \text{and} \quad G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \tag{2.11}$$

The scale range is pre-defined $\{\sigma, k\sigma, \ldots, k^s\sigma\}$ to produce the set of scale space images (left column in the Figure 2.5) with a fixed number of convolved images per octave (5 in our illustration). In practice, k is taken in the form of $2^{\frac{1}{s}}$, and once the octave is completed with all convolved images, we down-sample the Gaussian image with the initial value of σ by a factor of 2, and start the process of convolution for the new octave. After all octaves are completed, we take the Difference-of-Gaussian operator inside each octave to produce DOG images $D(x, y, \sigma)$ from adjacent Gaussian images (right column in the Figure 2.5):

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \tag{2.12}$$

Finally, interest point candidates are identified as local minima/maxima of the Difference-of-Gaussian images $D(x, y, \sigma)$ in the scale space. To achieve this, each point in the scale space is compared with its 26 neighbors: 8 in the current image and 9 corresponding points in each of the 2 adjacent scales above and below in the scale space. Refer to Figure 2.6 for visualization.

As a result of the first stage, the SIFT interest point detector identified a list of interest points with specified image location, scale at which the interest point was detected and orientation (estimate based on dominant gradient orientation in the interest point region). These parameters impose a local 2D coordinate system for describing the interest point region that ensures invariance to image location, scale and rotation changes. In the next stage, we compute the SIFT descriptor for each interest point region, which has the properties of being highly distinctive and also (at least partially) invariant to the other variations such as illumination and 3D viewpoint.

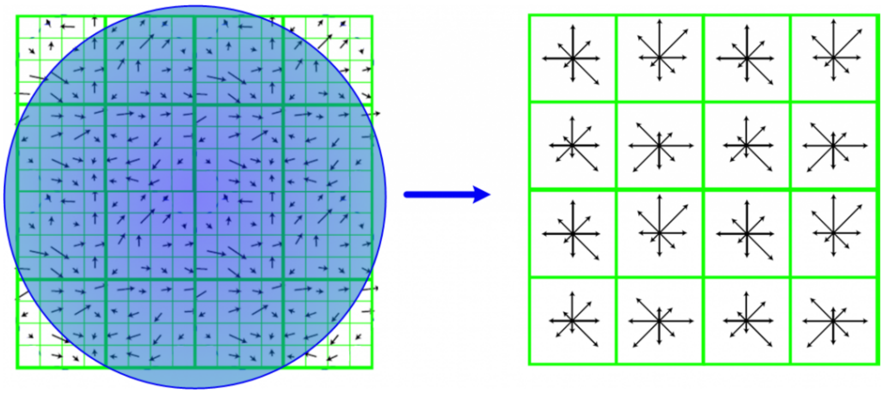


Figure 2.7: The SIFT descriptor is computed from the image patch around the interest point. It captures the appearance of the interest point region and serves as an example of the local image descriptor. The gradient orientation and magnitude are computed in the image patch (left), and then accumulated over 4x4 subregions into histograms with eight orientations (right) resulting in a 128-dimensional feature vector.

Stage 2: Computing SIFT descriptor The SIFT descriptor characterizes the appearance of the interest point by computing gradient information in the interest point region. This step is performed on the image L closest in scale to the interest point scale, so the image is first smoothed with the Gaussian kernel and all computations are performed in a scale-invariant manner. At each pixel location the gradient of the intensity value is computed (orientation and magnitude) and its orientation is aligned with the interest point:

$$\text{magnitude}(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$
$$\text{orientation}(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)/((L(x+1, y) - L(x-1, y)))$$

After this, each pixel in the interest point region is assigned to the histogram of 8 spatial orientations with respect to its gradient orientation. The score is weighted by its gradient magnitude and by a Gaussian-weighted circular window around the interest point (depicted with blue circle in Figure 2.7). The latter is the

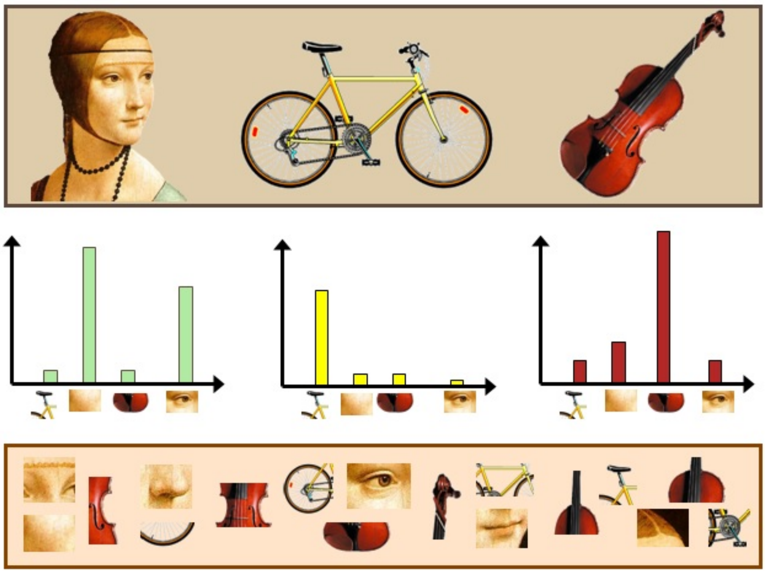


Figure 2.8: The bag-of-words histogram visualization for images of *face*, *bike*, and *violin* (top row). The vocabulary of visual words is collected from all three images (last row) and the histogram that counts words appearance is formed for each image (middle row). Image credit: Fei Fei Li.

Gaussian weighting function that is applied to give less importance to gradients farther away from the interest point center. In SIFT, one histogram is collected per 4x4 pixel subregion, and in total 4x4 subregions are formed in the interest point region. These 16 histograms with 8 bins each are concatenated to form the 128-dimensional SIFT descriptor. Refer to Figure 2.7 for illustration.

Using gradient information makes SIFT descriptor invariant to affine changes in brightness (adding a constant value to the image intensities does not affect its gradients). When normalized to unit length, it also enhances invariance to image contrast (multiplying the image intensities by a constant will vanish). The limits of the SIFT descriptor were tested empirically and showed reliable performance up to a 50 degree change in viewpoint. In this thesis, we also use a fast version of the SIFT features called Speeded Up Robust Features (SURF) (Bay et al., 2008), which computes gradients and magnitudes in only two orientations (x and y) and with some other modifications overcomes computational issues of SIFT features.

Stage 3: Forming the bag-of-words histogram of SIFT The bag-of-words model forms the last stage of our feature extraction procedure and represents the image as a collection of visual words formed from the interest point regions.

It is inspired by the successful feature representation in the natural language processing domain, where each document is characterized by its words frequency, i.e. how often words appear in the document, and neglects the general text structure. The BoW histogram of word frequencies is formed to represent the document and has the important property of invariance to the order of words.

In computer vision, the bag-of-*visual*-words model was first introduced for the task of image retrieval (Sivic & Zisserman, 2003) and object categorization (Csurka et al., 2004) as a way to overcome the problems with occlusions (at least partially) and intra-class variation. As in our case, the visual words correspond to image patches around interest point regions, and the BoW histogram counts how many times each word occurs in the given image. Refer to Figure 2.8 for illustrative example.

In order to count visual words that could appear in the image, we first need to create the vocabulary of possible words from the collection of images. In our case, we build the vocabulary of interest point regions in the following steps:

1. for all images in the collection extract the SIFT features from the interest points regions;

2. cluster all extracted SIFT features into k groups using k-means algorithm;

3. create the vocabulary of k visual words by taking the mean value of each group as one word.

Finally, for a given image, we construct the bag-of-words histogram of SIFT descriptors by counting how many times each word from the vocabulary appears in the image:

1. extract the SIFT features from the interest points regions in the image;
2. assign each SIFT feature to the nearest of the k words in the vocabulary;
3. count how many times each word from the vocabulary appears in the image and form the histogram of counts.

Refer to Figure 2.9 for illustration of how to learn the vocabulary of visual words and to create the bag-of-words histogram.

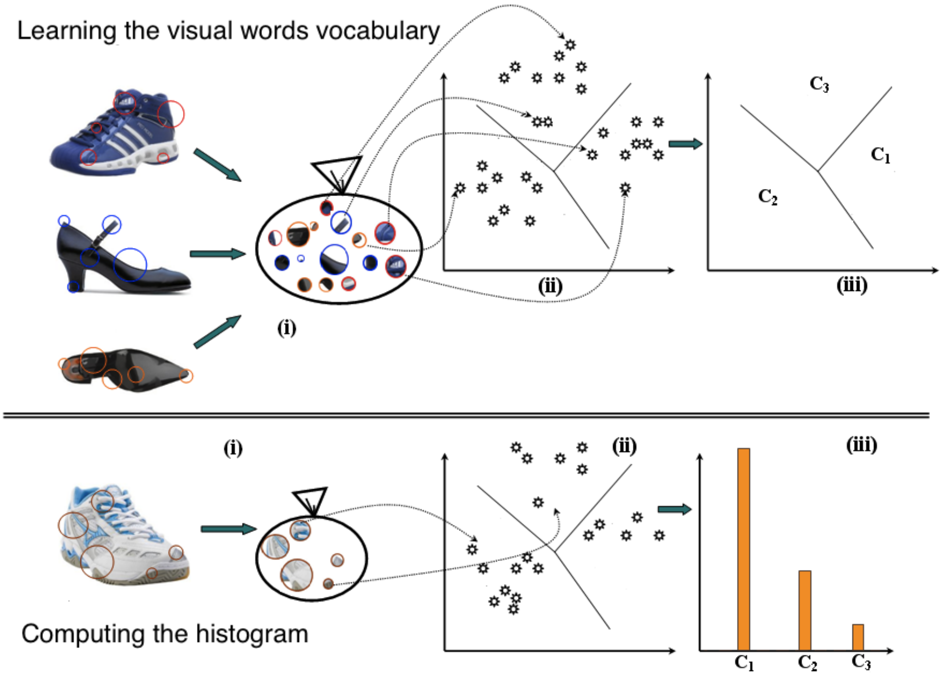


Figure 2.9: Illustration of learning the visual word vocabulary from the collection of images (top row) with $k = 3$ visual words. The bag-of-words histogram is formed for a new given image (bottom row) over the words in vocabulary C_1, C_2, C_3. Image adapted from (Tomasik et al., 2009).

2.2.2 Fisher Vectors

This representation extends the bag-of-words model of images beyond counting the frequencies of its word appearances. Fisher vectors explore the image representation from the generative probabilistic modeling perspective and encode the direction in which the parameters of the model should be moved to fit better the data, measured by the Fisher score. First, a generative probabilistic model parameterized by an unknown vector θ is used to generate the image features, and then, the Fisher score is computed as the natural gradient of the log-likelihood of this model with respect to its parameters θ using the image collection. Let us consider the image representation using Fisher vectors in more detail.

Following the description in (Perronnin & Dance, 2007; Perronnin et al., 2010; Sánchez et al., 2013) we use a Gaussian mixture model as the probabilistic generative model for the local feature appearance. Then, the probability distribution of the local descriptor x_i (for example, a SIFT descriptor at interest point i) is generated using a mixture of K Gaussians:

$$P(x_i|\theta) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}\,(x_i|\mu_k, \Sigma_k), \tag{2.13}$$

$$\text{subject to } \sum_{k=1}^{K} \pi_k = 1, \text{ and } \pi_k > 0 \text{ for all } k, \tag{2.14}$$

where π_k, μ_k, Σ_k are the mixing weight, the mean and the covariance matrix of the k-th Gaussian component. Here, θ encodes all parameters of the model: $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ for all K mixture components. To reduce the computational cost, the covariance matrix Σ_k is assumed to be diagonal with σ_k^2 variance vector for each Gaussian k. Additionally, we re-parameterize the mixing weights:

$$\pi_k = \frac{\exp(\alpha_k)}{\sum_{l=1}^{K} \exp(\alpha_l)}, \tag{2.15}$$

so that mixing weights automatically satisfy the non-negativity constraints and $\sum_{k=1}^{K} \pi_k = 1$.

In order to compute the likelihood model, we utilize the i.i.d. assumption over the set of local features, i.e. the local image features are independent and identically distributed random draws from the Gaussian mixture model:

$$P(x|\theta) = \prod_{i=1}^{N} P(x_i|\theta). \tag{2.16}$$

Its log-likelihood is

$$\mathcal{L} = \log P(x|\theta) = \sum_{i=1}^{N} \log P(x_i|\theta). \tag{2.17}$$

In the computer vision literature, the Gaussian mixture model is often referred to as a universal (probabilistic) visual vocabulary if used as generative probabilistic models for local descriptors (Winn et al., 2005; Sánchez et al., 2013). The model parameters θ are learned using the expectation-maximization algorithm that optimizes maximum likelihood criterion from the large pool of local image features (of training images). The gradient of the log-likelihood with respect to the parameter model has the following form:

$$\nabla_{\alpha_k}\mathcal{L} = \sum_{i=1}^{N}(\gamma_{ik} - \pi_k), \tag{2.18a}$$

$$\nabla_{\mu_k}\mathcal{L} = \sum_{i=1}^{N}\gamma_{ik}(x_i - \mu_k)\frac{1}{\sigma_k^2}, \tag{2.18b}$$

$$\nabla_{\sigma_k}\mathcal{L} = \sum_{i=1}^{N}\gamma_{ik}(\frac{(x_i - \mu_k)^2}{\sigma_k^3} - \frac{1}{\sigma_k}), \tag{2.18c}$$

$$\text{where } \gamma_{ik} = \frac{\pi_k\, \mathcal{N}\,(x_i|\mu_k, \sigma_k)}{\sum_{l=1}^{K}\pi_l\, \mathcal{N}\,(x_i|\mu_l, \sigma_l)}. \tag{2.18d}$$

γ_{ik} is the soft-assignment of x_i to Gaussian k, also known as “responsibility”, which the Gaussian component k takes to explain the local feature x_i (or posterior probability once we have observed data x_i with the prior probability encoded in the mixture weight π_k).

The Fisher vector representation takes one step further and *corrects* the gradient of the log-likelihood to have the form of the natural gradient (Amari, 1998). The fact that generative models generally lie on a non-linear Riemannian manifold means that in our case the small delta-change in the parameter μ_k and delta-change in the parameter σ_k have different effects. Hence, a different form of the gradient has to be considered. The information about the natural gradient of the log-likelihood is encoded in the Fisher score:

$$\mathcal{G}_\theta(x) = I_F^{-\frac{1}{2}}\ \nabla_\theta\ \mathcal{L}(x) \tag{2.19}$$

where I_F is the Fisher information matrix, $I_F = E_{x\sim P(x|\theta)}[\nabla_\theta\mathcal{L}(x)\ \nabla_\theta^T\mathcal{L}(x)]$. The square root of the inverse of the Fisher information matrix, $I_F^{-\frac{1}{2}}$, is obtained using Cholesky decomposition of the I_F^{-1}, which is positive semi-definite as inverse of the positive semi-definite matrix I_F. Another way to see the Fisher score is in terms of explicit feature map representation of the Fisher kernel:

$$K_F(x, y) = \mathcal{G}_\theta(x)^T\mathcal{G}_\theta(y), \tag{2.20}$$

defined as the inner product of the directions of natural gradients over the parameter manifold, and introduced by Jaakkola & Haussler (1998) as a principled way to use the power of probabilistic generative models in kernel methods. The rationale behind the Fisher kernel is that two similar objects induce similar natural gradients in the parameters of the generative model, and the Fisher information metric I_F corrects the similarity measurement due to the fact that generative models generally lie on a non-linear Riemannian manifold.

The setting is complete once we discuss how to compute the Fisher information matrix I_F. For the Gaussian mixture model, the exact solution of I_F cannot be computed, and it is usually approximated by the identity matrix, or sample average $I_F \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \mathcal{L}(x_i)\nabla_\theta^T \mathcal{L}(x_i)$. We use an analytical approximation of I_F described by Perronnin & Dance (2007), which relies on the following assumption about responsibilities γ_{ik}: the distribution of the assignment of x_i to Gaussian k is sharply peaked, meaning there is one Gaussian component k such that $\gamma_{ik} \approx 1$ and for all other components, $\gamma_{il} \approx 0$, $\forall l \neq k$. This approximation *corrects* the gradient of the Gaussian mixture model (2.18a) - (2.18c) by $\frac{1}{\sqrt{\pi_k}}$, $\frac{\sigma_k}{\sqrt{\pi_k}}$, $\frac{\sigma_k}{\sqrt{2\pi_k}}$ with respect to the mixing weights, the mean and the covariance of kth Gaussian, and the final Fisher score is:

$$\mathcal{G}_{\alpha_k} = \frac{1}{\sqrt{\pi_k}} \sum_{i=1}^{N} (\gamma_{ik} - \pi_k), \tag{2.21a}$$

$$\mathcal{G}_{\mu_k} = \frac{1}{\sqrt{\pi_k}} \sum_{i=1}^{N} \gamma_{ik}(x_i - \mu_k)\frac{1}{\sigma_k}, \tag{2.21b}$$

$$\mathcal{G}_{\sigma_k} = \frac{1}{\sqrt{2\pi_k}} \sum_{i=1}^{N} \gamma_{ik}(\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1). \tag{2.21c}$$

The Fisher Vector representation is a concatenation of all natural gradients: $[\mathcal{G}_{\alpha_k}, \mathcal{G}_{\mu_k}, \mathcal{G}_{\sigma_k}]_{k=1}^{K}$, and the dimension of this representation is $(2d + 1)K$, where d is the dimension of the local feature descriptor. Admittedly, this is a significantly larger representation than using the bag-of-words histogram of local feature appearances, which is K in case of K visual words, but it encodes much richer information about image content compared with the bag-of-words histogram, and shows superior performance in object recognition task in the challenging benchmark datasets. Recently, several approaches have been proposed to improve learning with the Fisher vector representation to make it scalable to larger datasets (Sánchez et al., 2013), and to learn a classifier with Fisher kernel at the same time as a task-specific data representation inspired by deep-layered architecture (Sydorov et al., 2014).

2.2.3 Deep Features

Despite years of research spent on improving feature representation of real-world images, the need for better and richer representation has never diminished. Historically, the progress was driven by two factors: data availability and computational power. Among the best performing strategies to handle challenging benchmark datasets was designing highly discriminative and robust image features like SIFT and bag-of-words representation, the combination of multiple features based on different aspects such as shape, color or texture, and learning very high-dimensional features like Fisher vectors. At the same time, technological progress has driven improvement in computational power and parallel GPU resources. This allowed even more complex models to be considered, and convolutional neural network took over the lead.

Instead of hand-engineered representations like SIFT and Fisher vectors, deep-layered compositional architectures offer a framework to automate feature extraction procedure. In 2012, the deep convolutional neural networks (CNN) proposed by Krizhevsky et al. (2012) achieved record results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) leaving all the other baselines far behind. ILSVRC-2010 dataset covers 1000 categories with roughly 1000 images in each category, which is a subset of the ImageNet large-scale dataset (with 22000 categories) used for benchmarking the baselines.

Results on ILSVRC Deep CNN (Krizhevsky et al., 2012) achieved 37.5% and 17.0% as `Top-1` and `Top-5` error rate performance over 1000 classes. The former state-of-the-art model by Sánchez & Perronnin (2011) achieved 45.7% and 25.7% error rates accordingly. Top-i error rate performance computes how many times the correct label was not among top-i most probable candidates. Sánchez & Perronnin (2011) averaged the predictions of two classifiers trained on Fisher Vectors from two types of densely-sampled SIFT features. And Krizhevsky et al. (2012) used 8-layered convolutional neural networks architecture to extract features and train a classifier in one unified framework.

The architecture The overall architecture proposed by Krizhevsky et al. (2012) is depicted in Figure 2.10 and maps the input 224×224 image via the series of 8 layers to the probability output over 1000 classes. The first five layers of the network are convolutional and the remaining three are fully connected layers with a softmax classifier as a final layer. Each layer consists of:

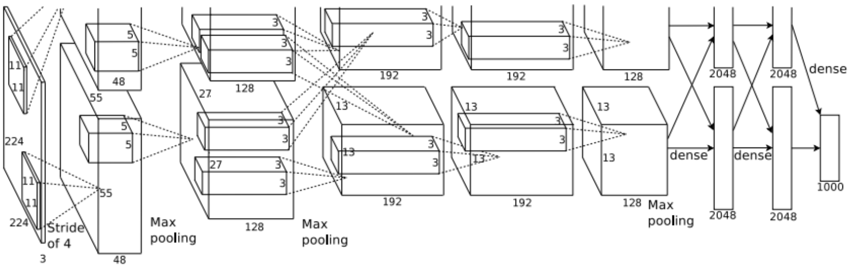


Figure 2.10: The AlexNet deep convolutional neural network (Krizhevsky et al., 2012). This illustration is from the original work by Krizhevsky et al. (2012).

- a convolution of the previous layer output (in the first layer it is the input image) with a set of learned filters,
- a non-linear transformation by passing the responses through a rectified linear function $f(x) = max(0, x)$ (classical sigmoid function could also be used here $f(x) = (1 + e^{-x})^{-1}$, however, the training time is shown to be much worse in this case),
- a response-normalization procedure to correct the contrast across 'adjacent' filters (local contrast operation),
- max pooling over local neighborhoods to summarize the responses in each filter map (with overlapping pooling scheme).

At the final layer, a softmax classifier is applied to learn the probabilistic classifier for each of the 1000 classes:

$$\begin{aligned} Pr(y = i|x, w_1, \ldots, w_{1000}) &= \text{softmax}(i, \langle x, w_1 \rangle, \ldots, \langle x, w_{1000} \rangle) \\ &= \frac{e^{\langle w_i, x \rangle}}{\sum_{k=1}^{1000} e^{\langle w_k, x \rangle}}. \end{aligned} \tag{2.22}$$

There are around 60 million parameters of the network (filters in the convolutional layers and weight matrices in the fully-connected layers) that are trained by back-propagation of the derivative of the loss with respect to the parameters of the network, and updating the parameters via stochastic gradient descent procedure. Admittedly, training deep networks of this size requires a large amount of data to avoid overfitting.

Visualization and interpretation of the network layers Zeiler & Fergus (2014) addressed the question of visualizing and interpreting the layers of the deep networks. Their findings confirm the commonly accepted interpretation of hierarchical structure of features: the first few layers capture high and low frequency information, which correspond to low-level features (corners and edges on the second layer, mesh patterns and texture on the third layer), whereas the later layers learn more class-specific or “high-level” features (a visualization of the forth layer often shows parts of objects like bird legs and dog faces, and the fifth layer shows entire objects with pose variations).

DeCAF features Recently, an open-source implementation of the deep convolutional activation features DeCAF was made publicly available[1](Donahue et al., 2014). In this thesis, we use the DeCAF feature representation extracted as the output of the seventh layer of the architecture in Figure 2.10. The seventh layer is right before fitting the signal to the last fully connected layer with soft-max class predictors, and the output of this layer is a 4096-dimensional feature vector.

2.3 Parametric versus Non-parametric View

In a typical machine learning setting, as, for example, learning the attribute classifier described in Section 2.1.2, we are given a set of N training data samples $\{(x_1, y_1), (x_2, y_2) \dots, (x_N, y_N)\}$, $x_i \in \mathcal{X}, y_i \in \mathcal{Y}$. The samples in the training set are assumed to be drawn i.i.d. from an underlying but unknown data distribution $p(x, y)$ over $\mathcal{X} \times \mathcal{Y}$. The data distribution $p(x, y)$ includes input variables x and output variables y, and the learning task is to infer the prediction function $f : \mathcal{X} \to \mathcal{Y}$.

In the *parametric* approach, we define apriori the mathematical form of this functional relationship, for example, linear, piecewise linear, polynomial, exponential, logarithm, or a combination of them. Subsequently, the weights that are also called model parameters are placed on the chosen form, and a prior distribution (in the probabilistic approach) or a regularization term (in the non-probabilistic approach) is defined over the parameters. For example, the linear attribute classifier $f(x) = \langle w, x \rangle$ parameterized by the weight vector w has the regularization term $\|w\|^2$ in the SVM learning objective, and the logistic regression classifier $f(x) = \frac{1}{1+\exp(-\langle w,x \rangle)}$ incorporates the Gaussian prior over the weight parameter vector w. The learning task is then reduced to the optimization over the parameters (in the non-probabilistic approach) or to the posterior estimation

[1] http://decaf.berkeleyvision.org/, http://caffe.berkeleyvision.org/

(in the probabilistic approach). Given the parameter vector w, the future prediction on a new data point $\bar{x}$ is independent of the training data, $f(\bar{x}) = \langle w, \bar{x} \rangle$. Hence, the parameters value capture everything about the training data that is relevant for predicting the future data. This approach, however, may not always be practical, as illustrated in Figure 2.11 (Data A and Data B), where the data distribution cannot be defined in terms of a finite set of parameters.

Statistical models with a potentially infinite number of parameters are called *non-parametric*. It is important to note that non-parametric property does not refer to models that have no parameters, but to models with a number of parameters that is not fixed but changes with the amount of available data. As illustrated in Figure 2.11, non-parametric models, the k-NN and the non-linear SVM classifiers, can adapt to the non-linear structure of Data A and Data B, whereas the linear SVM classifier failed. On the one hand, because of this property the non-parametric models are more flexible than the parametric ones. On the other hand, because the non-parametric models do not have an explicit parametric representation, to make predictions on new data points we need to have access to the training data (for the k-NN classifier all training data points have to be stored and for the non-linear SVM classifier the support vectors have to be stored). As it will be shown in later chapters, in practice, the computational complexity of inference for non-parametric methods will scale with the number of training data points instead of the number of parameters. Refer to our Table 2.1 for examples of parametric and non-parametric models for classification and feature learning tasks (when the output space is a discrete space).

Model	Task	Type
Linear SVM (Chapters 2, 5)	classification	parametric
Logistic Regression (Chapter 2)	classification	parametric
k-NN	classification	non-parametric
SVM with squared exponential kernel	classification	non-parametric
Autoencoder (Chapter 3)	feature learning	parametric
Indian Buffet Process (Chapter 4)	feature learning	non-parametric
Gaussian Processes(Chapter 6)	classification	non-parametric

Table 2.1: Examples of parametric and non-parametric models described in this thesis for object classification and image feature learning tasks.

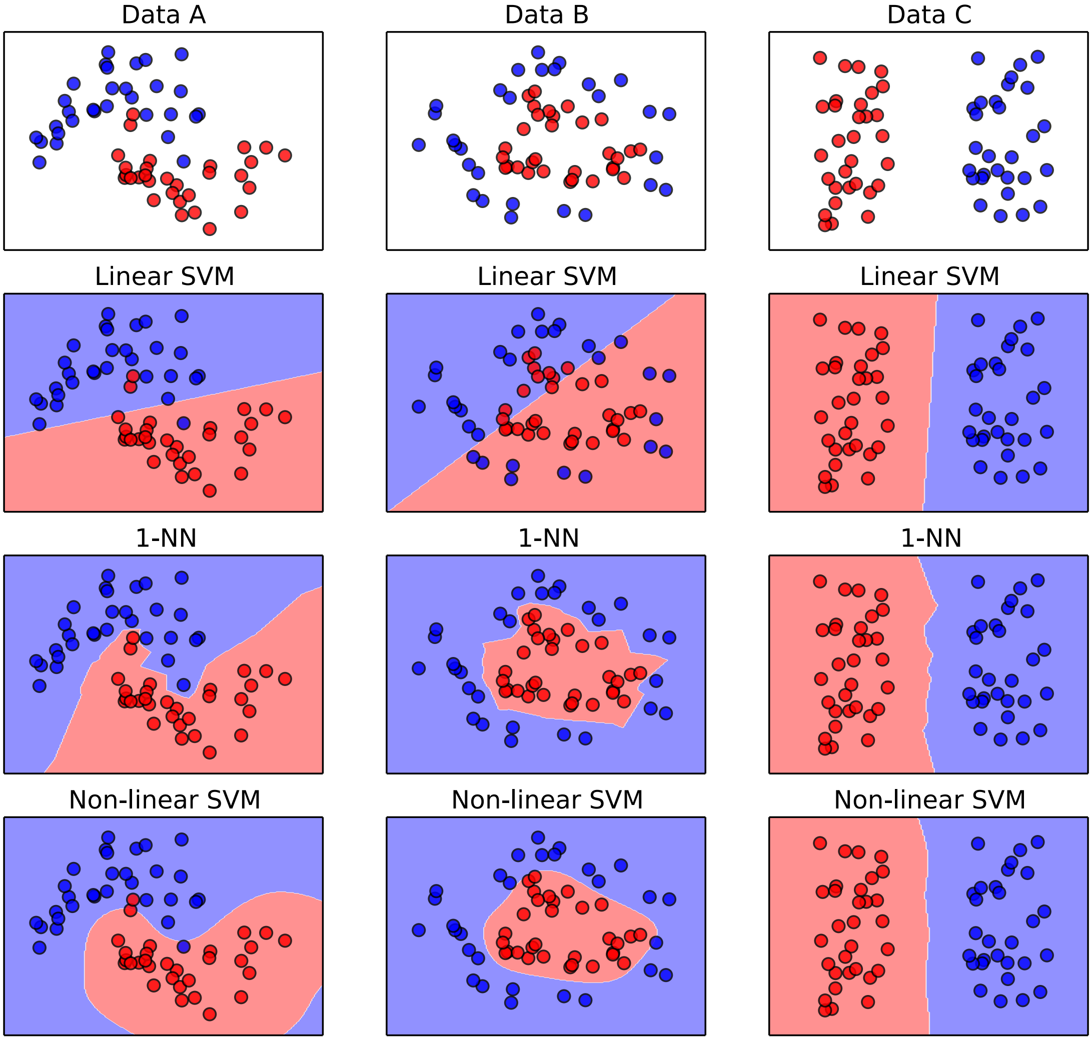


Figure 2.11: First row: Three different datasets for learning a binary classifier that can distinguish between red and blue samples. Here, we have $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{Y} = \{\text{blue}, \text{red}\}$. Second row: In the parametric approach, we fix apriori the form of the classifier $f : \mathcal{X} \to \mathcal{Y}$ to be, for example, a linear model: $f(x) = \langle w, x \rangle$, where w is the vector of parameters. In all three datasets the decision boundary of the classifier ($f(x) = 0$) can only be linear. In an ideal situation when red and blue samples are linearly separable (Data C), the linear parametric approach trained with the SVM model learns the boundary correctly. If the true relationship is not known apriori or cannot be described using a finite set of parameters, this approach may fail. For example, in Data A and Data B, non-parametric models with a potentially infinite number of parameters might be required to recover the functional relationship. Third-forth row: Non-parametric learning models, for example, k-NN and non-linear SVM, can adjust to the complexity of the training data, and adapt their decision boundaries to the complex data structure (two half-moons in Data A, two concentric circles in Data B).

Part I

Mid-level Discriminative Feature Learning with Attributes: Parametric and Non-parametric Views

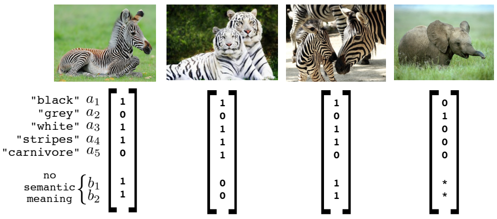
All models are wrong, but some are useful.
George Box

Chapter 3

Augmented Attributes: Parametric View

In this chapter, we propose a parametric learning method to infer a mid-level feature representation that combines the advantage of semantic attribute representations with the higher expressive power of non-semantic features. The idea lies in augmenting an existing attribute-based representation with additional dimensions for which an autoencoder model is coupled with a large-margin principle. This construction allows a smooth transition between the zero-shot regime with no training examples, the unsupervised regime with training examples but without class labels and the supervised regime with training examples and with class labels. The resulting optimization problem can be solved efficiently, because several of the necessity steps have closed-form solutions. Through extensive experiments we show that using the augmented representation is advantageous to using the attribute representation alone for solving the object classification task.

This chapter is based on:



V. Sharmanska, N. Quadrianto, C.H. Lampert:
Augmented attribute representations,
ECCV 2012, Firenze, Italy.

3.1 Introduction

In this chapter, we build on the attribute-based models for zero shot learning as described by Lampert et al. (2009) and extend it transparently to the case when few training examples are given (*small shot*), either with class annotation (*supervised*), or without it (*unsupervised*). The underlying idea is to extend the attribute representation with additional discriminative features, which are not necessarily semantic by themselves, but that augment the semantic features minimally in the sense that they offer additional representative power where necessary, and only there.

Figure 3.1 illustrates this concept: assume we are given semantic representations $(a_1, \dots, a_n)$, $n = 5$ in this example, for three object classes *zebra*, *white tiger* and *elephant*. As zebras and white tigers differ only in one entry in this representation, they will easily be confused when performing zero-shot classification with an imperfect, image-based attribute predictor. The representation of elephants, on the other hand, is clearly distinct from the other two classes, and classification errors are unlikely for them. The objective of our work is to reduce the total risk of misclassifications by learning an augmentation of the attributes with features $(b_1, \dots, b_m)$, which are learned automatically, even if this causes them not to be semantic anymore ($m = 2$ in our example). Specifically, we obtain values $(b_1, \dots, b_m)$ for each image by enforcing a large-margin criterion: the distance between representations of any pair of images of different classes should differ by at least a constant (here 3). As a result, different values are chosen for $(b_1, \dots, b_m)$ for the zebra images than for the white tiger image. For the elephant, the semantic representation alone is already sufficient to enforce the distance criterion to the other classes. Therefore, no specific values for $(b_1, \dots, b_m)$ are enforced.

To implement the above intuition, we rely on two successful concepts for learning the image representation: the *autoencoder* framework (Hinton & Salakhutdinov, 2006) and the *large margin* concept (Weinberger & Saul, 2009). The autoencoders follow a generative approach to learning an intermediate representation by identifying features that allow reconstruction of the image representation with only a small error. In the large margin nearest neighbor framework, we learn a representation in a discriminative way by trying to reduce the nearest neighbor classification on a training set in a robust way. In the rest of the chapter, we formalize these concepts and formulate them as a joint optimization problem over projection matrices. We show how to solve the optimization problem using alternating optimization in which some parts have efficient closed form solutions.

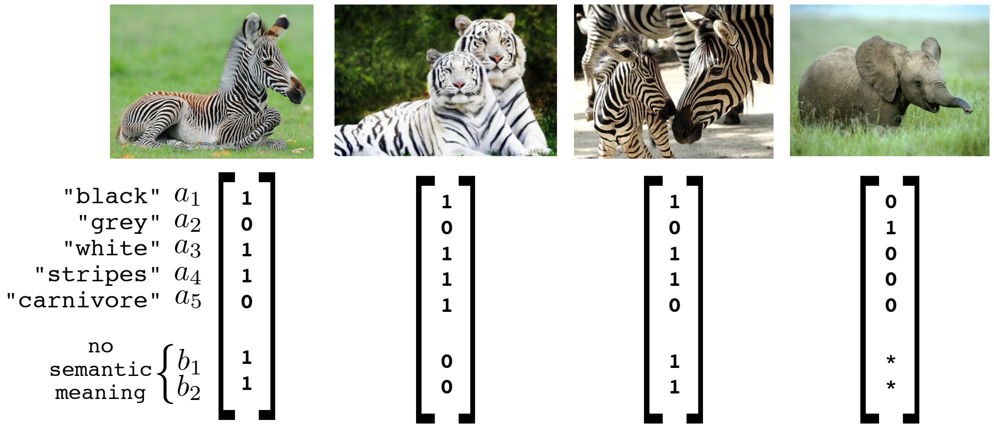


Figure 3.1: Proposed hybrid representation: a fixed semantic $(a_1, \dots, a_n)$ part is augmented by a non-semantic $(b_1, \dots, b_m)$ part, where the latter is learned by enforcing a large margin separation criterion between classes. In this example we use $n = 5$ and $m = 2$. See Section 3.1 for a detailed description.

We perform an experimental evaluation on the *Animals with Attributes* dataset, which shows that the learned hybrid representations improve over the representation purely in terms of semantic attributes when additional training data is available.

3.2 Learning to Augment Features

For the rest of this chapter we assume the following: we are given N images in a d-dimensional feature representation, $x_1, \dots, x_N$, for example, a *bag-of-visual-words*, from which we form a *data matrix* $X = (x_1, x_2, .., x_N) \in \mathbb{R}^{d \times N}$. Each $x_i \in X$ has a known attribute representation $a_i \in \mathcal{A}$, e.g. obtained from an existing set of attribute classifiers, such as (Lampert et al., 2009, 2013) discussed in Section 2.1.3. Our goal is to augment a_i with a non-semantic $b_i \in \mathcal{B}$, forming a hybrid representation $[a_i, b_i] \in \mathcal{AB}$, where $[\cdot]$ denotes the concatenation of vectors and $\mathcal{AB} = \mathcal{A} \times \mathcal{B}$. From the new, hybrid representation, we expect better properties than from the semantic part alone with respect to a target task. For simplicity, in this chapter we consider only a binary representation for the semantic attribute space $\mathcal{A} = \{0, 1\}^n$, and binary or probabilistic representations for the non-semantic space $\mathcal{B} = [0, 1]^m$, and we assume that the target task is nearest-neighbor based object classification.

In learning the augmented representations, we consider two scenarios: *unsupervised* and *supervised*. The unsupervised case is applicable whenever we have

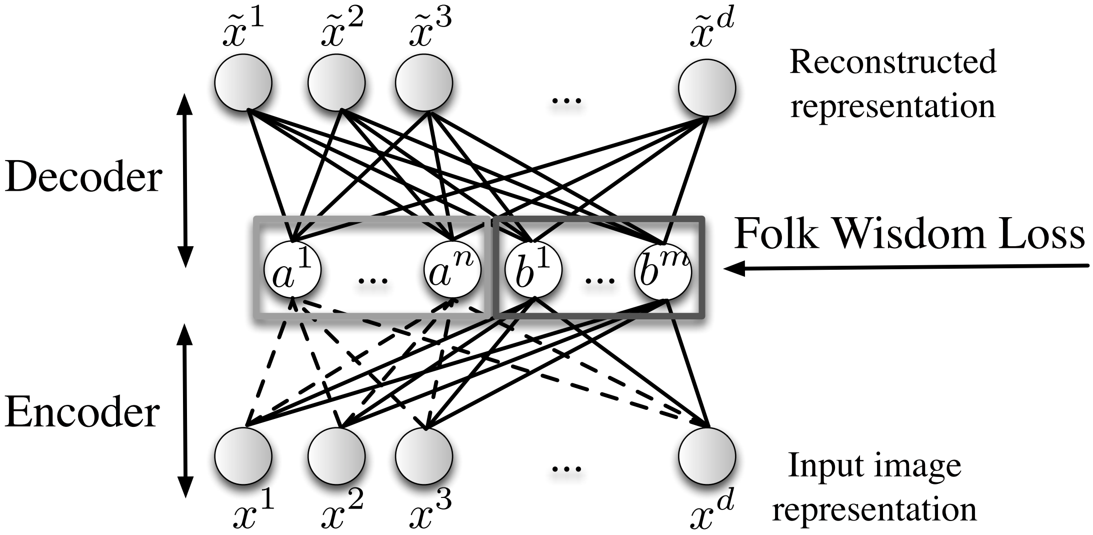

$\tilde{x}^1$ $\tilde{x}^2$ $\tilde{x}^3$ ... $\tilde{x}^d$
Reconstructed representation
Decoder
Folk Wisdom Loss
a^1 ... a^n b^1 ... b^m
Encoder
Input image representation
x^1 x^2 x^3 ... x^d

Figure 3.2: Autoencoder model for learning hybrid representations: input image $x \in \mathbb{R}^d$, (encoded) hybrid representation $[a, b] \in \mathbb{R}^{n+m}$, (decoded) reconstructed image $\tilde{x} \in \mathbb{R}^d$. The reconstruction criterion guides the learning, and the folk wisdom principle influences good discrimination between classes in the augmented attribute space.

training examples, regardless of whether we know their labels, whereas for the supervised case, we need to know the class labels. It will become clear from the description that a *semi-supervised* case that combines properties of the unsupervised and supervised can easily be obtained by a suitable choice of loss function, but we do not explore such an option in this chapter.

3.2.1 Unsupervised Learning of a Feature Space Augmentation

As the main idea in learning the augmenting features in an unsupervised way, we use the *autoencoder* principle. In general, the autoencoder aims to find a latent representation for a set of data that 1) is low-dimensional and, therefore, compact, and 2) preserves as much of the information in the original input signal as possible. This is achieved by forming a two-layered construction, in which a first layer *encodes* the input data into the latent representation, and a second layer *decodes* this representation back into the original data space. Each of the layers is parameterized by a weight matrix, and training the autoencoder means to identify parameters for both layers such that the overall reconstruction error for a set of training examples is minimized. Intuitively, a representation that allows good reconstruction of the input sample has captured more of the contained information than one that does not.

In our case, we are interested not in any ad-hoc latent representation, but we want to augment the existing semantic attributes. We achieve this by making the attribute vector a_i, a fixed part of the latent representation for any x_i, and learning an encoding only for the second part, b_i. In the decoding layer, we try to reconstruct x_i from the joint $[a_i, b_i]$, which has the effect that the b_i representation only needs to encode the information that a_i lacks; see Figure 3.2. Consequently, we have found a simple way to factorize the information in x_i into a semantic part in $\mathcal{A}$, and an additional, potentially non-semantic, part in $\mathcal{B}$.

Encoding function. As described above, the encoder function, e, maps an input $x \in \mathbb{R}^d$ to the latent space $\mathcal{AB}$. As the first, semantic, component $a \in \mathcal{A}$ is obtained from a separate method for attribute-prediction, we only parameterize the second, non-semantic component as $b = \sigma_e(W_B x)$, where $W_B \in \mathbb{R}^{m \times d}$ contains all parameters, and $\sigma_e(z) = \frac{1}{1+\exp(-z)}$ is a sigmoid non-linearity which we apply component-wise to ensure that the latent layer takes values in a range comparable with the binary-valued a. Together, we write

$$e(x) \;=\; [a, b] \;=\; [a,\; \sigma_e(W_B x)] \tag{3.1}$$

Decoding function. The decoder function $g : \mathcal{AB} \to \mathbb{R}^d$ aims at reconstructing the image in its original input space $\mathcal{X}$ from the latent space $\mathcal{AB}$. We assume the following linear form:

$$g([a, b]) \;=\; U[a, b] \tag{3.2}$$

parameterized by a matrix, $U \in \mathbb{R}^{d \times (n+m)}$, which we decompose as $U = [U_A, U_B]$ with $U_A \in \mathbb{R}^{d \times n}, U_B \in \mathbb{R}^{d \times m}$. To simplify notation, we denote the result of first encoding x then decoding it again by $\tilde{x}$. For the complete data X, we can write this as

$$\tilde{X} = U_A A + U_B B \tag{3.3}$$

where $A \in \mathcal{A}^N$, and $B \in \mathcal{B}^N$ are the encoded representations of the data X.

Reconstruction loss. The reconstruction loss measures the loss incurred by mapping the input data to the latent space and then reconstructing the input from the latent space. As such, it can be used to judge the quality of a choice of parameters W_B and U. We follow the usual choice for real-valued $x \in \mathbb{R}^d$ and use a squared error loss (Vincent et al., 2010) that has the form:

$$L_R = \sum_{i=1}^{N} \|x_i - \tilde{x}_i\|^2 \;=\; \|X - \tilde{X}\|^2_{Fro} \tag{3.4}$$

where $\|\cdot\|_{Fro}$ denotes Frobenius matrix norm.

3.2.2 Supervised Learning of a Feature Space Augmentation

If we have access to ground truth annotation during the learning phase, we can improve the augmented representation by adding an additional loss term that more directly reflects the object categorization task than the reconstruction loss.

Folk Wisdom Loss. This loss term is inspired by the intuitive principle “stay close to your friends and stay away from your enemies”. We can incorporate this loss for learning in the latent space $\mathcal{AB}$, because in a supervised setup a natural friendship (enemy) relation between samples is given by having the same (different) class labels. The folk wisdom loss (Quadrianto & Lampert, 2011) then directly reflects the idea that we would like to make few mistakes in nearest neighbor classification.

The idea of preserving the friendship while projecting the data to the latent space was earlier described by Weinberger & Saul (2009), where the authors showed how to learn a linear transformation over the data such that k-nearest neighbors belong to the same class, while examples from different classes are separated by a large margin. In our work we rely on the *large margin nearest neighbor (LMNN)* formulation that they propose. First, for each sample we identify a set of friends and non-friends based on their class label. We use the notation $i \sim j$ to indicate that x_i and x_j are friends, and the notation $i \not\sim k$ to indicate that x_i and x_k are non-friends. The folk wisdom loss can then be formalized as:

$$L_{FW} = \sum_{i \sim j} d^2_{\mathcal{AB}}(x_i, x_j) + \sum_{\substack{i \sim j,\\ i \not\sim k}} \max\left\{0, C + d^2_{\mathcal{AB}}(x_i, x_j) - d^2_{\mathcal{AB}}(x_i, x_k)\right\}, \tag{3.5}$$

where $d_{\mathcal{AB}}$ denotes the Euclidean distance in the $\mathcal{AB}$ space, i.e. $d^2_{\mathcal{AB}}(x_i, x_j) = \|[a_i, b_i] - [a_j, b_j]\|^2 = \|a_i - a_j\|^2 + \|b_i - b_j\|^2$. The first term in (3.5) penalizes large distances between objects of the same class. The second term penalizes small distances between objects of different classes, i.e. each sample is enforced to be C-units further from its non-friends than from its friends, where C is an application dependent parameter. We set it to be the median of the square distance between classes in the $\mathcal{A}$ space, where each class is represented by the mean over training samples that belong to this class.

3.2.3 Regularized Risk Functional

To avoid overfitting, especially in the regime when little data is available, we introduce regularizers on all parameters:

$$\Omega(W_B) = \|W_B\|^2_{Fro}, \quad \Omega(U_A) = \|U_A\|^2_{Fro}, \quad \Omega(U_B) = \|U_B\|^2_{Fro} \qquad (3.6)$$

In combination, we obtain the following regularized risk functional for learning the hybrid attribute representations:

$$L(W_B, U) = L_R(W_B, U) + \eta L_{FW}(W_B) + \alpha\Omega(U_A) + \beta\Omega(U_B) + \gamma\Omega(W_B) \qquad (3.7)$$

where we have made the dependence of the loss terms on the unknowns W_B and U explicit. The objective function expresses the properties we expect from the latent representation: 1) it should be compact (automatic, because $\mathcal{A}$ and $\mathcal{B}$ are low-dimensional), 2) it should retain as much information as possible from X (enforced by L_R), 3) it should have higher discriminative power than $\mathcal{A}$ alone (enforced by the folk wisdom loss L_{FW}) and 4) it should generalize from X to unseen data (enforced by the regularization). The trade-off variables η, α, β and γ control the relative influence of the aspects 2)–4). Setting $\eta = 0$, we obtain a formulation that allows unsupervised feature learning, because only the folk wisdom loss requires knowledge of labels (through the definition of friends and non-friends). Even though we do not enforce property 3) in this case, we can still hope for better classification performance, because property 2) will cause additional information to be present in the hybrid representation than in the semantic representation alone.

3.3 Optimization

Minimizing the expression (3.7) is a non-convex optimization problem. The reconstruction loss is non-convex with respect to the weight matrix W_B due to the nonlinear transformation in the encoder function (3.1). The folk wisdom loss is non-convex with respect to the weight matrix W_B when optimizing the non-friends relation part, i.e. the second term in (3.5). One potential approach to solve the optimization problem is to use alternating optimization with respect to one weight matrix at a time while fixing the others.

The key observation is that when the weight matrices W_B, U_B in (3.7) are fixed, we can obtain the closed form solution for updating the matrix U_A by solving a ridge regression problem. The closed form solution to:

$$\min_{U_A \in \mathbb{R}^{d \times n}} \|U_A A + U_B B - X\|_{Fro}^2 + \alpha \|U_A\|_{Fro}^2 \tag{3.8}$$

for fixed X, and U_B, A, B is:

$$U_A = (X - U_B B) A^T (A A^T + \alpha I_n)^{-1} \tag{3.9}$$

where I_n is the identity matrix of size n, and αI_n reflects the regularization on the matrix U_A. Essentially, U_A aims to capture the information, which was lost by decoding from the latent space $\mathcal{B}$, i.e. $X - U_B B$. By analogy, for fixed X, and U_A, A, B, we obtain the closed form solution for updating the matrix U_B:

$$U_B = (X - U_A A) B^T (B B^T + \beta I_m)^{-1} \tag{3.10}$$

where I_m is the identity matrix of size m and βI_m regularizes the matrix U_B.

For W_B the non-linearity of encoding prevents a closed form expression. After updating U_A, U_B several existing optimization solvers can be used for updating the matrix W_B. In our case, we use Broyden-Fletcher-Goldfarb-Shanno gradient descent method with limited-memory variation (L-BFGS). In the unsupervised learning, we solve:

$$\underset{W_B}{\operatorname{argmin}} \quad L_R(W_B) + \gamma \Omega(W_B), \tag{3.11}$$

and the gradient with respect to W_B has the form:

$$\nabla_{W_B} = 2 U_B^T (\tilde{X} - X) \odot B \odot (1 - B) X^T + 2 \gamma W_B, \tag{3.12}$$

where $\odot$ denotes the Hadamard or the entrywise product of the matrices, 1 is a matrix of ones and $B \odot (1 - B)$ appears when computing the gradient of the

Algorithm 1 Learning Feature Augmentation

```
Input Training set $X$ with attribute representation $A$
Input Regularization constants $\alpha$, $\beta$, $\gamma$
Input If supervised: training labels $Y$, regularization constant $\eta$
repeat
  $U_A \leftarrow$ update from closed form solution (3.9)
  $U_B \leftarrow$ update from closed form solution (3.10)
  if supervised then
    Randomly pick friend and non-friend pairs based on class label $Y$
    $W_B \leftarrow \text{argmin}_{W_B} \quad L_R(W_B) + \eta L_{FW}(W_B) + \gamma\Omega(W_B)$
  else
    $W_B \leftarrow \text{argmin}_{W_B} \quad L_R(W_B) + \gamma\Omega(W_B)$
  end if
until convergence, or for a maximal number of iterations
Return $W_B, U_A, U_B$
```

logistic sigmoid. In the supervised learning, in order to update W_B, we solve:

$$\underset{W_B}{\text{argmin}} \quad L_R(W_B) + \eta L_{FW}(W_B) + \gamma\Omega(W_B), \tag{3.13}$$

and the gradient has an additional term from the folk wisdom loss L_{FW}. The folk wisdom loss (3.5) is computed based on the pairwise distance between samples, such as $d^2_{\mathcal{AB}}(x_i, x_j)$ for friends $i \sim j$ and similar for non-friends $i \not\sim j$. For clarity, we only specify the gradient form for pairwise distance $d^2_{\mathcal{AB}}(x_i, x_j)$ between x_i and x_j here:

$$2(b_i - b_j) \odot b_i \odot (1 - b_i)x_i^T + 2(b_j - b_i) \odot b_j \odot (1 - b_j)x_j^T, \tag{3.14}$$

where b_i, b_j are the encoded representation of samples x_i and x_j in the $\mathcal{B}$ space, 1 is a vector of ones.

Note, we do not need to run full L-BFGS procedure at each pass to update the matrix W_B, few steps only. To speed up the training, we use 2 steps in our experiments. While training the autoencoder, we expect U_A to vary less strongly, because A is fixed, whereas B is learned, so we can accelerate the optimization by updating the matrix U_A less frequently, e.g. at every t-th iteration. The proposed training procedure is summarized in the Algorithm 1.

3.4 Related Work

A characteristic aspect of our work is that we want to extend the set of semantic attributes. Prior approaches were aimed at preserving the property that all attributes have a semantic meaning. Therefore, they required additional human knowledge, obtained either by external field expertise (Lampert et al., 2009, 2013), the analysis of textual sources (Berg et al., 2010), or by interaction with human users (Parikh & Grauman, 2011a). By adopting a hybrid model in which semantic and non-semantic attributes occur together, there is no need for such an additional source of human input.

Our approach of using an autoencoder to find a useful feature representation follows the recent trend of learning feature representations in an unsupervised way (Welling et al., 2005; Hinton & Salakhutdinov, 2006; Ranzato et al., 2007). Splitting the feature representation of the autoencoder into heterogeneous groups has been discussed by Gregor & LeCun (2010); Hinton et al. (2011), among others. However, in our case factorization of the autoencoder is different due to asymmetry of the semantic and non-semantic parts. The semantic part reflects the human-interpretable attributes and is fixed, whereas the non-semantic part is learned to overcome the shortcomings of the semantic attributes at the expense of not being interpretable.

Learning a lower dimensional feature representation with an autoencoder model is also related to the principal component analysis (PCA). PCA learns an orthogonal projection of the data points onto a lower dimensional subspace, such that the variance of the projected data is maximized. Alternatively, PCA can be viewed as an autoencoder model with linear encoder and decoder, so that the squared error reconstruction loss between a given sample and the sample reconstructed by the autoencoder is minimal (Bishop, 2006). The non-linear version of PCA is closely related to our autoencoder model with a non-linear decoder. It learns, however, a representation without incorporating the information about semantic attributes and without the folk wisdom principle.

To our knowledge, our work is the first that explores the idea of autoencoders jointly with the large margin nearest neighbor principle (Weinberger & Saul, 2009). Other approaches to preserve class structure during feature learning exist, however. For example, Salakhutdinov & Hinton (2007b) train a deep network and afterwards use Neighborhood Component Analysis (NCA) to improve the k-NN classification accuracy. NCA is also the basis of the work by Tang et al. (2010) that aims at learning a feature representation which is suitable for the object categorization with a small training set. Its focus, however, does not lie on leveraging existing attribute annotation, but to make optimal use of the available training examples by constructing many virtual training sets.

Recently, there has been much interest in learning discriminative attributes (Rastegari et al., 2012; Yu et al., 2013; Mittelman et al., 2013), also known as mid-level features, which are both detectable and discriminative. In this way, one could compensate for unreliable predictions from the semantic representation, and aim directly for good classification performance. While attributes have received much attention, most of the existing work studies either zero-shot learning with no training examples (Lampert et al., 2009; Palatucci et al., 2009; Russakovsky & Fei-Fei, 2010; Akata et al., 2013; Jayaraman & Grauman, 2014), or the more classical case of many training examples, which allow training of discriminative probabilistic or maximum-margin classifiers (Wang & Mori, 2010; Mahajan et al., 2011; Patterson & Hays, 2012). Our interest lies in the case in-between, where some, but few examples per class are available. It appears wasteful to use zero-shot learning in this case, but it has also been observed previously that discriminative techniques tend to fail in this regime (Fei-Fei et al., 2006; Wolf et al., 2008), unless specific transfer learning techniques can be incorporated (Tommasi et al., 2010; Rohrbach et al., 2010, 2011; Mensink et al., 2014).

3.5 Experiments

We use the *Animals with Attributes (AwA)* dataset (Lampert et al., 2009, 2013) described in our Section 2.1.3. The dataset consists of 30475 images, and each image has a category label attached to it which corresponds to the animal class. There are 50 animal classes in this dataset. The dataset also contains semantic information in the form of an 85-dimensional Osherson's (Osherson et al., 1991) attribute vector for each animal class. Following the studies of Lampert et al. (2009, 2013), we use 24295 images from 40 classes to learn the semantic attribute predictors. From the remaining 10 classes, we take 4680 images for training the autoencoder model, and use the rest of the 1500 images, i.e. 150 from each class, to test the performance of the model. We repeat the whole procedure of training and test 5 times to get better statistics of the performance. In our experiments, we use the representation by SURF descriptors (Bay et al., 2008) provided with the dataset and referred to as original feature representation. We further normalize the features to have zero mean and unit standard deviation for each dimension.

Algorithms. We analyze two variants of our proposed method: in the first variant the hybrid representation is learned in an unsupervised way via the autoencoder architecture while minimizing *only* the reconstruction loss; in the second variant the hybrid image representation is learned with additional supervision via the folk wisdom principle. The supervision comes from friendship and non-friendship relations based on class label. We define friends to be samples coming from the same class and non-friends to be from different classes. To keep the terms in balance we sample the pairs such that the cardinality of the non-friends set has the same order as the friends set. We find that 3 friends and 3 non-friends for each sample is a good balance between computational efficiency and accuracy performance. Further, we stochastically change the pairs of friends and non-friends as the optimization solver cycles through the steps.

Evaluation metric. We use k-nearest neighbor classification accuracy as the evaluation metric with $k = 3$. We compare the classification performances of our proposed unsupervised and supervised hybrid representations to baselines using original bag-of-visual-words image representation and pure semantic attribute representation (Lampert et al., 2009, 2013). The semantic attribute representation is the established method that is able to predict a class label without seeing any examples of the class and thus shows significant advantage in the small shot setting over bag-of-visual-words representation. However, the latter, in principle, can benefit from the availability of more training data points.

Attribute predictors. For the semantic attribute baseline, we learn a predictor for all of the 85 attributes based on samples and semantic information on the set of 40 animal classes. We use an ℓ_2-regularized logistic regression model with the 2000-dimensional bag-of-visual-words image representations.

Model selection. We also perform a cross validation model selection approach in choosing the regularization parameters for our unsupervised learning variant, α, β and γ, and then for our supervised variant η given the trade-off parameters of the unsupervised from the previous model selection.

Results. We demonstrate the performance in a small shot setting, when we have $2, 4, 6, 8, 10, 20, 30$ number of training samples per class. These are the only samples used to train the autoencoder model and to assess k-nearest neighbor performance. We randomly select the required number of training samples from the available training samples per class, which in total is 4680 images. We are interested in exploring how the latent attribute space $\mathcal{AB}$ benefits when augmenting the $\mathcal{A}$ with only few dimensions, and up to the case when we double the

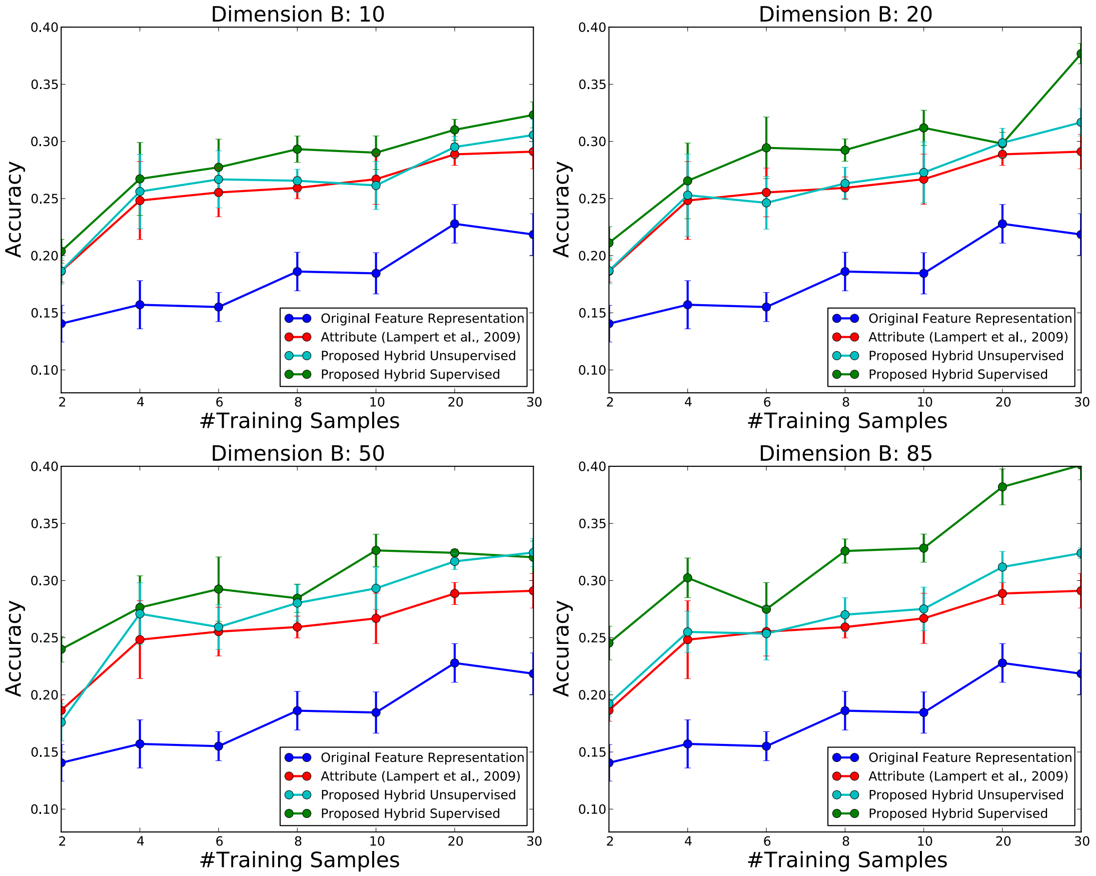


Figure 3.3: Augmented attribute representations using proposed hybrid unsupervised and supervised methods. Comparison with baseline methods using original feature representation (SURF), and predicted attribute representation (Lampert et al., 2009, DAP). We use accuracy as performance measure, and plot mean and standard deviation results over 5 runs. View of the classification performance across dimensions of the latent space $\mathcal{B}$.

size of the latent space representation compared with semantic attribute space $\mathcal{A}$. Guided by this interest, we augment the semantic attribute space $\mathcal{A}$ with a $m = 10, 20, 50, 85$ dimensional space $\mathcal{B}$, and we study the performance of the methods across dimensions. The results are summarized in Figure 3.3.

Our experiments show that the categorization performance of the proposed unsupervised variant is always better or on par with semantic attribute representation. In a majority of cases we observe that our supervised variant shows an increased improvement over the unsupervised counterpart. As expected, in the small training samples regime, performance of both proposed hybrid models and semantic attribute representation are significantly better than the original representation.

Looking at Figure 3.3 more closely for $m = 10$ dimensional space $\mathcal{B}$, we can see that our hybrid unsupervised model shows only minute improvements over semantic attributes representation when augmenting with only few dimensions. This is expected as the effect of few additional dimensions is overwhelmed by a strong semantic prior, which is by itself already discriminative enough. In comparison, at higher dimensions such as $m = 50$, the unsupervised variant becomes clearly better than the semantic attribute representation alone. When we double the size of the latent space, i.e. $m = 85$, we observe saturation in improvements when using a small number of training samples, due to highly redundant information in the encoded $\mathcal{B}$ space. As the number of samples grows, the trend of increased recognition performance continues.

We also observe a more positive effect of incorporating the folk wisdom principle into the learning of latent attribute space when more samples become available. The proposed hybrid supervised representation integrates the knowledge about object classes by enforcing a large margin between images from different classes. This can be seen also in the visualization of the AB representation for one of the runs with $m = 85$ dimensional $\mathcal{B}$ space and $N = 30$ training samples per class. Please, refer here to our Figure 3.4.

From the visualization, we observe certain structure in the representation learned in the $\mathcal{B}$ space with respect to different classes. We credit this to the influence of the large margin between images from different classes encoded in the folk wisdom criterion, which forces certain attributes to be on/off across the whole class (vertical stripes) supporting class discrimination.

The margin concept also helps to improve recognition performance at low dimension of the $\mathcal{B}$ space. However, we note that in some cases the performance of our supervised method only matches the unsupervised counterpart. Such cases can be seen in Figure 3.3 at dimension $m = 20$, and at dimension $m = 50$. This is caused by the sensitivity of the method to the model selection on the trade-off variables between reconstruction loss and folk wisdom loss.

We also consider the case where we are given the *ground truth* semantic attributes of the input images for training the autoencoder model. One could hope that this leads to better results, as it eliminates the effect of noisy predictions at training time. On the other hand, using the ground truth attributes prevents the training stage from learning to compensate for such errors. The results of these experiments for $m = 10$ and $m = 85$ dimensional space $\mathcal{B}$ are shown in Figure 3.5. Note, because the ground truth attributes are defined per class, the semantic

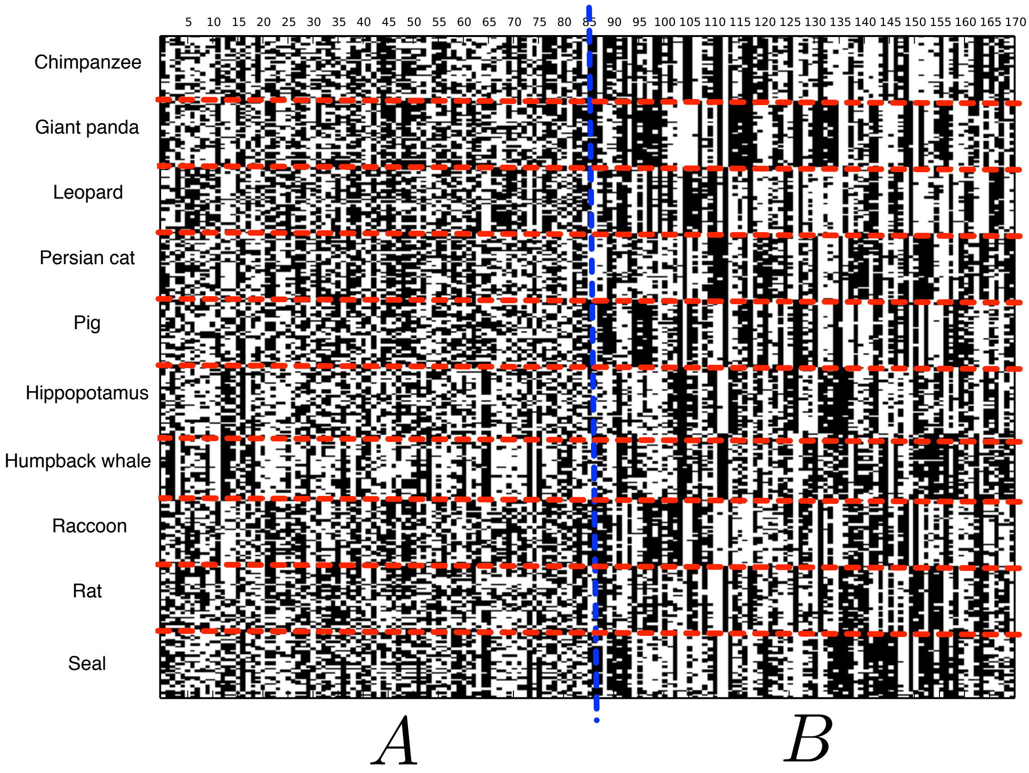


Figure 3.4: Visualization of the augmented attribute representation learned by our proposed hybrid supervised method with $N = 30$ training samples per class (vertical axes). $n = 85$ dimensional $\mathcal{A}$ space is augmented with $m = 85$ dimensional $\mathcal{B}$ space (horizontal axes). For clarity, we group the training samples from the same class together (between red lines) and annotate the class name accordingly. The matrix encodes binary attribute representation AB of the augmented space, where we threshold the sigmoid scores in B at 0.5.

attribute representation of the image directly corresponds to its class representation, and therefore prevents a completely unsupervised setting. Moreover, the nearest neighbor performance using semantic attribute representation (red line) does not gain from more training data because all examples of one class have the same ground truth attribute representation. We observe an advantage of using the hybrid representations with and without folk wisdom loss over the baseline methods for higher dimensional $\mathcal{B}$ space, as for $m = 85$ in Figure 3.5. Similar to the case with predicted semantic attributes, augmenting the semantic attribute space only with few dimensions, as for $m = 10$ in Figure 3.5, does not give essential advantage in performance, which highlights again the discrimination power of the semantic attribute representation alone.

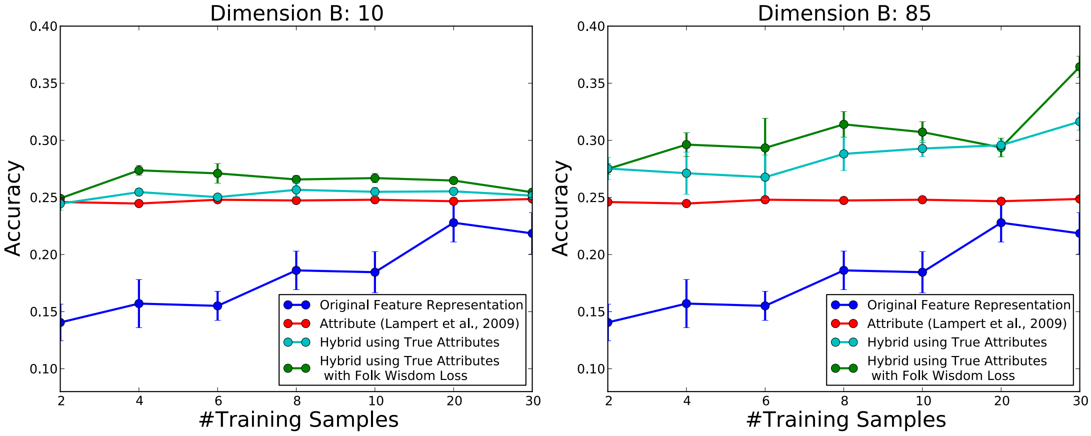


Figure 3.5: Learning hybrid representations with and without folk wisdom loss using ground truth semantic attributes. Comparison with baseline methods using original feature representation (SURF features), and ground truth attribute representation Lampert et al. (2009) (mean and standard deviation over 5 runs). View across dimensions of the latent space $\mathcal{B}$.

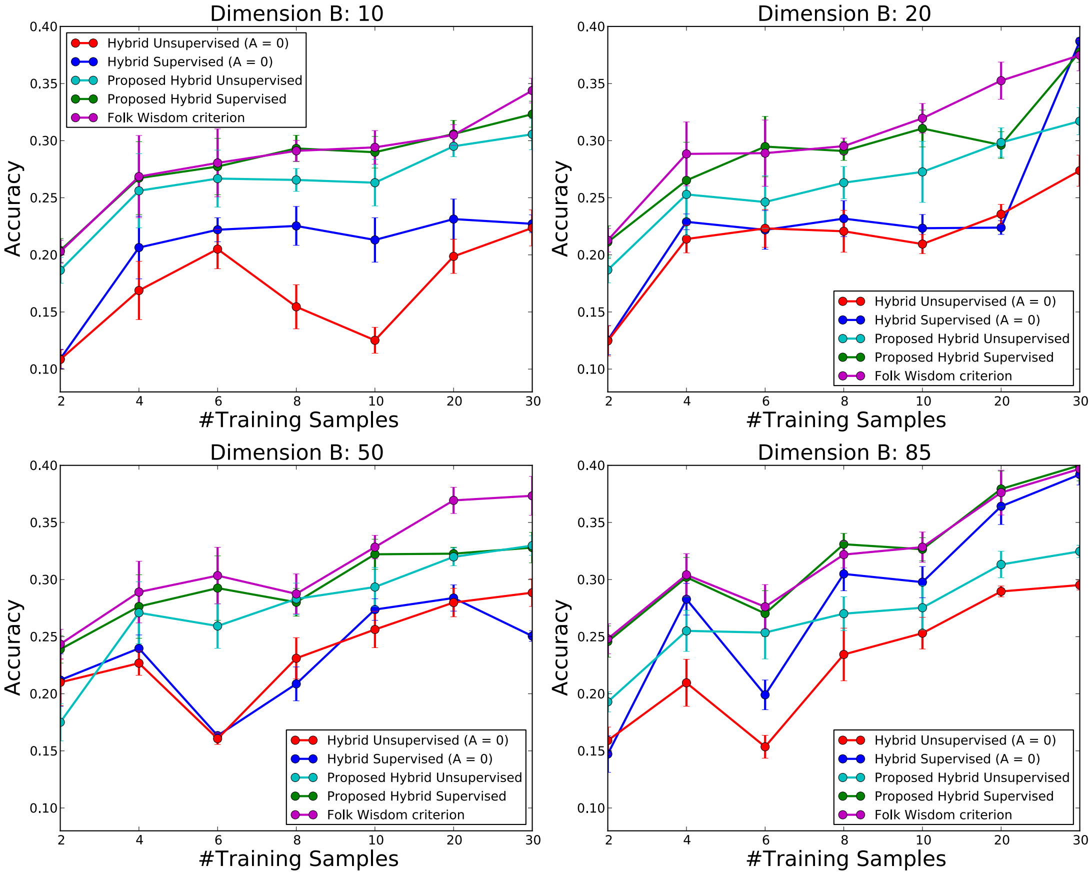


Figure 3.6: In-depth analysis of the model components. We compare the proposed methods to augment semantic attribute representations with learning feature representations without semantic attributes ($A = 0$). We also visualize the role of the folk wisdom criterion in the proposed hybrid supervised method. View across dimensions of the latent space $\mathcal{B}$ (mean and standard deviation over 5 runs).

We also provide more extensive experimental analysis of the impact of different model components in Figure 3.6. As can be seen in our setting, augmenting the semantic attributes with proposed hybrid unsupervised and supervised methods is clearly better than learning a representation "from scratch" (baselines with $A = 0$). We also illustrate the dominating role of the folk wisdom criterion over the reconstruction criterion in the proposed hybrid supervised model. In this case, the augmented attribute representations are learned using the folk wisdom criterion while eliminating the reconstruction term in (3.7).

Comparison to related work. Earlier works on object categorization for the Animals with Attributes dataset followed different experimental setups than the one we use, so numeric results are not directly comparable with ours. The original work of Lampert et al. (2009) reports classification accuracy of 40.5% in a zero-shot setup with DAP model (direct attribute prediction). However, the work makes use of multiple kernels with six different feature types, whereas we rely only on a single feature type. Note that the "Attribute" baseline we report in Figure 3.5 corresponds approximately to the DAP model. Ebert et al. (2010) performed experiments on 1– and 10–nearest neighbor classification accuracy. For the same SURF features that we use, the authors achieve 11.7% accuracy for the ℓ_2-norm and 16.4% accuracy for the ℓ_1-norm, which is comparable with the "Original Feature Representation" we report in Figure 3.3 and Figure 3.5. Tang et al. (2010) learned feature representations in a one-shot setup. Using the same combination of 6 different feature descriptors as Lampert et al. (2009), the authors report 23.7% for linear representations, 27.2% for non-linear and 29.0% for a combination of non-linear with semantic attribute features.

3.6 Summary and Discussion

In this chapter we introduced a method to augment a semantic attribute representation by additional (non-semantic) mid-level features. The main idea is to learn only the non-semantic part of the representation by an autoencoder in combination with an (optional) maximum-margin loss term, while keeping the semantic part fixed. The effect is that the additional feature dimensions overcome the shortcomings of the semantic original ones, but do not copy their behavior. We interpret the result as an orthogonal decomposition of the image features into semantic and non-semantic information.

Our experiments showed that the additional flexibility offered by the hybrid features improves the nearest neighbor classification accuracy over the purely semantic representation. In particular, they allow for a smooth transition between the zero-shot case (no training images), the unsupervised case (training images without labels) and the supervised case (training images including their labels).

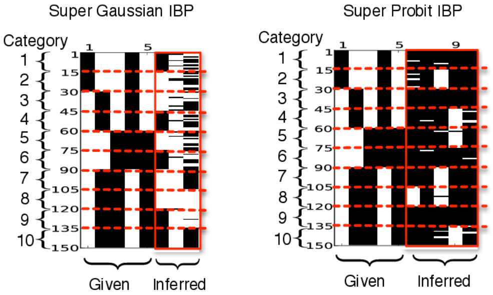
There are several limitations of the setup we chose. A first aspect is that our model requires regularization on the parameter matrices, and therefore the choice of regularization parameters. We used standard cross validation for this, but if the number of training examples is small – which is exactly the case of interest – this step can become unreliable. Instead, it could be promising to decide on free parameters using a Bayesian criterion that does not require splitting the available data into parts. A second aspect is that our model is parametric: we fixed the dimension of the augmented attribute representation *a priori*, and it does not scale as the number of data points. In more realistic setup, one would hope to have a model that can automatically decide on the dimension of the augmented attribute space from the available data. This brings us to the non-parametric aspect of learning the mid-level feature representations, which we address in the next chapter of this thesis.

Chapter 4

Augmented Attributes: Non-parametric View

In this chapter, we propose a probabilistic view on modeling discriminative non-semantic latent variables from observed data. Our model allows simultaneous inference of the number of binary latent variables and their values. The latent variables preserve the neighborhood structure of the data in the sense that objects in the same class have similar latent values and objects in different classes have dissimilar latent values. Inspired by the folk wisdom principle, we formulate the supervised latent variable problem based on an intuitive principle of pulling objects together if they are of the same type, and pushing them apart if they are not. We then combine this principle with a flexible Indian Buffet Process prior over (potentially) infinite latent variables. We show that the inferred supervised latent variables can be directly used to perform a nearest neighbor search for the purpose of classification or retrieval. We explore the application of augmenting semantic attributes, and show how to couple effectively the structure of the semantic space with continuously growing structure of the neighborhood preserving infinite latent feature space.

This chapter is based on:



N. Quadrianto, V. Sharmanska, D.A. Knowles, Z. Ghahramani:
The Supervised IBP: Neighbourhood Preserving Infinite Latent Feature Models,
UAI 2013, Seattle, USA.

4.1 Introduction

In statistical data analysis, latent variable models are used to represent components or properties of data that have not been directly observed, or to represent hidden causes that explain the observed data. In many cases, a natural representation of an object would allow each object to admit multiple latent features. This means for each data sample x we introduce a K-dimensional attribute vector z from a binary latent space, such as $\mathcal{B}$, for example. Classical statistical techniques require the number of latent features K to be fixed a priori. Lately, non-parametric Bayesian models have emerged as an elegant approach to deal with this issue by allowing the number of features to be inferred from data. One class of these models utilizes the Indian Buffet Process (IBP) prior (Griffiths & Ghahramani, 2005) to allow a potentially unbounded number of features. Almost all IBP-based statistical models are geared towards *unsupervised* latent feature learning, for example, as an exploratory tool for discovering compact hidden structures in observed data. However, in many practical settings we seek *supervised* learning of latent variables that are semantically meaningful and encode supervised information in the form of class labels, attributes, friendship relatedness, for example. We follow the same scenario as in our previous chapter and explore the supervised information in terms of friends and non-friends relationship between data points from the same/different classes. This can be directly used for nearest neighbor search and also for classification if we know the class labels of friends and non-friends samples.

Our supervised latent variable model enforces the folk wisdom principle, such that latent variables z associated with objects of the same class possess similar values, and latent variables associated with objects of different classes possess dissimilar ones. For example, two sample images of *zebra* class would have similar placement of 0s and 1s in their z representations, and samples of *zebra* and *white tiger* would have the opposite. To achieve this, we define a neighborhood likelihood function in Section 4.2 that views this criterion as *preference* relation. When coupled with a flexible prior on infinite sparse binary matrices and a data likelihood, we are able to characterize a probabilistic model for supervised infinite latent variables problems. For the data likelihood, we explore two directions: a standard linear Gaussian model, and our proposed linear probit dependent model, detailed in Section 4.3. We discuss inference in Section 4.4 and predictive distribution in Section 4.5. Finally, we present two synthetic data experiments and the application of augmenting semantic attributes in Section 4.7.

4.2 The Neighborhood Likelihood Model

We are given a set of N observed data samples $\{x_1, \ldots, x_N\} \subset \mathcal{X}$. As before, for image objects, $\mathcal{X}$ are features extracted based on the content of the image. We further assume that the supervised information is available in the form of triplets:

$$\mathcal{T} = \{(i, j, l) : \; i \sim j, \; i \not\sim l\}, \tag{4.1}$$

which contains indices of friends, i.e. samples i and j from the same category, and indices of non-friends, i.e. samples i and l from different categories. In fact, this type of supervision requires class label information only because we are interested in nearest neighbor classification as our final goal, and not object comparison or retrieval in general.

For each data point x_i, we introduce a K-dimensional vector z_i from a *binary* latent space, where $z_i^k = 1$ denotes that object i possesses feature k, and $z_i^k = 0$ otherwise, and K is inferred from data. The collection of all latent binary features z_i, $i = 1, \ldots, N$ form a matrix Z. Targeting directly our goal of learning neighborhood preserving latent space that is suitable for nearest neighbor search, we require that z_i is similar to z_j to model $i \sim j$, and z_i is dissimilar to z_l to model $i \not\sim l$. Now, we formalize the folk wisdom principle (Goldberger et al., 2004; Weinberger & Saul, 2009; Quadrianto & Lampert, 2011) for supervised learning of the latent variable representations as a preference relation.

When we observe that objects i and j are friends, while objects i and l are non-friends, we say that object i prefers object j to object l, and use a notation $j \underset{i}{\succ} l$. Let T be a $N \times N \times N$ preference tensor with entries $\{t_{jl}^i\}$, where $t_{jl}^i = 1$ whenever $j \underset{i}{\succ} l$ is observed. Let w be a $K \times 1$ *non-negative* weight vector that affects the probability of preference relations among object i, j and l. We assume that preference relations are independent when conditioned on latent features Z and weights w. Also, we assume that only the latent representation of objects i, j and l influence the tendency of i preferring j to l. With the above assumptions, the label preference likelihood function is given by:

$$\Pr(T|Z, w) = \prod_{(i,j,l) \in \mathcal{T}} \Pr(t_{jl}^i = 1 | z_i, z_j, z_l, w). \tag{4.2}$$

We will denote $\Pr(t_{jl}^i = 1 | z_i, z_j, z_l, w)$ as p_{jl}^i and define the individual preference probability as follows:

$$p_{jl}^i = \frac{1}{C} \sum_k w_k \mathbb{I}[z_i^k = z_j^k](1 - \mathbb{I}[z_i^k = z_l^k]), \tag{4.3}$$

where we make use of Iverson's bracket notation: $\mathbb{I}[P] = 1$ for the condition P is true and it is 0 otherwise, and C is the normalizing constant:

$$C = \sum_k w_k \mathbb{I}[z_i^k = z_j^k](1 - \mathbb{I}[z_i^k = z_l^k]) + \sum_k w_k (1 - \mathbb{I}[z_i^k = z_j^k])\mathbb{I}[z_i^k = z_l^k]. \quad (4.4)$$

In (4.3), the term $\sum_k w_k \mathbb{I}[z_i^k = z_j^k](1 - \mathbb{I}[z_i^k = z_l^k])$ collects the weights for all features that object i and j have but object l does not have, i.e. $z_i^k = z_j^k = 1$ and $z_l^k = 0$, and the weights for all features that object i and j do not have but l has, i.e. $z_i^k = z_j^k = 0$ and $z_l^k = 1$. Thus, the choice between two alternatives j and l from point-of-view i depends on latent features that are shared between i and j but not l. This type of preference model is inspired by the choice model of Görür et al. (2006) and is based on a standard Restle's choice model in psychology (Restle, 1961).

We take a fully Bayesian approach by treating latent variables Z and w as random variables, and computing the posterior distribution over them by invoking Bayes' theorem. We discuss the selection of prior probabilities on Z and w in detail in the next section.

4.3 The Generative Process

We want to define a flexible prior on Z that allows simultaneous inference of the number of features and all the entries in Z at the same time. Thus, we put the Indian Buffet Process (IBP) prior (Griffiths & Ghahramani, 2005) on Z:

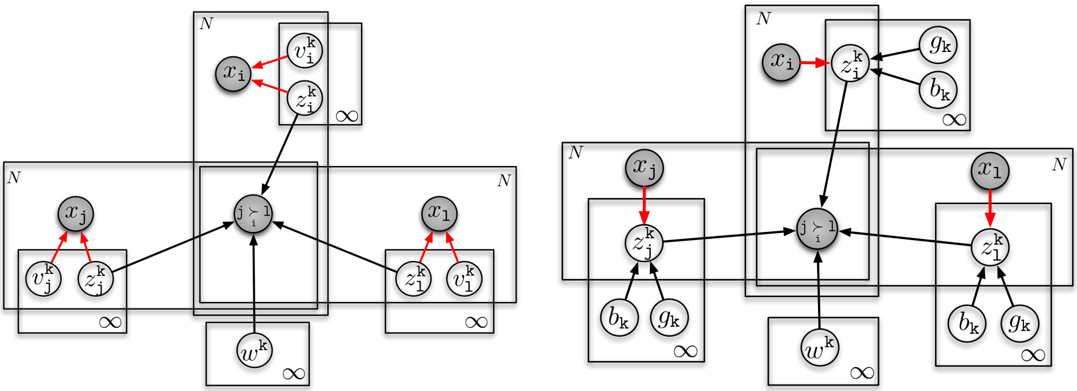
$$Z \sim \text{IBP}(\alpha), \quad \text{with parameter } \ \alpha \geq 0. \quad (4.5)$$

The IBP is a prior on binary matrices with a finite number of rows (N in our case) and an unbounded number of columns, such that with probability one, a feature matrix Z drawn from it will only have a finite number of non-zero entries. This distribution is suitable for use as a prior in probabilistic models that represent objects with potentially infinite number of features.

For the elements of w, we choose to put a Gamma distribution as a natural prior for a non-negative weight vector:

$$w_k \overset{\text{i.i.d.}}{\sim} \mathcal{G}(w|\gamma_w, \theta_w) = \frac{w^{\gamma_w - 1}\exp^{-\frac{w}{\theta_w}}}{\theta_w^{\gamma_w}\Gamma(\gamma_w)} \ \text{ with } \ \gamma_w \geq 0, \ \ \theta_w \geq 0. \quad (4.6)$$

The last required modeling part is to define the data likelihood. For this, we explore two directions: first, we use a standard linear Gaussian model which



(a) Linear Gaussian Model

(b) Linear Probit Model

Figure 4.1: Graphical model for learning our supervised infinite latent variable models based on preference relation. Left: Linear Gaussian model is used to generate the observed data based on latent features. Right: Linear probit model is used to generate latent features based on the observed data. The difference between left and right figures is encoded in the direction of red arrows modeling the dependency between data X and latent features Z. Shade indicates the observed variables.

assumes data is generated via a linear superposition of latent features. Second, we propose to make the latent features dependent on observed data via a linear probit model. We discuss both models in the next two sections.

4.3.1 Z → X Linear Gaussian Feature Model

This data generating model was initially explored for the IBP in the *unsupervised* context by Griffiths & Ghahramani (2005). In this model, for an d-dimensional input space $\mathcal{X} = \mathbb{R}^d$, the data point $x_i \in \mathbb{R}^d$ is generated as follows:

$$x_i = Vz_i + \sigma_x \epsilon, \text{ where } \epsilon \sim \mathcal{N}(\epsilon|0, I). \tag{4.7}$$

In the above, V is a real-valued $d \times K$ matrix of weights, for which we define a spherical Gaussian conjugate prior with a covariance matrix $\sigma_v^2 I$: $V \sim \mathcal{N}(0, \sigma_v^2 I)$. The generative process for our preference model with a linear Gaussian likelihood is then:

$$Z \sim \text{IBP}(\alpha); \quad V \sim \mathcal{N}(0, \sigma_v^2 I); \quad x_i|z_i, V \sim \mathcal{N}(Vz_i, \sigma_x^2 I); \tag{4.8a}$$

$$w_k \overset{\text{i.i.d.}}{\sim} \mathcal{G}(\gamma_w, \theta_w); \quad j \underset{i}{\succ} l|Z, w \sim \text{Bernoulli}(p_{jl}^i), \tag{4.8b}$$

where p^i_{jl} is preference probability defined in Equation (4.3). The hyperparameter α influences the number of non-zero features in the matrix drawn from the IBP distribution. The role of α will become clearer in the next section, where we describe the stick breaking construction of the IBP.

We can subsequently compute the posterior distribution of the latent feature matrix Z and the weights w using the conditional independence assumptions depicted in Figure 4.1(a):

$$\begin{aligned}&\Pr(Z, w|X, T) \propto \\ &\int \Pr(T|Z, w)\Pr(X|Z, V, \sigma_x)\Pr(Z|\alpha)\Pr(V|\sigma_v)\Pr(w|\gamma_w, \theta_w)dV. \end{aligned} \tag{4.9}$$

From Figure 4.1(a), we can observe that the neighborhood model with linear Gaussian data likelihood requires the latent features Z to *explain* the preference relation in given triplets (arrows from z^k_i, z^k_j and z^k_l to $j \underset{i}{\succ} l$), and to *generate* the observed data X (red arrows from z_i to x_i and similar for z_j, z_l). Modeling observed data is a hard task by itself. Instead, in the next section, we propose an IBP model that is *conditioned* on object covariate information X and let the latent features Z model only the supervised preference relation.

4.3.2 X → Z Linear Probit Dependent Model

In order to define the dependent IBP model, we first formalize the original IBP distribution. We overview its explicit construction called the stick breaking construction of the IBP (Teh et al., 2007):

$$z^k_i|b_k \sim \text{Bernoulli}(b_k); \quad b_k := v_k b_{k-1} = \prod_{j=1}^{k} v_j \tag{4.10a}$$

$$v_j \sim \text{Beta}(\alpha, 1) \quad \text{and } b_0 = 1. \tag{4.10b}$$

In this model, b_k is the probability of the kth feature z^k_i being activated, also considered as a length of the stick. In the beginning the length is 1. As k increases, the feature presence probabilities b_k decrease exponentially fast to zero, and this process can be seen as breaking the stick into two at each iteration k. Williamson et al. (2010) observe that a Bernoulli random variable z^k_i can be represented as:

$$z^k_i = \mathbb{I}[u^k_i < \Phi^{-1}_{\mu,\sigma^2}(b_k)] \tag{4.11a}$$

$$u^k_i \sim \mathcal{N}(\mu, \sigma^2), \tag{4.11b}$$

where $\Phi_{\mu,\sigma^2}(\cdot) := \Phi(\cdot|\mu,\sigma^2)$ is a Gaussian cumulative distribution function (CDF). We propose a covariate dependent IBP model by linearly parameterizing the cut off variable u_i^k as follows:

$$z_i^k = \mathbb{I}[u_i^k < \Phi^{-1}_{\mu,\sigma^2}(b_k)] \tag{4.12a}$$

$$u_i^k = -x_i^\top g_k + \epsilon, \quad \epsilon \sim \mathcal{N}(0,1). \tag{4.12b}$$

Here, $g_k \in \mathbb{R}^d$ is a vector of regression coefficients for each feature k, hence,

$$u_i^k|\, x_i, g_k \sim \mathcal{N}(u_i^k| - x_i^\top g_k, 1). \tag{4.13}$$

When Φ_{μ,σ^2} in (4.12a) is the Gaussian CDF of $\mathcal{N}(u_i^k| - x_i^\top g_k, 1)$ in (4.12b), then we recover the standard IBP model (4.11). By varying the parameters (μ, σ^2), we enforce the latent feature presence probability to depend on the covariate data x_i. For simplicity, here, we focus on the standard Gaussian CDF, $\Phi_{0,1}(\cdot)$. From (4.12) we compute the feature presence probability:

$$\begin{aligned}&\Pr(z_i^k = 1|x_i, g_k, b_k) = \Pr(\mathbb{I}[-x_i^\top g_k + \epsilon < \Phi^{-1}_{0,1}(b_k)] = 1) = \\ &\Pr(\epsilon < x_i^\top g_k + \Phi^{-1}_{0,1}(b_k)) = \Phi_{0,1}(x_i^\top g_k + \Phi^{-1}_{0,1}(b_k)).\end{aligned} \tag{4.14}$$

The interpretation of the dependent model above is that the feature k is present depending on *probit regression* model $\Phi_{0,1}(x_i^\top g_k + \Phi^{-1}_{0,1}(b_k))$, with decreasing biases $\Phi^{-1}_{0,1}(b_k)$, which will ensure that finitely many features are used only. Note the similarity between the probit IBP model of the latent binary representation and logistic encoding used in the autoencoder model in the previous chapter.

We use a spherical Gaussian prior with a covariance matrix $\sigma_g^2 I$ for the regression coefficient matrix, $[g_1, g_2, \ldots, g_K] = G$. With the above construction, the generative process for our preference model with linear probit likelihood is:

$$v_j \sim \text{Beta}(\alpha, 1); \quad b_k = \prod_{j=1}^{k} v_j; \quad G \sim \mathcal{N}(0, \sigma_g^2 I); \tag{4.15a}$$

$$z_i^k|\, x, g, b \sim \text{Bernoulli}(\ \Phi_{0,1}(x_i^\top g_k + \Phi^{-1}_{0,1}(b_k))\ ); \tag{4.15b}$$

$$w_k \overset{\text{i.i.d.}}{\sim} \mathcal{G}(\gamma_w, \theta_w); \quad j \underset{i}{\succ} l|\, Z, w \sim \text{Bernoulli}(p^i_{jl}). \tag{4.15c}$$

The joint posterior distribution of the latent feature matrix Z, the features presence probability b, the weights w and the regression coefficient matrix G using the conditional independence assumptions depicted in Figure 4.1(b) is:

$$\begin{aligned}&\Pr(Z, b, w, G|\, X, T) \propto \\ &\Pr(T|Z, w)\ \Pr(Z|X, G, b)\ \Pr(w|\gamma_w, \theta_w)\ \Pr(G|\sigma_g)\ \Pr(b|\alpha).\end{aligned} \tag{4.16}$$

Feature presence probability Similarly to the IBP model, we show that the feature presence probability in the linear probit dependent model $\Pr(z_i^k = 1|x_i, g_k, b_k)$ has a decreasing order with exponential rate. We use the Chernoff bound of the Gaussian CDF to upper bound $\Pr(z_i^k = 1|x_i, g_k, b_k)$.

Definition 1 *Chernoff bound of the Gaussian CDF:*

$$\Phi_{0,1}(x) \leq \frac{1}{2} e^{-\frac{1}{2}x^2}, \quad x \leq 0. \tag{4.17}$$

Lemma 2 *Given a covariate x_i, the stick length b_k, and the vector of regression coefficients g_k, that is bounded, the probability of the k-th feature being present decreases exponentially fast $Pr(z_i^k = 1|g_k, b_k, x_i) \to 0$ when $k \to \infty$.*

Proof We apply the Chernoff bound (4.17) to $\Pr(z_i^k = 1|g_k, b_k, x_i)$:

$$\Phi_{0,1}(x_i^\top g_k + \beta_k) \leq \frac{1}{2} \mathrm{e}^{-\frac{1}{2}(x_i^\top g_k + \beta_k)^2}, \ x_i^\top g_k + \beta_k \leq 0, \tag{4.18}$$

where $\beta_k = \Phi_{0,1}^{-1}(b_k)$ and $\beta_k \to -\infty$ when $k \to \infty$ due to exponential decrease of the stick lengths b_k in the stick breaking IBP. For the bounded covariate x_i and the bounded vector of regression coefficients g_k, the condition $x_i^\top g_k + \beta_k \leq 0$ holds because $|x_i^\top g_k|$ is bounded:

$$|x_i^\top g_k| \leq \gamma |\beta_k|, \quad 0 \leq \gamma < 1, \quad k \to \infty. \tag{4.19}$$

Therefore, $-\frac{1}{2}(x_i^\top g_k + \beta_k)^2 \leq -\frac{1}{2}(|\beta_k| - |x_i^\top g_k|)^2 \leq -\frac{1}{2}(|\beta_k|(1-\gamma))^2$ and we can simplify the bound,

$$\Pr(z_i^k = 1|g_k, b_k, x_i) \leq \frac{1}{2} \mathrm{e}^{-\frac{1}{2}((1-\gamma)\beta_k)^2} \xrightarrow[k \to \infty]{} 0. \tag{4.20}$$

■

Note that the lemma also recovers the standard IBP model when $\gamma = 0$, because $g_k = 0$ for all k according to (4.19).

4.4 Inference

In the inference phase, the goal is to compute the joint posterior over the latent binary feature matrix Z, the non-negative weights w and the regression coefficient matrix G (for the linear probit dependent model) as expressed in (4.9) and (4.16). For our proposed model, exact inference is computationally intractable. Thus, we employ a Markov Chain Monte Carlo (MCMC) method (Andrieu et al., 2003) to explore the posterior distributions.

4.4.1 Sampling for Linear Gaussian Model

Sampling of Z The sampler for the binary feature matrix Z consists of sampling existing features, proposing new features with corresponding weights and accepting or rejecting them based on the Metropolis-Hasting (M-H) criterion. We sample each row z_i one after another. For sampling existing features, we use:

$$\Pr(z_i^k = 1|X, T, w) \propto \\ \int m_{-i,k} \Pr\left(T\,|w, Z_{-i,k}, z_i^k = 1\right) \Pr\left(X\,|Z_{-i,k}, z_i^k = 1, V\right) \Pr\left(V\right) dV, \qquad (4.21)$$

where $Z_{-i,k}$ stands for column k in the latent feature matrix Z excluding row i, and $m_{-i,k}$ is the number of non-zero entries in it.

For sampling new features, we simultaneously propose $(K_{\text{new}}, Z_{\text{new}}, w_{\text{new}})$, where number of new features K_{new} is sampled from the Poisson(α/N) prior, w_{new} is the current weight vector for K existing features appended with i.i.d weights from its Gamma prior for K_{new} features and Z_{new} is the current feature matrix Z augmented with K_{new} column vectors of ones. We evaluate whether to accept the proposal $(K_{\text{new}}, Z_{\text{new}}, w_{\text{new}})$ using a M-H acceptance ratio, which reduces to the ratio of the likelihoods (Meeds et al., 2007):

$$\frac{\Pr(T|w_{\text{new}}, Z_{\text{new}}) \Pr(X|Z_{\text{new}}, V)}{\Pr(T|w, Z) \Pr(X|Z, V)} > 1. \qquad (4.22)$$

Due to our Gaussian assumptions, the real-valued weight matrix V in (4.21) and (4.22) can be marginalized analytically (Griffiths & Ghahramani, 2005).

Sampling of w For the weights that correspond to the non-zero features, we have:

$$\Pr(w|Z, X, T) \propto \int \Pr(T|Z, w)\Pr(X|Z, V)\Pr(V)\Pr(w)dV. \qquad (4.23)$$

We use a slice sampling procedure of Neal (2003) and sample each of the non-negative weights that correspond to the non-zero features and drop the weights that correspond to zero features.

4.4.2 Sampling for Linear Probit Model

For this model, we adapt a slice sampling procedure with stick breaking representation described by Teh et al. (2007).

Sampling of b The form of the conditional distribution of b can be found in (Teh et al., 2007). As suggested by the authors, we use the adaptive rejection sampling procedure (Gilks & Wild, 1992) to draw samples from it, due to the log-concavity of the distribution.

Sampling of Z Given the auxiliary slice variable, we will only update the latent feature for each observation and each dimension where its feature presence probability is below the slice. The required conditional distributions are:

$$\begin{aligned}&\Pr(z_i^k = 1 | x_i, T, w, g_k, b_k) = \\ &\Phi_{0,1}(x_i^\top g_k + \Phi_{0,1}^{-1}(b_k))\Pr(T|w, Z_{-i,k}, z_i^k = 1).\end{aligned} \tag{4.24}$$

Sampling of w Similar to the linear Gaussian case, we update the non-negative weights that correspond to the non-zero features by using a slice sampling procedure.

Sampling of G We sample each component g_k of the regression coefficient matrix $[g_1, g_2, \ldots, g_K]$ using elliptical slice sampling (Murray et al., 2010), an efficient MCMC procedure for the training of tightly coupled latent variables with a Gaussian prior.

4.5 Prediction on Test Data

4.5.1 Linear Gaussian Model

For a *previously unseen* test point $\bar{x} \in \mathbb{R}^d$, the joint predictive distribution for the latent variable $\bar{z}$ and the preference relation variable $\bar{t}$ is:

$$\Pr(\bar{z}, \bar{t}|X, T, \bar{x}) = \int \sum_Z \Pr(\bar{z}, \bar{t}|Z, w, X, \bar{x})\Pr(Z, w|X, T)dw, \tag{4.25}$$

where $\Pr(\bar{z}, \bar{t}|Z, w, X, \bar{x}) = \Pr(\bar{t}|\bar{z}, w)\Pr(\bar{z}|Z, X, \bar{x})$. This involves averaging over the predictions made by each of the posterior samples of Z and w. The preference relation variable $\bar{t}$ is a binary variable representing whether the object $\bar{x}$ is preferred in some triplet (or not). Since we have trained the binary latent space in a supervised manner, we could predict the label of the new test point based on its neighbors and non-neighbors. For this, we perform a *nearest neighbor* classification of the inferred test latent variable $\bar{z}$ with respect to the training latent variables Z. Therefore, we are interested only in the predictive distribution over the latent variable $\bar{z}$, and it has the form:

$$\Pr(\bar{z}|X,\bar{x}) = \sum_{Z} \Pr(\bar{z}|Z,X,\bar{x})\Pr(Z|X), \text{ where} \tag{4.26a}$$

$$\Pr(\bar{z}|Z,X,\bar{x}) \propto \Pr(\bar{x}|\bar{z},Z,X)\Pr(\bar{z}|Z). \tag{4.26b}$$

The explicit form of $\Pr(\bar{x}|\bar{z},Z,X)$ can be found from the joint Gaussian distribution of observed (training) and unobserved (test) data samples:

$$\begin{bmatrix} X \\ \bar{x} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} ZZ^\top + \sigma_x^2/\sigma_v^2 I & Z\bar{z}^\top \\ \bar{z}Z^\top & \bar{z}\bar{z}^\top + \sigma_x^2/\sigma_v^2 I \end{bmatrix}\right). \tag{4.27}$$

The conditional probability distribution has also a Gaussian form:

$$\Pr(\bar{x}|\bar{z},Z,X) \sim \mathcal{N}(\bar{\mu},\bar{\Sigma}) \tag{4.28a}$$

$$\bar{\mu} = \bar{z}(Z^\top Z + \sigma_x^2/\sigma_v^2 I)^{-1} Z^\top X \tag{4.28b}$$

$$\bar{\Sigma} = \bar{z}\bar{z}^\top - \bar{z}(Z^\top Z + \sigma_x^2/\sigma_v^2 I)^{-1} Z^\top Z \bar{z}^\top. \tag{4.28c}$$

The above predictive distribution $\Pr(\bar{x}|\bar{z},Z,X)$ defines a distribution of the mapping from a latent space to the observed data space.

Fast approximation In cases where we are only interested in a maximum a posteriori (MAP) estimate of the latent variables, it is desirable to avoid sampling from the predictive distribution, and directly find an approximate MAP estimate in a computationally efficient way. In our case, we use the predictive mean of $\Pr(\bar{x}|\bar{z},Z,X)$ in (4.28b) to approximate $\bar{z}$ by solving a linear system of equations, resulting in a continuous estimate of the binary vector $\bar{z}$:

$$\bar{x} \approx \bar{z}(Z^\top Z + \sigma_x^2/\sigma_v^2 I)^{-1} Z^\top X, \tag{4.29a}$$

$$\text{thus, } \bar{z} \approx \bar{x} X^\top Z (Z^\top X X^\top Z)^{-1} (Z^\top Z + \sigma_x^2/\sigma_v^2 I). \tag{4.29b}$$

4.5.2 Linear Probit Dependent Model

Similar to the linear Gaussian model, but with explicit representation of the regression coefficient matrix, the joint predictive distribution of the latent variable $\bar{z}$ and the preference variable $\bar{t}$ for a *new* test point $\bar{x} \in \mathbb{R}^d$ is:

$$\Pr(\bar{z},\bar{t}|T,\bar{x}) = \iiint \Pr(\bar{z},\bar{t}|G,w,b,\bar{x}) \sum_{Z} \Pr(Z,b,w,G|X,T)\; db\; dw\; dG, \tag{4.30}$$

with test likelihood given as follows:

$$\Pr(\bar{z},\bar{t}|G,w,b,\bar{x}) = \Pr(\bar{t}|\bar{z},w)\Pr(\bar{z}|G,b,\bar{x}). \tag{4.31}$$

As earlier, we are only concerned with the predictive distribution over the latent variable $\bar{z}$ for the new input $\bar{x}$, which is $\Pr(\bar{z}|G,b,\bar{x})$. Based on our linear probit model, this is $\Pr(\bar{z}^k = 1|G,b_k,\bar{x}) = \Phi_{0,1}(\bar{x}^\top g_k + \Phi_{0,1}^{-1}(b_k))$.

4.6 Related work

Infinite Latent Feature Models We are interested in Indian Buffet Process (IBP) based models that allow the number of latent features to be learned from data. By defining appropriate data generating likelihood functions, the IBP can be used in, among others, binary factor analysis (Griffiths & Ghahramani, 2005), choice behavior modeling (Görür et al., 2006), sparse factor and independent component analysis (Knowles & Ghahramani, 2007), link prediction (Miller et al., 2009) and invariant features (Austerweil & Griffiths, 2010; Zhai et al., 2012). For a recent comprehensive review of the IBP models, refer to Griffiths & Ghahramani (2011). Lately, there is also growing interest in learning correlated non-parametric feature models (Doshi-Velez & Ghahramani, 2009; Miller et al., 2008), or in the direction of supervised modeling, as for dimensionality reduction (Rai & Daume III, 2009). We should underline that our scenario of dependence on per object covariate x_k is not covered by the dependent IBP of Williamson et al. (2010). Their model defines a prior over multiple IBP matrices which (for certain settings of the model) are marginally IBP: a similar statement for our construction is meaningless since we only have one IBP matrix. However, our model does have the property that Z is IBP distributed conditional on vector of regression coefficients $g_k = 0$ for all k.

Gershman et al. (2015) expressed a goal closely related to ours, that nearby data is more likely to share latent features than distant data (as induced by distances between data in time or space, for example). However, encouraging sharing features between nearby data does not provide sufficient margin of separation between features of distant data. Our goal is to discover a binary latent space where meaningful notions of similarity and difference are preserved in terms of metric distances. Recently, the supervised IBP model was also proposed for cross-modal retrieval (Zhen et al., 2015), where, among others, the preference relation is extended to satisfy the cross-modal constraints. Also, IBP based models get moderate attention in the computer vision community, for example, as a dictionary learning framework (Feng et al., 2014) for learning discriminative mid-level features with efficient inference procedure, and as a weakly supervised method for learning objects, attributes and object-attribute relations in (Shi et al., 2014).

Binary hash coding Our augmented attribute representation can be viewed as a binary hash code representation for efficient image search. Binary representations are very attractive for reducing storage requirements and accelerating search and retrieval in large collections of high dimensional data. In recent years, there has been much interest in designing compact binary hash codes such that vectors that are similar in the original data space are mapped to similar binary strings as measured by Hamming distance (Salakhutdinov & Hinton, 2007a). Hash code is a short binary string that can act as an index to access directly elements in a database. Several machine learning methods have been developed to learn a *compact* hash code (Salakhutdinov & Hinton, 2007a; Torralba et al., 2008; Weiss et al., 2009; Norouzi et al., 2012), and to learn hash codes with better discrimination power (Mu et al., 2010; Wang et al., 2012), for example.

4.7 Experiments

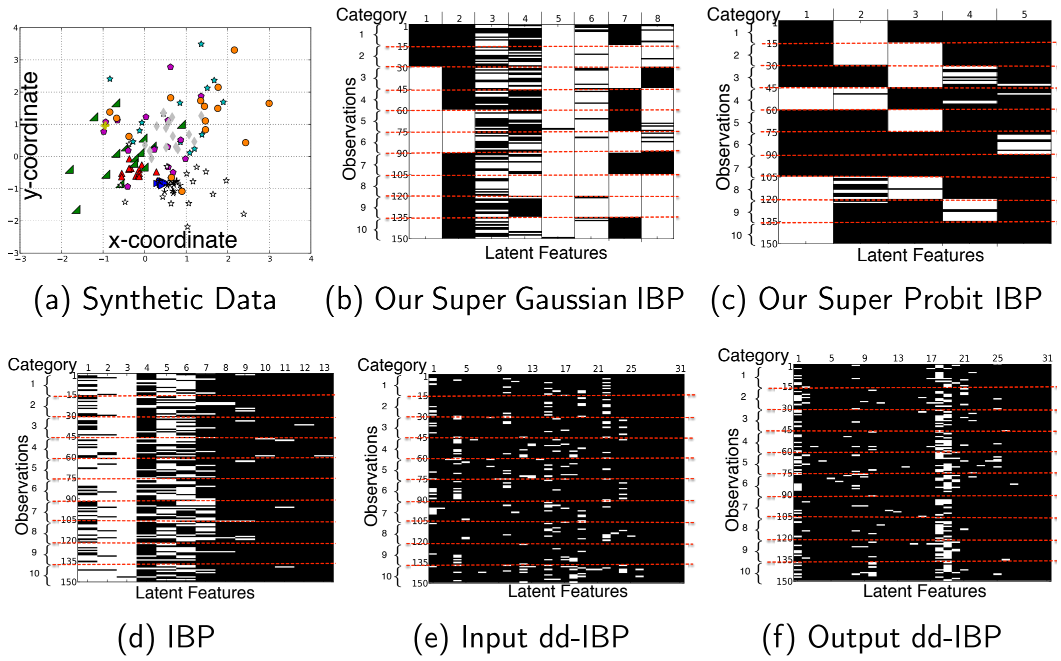
We start with a synthetic data experiment to explore the structure of the latent space Z produced by the proposed models (Section 4.7.1-4.7.2). Then, we continue with the application of augmented attribute representations in Section 4.7.3.

4.7.1 Visualization of the Binary Latent Space

Data We generate 150 synthetic data points with 10 categories from a mixture of 2-D multivariate Gaussians with uniformly drawn standard deviations in the range $[0, 1]$. The means are uniformly drawn in the range $[-1, 1]$ per category. The visualization of the generated data is provided in Figure 4.2(a).

Algorithms We compare the generated latent space of our supervised linear Gaussian (`Super Gaussian IBP`) and supervised linear probit (`Super Probit IBP`) models with the Indian buffet process (`IBP`), and the distance dependent Indian buffet process with distance defined on $\mathcal{X}$ (`Input dd-IBP`)[1], and on the

[1] We use the implementation provided by Gershman et al. (2015) at `http://www.princeton.edu/~sjgershm/ddIBP_release.zip`

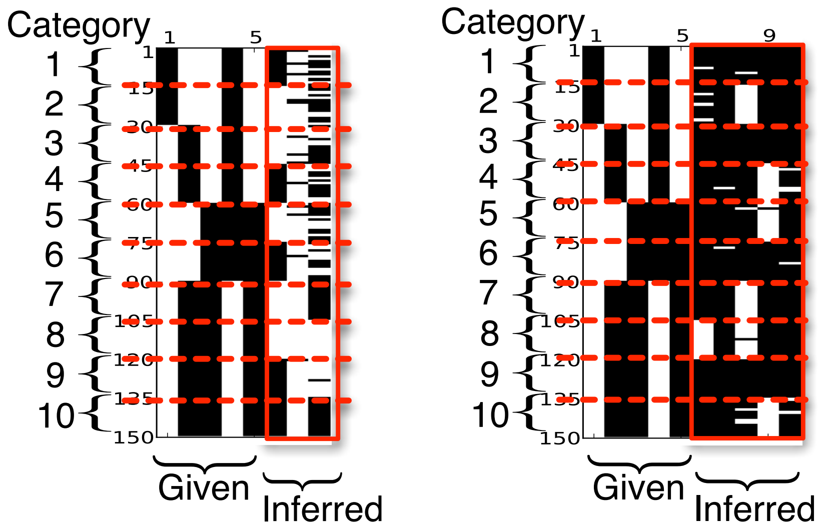


(a) Synthetic Data (b) Our Super Gaussian IBP (c) Our Super Probit IBP

(d) IBP (e) Input dd-IBP (f) Output dd-IBP

Figure 4.2: Visualization of the binary latent space. 4.2(a): 150 synthetic data points of 10 categories coming from a mixture of 2-D multivariate Gaussians. 4.2(b)-4.2(f): the corresponding latent binary representations generated by various methods ('1' is white, and '0' is black). The supervised information is given in terms of friends and non-friends relationship between data points from the same/different classes. In this example, each training data point has 14 friends and 14 non-friends. For clarity, we group the training samples from the same category together (between red lines) and annotate the category name accordingly. Our methods, Super Gaussian IBP and Super Probit IBP, clearly enforce neighborhood structure by assigning distinct features for different categories.

labels (`Output dd-IBP`). On a practical note, the supervised information in terms of triplets forms only a small number of observed entries in T. This translates to a minor computational overhead compared with the standard IBP inference. For all methods, we place conjugate priors on the hyperparameters, and subsequently perform posterior inference over them. The results are shown in Figure 4.2. In comparison with the IBP 4.2(d), dd-IBP input 4.2(e), and dd-IBP output 4.2(f) models, the inferred feature matrix Z of the proposed models `Super Gaussian IBP` 4.2(b) and `Super Probit IBP` 4.2(c) discovers a discriminative class structure for all categories by assigning distinct features for different categories.



(a) Super Gaussian IBP (b) Super Probit IBP

Figure 4.3: Extending observed binary variables. The first 5 *given* binary variables do not respect the neighborhood structure, for example, categories 1 and 2 have the same '01101' representation. Our supervised models allow *coupling* between given and inferred latent features. As a result, the inferred latent features enforce separation among categories and amend shortcomings that the observed binary representation might have in preserving the class neighborhood structure.

4.7.2 Model with Observed Binary Variables

We are interested to explore how this model can be used for the attribute augmentation task previously described in Section 3.2. We assume that we are given 5 binary attributes representation that can partially discriminate the objects according to their categories. Refer here to Figure 4.3(a). In this setup, the given attributes of categories 1 and 2 form binary representation '01101', of the categories 3 and 4 have '10101' representations, of the categories 5 and 6 have '11000', and of the categories $7, 8, 9$ and 10 have '10010'. The task is to augment these binary representations with latent features that can solve the problem of class separation by utilizing the supervised neighborhood information.

Formally, for each training sample x_i, we want to augment its observed binary representation $a_i \in \mathcal{A}$, where $\mathcal{A} \in \{0, 1\}^5$ with a latent binary feature vector z_i, forming the augmented representation $[a_i, z_i] diff$. Let A be the observed $N \times 5$ binary representation matrix, and w_A be a 5×1 *non-negative* weight vector. The preference probability p^i_{jl} defined in Equation (4.3) has the following form:

$$p^i_{jl} = \frac{1}{C}\{\sum_{n=1}^{5} w^n_A \, \mathbb{I}[a^n_i = a^n_j] \, (1 - \mathbb{I}[a^n_i = a^n_l]) + \sum_{k=1}^{K} w_k \, \mathbb{I}[z^k_i = z^k_j] \, (1 - \mathbb{I}[z^k_i = z^k_l]) \}, \tag{4.32}$$

where the normalizing constant C ensures $p^i_{jl} + p^i_{lj} = 1$. The latent features are inferred to enforce separation among categories and amend shortcomings that the observed binary variables might have in preserving the neighborhood structure.

Results. The supervised models are trained to utilize the *given* binary features, and to add additional binary latent representations only when it is needed to support the discrimination between categories (see Figure 4.3). As an example, in case of categories 1 and 2 that are indistinguishable under the first 5 binary features, 4.3(a)-4.3(b) learn at least unit distance in the extended representation for these categories, and improve the separation from the remaining categories. `Super Gaussian IBP` (Figure 4.3(a)) discovers additional 3 binary latent variables where category 1 has '$0**$' and category 2 has '$1**$'. While `Super Probit IBP` (Figure 4.3(b)) discovers 5 more binary latent variables with '$**0**$' and '$**1**$' assigned to category 1 and 2, respectively.

4.7.3 Augmenting Semantic Attributes

We use the *Animals with Attributes (AwA)* dataset (Lampert et al., 2009, 2013) described in our background Section 2.1.3 for this application. We follow closely our previous synthetic data experiment and define 2 settings to demonstrate the performance of the proposed models in the *difficult* scenario, when semantic attribute representation alone is not good enough to discriminate the classes.

1. We use 27,032 images from 45 classes to learn the semantic attribute predictors. From the remaining 5 classes – *dolphin*, *humpback whale*, *killer whale*, *chimpanzee* and *gorilla* – we randomly sample 150 images for training and 150 images for testing the algorithms (30/30 from each class).

2. We use 23,266 images from 40 classes to learn the semantic attribute predictors. And from the remaining 10 classes – *dolphin*, *humpback whale*, *killer whale*, *seal*, *chimpanzee*, *gorilla*, *giraffe*, *leopard*, *bobcat* and *squirrel* – we randomly sample 300 images for training and 300 images for testing the performance of the algorithms (30/30 from each class).

In both cases, we repeat the procedure of train and test splits 5 times to get better statistics of the performance. In this experiment, we use the representation by color histograms of quantized RGB pixels with a codebook of size 128 provided with the dataset and referred to as original feature representation.

Attribute predictors. We use the color histograms to represent images, and we focus on augmenting *color attributes*. Specifically, we take the first 5 entries in the Osherson's attribute vector that correspond to *black*, *white*, *blue*, *brown* and *gray* attributes. We again use the ℓ_2-regularized logistic regression model to train the attribute predictors using the original feature representation of 128-dimensional color histograms. The difficulty of our scenarios comes from the type of animals used and their attribute representation. In the case of 5 classes, the first 3 classes are animals which live in water (*humpback whale*, *dolphin*, *killer whale*), thus expected to have similar appearance, and the other 2 classes (*chimpanzee* and *gorilla*) have the same attribute representation in terms of color. In the case of 10 classes, the representation of different classes often differs in only one bit, or they are even identical. So our goal is to extend these observed semantic attributes with latent binary features that help the class separation, i.e. the augmented representation is favorable for nearest neighbor classification.

Algorithms. We compare the performance of our supervised linear Gaussian (`Super Gaussian IBP`) and supervised linear probit (`Super Probit IBP`) models with the Indian buffet process (`IBP`), and the distance dependent Indian buffet process with distance defined on $\mathcal{X}$ (`Input dd-IBP`), and on the labels $\mathcal{Y}$ (`Output dd-IBP`). During training, all these models learn to augment the true Osherson's attribute representation of the data points. Our proposed models utilize the supervised information in terms of triplets of friends and non-friends, and the remaining three models (`IBP`, `Input dd-IBP`, `Output dd-IBP`) do this in an unsupervised way. At test time, we predict the attributes on the test set using the attribute predictors and augment this representation with the latent features learned using the baseline methods. The supervised information is given by a set of triplets generated the same way as in the synthetic data experiment: for each data point we define 29 friends (all remaining samples in this class) and 29 non-friends (randomly sampled from other classes). The costly MCMC procedure is performed offline at the training phase. At test time, we simply perform a fast approximation via matrix vector multiplication in the linear Gaussian model (Section 4.5.1) or compute probit regression in the linear probit model (Section 4.5.2).

NN	`Semantic Attributes`	`IBP`	`Input dd-IBP`	`Output dd-IBP`	`Super Gaussian IBP`	`Super Probit IBP`	`Reference`
5 classes							
1	26.3 ± 2.2	30.3 ± 2.0	27.9 ± 6.3	29.7 ± 3.5	33.8 ± 1.6	$\mathbf{42.8 \pm 2.4}$	$\mathbf{40.9 \pm 4.7}$
3		29.5 ± 2.9	29.6 ± 3.6	31.0 ± 1.4	34.6 ± 2.3	$\mathbf{41.9 \pm 3.4}$	$\mathbf{40.2 \pm 3.4}$
15		31.5 ± 2.6	27.8 ± 2.8	28.1 ± 3.2	35.5 ± 1.0	$\mathbf{44.5 \pm 2.1}$	39.3 ± 3.7
30		29.5 ± 3.2	24.3 ± 3.0	23.6 ± 3.4	33.8 ± 0.7	$\mathbf{45.9 \pm 4.1}$	36.1 ± 2.8
10 classes							
1	12.7 ± 2.5	17.1 ± 3.1	12.9 ± 2.6	15.9 ± 2.3	17.3 ± 1.2	$\mathbf{25.0 \pm 2.9}$	$\mathbf{25.0 \pm 2.2}$
3		17.9 ± 2.8	13.1 ± 2.4	15.3 ± 2.3	18.2 ± 1.2	$\mathbf{25.1 \pm 3.0}$	$\mathbf{26.0 \pm 1.7}$
15		16.4 ± 2.5	14.7 ± 2.3	15.1 ± 1.8	18.0 ± 1.5	$\mathbf{26.6 \pm 2.7}$	$\mathbf{27.8 \pm 2.8}$
30		17.7 ± 3.4	14.5 ± 1.9	14.0 ± 1.9	18.3 ± 1.4	$\mathbf{27.5 \pm 2.4}$	$\mathbf{25.8 \pm 1.4}$

Table 4.1: Application of augmenting semantic attributes. `IBP`: standard IBP algorithm (Griffiths & Ghahramani, 2005); `dd-IBP`: distance dependent IBP (Gershman et al., 2015), where `Input`: distance on $\mathcal{X}$, and `Output`: distance on the labels; `Super Gaussian IBP`: our proposed supervised IBP with linear Gaussian feature model; `Super Probit IBP`: our proposed supervised IBP with linear probit dependent model; `Reference`: original 128 real-valued feature representation; `Semantic Attributes`: semantic attribute representation alone. The numbers are k-NN accuracy mean ± std over 5 repeats. The best result and those not significantly worse than it (using one-sided paired t-test with 95% confidence) are highlighted in **boldface**.

Evaluation metric. We use k-nearest neighbor classification accuracy as the evaluation metric with $k = 1, 3, 15, 30$ (the latter corresponds to total number of samples per class). We compare the classification performance using augmented attribute representations, using semantic attribute representation alone and using original feature representation (as a reference baseline). The k-NN performance using semantic attribute representation does not depend on k, because we use the true Osherson's color attributes during training. Thus, nearest neighbor search is equivalent to finding nearest class-attribute representation.

Results. The results of all the baselines are summarized in Table 4.1. We observe that our proposed models, `Super Gaussian IBP` and `Super Probit IBP`, exceed the performance of IBP and dd-IBP in *all cases*. We further notice that Super Probit IBP is far superior to Super Gaussian IBP. We credit this to the fact that linear Gaussian models are less suitable for modeling real-valued im-

	5 classes	**10 classes**
`Linear SVM`	$\mathbf{43.3 \pm 4.0}$	$\mathbf{29.0 \pm 3.3}$
`Super Probit IBP`	$\mathbf{45.9 \pm 4.1}$ (30 NN)	$\mathbf{27.5 \pm 2.4}$ (30 NN)

Table 4.2: Accuracy comparison between linear SVM and k-NN performance of our Super Probit IBP. According to one-sided paired t-test with 95% confidence, the results are not significantly different for both cases, 5 classes and 10 classes.

	5 animal categories		**10 animal categories**	
	`last`	`average`	`last`	`average`
1 NN	42.8 ± 2.4	42.7 ± 3.2	25.0 ± 2.9	25.5 ± 3.9
3 NN	41.9 ± 3.4	$\mathbf{44.0 \pm 2.7}$	25.1 ± 3.0	$\mathbf{27.1 \pm 2.9}$
15 NN	44.5 ± 2.1	$\mathbf{46.5 \pm 2.3}$	26.6 ± 2.7	27.7 ± 2.3
30 NN	45.9 ± 4.1	46.5 ± 2.3	27.5 ± 2.4	27.2 ± 2.7

Table 4.3: Effect of Bayesian Averaging on Super Probit IBP. `last`: using a sample from the last iteration; `average`: using samples from the last 50 iterations. The numbers show accuracy performance (mean±std). **boldface** means significantly better using one-sided paired t-test with 95% confidence.

age representations (Austerweil & Griffiths, 2010; Zhai et al., 2012). One of the possible solutions could be to define a more complex likelihood function (Austerweil & Griffiths, 2010; Zhai et al., 2012). Instead, we focus on generating binary features that depend on the observed images via probit regression. As a reference, we also provide k-NN performance in the original 128 real-valued features. Original features will require storage of $8,192$ $(128 * 64)$ bits per image, while our Super Probit IBP code with 80 inferred binary latent dimensions will only consume approximately 80 bits per image and gives better results. Further, to view our results from a wider perspective, we also run the standard SVM baseline[1] using original feature representation (multi-class classification setting) and present the results in Table 4.2.

Bayesian averaging. We explore the advantage of the fully Bayesian approach, which allows us to learn the distribution over latent variables during the MCMC runs. In our experiments, we run MCMC until a fixed number of iterations, and subsequently consider the latent features given by the last iteration as the outcome of the model. Instead, we can exploit the performance of the full distribution by

[1] We use the LIBSVM library available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

averaging the nearest neighbor performances over MCMC runs. The results of Bayesian averaging on Super Probit IBP are summarized in Table 4.3. We can see that averaging has a *positive effect* on the performance of the model, however, it comes with a price in storage requirement where now the MCMC outcomes have to be maintained.

4.8 Summary and Discussion

In this chapter, we have presented the non-parametric probabilistic approach to learning mid-level feature representation of discriminative attributes that augment the given semantic attributes. With our proposed models, we overcome two limitations of the parametric approach described in the previous Chapter 3: there is no cross validation model selection procedure when using Bayesian learning framework, and the dimension of the augmented attribute space $\mathcal{B}$ is inferred automatically from the data. However, it comes at the price of slow inference procedure: the sampling mechanism is very costly and does not scale to large datasets. Recently, Feng et al. (2014) proposed an asymptotic analysis of the IBP model that can reduce the inference procedure to an efficient MAP solution. However, it is currently restricted to the Linear Gaussian model case, which is disadvantageous compared with the Linear Probit Model based on our experience. Therefore, in future it will be beneficial to adapt the asymptotic analysis of Feng et al. (2014) to the Linear Probit model, and to explore the application of scalable IBP model for continuous streams of data.

To make continuous updates of the Linear Probit model parameters, we have to compute efficiently the posterior distribution of regression coefficients g_k forming the regression matrix G and the vector of features presence probability b:

$$\Pr(G, b|X, Z) = \prod_{k=1}^{K} \frac{\Pr(Z^k|X, g_k, b_k)\Pr(g_k, b_k|b_{k-1}, X)}{\Pr(Z^k|X)} \tag{4.33a}$$

$$= \prod_{k=1}^{K} \frac{\Pr(Z^k|X, g_k, b_k)\Pr(g_k)\Pr(b_k|b_{k-1})}{\Pr(Z^k|X)} \tag{4.33b}$$

$$= \prod_{k=1}^{K} \left[\frac{\prod_{i=1}^{N} \Pr(z_i^k|x_i, g_k, b_k)\Pr(g_k)\Pr(b_k|b_{k-1})}{\prod_{i=1}^{N} \Pr(z_i^k|x_i)} \right] \tag{4.33c}$$

$$= \prod_{k=1}^{K} \left[\frac{\prod_{i=1}^{N} \Phi_{0,1}(x_i^T g_k + \Phi_{0,1}^{-1}(b_k)) \mathcal{N}(g_k | 0, \sigma_g^2 I) b_{k-1} \mathrm{Beta}(\alpha, 1)}{\prod_{i=1}^{N} \Pr(z_i^k | x_i)} \right], \tag{4.33d}$$

where computation of the normalization constant (in the denominator) requires integration over g_k and b_k, and is computationally expensive:

$$\Pr(z_i^k|x_i) = \int_{g_k} \int_{b_k} \Pr(z_i^k|x_i, g_k, b_k) \Pr(g_k) \Pr(b_k) \, dg_k \, db_k. \tag{4.34}$$

Replacing our sampling procedure for posterior computation of G, b with asymptotic analysis, or efficient approximate inference such as Expectation Propagation (Minka, 2001), or recently proposed spectral methods for IBP (Tung & Smola, 2014) would be attractive future directions to explore.

Part II

Attributes in Learning with Privileged Information: Parametric and Non-parametric Views

For 2,000 years, we believed logic was the only instrument for solving intellectual problems. Now, our analysis of machine learning is showing us that to address truly complex problems, we need images, poetry, and metaphors as well.

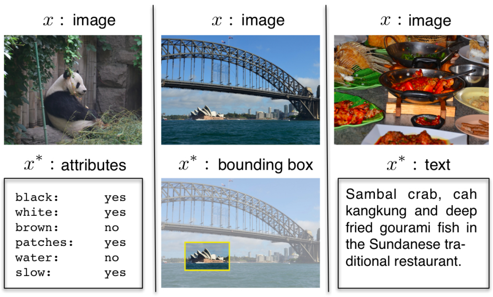
Vladimir Vapnik

Chapter 5

Learning with Privileged Information: Parametric View

In this chapter, we focus on semantic attributes as a source of additional information about image data. This information is privileged to image data as it is not available at test time. We bring a learning framework called *learning using privileged information (LUPI)* to the computer vision field to solve the object recognition task in images. We want the computers to be able to learn more efficiently at the expense of providing extra information during training time. Besides semantic attributes, we look at bounding boxes and image tags annotations as additional information about the image data alone. We explore two maximum-margin LUPI techniques for binary and multiclass settings. We interpret these methods as learning to identify easy and hard objects in the privileged space and transferring this knowledge to train a better classifier in the original data space. We provide a thorough analysis and comparison of information transfer from privileged to the original data spaces for both LUPI methods. Our experiments show that incorporating privileged information can improve the classification accuracy. Finally, we conduct user studies to understand which images are easy and which are hard for human learning, and explore how this information is related to easy and hard samples when training a classifier from images.

This chapter is based on:



black:	yes
white:	yes
brown:	no
patches:	yes
water:	no
slow:	yes

Sambal crab, cah kangkung and deep fried gourami fish in the Sundanese traditional restaurant.

V. Sharmanska, N. Quadrianto, C.H. Lampert: *Learning to Rank Using Privileged Information,* ICCV 2013, Sydney, Australia.

V. Sharmanska, N. Quadrianto, C.H. Lampert: *Learning to Transfer Privileged Information,* 2014, under journal review.

5.1 Introduction

The framework called *learning using privileged information (LUPI)* was formally introduced by Vapnik & Vashist (2009), and it has not been recognized in the computer vision community until very recently. The concept is inspired by human experience of learning with a teacher, when during learning we have access to training examples and to an additional source of explanation from the teacher. For example, learning a new concept in mathematics is faster when the teacher explains it to us rather than if we only get questions and right answers. After the course, the students should be able to solve new tasks themselves and not rely on the teacher's expertise anymore. Training with a teacher can significantly improve the learning process and ability to generalize for humans and machines (Vapnik & Vashist, 2009).

As a general framework, LUPI has been successfully applied to a variety of tasks: handwritten digit images with poetic descriptions as the privileged source for a data clustering task (Feyereisl & Aickelin, 2012); head pose or gender as privileged information for facial feature detection(Yang & Patras, 2013); facial expression recognition from low resolution images with high resolution images as the privileged source (Chen et al., 2013); metric learning (Fouad et al., 2013); our ranking framework with attributes, bounding box annotation, textual description, and rationales as privileged information (Sharmanska et al., 2013); counting with back-propagation (Chen & Kämäräinen, 2014).

In the standard learning setting, we are given input–output training pairs about the task we want to learn, for example, images and category labels for object classification. In the LUPI setting, we have the input–output training pairs plus additional information for each training pair that is *only available during training*. There is no direct limitation on the form of privileged information, i.e. it could be yet another feature representation, or a completely different modality like text or human annotation in addition to image data, that is specific for each training instance.

LUPI in its original formulation does not tell us what kind of privileged information is useful, and will lead to better performance, or how to measure the quality of it. In this chapter, we examine the three different types of privileged information in the context of object classification task: *attributes* that describe semantic properties of an object, *bounding boxes* that specify the exact localization of the target object in an image, and *image tags* that describe the context of an image in textual form. Figure 5.1 illustrates these three modalities.

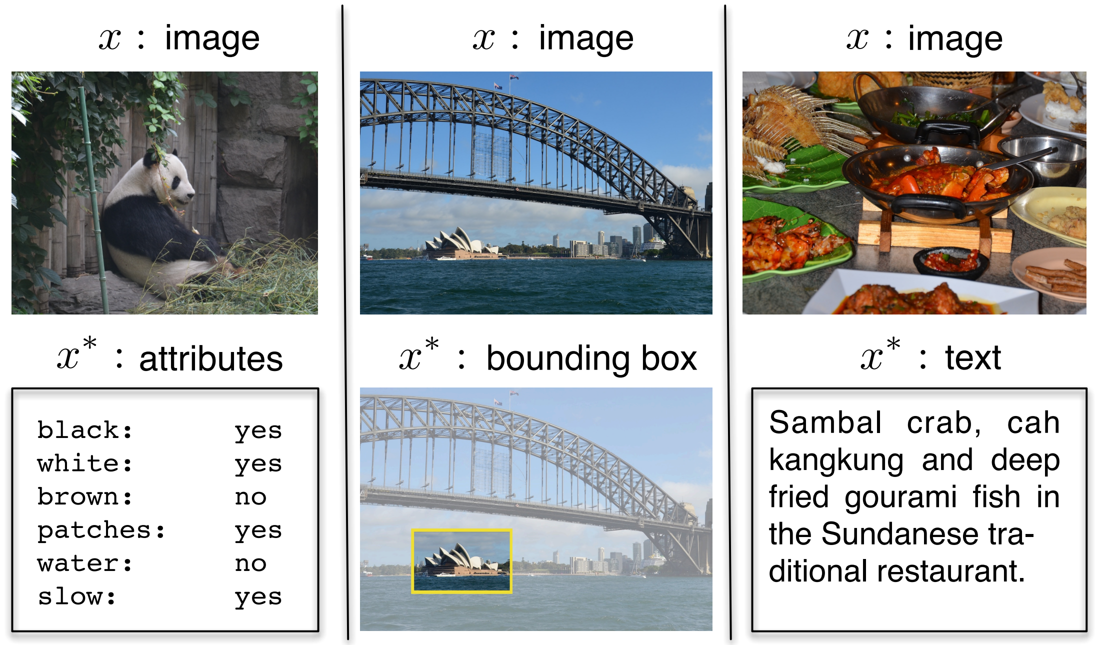


Figure 5.1: Three different forms of *privileged* information that can help learning better object recognition systems: *attributes*, *object bounding boxes*, and *textual descriptions*.

Main idea In order to do LUPI, we have to understand how to make use of the data modality that is not available at test time. At first glance, training a classifier on the privileged data does not appear sensible, since there is no way to evaluate the resulting classifier on the test data. At the core of our work lies the assumption that *privileged information allows us to distinguish between* easy *and* hard *examples in the training set.* Assuming that examples that are easy or hard with respect to the privileged information will also be easy or hard with respect to the original data, we enable information transfer from the privileged to the original data modality. Specifically, we first identify which samples are easy and which are hard for the classification task, and then we incorporate this information into the sample weights and train the classifier in the original space with encoded easy-hard weights.

We formalize the above observation in Section 5.3, where we study and compare two maximum-margin learning techniques for LUPI. The first, SVM+, was originally described by Vapnik & Vashist (2009). The second, *Margin Transfer*, is our contribution to the classification setting, and is an adaptation of our earlier method proposed for ranking framework (Sharmanska et al., 2013). We analyze the core difference of the information transfer in the proposed methods, and how this kind of knowledge about the learning problem can guide the training of an image-based predictor to a better solution. In Section 5.4, we report on experiments in the three privileged information scenarios introduced earlier: attributes, bounding boxes and image tags. We demonstrate how to avoid handcrafted methods designed for a specific type of additional information, and handle all the situations in a unified framework. In Section 5.5 we show that our method is naturally suitable for a multiclass classification setting with the one-versus-rest strategy. Additionally, we conduct user studies to identify easy and hard samples in human learning of object categories. We then utilize this data as ground truth information to analyze and compare with easy and hard samples learned by the proposed LUPI methods in Section 5.6. We end with the discussion and conclusions in Section 5.7.

5.2 Related work

In computer vision problems it is common to have access to multiple sources of information. Sometimes all of them are visual, such as when images are represented by color features as well as by texture features. Sometimes, the modalities are mixed, such as for images with text captions. If all modalities are present both at training and at test time, it is rather straight-forward to combine them for better prediction performance. This is studied, e.g., in the fields of multi-modal or multi-view learning. Methods suggested here range from stacking, where one simply concatenates the feature vectors of all data modalities, to complex adaptive methods for early or late data fusions (Snoek et al., 2005), including multiple kernel learning (Vedaldi et al., 2009) and LP-β (Gehler & Nowozin, 2009).

Situations with an asymmetric distribution of information have also been explored. In weakly supervised learning, the annotation available at training time is less detailed than the output one wants to predict. This situation occurs, e.g., when trying to learn an image segmentation system using only per-image or bounding box annotation (Kuettel et al., 2012). In multiple instance learning, training labels are given not for individual examples, but collectively for groups of examples (Maron & Ratan, 1998). The inverse situation also occurs: for example in the PASCAL object recognition challenge, it has become a standard technique to incorporate strong annotation in the form of bounding boxes or per-pixel segmentations, even when the goal is object categorization (Everingham et al., 2010;

Russakovsky et al., 2012). Similar to strong and weak supervision, situations in which the data representations differ between training and testing phase can be distinguished by whether one has less or more information available at training time than at test time. The first situation occurs, e.g., in tracking, where temporal continuity can be used at test time that might not have been available at training time (Kalal et al., 2009). Similarly, it has been shown that image metadata (geolocation, capture time) (Chen & Grauman, 2011) and an auxiliary feature modality (Khamis & Lampert, 2014) can provide additional information at test time compared to only the image information available at training time.

The situation we are interested in occurs when at training time we have an additional data representation compared to test time. Different settings of this kind have appeared in the computer vision literature, but each was studied in a separate way. For example, for clustering with multiple image modalities, it has been proposed to use CCA to learn a shared representation that can be computed from either of the representations (Blaschko & Lampert, 2008). Similarly the shared representation is also used for cross-modal retrieval by Quadrianto & Lampert (2011). Alternatively, one can use the training data to learn a mapping from the image to the privileged modality and use this predictor to fill in the values missing at test time (Christoudias et al., 2008). Donahue & Grauman (2011) showed that annotator rationales can act as additional sources of information during training, as long as the rationales can be expressed in the same data representation as the original data (e.g. characteristic regions within the training images).

We follow a different route than the above approaches. We are not looking for task-specific solutions applicable to a specific form of privileged information. Instead, we aim for a generic method that is applicable to any form of privileged information that is given as additional representations of the training data. We show in the following sections that such frameworks do indeed exist, and in Section 5.4 we illustrate that the individual situations described above can naturally be expressed in these frameworks.

5.3 Learning using Privileged Information

In the following we formalize the LUPI setup for the task of supervised binary classification. We describe a simple extension of LUPI for a multiclass setting using the one-versus-rest procedure in Section 5.5. Similarly to the settings before, we are given a set of N training images, represented by feature vectors $X = \{x_1, \ldots, x_N\} \subset \mathcal{X} = \mathbb{R}^d$, their label annotation, $Y = \{y_1, \ldots, y_N\} \in \mathcal{Y} = \{+1, -1\}$, and in addition to this, we are given a set of feature vectors, $X^* = \{x_1^*, \ldots, x_N^*\} \subset \mathcal{X}^* = \mathbb{R}^{d^*}$, where x_i^* encodes the privileged information we have about sample x_i. We do not make any specific assumption about the privileged data space $\mathcal{X}^*$, and keep general notations for the features extracted from visual, verbal or semantic form of privileged information. We refer to $\mathcal{X}$ and $\mathcal{X}^*$ as original and privileged data spaces, accordingly.

Our binary classification task is to learn a prediction function $f : \mathcal{X} \to \mathcal{Y}$ from a space $\mathcal{F}$ of possible functions, e.g. all linear classifiers. The goal of LUPI is to use the privileged data, X^*, to learn a better classifier in the original data space $f : \mathcal{X} \to \mathcal{Y}$, than one would learn without it. Since the privileged data is only available during training time and comes from a different domain, $\mathcal{X}^*$, than the original space $\mathcal{X}$, it is not possible, e.g., to apply functions defined on $\mathcal{X}$ to $\mathcal{X}^*$ or vice versa. In this chapter, we describe how to use the privileged data to characterize the training samples in the original data space into easy and hard cases. Knowing this will help us to direct the learning procedure towards better generalization and to learn a function of higher prediction quality.

In the following, we will explain two maximum-margin methods for learning with privileged information that fit to this interpretation. The first method was proposed by Vapnik & Vashist (2009), and the second method is our alternative model for solving LUPI.

5.3.1 Maximum Margin Model 1: SVM+

The first model for learning with privileged information, SVM+, (Vapnik & Vashist, 2009; Pechyony & Vapnik, 2010) is based on a direct observation that soft constraints in a soft-margin SVM could be turned into hard constraints that resemble a hard-margin SVM if one had access to a so-called *slack oracle*. Standard soft-margin SVM classifier was previously described in our background Section 2.1.2 for training the attribute models. For clarity of presentation, here we include both the soft-margin and the hard-margin formulations of SVM:

Soft-margin SVM

$$\underset{\substack{w\in\mathbb{R}^d, b\in\mathbb{R}\\ \xi_1,\dots,\xi_N}}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \quad (5.1\text{a})$$

subject to, for all $i = 1, \dots, N$,

$$y_i[\langle w, x_i\rangle + b] \geq 1 - \xi_i, \quad (5.1\text{b})$$

$$\xi_i \geq 0.$$

Hard-margin SVM

$$\underset{w\in\mathbb{R}^d, b\in\mathbb{R}}{\text{minimize}} \ \frac{1}{2}\|w\|^2 \quad (5.2\text{a})$$

subject to, for all $i = 1, \dots, N$,

$$y_i[\langle w, x_i\rangle + b] \geq 1. \quad (5.2\text{b})$$

The soft-margin SVM classifier is fully characterized by its weight vector w and bias parameter b. However, in the training phase, N slack variables ξ_i – one for each training sample – also need to be estimated. When the number of training examples increases, soft-margin SVM solutions are known to converge with a rate of $\mathrm{O}\left(\frac{1}{\sqrt{N}}\right)$ to the optimal classifier (Vapnik, 1999). This is in sharp contrast to the hard-margin solutions that converge with a rate of $\mathrm{O}\left(\frac{1}{N}\right)$. Then one could wonder whether it is possible for the soft-margin SVM to have a faster convergence rate, ideally at the same rate as the hard-margin SVM. If the answer is positive, the improved soft-margin SVM would require fewer training examples to reach a certain prediction accuracy than a standard one. Intuitively, with $\mathrm{O}\left(\frac{1}{N}\right)$ rate, we will only require 100 samples instead of $10,000$ to achieve the same level of predictive performance.

It might not come as a surprise that if we knew the optimal slack values ξ_i in the optimization problem (5.1), for example from an *oracle*, then the formulation can be reduced to the Oracle SVM that resembles the hard-margin case (5.2) with the convergence rate $\mathrm{O}\left(\frac{1}{N}\right)$.

Oracle SVM

$$\underset{w\in\mathbb{R}^d, b\in\mathbb{R}}{\text{minimize}} \ \frac{1}{2}\|w\|^2 \quad (5.3\text{a})$$

subject to, for all $i = 1, \dots, N$,

$$y_i[\langle w, x_i\rangle + b] \geq r_i, \quad (5.3\text{b})$$

r_i is known ($r_i = 1 - \xi_i$).

Instead of $N+d+1$ unknowns which include slack variables, we are now estimating only $d+1$ unknowns which are the actual object of interest, our classifying hyperplane. The interpretation of slack variables is to tell us which training examples are *easy* and which are *hard*. In the above Oracle SVM, we do not have to infer those variables from the data as they are given by the oracle.

The idea of the SVM+ method is to use the privileged information as a proxy to the oracle. For this the slack variable is parameterized with unknown w^* and b^*:

$$\xi_i = \langle w^*, x_i^* \rangle + b^*, \tag{5.4}$$

and we obtain the following formulation of the SVM+ training problem:

$$\underset{\substack{w \in \mathbb{R}^d, b \in \mathbb{R} \\ w^* \in \mathbb{R}^{d^*}, b^* \in \mathbb{R}}}{\text{minimize}} \; \frac{1}{2}\left(\|w\|^2 + \gamma \|w^*\|^2\right) + C \sum_{i=1}^{N} \langle w^*, x_i^* \rangle + b^* \tag{5.5a}$$

subject to, for all $i = 1, \ldots, N$,

$$y_i[\langle w, x_i \rangle + b] \geq 1 - [\langle w^*, x_i^* \rangle + b^*] \tag{5.5b}$$

$$\text{and} \;\; \langle w^*, x_i^* \rangle + b^* \geq 0. \tag{5.5c}$$

The above SVM+ parameterizes the slack variables with a finite hypothesis space (a scalar and a weight vector with dimension d^*, for example), instead of allowing them to grow with the number of examples N.

Numerical optimization The SVM+ optimization problem (5.5) is convex, and can be solved in the dual representation using a standard quadratic programming (QP) solver. For a medium size problem (thousands to hundreds of thousands of samples), a general purpose QP solver might not suffice, and special purpose algorithms have to be developed to solve the QP. Pechyony & Vapnik (2011) derived suitable sequential minimal optimization (SMO) algorithms to tackle the problem. However, for the problem size that we are experimenting with (hundreds of samples), we find that using a general purpose QP provided in the `CVXOPT`[1] package is faster than the specialized SMO solver. Therefore, we use the `CVXOPT`-based QP solver for our experiments (Section 5.4).

5.3.2 Maximum Margin Model 2: Margin Transfer

In this framework, we propose a different model called *Margin Transfer* that: 1) can be solved by off-the-shelf SVM packages; and 2) *explicitly* enforces an easy-hard interpretation for transferring information from the privileged to the original data space. Our strategy is to check whether each example is easy-to-classify or hard-to-classify based on the margin distance to the classifying hyperplane in the privileged space. Subsequently, we transfer this knowledge to the original space. We hypothesize that knowing a priori which examples are easy to classify and

[1] http://cvxopt.org

Algorithm 2 Margin Transfer from $\mathcal{X}^*$ to $\mathcal{X}$

```
**Input** original data $X$, privileged data $X^*$, labels $Y$, tolerance $\epsilon \geq 0$
$f^* \leftarrow$ SVM (5.1) trained on $(X^*, Y)$
$\rho_i = \max\{y_i f^*(x_i^*),\ \epsilon\}$    (per-sample margin)
$f \leftarrow$ SVM (5.6) trained on $(X, Y)$ using $\rho_i$ instead of unit margin.
**Return** $f : \mathcal{X} \rightarrow \mathbb{R}$
```

which are hard during learning should improve the prediction performance in the original data space. This consideration leads us to the Margin Transfer method, summarized in Algorithm 2.

First, we train an ordinary SVM (5.1) on X^*. The resulting prediction function $f^*(x^*) = \langle w^*, x^* \rangle$ is used to compute the margin distance from the training samples to the classifying hyperplane in the privileged space[1] $\rho_i := y_i f^*(x_i^*)$. We omit the explicit computation of the bias term b in the Algorithm 2, assuming it is implicitly added to the weight vector w, and all data points are augmented with a unit element. Examples with large values of ρ_i are considered easy to classify, whereas small or even negative values of ρ_i indicate hard or even impossible to classify samples. We then train a standard SVM on X, aiming for a *data-dependent margin* ρ_i transferred from the privileged space rather than enforcing a constant margin of 1. The corresponding optimization problem is:

$$\underset{w \in \mathbb{R}^d,\, \xi_i \in \mathbb{R}}{\text{minimize}} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \tag{5.6a}$$

$$\text{subject to, for all } \; i = 1, \ldots, N$$

$$y_i \langle w, x_i \rangle \geq \rho_i - \xi_i \quad \text{and} \quad \xi_i \geq 0. \tag{5.6b}$$

One can see that examples with small and negative values of ρ_i have limited influence on w compared to the standard SVM, because their slacks ξ_i can easily compensate for the inequality constraint. We threshold the negative values of margin at certain tolerance value ϵ, $\epsilon \geq 0$. Our interpretation is that if it was not possible to correctly classify a sample in the privileged space, it will also be impossible to do so in the, presumably weaker, original space. Forcing the optimization to solve a hopeless task would only lead to overfitting and reduced prediction accuracy.

[1] Note that in the standard SVM formulation one would compute the values of slack variables to know how far the sample is from the hyperplane. As slack variables appear only at the training phase, we deliberately evaluate the prediction function on the same data it was trained on to identify easy and hard samples *at train* time.

Numeric Optimization Both learning steps in the *Margin Transfer* method are convex optimization problems. Furthermore, in contrast to SVM+, we can use standard SVM packages to solve them, including efficient methods working in primal representation (Chapelle, 2007), and solvers based on stochastic gradient descent (Shalev-Shwartz et al., 2007).

For the SVM with data-dependent margin (5.6), we do the following reparameterization: we divide each constraint (5.6b) by the corresponding ρ_i, which is possible after thresholding at the non-negative tolerance value. For our experiments, we threshold at $\epsilon = 0.1$, thereby preventing numeric instabilities and increasing the computational efficiency of the method. Changing variables from x_i to $\hat{x}_i = \frac{x_i}{\rho_i}$ and from ξ_i to $\hat{\xi}_i = \frac{\xi_i}{\rho_i}$ we obtain the equivalent optimization problem:

$$\underset{w \in \mathbb{R}^d,\, \hat{\xi}_i \in \mathbb{R}}{\text{minimize}} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \rho_i \hat{\xi}_i \tag{5.7a}$$

$$\text{subject to, for all } \; i = 1, \dots, N$$

$$y_i \langle w, \hat{x}_i \rangle \geq 1 - \hat{\xi}_i \quad \text{and} \quad \hat{\xi}_i \geq 0. \tag{5.7b}$$

This corresponds to standard SVM optimization with training examples $\hat{x}_i$, where each slack variable has an individual weight $C\rho_i$ in the objective. Many existing SVM packages support such per-sample weights, in our experiments we use LIBLINEAR (Fan et al., 2008). Additionally we would like to position our model in support of recent results that SVM+ classifiers can be reformulated as a special form of example-weighted binary SVMs (Lapin et al., 2014).

5.3.3 How is Information being Transferred?

We elaborate on how SVM+ and Margin Transfer instantiate the easy-hard interpretation and how they differ from each other.

Observation 1: Both methods, SVM+ and Margin Transfer, concentrate on learning easy samples and de-emphasizing the hard ones. Though SVM+ and Margin Transfer aim at the same goal, the way this is achieved is different in these two methods. Let us illustrate this by using the oracle analogy. In the SVM+, the oracle gives us the value of the slack function $\texttt{oracle}_{\text{svm+}}(x_i) := \langle w^*, x_i^* \rangle + b^*$ for example x_i, and in the Margin Transfer, the oracle gives us the margin distance to the classifying hyperplane $\texttt{oracle}_{\text{margin transfer}}(x_i) := y_i f^*(x_i^*)$.

Suppose we only have two training samples, x_1 and x_2, and we ask the oracles what they perceive about the two samples. Say, in case of SVM+, we get back the following answers: $\texttt{oracle}_{\text{svm+}}(x_1) = 10.0$ and $\texttt{oracle}_{\text{svm+}}(x_2) = 0.0$. This means that the first sample is hard (its slack variable is high) and the second one is easy (its slack variable is zero). When we encode this into the optimization problem of SVM+, we can see that the constraint (5.5b) becomes $y_1[\langle w, x_1\rangle + b] \geq -9$, (effort*less* to satisfy compared to the unit margin in the standard SVM method) for the first sample and $y_2[\langle w, x_2\rangle + b] \geq 1$ (effort*ful* to satisfy compared to the standard SVM method) for the second one. So this means that the optimization task would more or less ignore the constraint of the first sample (that is hard) and concentrate on satisfying the constraint about the second sample (that is easy).

We repeat the questions to the Margin Transfer oracle and say the answers are: $\texttt{oracle}_{\text{margin transfer}}(x_1) = -5$ and $\texttt{oracle}_{\text{margin transfer}}(x_2) = 8$. Interpreting the oracle's answers lead us to conclude that the first sample is hard (its margin distance is zero or negative) and the second one is easy (its margin distance is positive). When we encode this into the optimization problem of Margin Transfer, the constraint (5.6b) becomes $y_1 \langle w, x_1\rangle \geq \epsilon - \xi_1$ (effortless to satisfy) for the first sample and $y_2 \langle w, x_2\rangle \geq 8 - \xi_2$ (effortful to satisfy) for the second one. As before, the optimization task would ignore the constraints of the hard samples and concentrate on learning the easy ones. This is despite the fact that the SVM+ oracle returns high values for hard samples while the Margin Transfer oracle returns low values for hard samples, and vice versa for easy ones.

Observation 2: Classification performance in the privileged space matters for Margin Transfer but not for SVM+. At the core of SVM+ lies the idea of imitating the oracle by learning the non-negative linear regression slack function defined in the privileged space. The information about labels does not come into play when modeling the slack function, so in a sense, we never validate the classification performance in the privileged space. In contrast, in the Margin Transfer method, the performance in the privileged space explicitly guides the training of the predictor in the original data space. Samples that are easy and hard to classify in the privileged space directly define the margin for the samples in the original data space.

In our Rank Transfer method (Sharmanska et al., 2013) we observe another way to do information transfer considering pairs of samples. For any pair of samples from different classes we estimate whether it is easy-to-separate or hard-to-separate pair based on the rank margin between the samples in the privileged space. We transfer this information into the ranking SVM objective and completely ignore pairs that got swapped. In this framework, we deal with pairs of samples and therefore suffer from quadratic amount of constraints to be satisfied.

5.4 Experiments

In our experimental setting we study three different types of privileged information, showing that all of these can be handled in a unified framework, where previously hand crafted methods were used. We consider semantic attributes, bounding box annotation and textual description as sources of privileged information if these are present at training time but not at test time. As we will see, some modalities are more suitable for transferring the margin than others. We will discuss this in the following subsections.

Methods. We analyze two methods of learning using privileged information: our proposed Margin Transfer method and the SVM+ method (Pechyony & Vapnik, 2011). We compare the results with the ordinary SVM method when learning on the original space $\mathcal{X}$ directly. We also provide as a reference the performance of SVM in the privileged space $\mathcal{X}^*$, as if we had access to the privileged information during testing.

Evaluation metric. To evaluate the performance of the methods we use accuracy, and we report mean and standard error across 20 repeats.

Model selection. For the LUPI methods, we perform a joint cross validation model selection approach for choosing the regularization parameters in the original and privileged spaces. In the SVM+ method these are C and γ (5.5a), and in the Margin Transfer these are C's in the two-stage procedure (5.1a), (5.6a). For the methods that do not use privileged information there is only a regularization parameter C to be cross validated. In the privileged space we select over 7 parameters $\{10^{-3}, \dots, 10^{3}\}$. We use the same range in the original space if the data is L_2 normalized, and the range $\{10^{0}, \dots, 10^{5}\}$ for L_1 normalized data. In our experiments we use 5x5 fold cross validation scheme for binary classification and 5 fold cross validation for the multiclass setting. The best parameter (or pair of parameters) found is used to retrain the complete training set. Based on our experience, LUPI methods require very thorough model selection. To couple the modalities of privileged and original data spaces properly, the grid search over both parameter spaces has to be exploited.

5.4.1 Attributes as privileged information

We use the *Animals with Attributes (AwA)* dataset (Lampert et al., 2009, 2013) and focus on the default 10 test classes: *chimpanzee, giant panda, leopard, persian cat, pig, hippopotamus, humpback whale, raccoon, rat, seal* with 6180 images

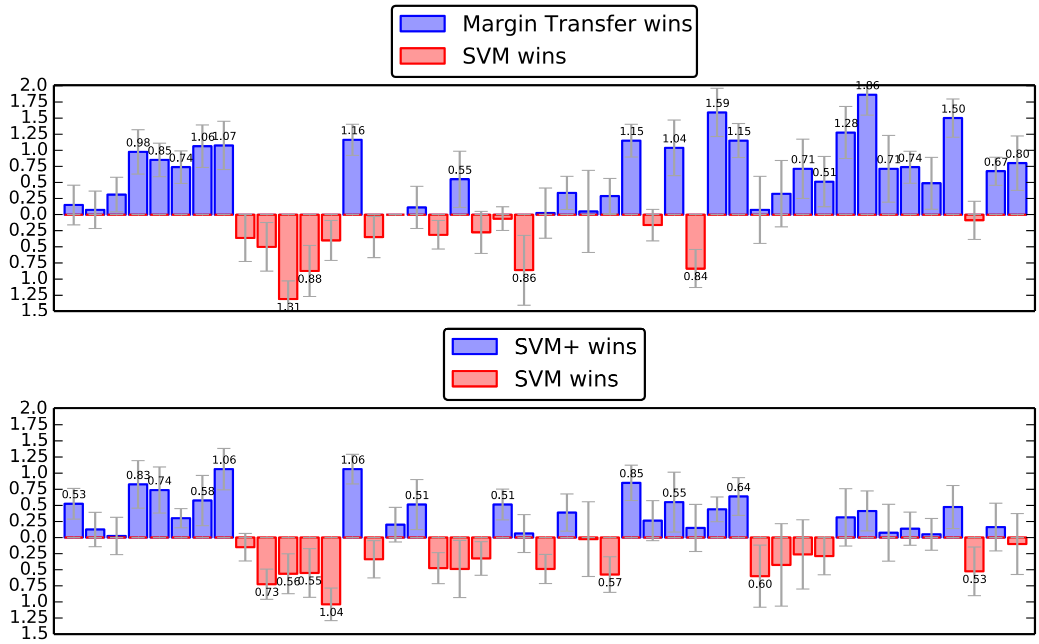


Figure 5.2: AwA dataset (attributes as privileged information). Pairwise comparison of the methods that utilize privileged information (Margin Transfer, SVM+) and their baseline counterpart (SVM) is shown via differences in accuracy performance. The length of the 45 bars corresponds to relative improvement of the accuracy for 45 cases.

in total. We use L_1-normalized 2000-dimensional SURF descriptors as original features, and 85-dimensional predicted attributes as the privileged information. The values of the semantic attributes are obtained from the DAP model (Lampert et al., 2009, 2013) described in Section 2.1.3, and are provided together with the dataset. We train 45 binary classifiers, one for each pair of the 10 classes with 100 images per class as training data. We use 200 samples per class for testing. To get better statistics of the performance we repeat the procedure of train/test split 20 times.

Results. As we can see from the Figure 5.2, utilizing attributes as privileged information for object classification task is useful. Margin Transfer outperforms SVM in 32 out of 45 cases, and SVM+ outperforms SVM in 27 out of 45 cases. Noticeably, the Margin Transfer model is able to utilize privileged information better than the SVM+. We observe partial overlap of cases where Margin Transfer and SVM+ are not able to utilize privileged information (location of the red bars).

		SVM	Margin Transfer	SVM+	Reference
		image	image+attributes	image+attributes	*(SVM attributes)*
1	Chimpanzee vs Giant panda	83.25 ± 0.53	83.40 ± 0.43	**83.77 ± 0.48**	*85.00 ± 0.42*
2	Chimpanzee vs Leopard	86.63 ± 0.35	86.71 ± 0.38	**86.76 ± 0.35**	*92.95 ± 0.27*
3	Chimpanzee vs Persian cat	83.91 ± 0.46	**84.22 ± 0.41**	83.93 ± 0.49	*91.42 ± 0.31*
4	Chimpanzee vs Pig	79.72 ± 0.35	**80.70 ± 0.26**	80.55 ± 0.27	*86.53 ± 0.43*
5	Chimpanzee vs Hippopotamus	81.05 ± 0.28	**81.90 ± 0.27**	81.78 ± 0.29	*88.12 ± 0.29*
6	Chimpanzee vs Humpback whale	94.45 ± 0.26	**95.18 ± 0.21**	94.75 ± 0.24	*98.32 ± 0.16*
7	Chimpanzee vs Raccoon	80.11 ± 0.50	**81.17 ± 0.48**	80.68 ± 0.45	*85.47 ± 0.38*
8	Chimpanzee vs Rat	80.15 ± 0.43	**81.22 ± 0.43**	81.21 ± 0.42	*90.03 ± 0.48*
9	Chimpanzee vs Seal	**85.80 ± 0.26**	85.43 ± 0.44	85.65 ± 0.31	*91.12 ± 0.24*
10	Giant panda vs Leopard	**87.82 ± 0.32**	87.32 ± 0.37	87.10 ± 0.35	*92.52 ± 0.31*
11	Giant panda vs Persian cat	**87.66 ± 0.37**	86.35 ± 0.30	87.10 ± 0.28	*88.92 ± 0.40*
12	Giant panda vs Pig	**80.80 ± 0.44**	79.92 ± 0.41	80.25 ± 0.39	*84.57 ± 0.40*
13	Giant panda vs Hippopotamus	**85.36 ± 0.41**	84.96 ± 0.51	84.32 ± 0.43	*90.36 ± 0.34*
14	Giant panda vs Humpback whale	94.30 ± 0.30	**95.46 ± 0.25**	95.36 ± 0.24	*98.36 ± 0.15*
15	Giant panda vs Raccoon	**83.52 ± 0.44**	83.17 ± 0.44	83.18 ± 0.44	*84.08 ± 0.30*
16	Giant panda vs Rat	81.76 ± 0.45	81.76 ± 0.38	**81.96 ± 0.43**	*87.60 ± 0.27*
17	Giant panda vs Seal	85.47 ± 0.41	85.58 ± 0.40	**85.98 ± 0.31**	*89.42 ± 0.35*
18	Leopard vs Persian cat	**90.18 ± 0.23**	89.87 ± 0.26	89.71 ± 0.28	*93.32 ± 0.25*
19	Leopard vs Pig	81.20 ± 0.42	**81.75 ± 0.43**	80.71 ± 0.22	*91.01 ± 0.26*
20	Leopard vs Hippopotamus	**86.37 ± 0.33**	86.10 ± 0.31	86.05 ± 0.26	*91.36 ± 0.31*
21	Leopard vs Humpback whale	95.26 ± 0.34	95.20 ± 0.36	**95.77 ± 0.20**	*98.60 ± 0.10*
22	Leopard vs Raccoon	77.40 ± 0.51	76.53 ± 0.76	**77.46 ± 0.50**	*81.36 ± 0.35*
23	Leopard vs Rat	81.82 ± 0.26	**81.85 ± 0.40**	81.33 ± 0.35	*90.55 ± 0.23*
24	Leopard vs Seal	87.28 ± 0.36	87.62 ± 0.36	**87.67 ± 0.31**	*92.36 ± 0.31*
25	Persian cat vs Pig	76.28 ± 0.61	**76.33 ± 0.62**	76.26 ± 0.58	*78.16 ± 0.35*
26	Persian cat vs Hippopotamus	84.85 ± 0.56	**85.13 ± 0.46**	84.27 ± 0.48	*90.62 ± 0.35*
27	Persian cat vs Humpback whale	91.81 ± 0.32	**92.96 ± 0.26**	92.66 ± 0.30	*97.87 ± 0.16*
28	Persian cat vs Raccoon	84.50 ± 0.44	84.33 ± 0.44	**84.76 ± 0.42**	*85.65 ± 0.32*
29	Persian cat vs Rat	65.32 ± 0.38	**66.36 ± 0.51**	65.87 ± 0.48	*65.51 ± 0.71*
30	Persian cat vs Seal	80.61 ± 0.38	79.77 ± 0.48	**80.76 ± 0.57**	*86.65 ± 0.35*
31	Pig vs Hippopotamus	71.26 ± 0.38	**72.85 ± 0.42**	71.70 ± 0.37	*77.77 ± 0.61*
32	Pig vs Humpback whale	91.31 ± 0.38	**92.46 ± 0.32**	91.95 ± 0.44	*97.41 ± 0.14*
33	Pig vs Raccoon	75.63 ± 0.51	**75.71 ± 0.39**	75.03 ± 0.28	*82.28 ± 0.36*
34	Pig vs Rat	67.80 ± 0.44	**68.12 ± 0.40**	67.37 ± 0.59	*74.12 ± 0.38*
35	Pig vs Seal	76.10 ± 0.50	**76.81 ± 0.38**	75.83 ± 0.37	*82.11 ± 0.24*
36	Hippopotamus vs Humpback whale	86.67 ± 0.40	**87.18 ± 0.38**	86.38 ± 0.38	*95.02 ± 0.25*
37	Hippopotamus vs Raccoon	80.61 ± 0.61	**81.88 ± 0.60**	80.92 ± 0.52	*87.62 ± 0.28*
38	Hippopotamus vs Rat	77.83 ± 0.45	**79.70 ± 0.41**	78.25 ± 0.48	*88.41 ± 0.37*
39	Hippopotamus vs Seal	67.91 ± 0.60	**68.62 ± 0.55**	67.98 ± 0.49	*73.32 ± 0.50*
40	Humpback whale vs Raccoon	92.52 ± 0.22	**93.26 ± 0.30**	92.66 ± 0.30	*97.21 ± 0.19*
41	Humpback whale vs Rat	89.22 ± 0.45	**89.71 ± 0.33**	89.27 ± 0.37	*97.27 ± 0.15*
42	Humpback whale vs Seal	80.67 ± 0.37	**82.17 ± 0.34**	81.15 ± 0.33	*89.37 ± 0.34*
43	Raccoon vs Rat	**73.75 ± 0.38**	73.66 ± 0.43	73.22 ± 0.52	*77.92 ± 0.36*
44	Raccoon vs Seal	83.86 ± 0.42	**84.53 ± 0.41**	84.02 ± 0.45	*87.32 ± 0.25*
45	Rat vs Seal	74.03 ± 0.56	**74.83 ± 0.52**	73.93 ± 0.46	*86.46 ± 0.23*

Table 5.1: AwA dataset (attributes as privileged information). The numbers shown are mean and standard error of accuracy over 20 runs. The best result is highlighted in **boldface**, which in total is **9** for `SVM`, **27** for `Margin Transfer`, and **9** for `SVM+`. Blue highlighting indicates significant improvement of the methods that utilize privileged information over the methods that do not. We used a paired Wilcoxon test with 95% confidence level as a reference. Additionally, we provide the SVM performance on $\mathcal{X}^*$ (last column).

The red bars coincide mostly in pairs with *giant panda* or *leopard* versus other animals. Full comparison of the accuracy of all methods is shown in Table 5.1. We also notice that the gain of the Margin Transfer method is higher in the regime when the problem is hard, i.e. when accuracy is below 90%. As a further analysis, we also check the hypothetical performance of SVM in the privileged space $\mathcal{X}^*$. The privileged information has consistently higher accuracy than SVM in the original space $\mathcal{X}$. In most cases, higher accuracy in the privileged space than in the original space translates to a positive effect in Margin Transfer. We credit this to the fact that Margin Transfer relies on the performance in the privileged space in order to explore easiness and hardness of the samples. And it is successful if the underlying assumption that the same examples are easy or hard in both modalities is fulfilled, as it is in most of the cases here.

5.4.2 Bounding box as privileged information

Bounding box annotation is designed to capture the exact location of an object in the image. It is usually represented as a box around the object. When performing image-level object recognition, knowing the exact location of the object in the training data is privileged information. We use a subset of the categories from the ImageNet 2012 challenge (ILSVRC2012) for which bounding box annotation is available[1]. We define two groups of interest: one with a variety of snakes, and the other with balls in different sport activities. The "snakes" group has 17 classes: *thunder snake, ringneck snake, hognose snake, green snake, king snake, garter snake, water snake, vine snake, night snake, boa constrictor, rock python, indian cobra, green mamba, sea snake, horned viper, diamondback, sidewinder*, and has 8254 images in total, on average 500 samples per class. We ignore a few images where the bounding box region is too small, and use 8227 images for further analysis. The "balls" group has 6 classes: *soccer ball, croquet ball, golf ball, ping-pong ball, rugby ball, tennis ball*, and has 3259 images in total, on average 500 samples per class. Here, we also ignore images with uninformative bounding box annotation and use 3165 images instead. We consider a one-versus-rest scenario for each group separately. We use L_2-normalized 4096-dimensional Fisher vectors (Perronnin et al., 2010) described in our Section 2.2.2, that are extracted from the whole images as well as from only the bounding box regions, and we use the former as the original data representation and the latter as privileged information. We train one binary classifier for each class, 17 in the first group and

[1] `http://www.image-net.org/challenges/LSVRC/2012/index`

		SVM	Margin Transfer	SVM+	Reference
		image	image+bbox	image+bbox	*(SVM bbox)*
1	Thunder snake	65.76 ± 0.85	**66.28 ± 0.86**	65.43 ± 0.82	*66.51 ± 0.85*
2	Ringneck snake	67.28 ± 0.67	67.68 ± 0.62	**68.40 ± 0.60**	*68.45 ± 0.48*
3	Hognose snake	65.10 ± 0.69	65.28 ± 0.66	**68.32 ± 0.43**	*65.92 ± 0.62*
4	Green snake	73.42 ± 0.44	**73.89 ± 0.38**	73.40 ± 0.40	*74.46 ± 0.36*
5	King snake	76.31 ± 0.45	76.67 ± 0.41	**78.51 ± 0.48**	*77.56 ± 0.46*
6	Garter snake	72.92 ± 0.55	73.20 ± 0.47	**76.70 ± 0.53**	*75.35 ± 0.52*
7	Water snake	67.48 ± 0.55	67.75 ± 0.57	**68.40 ± 0.42**	*65.35 ± 0.52*
8	Vine snake	79.42 ± 0.33	79.26 ± 0.33	**79.98 ± 0.29**	*80.67 ± 0.46*
9	Night snake	57.42 ± 0.65	57.62 ± 0.70	**58.07 ± 0.57**	*56.51 ± 0.53*
10	Boa constrictor	72.85 ± 0.57	72.68 ± 0.53	**75.34 ± 0.60**	*72.62 ± 0.49*
11	Rock python	63.26 ± 0.59	63.29 ± 0.64	**65.79 ± 0.44**	*63.20 ± 0.41*
12	Indian cobra	64.29 ± 0.57	**64.59 ± 0.59**	64.51 ± 0.62	*63.29 ± 0.69*
13	Green mamba	72.56 ± 0.46	72.89 ± 0.49	**73.14 ± 0.50**	*73.60 ± 0.51*
14	Sea snake	80.29 ± 0.41	80.23 ± 0.38	**80.82 ± 0.37**	*77.67 ± 0.36*
15	Horned viper	69.75 ± 0.51	69.73 ± 0.48	**71.43 ± 0.55**	*72.53 ± 0.42*
16	Diamondback	75.39 ± 0.50	75.64 ± 0.43	**77.21 ± 0.45**	*76.01 ± 0.51*
17	Sidewinder	**68.85 ± 0.42**	68.53 ± 0.57	68.53 ± 0.59	*69.84 ± 0.57*

Table 5.2: ImageNet dataset, the ”snakes“ group (bounding box annotation as privileged information). The numbers shown are mean and standard error of accuracy over 20 runs. The best result is highlighted in **boldface**. Blue highlighting indicates significant improvement of the methods that utilize privileged information over the methods that do not. We used a paired Wilcoxon test with 95% confidence level as a reference. Additionally, we also provide the SVM performance on $\mathcal{X}^*$ (last column).

6 in the second group. For training we balance the number of positive samples (from the desired class) and negative samples formed from the remaining classes, i.e. 16 and 5 for two groups accordingly. In the ”snakes“ group, we use 160 versus 160 images randomly drawn from the desired class and from the remaining 16 classes (10 from each). We used the same amount of samples for testing. In the ”balls“ group, we use 100 versus 100 images for training randomly drawn from the desired class and from the remaining 5 classes (20 from each). To keep the setting similar across datasets, we used twice as many samples for testing. To get better statistics of the performance we repeat each train/test split 20 times.

Results. As we can see from Table 5.2 and Table 5.3, utilizing bounding box annotation as privileged information for fine-grained classification is useful. In

		SVM	Margin Transfer	SVM+	Reference
		image	image+bbox	image+bbox	*(SVM bbox)*
1	Soccer ball	65.95 ± 0.66	65.95 ± 0.66	$\mathbf{67.42 \pm 0.67}$	$\mathit{69.78 \pm 0.44}$
2	Croquet ball	73.31 ± 0.38	73.70 ± 0.39	$\mathbf{73.80 \pm 0.40}$	$\mathit{74.76 \pm 0.39}$
3	Golf ball	$\mathbf{76.46 \pm 0.47}$	76.18 ± 0.52	75.95 ± 0.52	$\mathit{68.53 \pm 0.46}$
4	Ping-pong ball	71.80 ± 0.54	71.71 ± 0.50	$\mathbf{72.85 \pm 0.44}$	$\mathit{71.20 \pm 0.59}$
5	Rugby ball	76.08 ± 0.40	76.00 ± 0.43	$\mathbf{82.90 \pm 0.29}$	$\mathit{71.07 \pm 0.57}$
6	Tennis ball	67.57 ± 0.48	67.65 ± 0.44	$\mathbf{68.17 \pm 0.45}$	$\mathit{65.36 \pm 0.71}$

Table 5.3: ImageNet dataset, the ”balls“ group (bounding box annotation as privileged information). The numbers shown are mean and standard error of accuracy over 20 runs. The best result is highlighted in **boldface**. Blue highlighting indicates significant improvement of the methods that utilize privileged information over the methods that do not. We used a paired Wilcoxon test with 95% confidence level as a reference. Additionally, we also provide the SVM performance on $\mathcal{X}^*$ (last column).

both tables, the LUPI methods outperform the non-LUPI SVM baseline in all but 1 case. In the group of snakes, SVM+ clearly outperforms SVM in 14 cases, and Margin Transfer outperforms SVM in 12 cases out of 17. In this experiment, the SVM+ method is able to exploit the privileged information much better than the Margin Transfer method (in 13 out of 17 cases, and 1 tie in the case of Sidewinder snake). In the group of balls, we observe very similar results with clear advantage of the SVM+ method over all other methods. Margin Transfer shows only a minor difference with respect to standard SVM.

Noticeably, the performance in the privileged space is not superior to the original data space, sometimes it is even worse, especially in the ”balls“ group. Since our Margin Transfer method relies directly on the performance in the privileged space, its ability to exploit easy and hard samples is limited in this scenario. On the other hand, modeling the slacks in the form of a regression model, as SVM+ does, works well. We suspect it is more suitable when the privileged and original spaces are of the same modality, as in this case, where the privileged information is obtained from a subset of the same image features that are used for the original data representation.

5.4.3 Textual description as privileged information

Textual description provides a complementary view to a visual representation of an object. This can be used as privileged information in an object classification task. We use two datasets to explore textual description as the source of privileged information and we will describe them in turn. The first dataset is *IsraelImages*[1] introduced by Bekkerman & Jeon (2007). The dataset has 11 classes, 1823 images in total, with a textual description (up to 18 words) attached to each of the image. The number of samples per class is relatively small, around 150 samples, and varies from 96 to 191 samples. We merge the classes into three groups: nature (birds, trees, flowers, desert), religion (Christianity, Islam, Judaism, symbols) and urban (food, housing, personalities), and perform binary classification on the pairs of groups. We use L_2-normalized 4096-dimensional Fisher vectors (Perronnin et al., 2010) extracted from the images as the original data representation and bag-of-words representation of the text data as privileged information. We use 100 images per group for training and 200 per group for testing.

The second dataset is the *Attribute Discovery* dataset[2] introduced by Berg et al. (2010) for the purpose of automatic attribute discovery from the Web images and their textual descriptions. To avoid confusion with attribute descriptions as privileged information, we will call this dataset *Accessories* as it contains accessory products taken from variety of e-commerce sources with the images and their textual descriptions. The products are grouped into 4 broad shopping categories: *bags, earrings, ties, and shoes.* We randomly select 1800 samples from this dataset for our experiments, 450 samples from each category. We generated 6 binary classification tasks for each pair of the 4 classes with 100 samples per class for training and 200 per class for testing. This second dataset contains longer text descriptions than the *IsraelImages* dataset, which allows us to use advanced features in terms of word-vectors instead of simple word frequency features. We extracted 200-dimensional word-vector representations using a neural network skip-gram architecture (Mikolov et al., 2013)[3]. Then we constructed a codebook of 100 word-vectors to convert this word representation into a fixed-length sentence representation and apply L_1 normalization. In the original data space, we use L_1-normalized bag-of-words histograms based on SURF descriptors with the 100 visual words codebook.

[1] http://people.cs.umass.edu/~ronb/image_clustering.html

[2] http://tamaraberg.com/attributesDataset/index.html

[3] https://code.google.com/p/word2vec/

		SVM	Margin Transfer	SVM+		Reference
		image	image+text	image+text		*(SVM text)*
1	Nature vs Religion	81.01 ± 0.49	$\mathbf{81.15 \pm 0.51}$	81.08 ± 0.50		*84.52 ± 0.47*
2	Religion vs Urban	65.55 ± 0.43	$\mathbf{65.62 \pm 0.50}$	65.52 ± 0.57		*88.12 ± 0.44*
3	Nature vs Urban	78.97 ± 0.42	78.73 ± 0.49	$\mathbf{79.15 \pm 0.53}$		*83.11 ± 0.56*

Table 5.4: Israeli dataset (textual description as privileged information). The numbers shown are mean and standard error of the accuracy over 20 runs. As reference we also provide the SVM performance on the $\mathcal{X}^*$ (last column).

		SVM	Margin Transfer	SVM+		Reference
		image	image+text	image+text		*(SVM text)*
1	Earrings vs Bags	90.25 ± 0.27	$\mathbf{90.48 \pm 0.30}$	89.73 ± 0.32		*97.87 ± 0.18*
2	Earrings vs Shoes	92.65 ± 0.20	$\mathbf{92.81 \pm 0.13}$	92.46 ± 0.23		*98.58 ± 0.16*
3	Bags vs Ties	87.70 ± 0.44	$\mathbf{88.08 \pm 0.43}$	87.80 ± 0.45		*98.62 ± 0.13*
4	Bags vs Shoes	$\mathbf{90.52 \pm 0.24}$	90.28 ± 0.25	88.61 ± 0.40		*96.26 ± 0.25*
5	Ties vs Earrings	$\mathbf{90.27 \pm 0.28}$	90.11 ± 0.24	88.97 ± 0.25		*99.57 ± 0.07*
6	Ties vs Shoes	$\mathbf{81.60 \pm 0.61}$	81.53 ± 0.48	80.90 ± 0.51		*98.96 ± 0.15*

Table 5.5: Accessories dataset (textual description as privileged information). The numbers shown are mean and standard error of the accuracy over 20 runs. As reference we also provide the SVM performance on the $\mathcal{X}^*$ (last column).

Results. As we can see from Table 5.4 and Table 5.5, utilizing textual privileged information as provided in the *IsraelImages* and *Accessories* datasets does not help in our scenario. All methods, LUPI and non-LUPI, have near equal performance, and there is no indication of privileged information being utilized in both LUPI methods. This might seem contradictory to the high performance of the reference baseline in the text domain, $\mathcal{X}^*$. However high accuracy in the privileged space does not necessarily mean that the privileged information is helpful. For example, assume we used the labels themselves as the privileged modality: classification would be trivial, but it would provide no additional information to transfer. In the *IsraelImages* dataset, the textual descriptions of the images are very sparse and contain many duplicates, and in the *Accessories* datasets the texts are "too easy". Therefore, the margin distance in the privileged space does not capture the easiness and hardness of different samples, and mainly preserves the class separation only. Nevertheless, the performance does not degrade.

5.5 Multiclass classification

We also explore the benefits of utilizing the LUPI methods in the multiclass setup with a one-versus-rest learning strategy. We train one binary classifier for each class to distinguish samples of this class (positive label) from samples of the remaining classes (negative label). For a test point, the label is assigned based on the class with maximum prediction value over all binary classifiers. For model selection, we use 5 fold cross validation and search over the range of regularization parameters the same as before. In order to calibrate the prediction scores from different classifiers we use one parameter value to train all the binary classifiers, and cross validate the multiclass performance. The best parameter (pair of parameters) is used to retrain all classifiers. We run the multiclass setting on all the datasets described previously for the binary classification task. The results are summarized in Table 5.6.

`Dataset`	$\mathcal{X}^*$	`SVM`	`Margin Transfer`	`SVM+`	`Reference`
AwA (10 classes)	attributes	45.41 ± 0.18	$\mathbf{46.44 \pm 0.27}$	42.07 ± 0.36	$\mathit{56.18 \pm 0.21}$
Snakes (17 classes)	bbox	30.41 ± 0.18	$\mathbf{31.61 \pm 0.19}$	31.09 ± 0.24	$\mathit{31.84 \pm 0.15}$
Sport Balls (6 classes)	bbox	51.78 ± 0.26	51.65 ± 0.36	$\mathbf{52.75 \pm 0.35}$	$\mathit{49.47 \pm 0.29}$
Israeli (3 groups)	text	60.16 ± 0.41	$\mathbf{60.65 \pm 0.46}$	60.14 ± 0.42	$\mathit{76.37 \pm 0.43}$
Accessories (4 classes)	text	76.45 ± 0.28	$\mathbf{76.48 \pm 0.26}$	72.68 ± 0.37	$\mathit{97.00 \pm 0.16}$

Table 5.6: Multiclass performance. The numbers shown are mean and standard error of accuracy over 20 runs. The best result is highlighted in **boldface**. Additionally, as reference we also provide the performance on $\mathcal{X}^*$ (last column).

Results. As we can see from Table 5.6, utilizing privileged information is useful in the studied multiclass setting. The LUPI methods outperform the non-LUPI baseline (SVM) in all datasets. Overall the Margin Transfer is superior to SVM+ in all but one case (Sport Balls); it is more stable contrary to performance drop of the SVM+ in the *AwA* and *Accessories* datasets; and it follows the tendency to outperform SVM when performance in the privileged space (column `Reference`) is better than in the original space (column `SVM`).

5.6 Human annotation as privileged data

For this experiment, we collect Mechanical Turk[1] annotation of images to define easy and hard samples for human learning. We analyze the advantages of having this information in comparison to the LUPI methods. We managed to collect reliable human annotation for 8 out of 10 classes in the AwA dataset: *chimpanzee, giant panda, leopard, persian cat, hippopotamus, raccoon, rat, seal.* For the remaining two classes, *pig* and *humpback whale* we could not obtain reliable annotations. Many images of the class *pig* are related to food products rather than the animal itself, which creates a clear bias in user judgments. Images in the class *humpback whale* lack variability across samples, which makes it difficult to distinguish between easy and hard ones. In our user study, the participant is shown a set of images of one particular class and is asked to select the most prominent (easiest) images first, then proceed to less obvious, and so on, until the most difficult samples are left. We aggregate this ranking information across overlapping sets of images to compute a global order of images per category. The score is in the range from 1 (hardest) to 16 (easiest). We observe that most of the time, the easiest instances are those with a clearly visible object of interest in the center of the image, whereas the hardest are occluded objects, small sized, or images that also show humans.

Evaluation. First, we analyze the advantage of transferring the information from human annotation by looking into accuracy performance. In order to map the easy-hard score into the margin distance ρ_i, we use linear scaling of the score to the interval $[0, 2]$, so values between 0 and 1 correspond to hard samples and values greater than 1 correspond to easy samples. After this we proceed directly to the second stage of the Margin Transfer method (5.6). We report the results over 28 pairs of classes in Table 5.7 (last column).

Secondly, we study whether the easy-hard score according to human understanding correlates with easy and hard samples that we identify when learning in the privileged space of attributes. We use the Kendall tau rank correlation analysis and compute the correlation coefficient across 28 learning tasks. For each task, we compute the correlation between the margin distance from the training samples to the classifying hyperplane in the privileged space of attributes, $\langle w^*, x_i^* \rangle$, and the easy-hard scores obtained from the human annotation $y_i \, \text{score}(x_i)$. Similarly we evaluate the correlation between easy and hard samples in the original

[1] https://www.mturk.com/mturk/

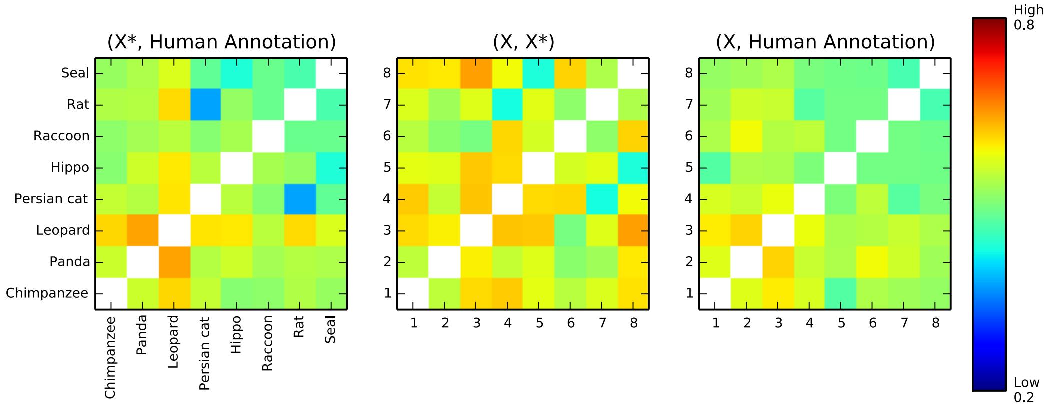


Figure 5.3: Human annotation as privileged information. Kendall tau rank correlation analysis is used to explore the correlation between easy-hard samples defined by human annotation and easy and hard samples that we identify when learning in the privileged space (left). We analyze the correlation between easy and hard samples in the privileged and original data spaces (center), and between human annotation and samples in the original data space (right).

and privileged spaces that the Margin Transfer method relies on. For this, we compute the correlation between the predicted values of the classifiers trained on data X from the original space and on X^* from the privileged space of attributes. To complete this analysis we also look at correlation between user defined easy-hard scores and easy and hard samples in the original space. For visualization, we aggregate the results into a symmetric table, where each entry is the tau coefficient computed for the corresponding pair of classes in the binary learning task, refer to Figure 5.3.

Results. As we can see from Table 5.7, collecting good quality human annotation can help to improve the classification performance, however it cannot solve the problem of negative transfer from privileged space to the original space. In some cases it clearly helps, such as classifying the categories *giant panda* and *leopard* versus others, and in other cases it does not, as in the category *chimpanzee.*

As we can see from Figure 5.3, overall the correlation between easy-hard samples in the privileged and the original data spaces (center) has higher signal than between human annotation versus privileged data space (left) and versus original data space (right). If we look closely at the case with *giant panda* (second column across the tables), we observe that indeed the correlation between human anno-

		SVM	Margin Transfer	Margin Transfer
		image	image+attributes	image+human annot.
1	Chimpanzee vs Giant panda	83.25 ± 0.53	83.40 ± 0.43	**83.72 ± 0.58**
2	Chimpanzee vs Leopard	86.63 ± 0.35	**86.71 ± 0.38**	86.43 ± 0.30
3	Chimpanzee vs Persian cat	83.91 ± 0.46	**84.22 ± 0.41**	83.93 ± 0.38
4	Chimpanzee vs Hippopotamus	81.05 ± 0.28	**81.90 ± 0.27**	80.88 ± 0.29
5	Chimpanzee vs Raccoon	80.11 ± 0.50	**81.17 ± 0.48**	80.76 ± 0.55
6	Chimpanzee vs Rat	80.15 ± 0.43	**81.22 ± 0.43**	79.91 ± 0.42
7	Chimpanzee vs Seal	**85.80 ± 0.26**	85.43 ± 0.44	85.60 ± 0.38
8	Giant panda vs Leopard	87.82 ± 0.32	87.32 ± 0.37	**88.11 ± 0.36**
9	Giant panda vs Persian cat	87.66 ± 0.37	86.35 ± 0.30	**88.12 ± 0.28**
10	Giant panda vs Hippopotamus	85.36 ± 0.41	84.96 ± 0.51	**85.90 ± 0.45**
11	Giant panda vs Raccoon	83.52 ± 0.44	83.17 ± 0.44	**83.77 ± 0.52**
12	Giant panda vs Rat	81.76 ± 0.45	81.76 ± 0.38	**82.20 ± 0.44**
13	Giant panda vs Seal	85.47 ± 0.41	85.58 ± 0.40	**85.72 ± 0.37**
14	Leopard vs Persian cat	**90.18 ± 0.23**	89.87 ± 0.26	89.76 ± 0.31
15	Leopard vs Hippopotamus	86.37 ± 0.33	86.10 ± 0.31	**86.43 ± 0.35**
16	Leopard vs Raccoon	77.40 ± 0.51	76.53 ± 0.76	**78.21 ± 0.47**
17	Leopard vs Rat	81.82 ± 0.26	81.85 ± 0.40	**82.11 ± 0.33**
18	Leopard vs Seal	87.28 ± 0.36	**87.62 ± 0.36**	87.56 ± 0.36
19	Persian cat vs Hippopotamus	84.85 ± 0.56	85.13 ± 0.46	**85.30 ± 0.48**
20	Persian cat vs Raccoon	**84.50 ± 0.44**	84.33 ± 0.44	84.42 ± 0.51
21	Persian cat vs Rat	65.32 ± 0.38	**66.36 ± 0.51**	65.05 ± 0.43
22	Persian cat vs Seal	**80.61 ± 0.38**	79.77 ± 0.48	79.80 ± 0.44
23	Hippopotamus vs Raccoon	80.61 ± 0.61	**81.88 ± 0.60**	81.20 ± 0.62
24	Hippopotamus vs Rat	77.83 ± 0.45	**79.70 ± 0.41**	78.07 ± 0.39
25	Hippopotamus vs Seal	67.91 ± 0.60	**68.62 ± 0.55**	66.96 ± 0.61
26	Raccoon vs Rat	**73.75 ± 0.38**	73.66 ± 0.43	73.60 ± 0.35
27	Raccoon vs Seal	83.86 ± 0.42	**84.53 ± 0.41**	84.06 ± 0.41
28	Rat vs Seal	74.03 ± 0.56	**74.83 ± 0.52**	74.31 ± 0.47

Table 5.7: Human annotation as privileged information. We incorporate human perception of easiness and hardness into the margin distance and perform Margin Transfer with easy-hard score annotation (last column). The numbers shown are mean accuracy and standard error over 20 runs.

tation and ranking in the original data X (right) is more expressed than between X^* and X (center), which possibly explains the performance gain when using human annotation as privileged information instead of attribute description. It is not always the case for the *leopard* class, when for example, in classification versus *seal* the correlation (X, X^*) is considerably stronger than (X, Human annotation), which also matches better performance gain when utilizing attributes as privileged information. The class *chimpanzee* (first column across the tables) seems to be more suitable to explore the privileged information based on attribute description (center) compared to human annotation (right). As we can see, in other classes on the right plot, there is little signal in the correlation (X, Human annotation), i.e. mostly blue color, which coincides with rather disadvantageous performance when doing Margin Transfer with human annotation as privileged information in comparison to Margin Transfer with attributes as privileged information. Low correlation (striking blue color), like in pairs *Rat versus Persian cat*, and *Seal versus Hippopotamus*, can be explained by low performance of the classifiers on these pairs. Low performance in our case means a lot of hard/misclassified samples that are on the wrong side of the classifier hyperplane which influences the margin score. In principle, this situation is not suitable for our ranking correlation analysis, because human defined easy hard sample scores do not account for such misclassifications.

5.7 Summary and Discussion

In this chapter, we have introduced the setting of learning using privileged information (LUPI) for an object classification task using images. Semantic attributes can be seen as one form of privileged information that is given in addition to the image data during training. We showed how LUPI can be applied to several situations that previously were handled by hand-crafted separate methods. We have studied two methods, the SVM+ and the Margin Transfer, that examine the max margin framework for solving LUPI. In this framework, the slack value or margin distance computed in the privileged space encodes how easy or difficult the sample is, and guides learning of the classifier in the original space. In its current form, when we use linear SVM+ and linear SVM classifiers in the Margin Transfer, it is the parametric learning framework as considered in our Section 2.3.

In our next chapter we will show how to address LUPI in a non-parametric framework. One obvious choice would be to kernelize the max margin models and have a direct non-parametric non-probabilistic extension. However this extension requires tuning the parameters of the kernels in the privileged and original space in addition to our costly cross validation procedure on trade-off regularization parameters C and γ (C and C for Margin Transfer). For example, if we consider Gaussian kernel in $\mathcal{X}$ and Gaussian kernel in $\mathcal{X}^*$:

$$k(x_i, x_j) = \exp\{-\lambda_{\mathcal{X}} \|x_i - x_j\|^2\}, \quad k(x_i^*, x_j^*) = \exp\{-\lambda_{\mathcal{X}^*} \|x_i^* - x_j^*\|^2\}, \tag{5.8}$$

parameterized by $\lambda_{\mathcal{X}}$ and $\lambda_{\mathcal{X}^*}$ correspondingly. If we have 5 values to cross validate for each parameter C, γ, $\lambda_{\mathcal{X}}$, $\lambda_{\mathcal{X}^*}$ in the case of SVM+, and C, C, $\lambda_{\mathcal{X}}$, $\lambda_{\mathcal{X}^*}$ in the case of Margin Transfer. Then using our grid search procedure, we will have to check 625 values during cross validation model selection, compared to 25 values in the linear case.

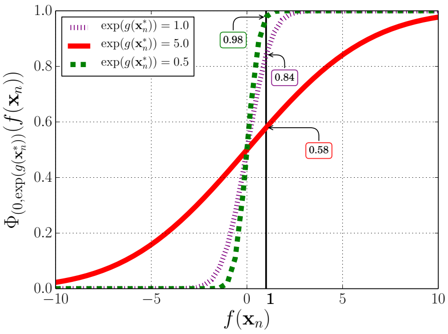
So we take one step further, and address LUPI in the non-parametric probabilistic framework of Gaussian process classification, and allow the parameters of the kernels to be learned from the data. Interestingly, we can still retain our interpretation of easy and hard samples learned from the privileged space in this framework. However, in the probabilistic formulation it will correspond to the confidence level of the likelihood model when observing the easy sample, and uncertainty when observing the hard one.

Chapter 6

Learning with Privileged Information: Non-parametric View

In this chapter, we propose a method for using privileged information based on the framework of Gaussian process classifiers (GPC). In contrast to the previous chapter, that used the privileged information mainly to assign different weights to each training example, in the Bayesian treatment the privileged data enters the model in the form of a latent variable, which modulates the noise term of the GPC. By this procedure, the influence of the privileged information stays interpretable: it models the confidence that the Gaussian process has about any training example, which can be directly read off from the slope of the sigmoid-shaped likelihood. Training examples that are easy to classify by means of their privileged data cause a faster increasing sigmoid, which means the GP trusts the training example and tries to fit it well. Examples that are hard to classify result in a slowly increase slope, so the GPC considers the training example less reliable and does not put a lot of effort into fitting its label well. Our experiments on multiple datasets show that this procedure leads not just to interpretable models, but also to significantly higher classification accuracy.

This chapter is based on:



D. Hernández-Lobato, V. Sharmanska,
K. Kersting, C.H. Lampert, N. Quadrianto:
Mind the Nuisance: Gaussian Process Classification using Privileged Noise,
NIPS 2014, Montreal, Canada.

6.1 Gaussian Process Classification with Privileged Noise

First, we will review the standard GPC model by Rasmussen & Williams (2006) with a particular emphasis on the noise-corrupted latent Gaussian process perspective. Then, we show how to incorporate privileged information as heteroscedastic noise in this latent process. The elegant aspect of this view is the intuition as how the privileged noise is able to distinguish between easy and hard samples and in turn to re-calibrate our uncertainty in the original space.

6.1.1 Gaussian process classifier with noisy latent process

Assume we are given a set of N training samples, represented by feature vectors $\{x_1, \ldots, x_N\} \subset \mathcal{X} = \mathbb{R}^d$ and their label annotation $\{y_1, \ldots, y_N\} \in \mathcal{Y} = \{+1, -1\}$. Furthermore, we assume that the class label y_i of sample x_i has been generated as $y_i = \text{sign}(\tilde{f}(x_i))$, where $\tilde{f}(\cdot)$ is a *noisy* latent function. Induced by the label generation process, we adopt the following form of label likelihood function for $\tilde{f} = (\tilde{f}(x_1), \ldots, \tilde{f}(x_N))$:

$$\Pr(y|\tilde{f}, x_1, \ldots, x_N) = \prod_{i=1}^{N} \Pr(y_i|x_i, \tilde{f}) = \prod_{i=1}^{N} \mathbb{I}[\ y_i \tilde{f}(x_i) \geq 0\ ], \tag{6.1}$$

where y is a label vector of all training samples, $\mathbb{I}[\cdot]$ is the Iverson's bracket notation, and the noisy latent function at sample x_i is given by:

$$\tilde{f}(x_i) = f(x_i) + \epsilon_i \tag{6.2}$$

with $f(x_i)$ being the *noise-free* latent function. The noise term ϵ_i is assumed to be independent and normally distributed with zero mean and variance σ^2:

$$\epsilon_i \sim \mathcal{N}(\epsilon_i|0, \sigma^2). \tag{6.3}$$

To make inference about $\tilde{f}(x_i)$, we need to specify a prior distribution over this function. We proceed by imposing a Gaussian process prior (Rasmussen & Williams, 2006) on the noise-free latent function $f(x_i)$. The Gaussian process prior defines a distribution over the function's space and can be seen as a generalization of the multivariate Gaussian distribution. It is fully specified by a mean function, in our case we take zero mean, and a covariance or kernel function $k(\cdot, \cdot)$, that specifies prior properties of $f(\cdot)$, that is $f(x_i) \sim \mathcal{GP}(0, k(x_i, \cdot))$.

A typical kernel function that stands for the non-linear smooth function is the squared exponential kernel $k_f(x_i, x_j) = \theta \exp(-\frac{1}{2l} \|x_i - x_j\|^2_{\ell_2})$. In this kernel function, the parameter θ controls the amplitude of function $f(\cdot)$ while l controls the smoothness of $f(\cdot)$. Given the prior distribution and the likelihood, Bayes' rule is used to compute the posterior of $\tilde{f}(\cdot)$:

$$\Pr(\tilde{f}|y, x_1, \ldots, x_N) = \frac{\Pr(y|\tilde{f}, x_1, \ldots, x_N)\Pr(\tilde{f})}{\Pr(y|x_1, \ldots, x_N)}. \tag{6.4}$$

We can simplify the above noisy latent process view by integrating out the noise term ϵ_i and writing down the individual likelihood at sample x_i in term of noise-free latent function $f(\cdot)$ as follows:

$$\begin{aligned}\Pr(y_i = 1|x_i, f) = \int \mathbb{I}[\tilde{f}(x_i) \geq 0]\mathcal{N}(\epsilon_i|0, \sigma^2)d\epsilon_i = \int \mathbb{I}[\epsilon_i \geq -f(x_i)]\mathcal{N}(\epsilon_i|0, \sigma^2)d\epsilon_i \\ = \Phi_{(0,\sigma^2)}(f(x_i)), \qquad (6.5)\end{aligned}$$

where $\Phi_{(\mu,\sigma^2)}(\cdot)$ is a Gaussian cumulative distribution function (CDF) with mean μ and variance σ^2. Typically the standard Gaussian CDF $\Phi_{(0,1)}(\cdot)$ is used in this case, and the full likelihood model is as follows:

$$\Pr(y|f, x_1, \ldots, x_N) = \prod_{i=1}^{N} \Pr(y_i|x_i, f) = \prod_{i=1}^{N} \Phi_{(0,\sigma^2)}(y_i f(x_i)). \tag{6.6}$$

Coupled with a Gaussian process prior on the latent function $f(\cdot)$, this results in the widely adopted noise-free latent Gaussian process view with probit likelihood. The equivalence between a noise-free latent process with probit likelihood (6.6) and a noisy latent process with step-function likelihood (6.1) is widely known (Rasmussen & Williams, 2006). The Gaussian Process classification model is formalized as follows:

$$\text{Likelihood model} : \Pr(y_i|x_i, \tilde{f}) = \mathbb{I}[\ y_i\tilde{f}(x_i) \geq 0\ ], \quad x_i \in \mathbb{R}^d. \tag{6.7a}$$

$$\text{Assume} : \tilde{f}(x_i) = f(x_i) + \epsilon_i, \tag{6.7b}$$

$$\text{Privileged noise model} : \epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(\epsilon_i|0, \sigma^2), \tag{6.7c}$$

$$\text{GP prior model} : f(x_i) \sim \mathcal{GP}(0, k_f(x_i, \cdot)). \tag{6.7d}$$

It is also widely accepted that the noisy latent function $\tilde{f}$ (or the noise-free latent function f) is a *nuisance* function as we do not observe the value of this function itself and its sole purpose is for a convenient formulation of the classification model by Rasmussen & Williams (2006). However, in the next section, we will show that by using privileged information as the noise term, the latent function $\tilde{f}$ plays a crucial role. The latent function with privileged noise adjusts the slope transition in the Gaussian CDF to be *faster* or *slower* corresponding to more *certainty* or more *uncertainty* about the samples in the original space.

6.1.2 Privileged information is in the Nuisance Function

In the learning using privileged information (LUPI) paradigm (Vapnik & Vashist, 2009), besides input data points $\{x_1, \ldots, x_N\}$ and associated outputs $\{y_1, \ldots, y_N\}$, we are given additional information $x_i^* \in \mathbb{R}^{d^*}$ about each training instance x_i during training. Our goal is to exploit how the additional data x^* can influence our choice of the latent function $\tilde{f}(\cdot)$. We achieve this naturally by treating the privileged information as a heteroscedastic (input-dependent) noise in the latent process. Our classification model with privileged noise called GPC+ is as follows:

$$\begin{aligned}
\text{Likelihood model} : \Pr(y_i|x_i, \tilde{f}) &= \mathbb{I}[\ y_i\tilde{f}(x_i) \geq 0\ ], \quad x_i \in \mathbb{R}^d. && (6.8a)\\
\text{Assume} : \tilde{f}(x_i) &= f(x_i) + \epsilon_i, && (6.8b)\\
\text{Privileged noise model} : \epsilon_i &\overset{i.i.d.}{\sim} \mathcal{N}(\epsilon_i|0, z(x_i^*) = \exp(g(x_i^*))), \quad x_i^* \in \mathbb{R}^{d^*}, && (6.8c)\\
\text{GP prior model} : f(x_i) &\sim \mathcal{GP}(0, k_f(x_i, \cdot)), \quad g(x_i^*) \sim \mathcal{GP}(0, k_g(x_i^*, \cdot)). && (6.8d)
\end{aligned}$$

In the above, the $\exp(\cdot)$ function in the privileged noise model is needed to ensure positivity of the noise variance. The term $k_g(\cdot, \cdot)$ is a positive-definite kernel function that specifies the prior properties of the latent function $g(\cdot)$ evaluated in the privileged space x^*. Crucially, the noise term ϵ_i is now *heteroscedastic*, that is it has a different variance $z(x_i^*)$ at each input point x_i. This is in contrast to the standard GPC approach described in the previous section, where the noise term is assumed to be homoscedastic, $\epsilon_i \sim \mathcal{N}(\epsilon_i|0, z(x_i^*) = \sigma^2)$. The input-dependent noise term is very common in the regression tasks with continuous output values $y_i \in \mathbb{R}$, resulting in the so-called heteroscedastic regression models, which have been proven to be effective in numerous applications including econometric and statistical finance. However, to our knowledge, there is no prior work on heteroscedastic classification models, and in the context of learning with privileged information heteroscedastic classification is actually a very sensible idea.

Let us illustrate the effect of privileged information in the equivalent formulation of the noise free latent process, when one integrates out the privileged

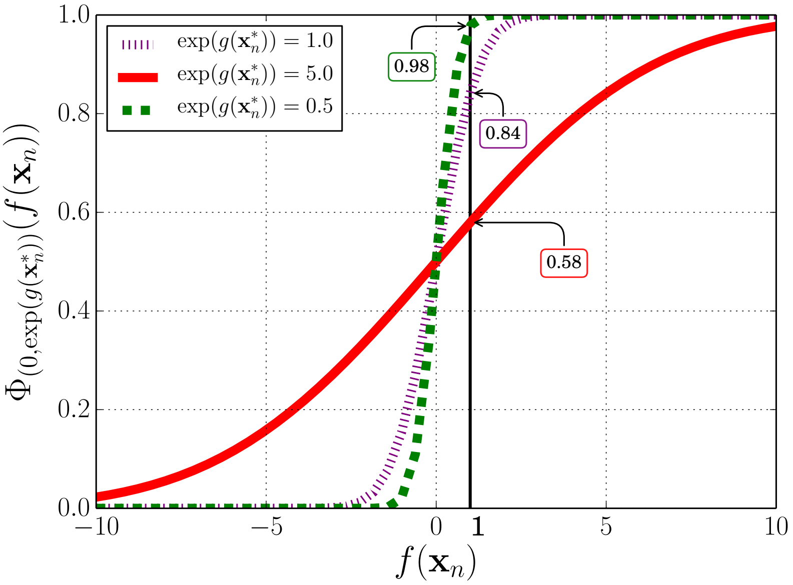


Figure 6.1: Effect of privileged noise on the nuisance function on synthetic data. Suppose for a positive input x_n, the latent function value is $f(x_n) = 1$. Now assume that the associated privileged information x_n^* for n-th data point deems the sample as *difficult*, say $\exp(g(x_n^*)) = 5.0$. Then the likelihood reflects this uncertainty as $\Pr(y_n = 1|f, g, x_n, x_n^*) = 0.58$. In contrast, if the associated privileged information considers the sample as *easy*, say $\exp(g(x_n^*)) = 0.5$, the likelihood reflects our certainty $\Pr(y_n = 1|f, g, x_n, x_n^*) = 0.98$. Best viewed in color.

input-dependent noise term:

$$\Pr(y_i = 1|x_i, x_i^*, f, g) = \int \mathbb{I}[\ \tilde{f}(x_i) \geq 0\ ]\mathcal{N}(\epsilon_i|0, \exp(g(x_i^*))d\epsilon_i \tag{6.9a}$$

$$= \int \mathbb{I}[\ \epsilon_i \leq f(x_i)\ ]\mathcal{N}(\epsilon_i|0, \exp(g(x_i^*))d\epsilon_i \tag{6.9b}$$

$$= \Phi_{(0,\exp(g(x_i^*)))}(f(x_i)). \tag{6.9c}$$

From (6.9c), it is clear that for each sample, privileged information adjusts the slope transition of the Gaussian CDF. For a difficult (noisy) sample, the latent function $g(x_i^*)$ is high and the slope transition is slow, thus emphasizing our uncertainty in the likelihood term $\Pr(y_i|x_i, x_i^*, f, g)$. For an easy sample, however, the latent function $g(x_i^*)$ is low, the slope transition is fast (steep slope), thus emphasizing our certainty in the likelihood term $\Pr(y_i|x_i, x_i^*, f, g)$. Please, refer here to the illustration in Figure 6.1.

Related work GPC+ constitutes the first Bayesian treatment of classification using privileged information. The resulting privileged noise approach is related to input-modulated noise commonly done in the regression task, and several Bayesian treatments of this heteroscedastic regression using Gaussian processes have been proposed. Since the predictive density and marginal likelihood are no longer analytically tractable, most works in heteroscedastic GPs deal with approximate inference. Techniques such as Markov Chain Monte Carlo (Goldberg et al., 1998), maximum a posteriori (Quadrianto et al., 2009), and lately a variational Bayes method (Lázaro-Gredilla & Titsias, 2011) have been proposed. To our knowledge, however, there is no prior work on heteroscedastic classification using GPs and consequently this work develops the first approximate inference based on expectation propagation for the heteroscedastic noise case in the context of classification.

6.1.3 Posterior and Prediction on Test Data

We denote a vector of per-sample noise variances as $g = (g(x_1^*), \ldots, g(x_N^*))$, a matrix of training samples in the original space as $X = (x_1, \ldots, x_N)$, and a matrix of training samples in the privileged space as $X^* = (x_1^*, \ldots, x_N^*)$. Given the likelihood $\Pr(y|X, X^\star, f, g) = \prod_{n=1}^N \Pr(y_i|f, g, x_i, x_i^*)$ with the individual terms $\Pr(y_i|f, g, x_i, x_i^*)$ defined in (6.9c) and the Gaussian process priors on the predictor functions, the posterior for f and g is as follows:

$$\Pr(f, g|\, y, X, X^\star) = \frac{\Pr(y|X, X^\star, f, g)\Pr(f)\Pr(g)}{\Pr(y|X, X^\star)}, \tag{6.10}$$

where $\Pr(y|\, X, X^\star)$ can be maximized with respect to a set of hyper-parameter values such as amplitude θ and smoothness l parameters of the squared exponential kernel functions (Rasmussen & Williams, 2006). For a previously unseen test point $\bar{x} \in \mathbb{R}^d$, the predictive distribution for its label $\bar{y}$ is given as:

$$\Pr(\bar{y} = 1|\, y, X, X^\star) = \int \mathbb{I}[\ \tilde{f}(\bar{x}) \geq 0\ ]\Pr(\bar{f}|f)\Pr(f, g|y, X, X^\star) df\, dg\, d\bar{f}, \tag{6.11}$$

where, since we do not have the privileged information associated to $\bar{x}$, we consider the homoscedastic noise at test time. This is a reasonable approach as there is no additional information for increasing or decreasing our confidence in the newly observed data $\bar{x}$. In (6.11), $\Pr(\bar{f}|f)$ is a Gaussian conditional distribution, that is obtained based on the joint Gaussian distribution of $\Pr([\bar{f}, f])$:

$$\begin{bmatrix} \bar{f} \\ f \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{\bar{x}\bar{x}} & K_{\bar{x}X} \\ K_{X\bar{x}} & K_{XX} \end{bmatrix}\right), \tag{6.12}$$

where $K_{XX} = \{k_f(x_i, x_j)\}_{i,j=1}^N$, $K_{\bar{x}X} = \{k_f(\bar{x}, x_i)\}_{i=1}^N$, and similarly for the remaining components. Then the Gaussian conditional distribution is as follows (Quadrianto et al., 2010):

$$\Pr(\bar{f}|f) = \mathcal{N}(K_{\bar{x}X}\, K_{XX}^{-1}\, f,\ K_{\bar{x}\bar{x}} - K_{\bar{x}X} K_{XX}^{-1} K_{X\bar{x}}). \tag{6.13}$$

Finally, we predict the label for a test point via Bayesian decision theory: the label being predicted is the one with the largest probability.

6.2 Expectation Propagation using Numerical Quadrature

Unfortunately, as for most interesting Bayesian models, inference in the GPC+ model is very challenging. Already in the homoscedastic case, the predictive density and marginal likelihood are not analytically tractable. In this chapter, we therefore adapt the expectation propagation (EP) procedure by Minka (2001) with numerical quadrature for approximate inference. Our choice is supported by the fact that EP is the preferred method for approximate inference in GPCs, in terms of accuracy and computational cost as shown by Nickisch & Rasmussen (2008); Kuss & Rasmussen (2005).

6.2.1 Posterior approximation in EP

We approximate the posterior distribution defined in (6.10) as a product of two multivariate Gaussians for f and g terms:

$$\Pr(f, g|\, y, X, X^\star) = \frac{\Pr(y|X, X^\star, f, g)\, \Pr(f)\, \Pr(g)}{\Pr(y|X, X^\star)} \tag{6.14a}$$

$$\approx \mathcal{N}(f|m_f, \Sigma_f)\, \mathcal{N}(g|m_g, \Sigma_g), \tag{6.14b}$$

where $\Pr(f)$ and $\Pr(g)$ are Gaussian process priors, the likelihood $\Pr(y|X, X^*, f, g)$ is equal to $\prod_{i=1}^N \Pr(y_i|x_i, x_i^*, f, g)$, with probit terms $\Pr(y_i|x_i, x_i^*, f, g)$ defined in (6.9c). The difference between the two probability distributions (true posterior and its approximation) is measured by the Kullback–Leibler divergence, which for continuous random variables with $p(x)$ and $q(x)$ probability density distributions is as follows:

$$\mathrm{KL}(p\|q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)}. \tag{6.15}$$

We cannot perform direct minimization of the KL divergence between the true posterior in (6.14a) and its approximation in (6.14b), because this would require numerical integration of the normalization constant (denominator term in the true posterior):

$$\Pr(y|X, X^\star) = \int_f \int_g \prod_{i=1}^{N} \Pr(y|X, X^\star, f, g) \Pr(f) \Pr(g) df\, dg, \tag{6.16}$$

which we want to avoid in the first place. Instead, the EP procedure offers an iterative scheme to approximate each of the non-normal factors in the true posterior with a normal distribution. Specifically, at each step of the EP procedure, we approximate one of the probit factors in the likelihood term by an un-normalized bi-variate Gaussian distribution of f and g, taking one factor at a time:

$$\begin{aligned}
\Pr(y_i|x_i, x_i^*, f, g) &= \Phi_{(0,\exp(g(x_i^*)))}(y_i f(x_i)) = \gamma(f_i, g_i) && \text{(6.17a)}\\
&\approx \overline{\gamma}(f_i, g_i) && \text{(6.17b)}\\
&= \overline{z}_i \mathcal{N}(f(x_i)|\overline{m}_f, \overline{v}_f)\mathcal{N}(g(x_i^*)|\overline{m}_g, \overline{v}_g)\,, && \text{(6.17c)}
\end{aligned}$$

where the parameters with the upper-script $^-$ are to be found by EP, and $\overline{z}_i$ is the scaling constant that will be detailed in the next section. The un-normalized posterior approximation $\mathcal{Q}$ computed by EP procedure is obtained by replacing each likelihood factor $\gamma(f_i, g_i)$ by the approximate factor $\overline{\gamma}(f_i, g_i)$:

$$\mathcal{Q}(f, g) = [\prod_{i=1}^{N} \overline{\gamma}(f_i, g_i)]\Pr(f)\Pr(g). \tag{6.18}$$

The normal distribution belongs to the exponential family of probability distributions and is closed under taking products and under divisions. It is hence possible to show that $\mathcal{Q}$ is the product of two multi-variate normals (Seeger, 2006), where the first normal approximates the posterior for f and the second normal the posterior for g:

$$\mathcal{Q}(f, g) = \mathcal{N}(f|m_f, \Sigma_f)\,\mathcal{N}(g|m_g, \Sigma_g). \tag{6.19}$$

Hence, during the EP procedure, the true posterior distribution $\Pr(f, g|X, X^*, y)$ is approximated by the product of two multivariate Gaussians for f and g terms, $\Pr(f, g|X, X^*, y) = Z^{-1}\mathcal{Q}(f, g)$, where Z^{-1} is the normalization constant that in fact approximates the model evidence $\Pr(y|X, X^*)$ in (6.14a).

6.2.2 The EP procedure

EP tries to fix the parameters of $\overline{\gamma}(f_i, g_i)$ so that it is similar to the exact factor $\Pr(y_i|x_i, x_i^*, f, g)$ in regions of high posterior probability (Minka, 2001). The complete EP algorithm involves the following steps until all $\overline{\gamma}(f_i, g_i)$ converge:

1. Select a particular factor $\overline{\gamma}(f_i, g_i)$ to be refined as:

$$\overline{\gamma}(f_i, g_i) = \overline{z}_i \mathcal{N}(f(x_i)|\overline{m}_f, \overline{v}_f)\mathcal{N}(g(x_i^*)|\overline{m}_g, \overline{v}_g). \tag{6.20}$$

   Compute a cavity distribution $\mathcal{Q}_{-i}(f, g)$ when dividing $\mathcal{Q}(f, g)$ by $\overline{\gamma}(f_i, g_i)$. The ith cavity distribution contains the information about all but the ith approximated likelihood term $\overline{\gamma}(f_i, g_i)$, that is:

$$\mathcal{Q}_{-i}(f, g) = [\prod_{j\neq i} \overline{\gamma}(f_j, g_j)]\Pr(f)\Pr(g). \tag{6.21}$$

2. Update the $\overline{\gamma}(f_i, g_i)$ with respect to the parameters $\overline{z}_i, \overline{m}_f, \overline{v}_f, \overline{m}_g, \overline{v}_g$ such that the posterior approximation

$$\mathcal{Q}(f, g) = [\prod_{i=1}^{N} \overline{\gamma}(f_i, g_i)]\Pr(f)\Pr(g) = \overline{\gamma}(f_i, g_i)\mathcal{Q}_{-i}(f, g) \tag{6.22}$$

   is close to the true yet intractable posterior

$$\Pr(f, g|X, X^*, y) \approx \gamma(f_i, g_i)[\prod_{j\neq i} \overline{\gamma}(f_j, g_j)]\Pr(f)\Pr(g) \tag{6.23a}$$

$$= \gamma(f_i, g_i)\mathcal{Q}_{-i}(f, g) \tag{6.23b}$$

   in the sense of extended KL divergence:

$$\mathrm{KL}(\ \gamma(f_i, g_i)\mathcal{Q}_{-i}(f, g) \parallel \overline{\gamma}(f_i, g_i)\mathcal{Q}_{-i}(f, g)\ ). \tag{6.24}$$

   We use the extended KL divergence that is applicable to un-normalized positive distributions, that is why we do not need to estimate the normalization constant of $\mathcal{Q}_{-i}(f, g)$. The positive constants $\overline{z}_i$ are introduced to guarantee that $\gamma(f_i, g_i)\mathcal{Q}_{-i}(f, g)$ and $\overline{\gamma}(f_i, g_i)\mathcal{Q}_{-i}(f, g)$ have the same scaling for $i = 1, \ldots, N$.

3. The KL divergence in (6.24) can be minimized with respect to the parameters of the factor $\overline{\gamma}(f_i, g_i)$ as follows. Since $\mathcal{Q}$ and all the $\overline{\gamma}(f_i, g_i)$ belong to the same family of the exponential distributions (Gaussians in our case),

the optimum is obtained by finding the parameters of $\overline{\gamma}(f_i, g_i)$ that guarantee that the zeroth moment $\overline{z}_i$, the first and the second moments are the same for $\gamma(f_i, g_i)\mathcal{Q}_{-i}(f, g)$ and $\overline{\gamma}(f_i, g_i)\mathcal{Q}_{-i}(f, g)$ in (6.24). The easiest way to match the first and the second moments (the expected sufficient statistics) is by taking the derivatives of the log partition function at ith step with respect to the parameters of $\overline{\gamma}(f_i, g_i)$ (Seeger, 2006). The log partition function at ith step, is the normalization of the $\gamma(f_i, g_i)\mathcal{Q}_{-i}(f, g)$ in (6.24):

$$\int_f \int_g \Phi_{(0,\exp(g(x_i^*)))}(y_i f(x_i))\mathcal{N}(g(x_i^*)|m_g, v_g)\mathcal{N}(f(x_i)|m_f, v_f)dg(x_i^*)df(x_i). \tag{6.25}$$

Unfortunately, the computation of $\log Z_i$ in closed form is intractable. We show here that it can be approximated by a *one dimensional quadrature*:

$$\int \Phi_{(0,1)}\left(\frac{y_i m_f}{\sqrt{v_f + \exp(g(x_i^*))}}\right)\mathcal{N}(g(x_i^*)|m_g, v_g)dg(x_i^*), \tag{6.26}$$

where we integrate out the $f(x_i)$ analytically. Thus, the EP algorithm only requires five quadratures to update each $\overline{\gamma}_i$: a first one to compute $\log Z_i$ and four extras to compute its derivatives with respect to m_f, v_f, m_g and v_g, and update the parameters $\overline{m}_f, \overline{v}_f, \overline{m}_g, \overline{v}_g$ when dividing by the old value of $\mathcal{Q}$. After convergence, $\mathcal{Q}$ can be used to approximate predictive distributions and the normalization constant Z can be maximized to find good values for the model's hyper-parameters. In particular, it is possible to compute the gradient of Z with respect to the parameters of the Gaussian process priors for f and g (Seeger, 2006).

6.3 Experiments

Our intention here is to investigate the performance of the GP with the privileged noise approach. To this aim, we considered three types of binary classification tasks corresponding to different privileged information using *Attribute Discovery* and *Animals with Attributes* datasets. We detail those experiments in turn in the following sections.

Methods We compared the proposed `GPC+` method with the kernelized (nonparametric) SVM-based method `SVM+` by Pechyony & Vapnik (2011) that we described in our previous chapter. As a reference, we also fit standard GPC and

SVM classifiers when learning on the original space $\mathbb{R}^d$ (`GPC` and `SVM` baselines). For *all four* methods, we used a squared exponential kernel with amplitude parameter θ and smoothness parameter l. For simplicity, we set $\theta = 1.0$ in all cases. For GPC and GPC+, we estimate the hyper-parameters using the maximum likelihood principle. There are two hyper-parameters in GPC, the smoothness parameter l and the noise variance σ^2, and also two in GPC+, the smoothness parameters l of the kernel $k_f(\cdot,\cdot)$ and of the kernel $k_g(\cdot,\cdot)$. For SVM and SVM+, we used cross validation to set the hyper-parameters. The SVM has two parameters, smoothness and regularization, and the SVM+ has four parameters, two smoothness and two regularization terms. It turned out that a grid search via cross validation was too expensive for searching the best parameters in SVM+. We discretized the hyper-parameter search space over 625 ($5 \times 5 \times 5 \times 5$) possible combination values and use the performance on a separate validation set to guide the search process. None of the other three methods used this separate validation set, this means that we give a competitive advantage to SVM+ over the other methods.

Evaluation metric To evaluate the performance of the methods we use classification accuracy on an independent test set. We perform 100 repeats of all the experiments to get better statistics of the performance and report the mean and the standard error.

6.3.1 Accessories Dataset

This dataset was used in our previous chapter to perform experiments with LUPI methods in the parametric learning setting. It is originally called *Attribute Discovery* dataset (Berg et al., 2010) and aggregates product data from a variety of e-commerce sources that includes both images and associated textual descriptions. The images and associated texts are grouped into 4 categories: *bags, earrings, ties, and shoes.* We generate 6 binary classification tasks, one for each pair of the 4 classes, with 200 samples for training, 200 samples for validation, and the remaining samples for testing the predictive performance.

Deep neural networks on texts as privileged information We use *images* as the *original* domain and *texts* as the *privileged* domain. As image representation, we extract SURF descriptors (Bay et al., 2008) and construct a codebook of 100 visual words using the k-means clustering. As text representation,

		GPC	GPC+ (Ours)	SVM	SVM+
		image	image+text	image	image+text
1	Bags vs Earrings	90.20 ± 0.12	$\mathbf{90.49 \pm 0.10}$	90.11 ± 0.13	90.10 ± 0.12
2	Bags vs Ties	89.64 ± 0.15	89.96 ± 0.14	$\mathbf{90.55 \pm 0.15}$	90.53 ± 0.13
3	Bags vs Shoes	90.34 ± 0.12	$\mathbf{90.78 \pm 0.10}$	90.68 ± 0.11	90.70 ± 0.13
4	Earrings vs Ties	89.15 ± 0.13	$\mathbf{89.44 \pm 0.13}$	88.84 ± 0.15	88.89 ± 0.15
5	Earrings vs Shoes	92.26 ± 0.10	$\mathbf{92.66 \pm 0.10}$	92.25 ± 0.12	92.37 ± 0.12
6	Ties vs Shoes	84.48 ± 0.16	84.46 ± 0.16	$\mathbf{85.10 \pm 0.20}$	84.89 ± 0.18
	Average accuracy	89.34 ± 0.11	$\mathbf{89.63 \pm 0.11}$	89.59 ± 0.11	89.58 ± 0.11
	Average ranking	3.0	**1.8**	2.7	2.5

Table 6.1: Accessories dataset (textual description as privileged information). The numbers are mean and standard error of the accuracy over 100 runs. We used *images* as the original domain and neural networks word-vector representation on *texts* as the privileged domain. The best method for each binary task is highlighted in **boldface**. An average rank equal to one means that the corresponding method has the highest accuracy on the 6 tasks (the lower the better).

we extract 200-dimensional continuous word-vector representation using a neural network skip-gram architecture (Mikolov et al., 2013)[1]. To convert this word representation to a fixed-length sentence representation, we construct a codebook of 100 word-vector using again k-means clustering. We note that a more elaborate approach to transform words to sentence or document features has recently been developed (Le & Mikolov, 2014), and we are planning to explore this in the future. We perform PCA for dimensionality reduction in the original and privileged domains and only keep the top 50 principal components. Finally, we standardize the data to have zero mean and unit standard deviation across features.

Results The experimental results are summarized in Table 6.1. On average over 6 tasks, the kernelized SVM baseline with hinge loss outperforms GPC with probit likelihood. However, GPC+ significantly improves over GPC providing the best results on average. This clearly shows that GPC+ is able to utilize the neural network textual representation as privileged information. To summarize our findings, when using text as privileged information, based on the results from our previous and current chapters, we can see that GPC+ is the only method that benefits from textual descriptions as provided in the *Accessories* dataset.

[1] https://code.google.com/p/word2vec/

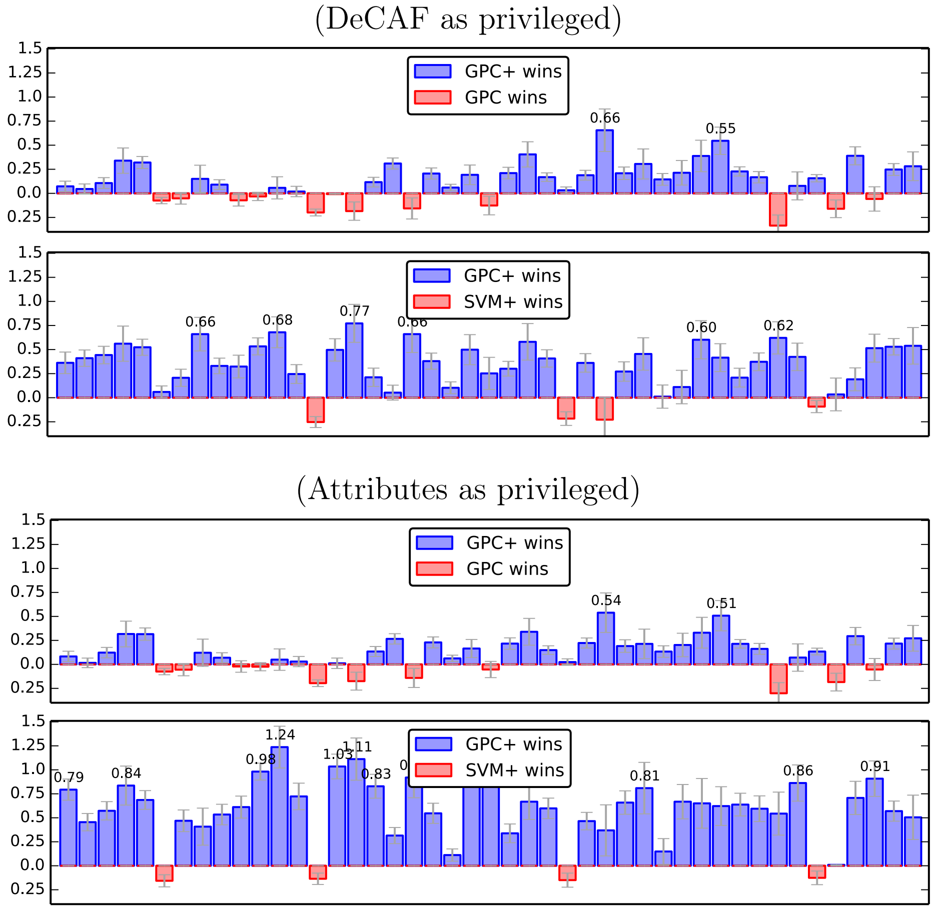
Similarly to the linear case on this dataset, SVM+ produced very similar results to SVM baseline also when kernelized. We suspect this is due to that fact that SVM has already shown strong performance on the original image space coupled with the difficulties in finding the best values of four hyper-parameters.

6.3.2 Animals with Attributes (AwA) Dataset

We focused on the default 10 test classes for which the predicted attributes are provided based on the probabilistic DAP model (Lampert et al., 2009, 2013). As in our previous chapter, we generate 45 binary classification tasks, one for each pair of the 10 classes, with 200 samples for training, 200 samples for validation, and the remaining samples for testing the predictive performance.

Deep neural networks on images as privileged information Deep learning methods have gained increased attention within the machine learning and computer vision communities over the recent years. This is due to their capability in extracting informative features and delivering strong predictive performance in many classification tasks. We are interested to explore the use of deep learning based features as privileged information so that their predictive power can be used even if we do not have access to them at prediction time. We used the standard SURF features with 2000 visual words as the original domain and used the DeCAF features (Donahue et al., 2014) extracted from the activation of a deep convolutional network trained in a fully supervised fashion as the privileged domain. We again perform PCA for dimensionality reduction in the original and privileged domains and keep the top 50 principal components, as well as standardizing the data so that each feature has zero mean and unit standard deviation.

Attributes as privileged information Following the experimental setting of our previous chapter, we also use *images* as the original domain and *attributes* as the privileged domain. Images are represented by 2000 visual words based on SURF descriptors and attributes are in the form of 85-dimensional predicted attributes based on probabilistic binary classifiers (Lampert et al., 2009, 2013). We perform PCA and keep the top 50 principal components in the original domain. Finally, we also standardize the data to have zero mean and unit standard deviation across features.

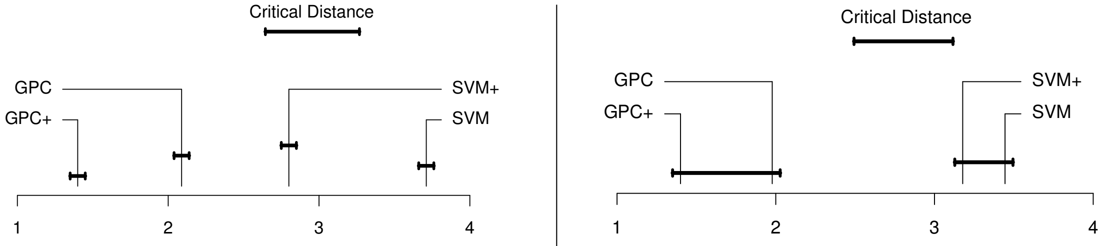


(DeCAF as privileged)

(Attributes as privileged)

Figure 6.2: Pairwise comparison of the proposed GPC+ method and main baselines is shown via relative differences in performance (top: GPC+ versus GPC, bottom: GPC+ versus SVM+). The length of the 45 bars corresponds to relative difference of the accuracy over 45 cases averaged over 100 repeated experiments.

Results The results are summarized in Figure 6.2 in term of pairwise comparison over 45 binary tasks between GPC+ and the main baselines, GPC and SVM+. In contrast to the results on the attribute discovery dataset, on the AwA dataset it is clear that GPC outperforms SVM in almost all of the 45 binary classification tasks. The average accuracy of GPC over 4500 (45 tasks and 100 repeats per task) experiments is much higher than SVM. On the AwA dataset, SVM+ can take advantage of privileged information – be it deep belief DeCAF features or semantic attributes – and shows significant performance improvement over SVM. However, GPC+ still shows the best overall results and further improves the already strong performance of GPC.



(DeCAF as privileged) (Attributes as privileged)

Figure 6.3: Average rank (the lower the better) of the four methods and critical distance for significant differences (Demšar, 2006) on the AwA dataset. An average rank equal to one means that particular method has the highest accuracy on the 45 tasks. Whenever the average ranks differ by more than the critical distance, there is statistical evidence (p-value $< 10\%$) to support a difference in the average ranks and thus in the performance. We also link two methods with a solid line if they are not statistically different from each other (p-value $> 10\%$). In DeCAF, there is statistical evidence that GPC+ performs best among the four methods considered, while in attributes, GPC+ still performs best but there is not enough evidence to reject that GPC+ performs comparably to GPC.

Additionally we analyzed our experimental results using the multiple dataset statistical comparison method described by Demšar (2006)[1]. The statistical tests are summarized in Figure 6.3. When DeCAF is used as privileged information, there is statistical evidence that GPC+ *performs best* among the four methods, while in semantic attributes as privileged information setting, GPC+ still performs best but there is not enough evidence to reject that GPC+ performs comparably to GPC. The full results with the accuracy of GPC, GPC+, SVM, and SVM+ on each problem are summarized in Table 6.2 and Table 6.3.

In this experiment, we also illustrate the slope of the sigmoid likelihood for one pair of classes *Chimpanzee versus Giant panda* in Figure 6.4. Similarly to what we observed in the synthetic data experiment, the privileged information modulates the slope of the sigmoid likelihood function differently for easy and difficult examples. Easy examples gain slope and hence importance whereas difficult ones lose it because of uncertainty about their labels in the classification framework.

[1] We are not able to use this method for our attribute discovery results in Table 6.1 as the number of methods being compared (4) is almost equal to the number of tasks or datasets (6).

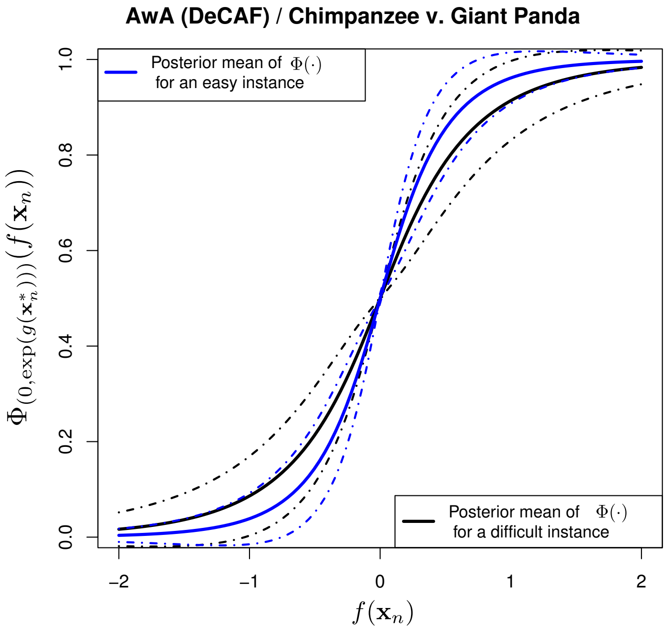


Figure 6.4: Effect of the privileged noise term on the nuisance function in our experiments on the AwA dataset (Section 6.3). The posterior means of the $\Phi(\cdot)$ function (solid) and its 1-standard deviation confidence interval (dash-dot) for easy (blue) and difficult (black) instances in the *Chimpanzee versus Giant panda* binary task. (Best viewed in color)

6.4 Summary and Discussion

In this chapter, we presented the Bayesian treatment of the *learning with privileged information* setting in the Gaussian process classification (GPC) framework called GPC+. The privileged information enters the latent noise layer of the GPC+, resulting in a data-dependent modulation of the sigmoid slope of the GP likelihood. As our experimental results demonstrate the GPC+ method is an effective way to make use of the privileged information, and the heteroscedastic noise term improves GPC.

Recently, Feyereisl et al. (2014) proposed a new structure learning framework with privileged information to perform the task of object localization in images. It would be a fruitful future direction to combine this framework with Bayesian structured learning with Gaussian processes by Bratieres et al. (2014), when incorporating privileged information. Also, as future work, we plan to extend the GPC+ to the multiclass situation and to speed up computation by devising a quadrature-free expectation propagation similar to (Riihimäki et al., 2013).

		GPC	GPC+ (Ours)	SVM	SVM+
		image	image+text	image	image+text
1	Chimpanzee vs Giant panda	84.06 ± 0.15	**84.13 ± 0.15**	83.09 ± 0.15	83.77 ± 0.14
2	Chimpanzee vs Leopard	85.25 ± 0.13	**85.30 ± 0.12**	84.71 ± 0.15	84.89 ± 0.12
3	Chimpanzee vs Persian cat	83.36 ± 0.12	**83.47 ± 0.11**	82.65 ± 0.11	83.02 ± 0.11
4	Chimpanzee vs Pig	80.17 ± 0.23	**80.51 ± 0.26**	79.47 ± 0.27	79.95 ± 0.25
5	Chimpanzee vs Hippopotamus	81.00 ± 0.11	**81.32 ± 0.11**	80.42 ± 0.11	80.79 ± 0.11
6	Chimpanzee vs Humpback whale	**94.27 ± 0.08**	94.20 ± 0.08	93.77 ± 0.11	94.14 ± 0.10
7	Chimpanzee vs Raccoon	**80.39 ± 0.13**	80.34 ± 0.13	79.96 ± 0.14	80.13 ± 0.13
8	Chimpanzee vs Rat	80.05 ± 0.27	**80.20 ± 0.26**	78.46 ± 0.30	79.54 ± 0.27
9	Chimpanzee vs Seal	85.89 ± 0.12	**85.98 ± 0.12**	85.05 ± 0.12	85.65 ± 0.13
10	Giant panda vs Leopard	**85.80 ± 0.12**	85.73 ± 0.13	85.14 ± 0.15	85.40 ± 0.15
11	Giant panda vs Persian cat	**87.03 ± 0.12**	87.00 ± 0.12	85.73 ± 0.13	86.47 ± 0.13
12	Giant panda vs Pig	79.76 ± 0.23	**79.82 ± 0.22**	77.45 ± 0.29	79.14 ± 0.24
13	Giant panda vs Hippopotamus	84.52 ± 0.12	**84.54 ± 0.13**	83.10 ± 0.17	84.29 ± 0.14
14	Giant panda vs Humpback whale	94.99 ± 0.09	94.79 ± 0.09	94.26 ± 0.11	**95.04 ± 0.09**
15	Giant panda vs Raccoon	**82.96 ± 0.13**	82.95 ± 0.13	81.62 ± 0.18	82.45 ± 0.14
16	Giant panda vs Rat	**82.90 ± 0.24**	82.72 ± 0.23	79.93 ± 0.30	81.95 ± 0.25
17	Giant panda vs Seal	85.86 ± 0.13	**85.98 ± 0.12**	84.83 ± 0.17	85.77 ± 0.13
18	Leopard vs Persian cat	87.91 ± 0.11	**88.22 ± 0.09**	88.20 ± 0.08	88.17 ± 0.09
19	Leopard vs Pig	**79.02 ± 0.24**	78.87 ± 0.24	78.11 ± 0.24	78.20 ± 0.26
20	Leopard vs Hippopotamus	83.89 ± 0.14	**84.10 ± 0.13**	83.63 ± 0.13	83.72 ± 0.14
21	Leopard vs Humpback whale	95.14 ± 0.07	**95.20 ± 0.07**	95.04 ± 0.08	95.10 ± 0.07
22	Leopard vs Raccoon	73.56 ± 0.17	**73.75 ± 0.16**	73.06 ± 0.20	73.25 ± 0.19
23	Leopard vs Rat	**82.37 ± 0.20**	82.24 ± 0.21	81.21 ± 0.25	81.99 ± 0.23
24	Leopard vs Seal	86.73 ± 0.10	**86.94 ± 0.10**	86.66 ± 0.10	86.64 ± 0.10
25	Persian cat vs Pig	75.64 ± 0.24	**76.04 ± 0.22**	75.47 ± 0.24	75.46 ± 0.25
26	Persian cat vs Hippopotamus	84.97 ± 0.10	**85.14 ± 0.10**	84.47 ± 0.12	84.73 ± 0.11
27	Persian cat vs Humpback whale	92.33 ± 0.09	92.36 ± 0.09	92.48 ± 0.09	**92.58 ± 0.10**
28	Persian cat vs Raccoon	84.45 ± 0.11	**84.64 ± 0.10**	84.19 ± 0.10	84.28 ± 0.13
29	Persian cat vs Rat	65.03 ± 0.30	65.69 ± 0.30	65.16 ± 0.32	**65.92 ± 0.32**
30	Persian cat vs Seal	81.20 ± 0.14	**81.41 ± 0.13**	81.02 ± 0.14	81.14 ± 0.13
31	Pig vs Hippopotamus	72.42 ± 0.27	**72.72 ± 0.27**	72.24 ± 0.25	72.27 ± 0.27
32	Pig vs Humpback whale	91.62 ± 0.15	**91.77 ± 0.16**	91.48 ± 0.17	91.75 ± 0.16
33	Pig vs Raccoon	75.54 ± 0.24	**75.76 ± 0.23**	75.59 ± 0.25	75.65 ± 0.25
34	Pig vs Rat	69.99 ± 0.26	**70.37 ± 0.26**	69.22 ± 0.30	69.77 ± 0.28
35	Pig vs Seal	76.08 ± 0.24	**76.62 ± 0.21**	75.64 ± 0.23	76.20 ± 0.22
36	Hippopotamus vs Humpback whale	85.96 ± 0.12	**86.19 ± 0.12**	85.98 ± 0.12	85.98 ± 0.10
37	Hippopotamus vs Raccoon	80.68 ± 0.14	**80.85 ± 0.13**	80.38 ± 0.15	80.47 ± 0.13
38	Hippopotamus vs Rat	**78.50 ± 0.27**	78.17 ± 0.26	77.32 ± 0.26	77.55 ± 0.27
39	Hippopotamus vs Seal	69.31 ± 0.17	**69.39 ± 0.18**	68.47 ± 0.18	68.96 ± 0.19
40	Humpback whale vs Raccoon	92.07 ± 0.09	92.22 ± 0.08	**92.38 ± 0.09**	92.31 ± 0.09
41	Humpback whale vs Rat	**89.01 ± 0.22**	88.85 ± 0.22	88.61 ± 0.24	88.82 ± 0.24
42	Humpback whale vs Seal	81.42 ± 0.16	**81.81 ± 0.16**	81.41 ± 0.18	81.62 ± 0.16
43	Raccoon vs Rat	**74.83 ± 0.27**	74.77 ± 0.25	74.09 ± 0.24	74.26 ± 0.25
44	Raccoon vs Seal	84.93 ± 0.13	**85.17 ± 0.13**	84.56 ± 0.12	84.64 ± 0.12
45	Rat vs Seal	75.08 ± 0.28	**75.36 ± 0.27**	74.75 ± 0.28	74.83 ± 0.28
	Average accuracy	82.40 ± 0.10	**82.52 ± 0.10**	81.79 ± 0.10	82.19 ± 0.10
	Average ranking	2.09	**1.40**	3.71	2.80

Table 6.2: Accuracy performance on the AwA dataset over 100 repeats (mean and standard error). We used *SURF* image features as the original data space and *DeCAF* deep neural network image features as the privileged space. The best method for each binary task is highlighted in **boldface**. An average rank equal to one means that the corresponding method has the highest accuracy on the 45 tasks (the lower the better).

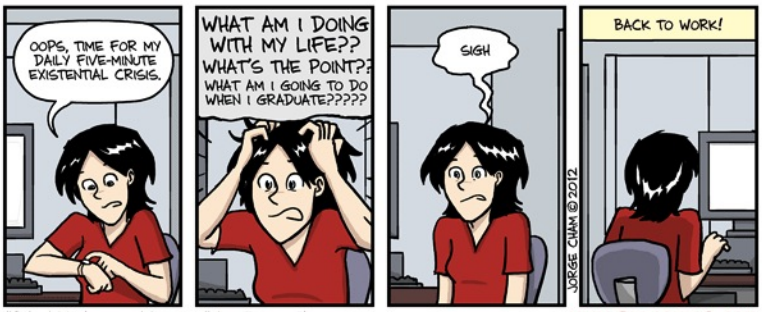
		GPC	GPC+ (Ours)	SVM	SVM+
		image	image+text	image	image+text
1	Chimpanzee vs Giant panda	84.06 ± 0.15	**84.14 ± 0.14**	83.09 ± 0.15	83.35 ± 0.15
2	Chimpanzee vs Leopard	85.25 ± 0.13	**85.27 ± 0.12**	84.71 ± 0.15	84.81 ± 0.12
3	Chimpanzee vs Persian cat	83.36 ± 0.12	**83.48 ± 0.11**	82.65 ± 0.11	82.91 ± 0.10
4	Chimpanzee vs Pig	80.17 ± 0.23	**80.49 ± 0.26**	79.47 ± 0.27	79.65 ± 0.24
5	Chimpanzee vs Hippopotamus	81.00 ± 0.11	**81.31 ± 0.11**	80.42 ± 0.11	80.63 ± 0.11
6	Chimpanzee vs Humpback whale	94.27 ± 0.08	94.20 ± 0.08	93.77 ± 0.11	**94.36 ± 0.08**
7	Chimpanzee vs Raccoon	**80.39 ± 0.13**	80.33 ± 0.13	79.96 ± 0.14	79.86 ± 0.13
8	Chimpanzee vs Rat	80.05 ± 0.27	**80.17 ± 0.26**	78.46 ± 0.30	79.77 ± 0.28
9	Chimpanzee vs Seal	85.89 ± 0.12	**85.96 ± 0.12**	85.05 ± 0.12	85.42 ± 0.15
10	Giant panda vs Leopard	**85.80 ± 0.12**	85.78 ± 0.13	85.14 ± 0.15	85.17 ± 0.14
11	Giant panda vs Persian cat	**87.03 ± 0.12**	87.01 ± 0.12	85.73 ± 0.13	86.03 ± 0.12
12	Giant panda vs Pig	79.76 ± 0.23	**79.81 ± 0.22**	77.45 ± 0.29	78.58 ± 0.26
13	Giant panda vs Hippopotamus	84.52 ± 0.12	**84.55 ± 0.13**	83.10 ± 0.17	83.83 ± 0.17
14	Giant panda vs Humpback whale	**94.99 ± 0.09**	94.79 ± 0.09	94.26 ± 0.11	94.93 ± 0.08
15	Giant panda vs Raccoon	82.96 ± 0.13	**82.97 ± 0.13**	81.62 ± 0.18	81.93 ± 0.15
16	Giant panda vs Rat	**82.90 ± 0.24**	82.73 ± 0.23	79.93 ± 0.30	81.62 ± 0.25
17	Giant panda vs Seal	85.86 ± 0.13	**86.00 ± 0.13**	84.83 ± 0.17	85.17 ± 0.15
18	Leopard vs Persian cat	87.91 ± 0.11	88.18 ± 0.09	**88.20 ± 0.08**	87.86 ± 0.11
19	Leopard vs Pig	**79.02 ± 0.24**	78.88 ± 0.24	78.11 ± 0.24	77.96 ± 0.29
20	Leopard vs Hippopotamus	83.89 ± 0.14	**84.12 ± 0.13**	83.63 ± 0.13	83.58 ± 0.14
21	Leopard vs Humpback whale	95.14 ± 0.07	**95.21 ± 0.07**	95.04 ± 0.08	95.09 ± 0.07
22	Leopard vs Raccoon	73.56 ± 0.17	**73.72 ± 0.17**	73.06 ± 0.20	72.68 ± 0.19
23	Leopard vs Rat	**82.37 ± 0.20**	82.32 ± 0.21	81.21 ± 0.25	81.15 ± 0.24
24	Leopard vs Seal	86.73 ± 0.10	**86.95 ± 0.10**	86.66 ± 0.10	86.61 ± 0.10
25	Persian cat vs Pig	75.64 ± 0.24	**75.98 ± 0.23**	75.47 ± 0.24	75.31 ± 0.25
26	Persian cat vs Hippopotamus	84.97 ± 0.10	**85.12 ± 0.10**	84.47 ± 0.12	84.52 ± 0.11
27	Persian cat vs Humpback whale	92.33 ± 0.09	92.36 ± 0.09	92.48 ± 0.09	**92.51 ± 0.09**
28	Persian cat vs Raccoon	84.45 ± 0.11	**84.67 ± 0.10**	84.19 ± 0.10	84.21 ± 0.10
29	Persian cat vs Rat	65.03 ± 0.30	**65.57 ± 0.29**	65.16 ± 0.32	65.20 ± 0.28
30	Persian cat vs Seal	81.20 ± 0.14	**81.39 ± 0.13**	81.02 ± 0.14	80.74 ± 0.16
31	Pig vs Hippopotamus	72.42 ± 0.27	**72.63 ± 0.26**	72.24 ± 0.25	71.82 ± 0.26
32	Pig vs Humpback whale	91.62 ± 0.15	**91.75 ± 0.16**	91.48 ± 0.17	91.61 ± 0.15
33	Pig vs Raccoon	75.54 ± 0.24	**75.75 ± 0.22**	75.59 ± 0.25	75.08 ± 0.22
34	Pig vs Rat	69.99 ± 0.26	**70.32 ± 0.26**	69.22 ± 0.30	69.66 ± 0.29
35	Pig vs Seal	76.08 ± 0.24	**76.58 ± 0.22**	75.64 ± 0.23	75.96 ± 0.23
36	Hippopotamus vs Humpback whale	85.96 ± 0.12	**86.17 ± 0.12**	85.98 ± 0.12	85.54 ± 0.13
37	Hippopotamus vs Raccoon	80.68 ± 0.14	**80.84 ± 0.13**	80.38 ± 0.15	80.25 ± 0.16
38	Hippopotamus vs Rat	**78.50 ± 0.27**	78.20 ± 0.26	77.32 ± 0.26	77.66 ± 0.26
39	Hippopotamus vs Seal	69.31 ± 0.17	**69.38 ± 0.18**	68.47 ± 0.18	68.52 ± 0.19
40	Humpback whale vs Raccoon	92.07 ± 0.09	92.20 ± 0.08	**92.38 ± 0.09**	92.33 ± 0.08
41	Humpback whale vs Rat	**89.01 ± 0.22**	88.83 ± 0.22	88.61 ± 0.24	88.82 ± 0.22
42	Humpback whale vs Seal	81.42 ± 0.16	**81.71 ± 0.16**	81.41 ± 0.18	81.00 ± 0.18
43	Raccoon vs Rat	**74.83 ± 0.27**	74.78 ± 0.25	74.09 ± 0.24	73.87 ± 0.26
44	Raccoon vs Seal	84.93 ± 0.13	**85.14 ± 0.12**	84.56 ± 0.12	84.57 ± 0.13
45	Rat vs Seal	75.08 ± 0.28	**75.35 ± 0.26**	74.75 ± 0.28	74.85 ± 0.27
	Average accuracy	82.40 ± 0.10	**82.51 ± 0.10**	81.79 ± 0.10	81.93 ± 0.10
	Average ranking	1.98	**1.40**	3.44	3.18

Table 6.3: Accuracy performance on the AwA dataset over 100 repeats (mean and standard error). We used *SURF* image features as the original data space and *attributes* in form of DAP features as the privileged space. The best method for each binary task is highlighted in **boldface**. An average rank equal to one means that the corresponding method has the highest accuracy on the 45 tasks (the lower the better).

Chapter 7

Conclusions and Future Directions

In this chapter, we summarize the main contributions of this thesis and discuss possible future directions for attribute-based models. In this thesis, we started with semantic attributes as an interpretable binary image representation and showed how to augment them with discriminative attributes that can be learned directly from the image content. Then, we showed how to use attributes as privileged information to the image data during learning. We explained the main principles of learning with privileged information, described our findings about what kind of information is privileged for image based recognition, and the lesson learned from our user studies on human learning versus machine learning from images. Despite huge progress over the last five years, learning the attributes from image data remains a challenging task. Among attractive future directions for attribute-based models is learning attributes from 3D object models instead of learning them from images. The 3D view on attributes opens a new principled way of understanding and analyzing what makes an attribute of the objects: is it difference in shape, in appearance, or the way we interact with the objects?


OOPS, TIME FOR MY DAILY FIVE-MINUTE EXISTENTIAL CRISIS.
WHAT AM I DOING WITH MY LIFE?? WHAT'S THE POINT?? WHAT AM I GOING TO DO WHEN I GRADUATE?????
SIGH
JORGE CHAM ©2012
BACK TO WORK!

"Piled Higher and Deeper" by Jorge Cham WWW.PHDCOMICS.COM

7.1 Conclusions and Discussion

Attributes is an active area of research in computer vision that involves a variety of tasks, such as describing objects in images, recognizing known and unknown classes of objects, comparing images of objects based on their attributes, intelligent image search, etc. In this thesis, we focused on the role of attributes in the context of image based object recognition.

The first part of this thesis addressed the attributes in the context of mid-level feature learning framework, where semantic attributes are combined with discriminative attributes to form a desirable representation for object recognition tasks in images. *The contributions* of this part of the thesis are parametric and non-parametric machine learning methods for data-driven augmentation of attribute representations forming Chapter 3 and Chapter 4 of the thesis:

1. Augmented Attributes: Parametric View.
   In this chapter, we introduced the parametric approach for augmenting semantic attribute representation with non-semantic but discriminative attributes that help to resolve the object recognition task in images. Our approach combines (i) the unsupervised augmentation using autoencoder model and (ii) the folk wisdom supervision using large margin nearest neighbor principle.

2. Augmented Attributes: Non-parametric View.
   In this chapter, we addressed the non-parametric probabilistic approach for learning discriminative attribute representations. Our probabilistic model combines (i) the neighborhood likelihood function with folk wisdom supervision as preference relation, (ii) the flexible Indian Buffet Process prior on infinite sparse binary matrices, and (iii) the data likelihood in form of linear Gaussian or linear probit model. This combination enables to couple effectively the structure of the semantic space with the continuously growing structure of the neighborhood preserving infinite latent feature space.

The augmented attribute representations form a suitable basement for addressing the next real-world challenges at the Internet scale. Among most attractive future explorations in this direction are (i) unsupervised and weakly supervised parametric models for attribute-based classification at large scale, (ii) learning mid-level feature representation that combines attribute semantics with discriminative features from the deep convolutional neural networks and (iii) scalable non-parametric models for continuous streams of data.

The second part of this thesis addresses the attributes in the context of new learning framework called learning using privileged information (LUPI), where attributes can be used as privileged information in addition to the image data during training. *The contributions* of this part include parametric and non-parametric machine learning methods for learning with privileged information, forming Chapter 5 and Chapter 6 of the thesis:

1. Learning with Privileged Information: Parametric View.
   In this chapter, we introduced the main principles of learning with privileged information for object recognition task, and explored the maximum-margin approach for solving LUPI. Our approach learned easy and hard objects in the privileged space, and then transferred this knowledge to train a better classifier in the original data space. In contrast to the standard SVM, where all training samples have equal weights, we used the margin distance to the separating hyperplane in the privileged space as the easy-hard score, and then re-weighted the training samples in the original data space according to this score.

2. Learning with Privileged Information: Non-parametric View.
   In this chapter, we showed the non-parametric Bayesian approach to address the framework of learning using privileged information. We placed LUPI in the context of Gaussian process classification, where the privileged information can be naturally treated as the noise term in the latent function of the Gaussian process classifier. In contrast to the standard Gaussian Processes classification setting, where the latent function is just a nuisance, in LUPI it becomes a natural measure of confidence about the training data by modulating the slope of the sigmoid-shaped likelihood function.

The framework for learning with privileged information was established very recently and, hence, raises a variety of questions to explore. For example: (i) what is the criterion of useful privileged information for the task at hand? In other words, how can we evaluate a priori the quality of the privileged information? (ii) How can we use privileged information when there is no direct correspondence between feature representation in original and privileged spaces? (iii) What is the relation between LUPI and domain adaptation, transfer learning and co-training frameworks studied in the computer vision literature before?

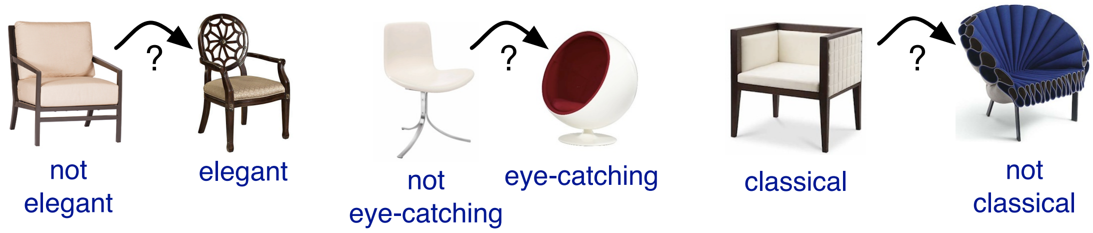


Figure 7.1: How to enable a computer to learn the semantic properties of different 3D objects, for example, *elegant*, *eye-catching*, *classical*. The principal idea is to combine state-of-the-art data-driven approach with advanced 3D shape modeling to enable computer understanding of semantic attributes and automatic synthesis of new smart 3D shapes.

7.2 Future Directions

Despite significant progress in the topic of learning the attributes for specific tasks, two key observations can summarize the current stage: (i) attributes are not easy to learn from image data and require much human annotation; (ii) very little is known about the relations between what was learned from data and what makes a semantic attribute of the object. Since in real life we observe objects in 3D space, an interesting future direction would be to learn the attributes from the 3D data directly instead of learning them from 2D projections onto the image plane. The underlying goal would be to develop a systematic approach of learning attributes from 3D data, understanding and analyzing what are the learnable attributes of the 3D objects.

Attributes of 3D objects The attributes of 3D objects can be naturally defined based on human preference criterion, shape geometry, affordance, etc. The attributes of human preference capture the *style* of the objects: elegant, eye-catching, modern, classical, etc. Attributes of shape geometry capture *shape* variations: round, v-shaped, curved, long. Attributes of affordance capture our *experience* when interacting with the object, for example, body pose and level of comfort: comfortable, relaxing.

3D shape modeling Several methods have been developed recently for discovering structure, style and variations in a large collection of 3D shape models (Kim et al., 2013; Fish et al., 2014; Su et al., 2014; Zheng et al., 2014; Umetani et al., 2012). The main idea is to model the 3D shape of the object as deformable part-based (DPB) template that captures parts, i.e. appearance of distinctive object parts, their configurations, i.e. how the parts relate to each other, and allows deformation of the parts configuration. For example, for modeling a chair, parts could be seat, leg, arm, backrest and the configuration could be the angle between adjacent parts. The advantages of using this approach are the following: (i) it does not require manual annotation of the shape parts, which are automatically discovered from the data collection; (ii) it is flexible to cover the intra-class variation of the objects, as the number of templates is not fixed and can grow as required forming the set of deformable part-based templates. Finally, the deformable part-based templates are fitted to data by finding the deformation of parts configuration that best matches the 3D shapes of objects in the data collection. It is formalized as optimizing an energy function measuring the fit of the template T with parts $p_i \in P$ and their configurations $(p_i, p_j \in E)$ to the shape S from dataset with corresponding locations l_i:

$$\mathrm{E}(T, S) = \sum_{p_i \in T} \mathrm{E}_{data}(p_i, l_i) + \sum_{(p_i, p_j) \in E} \mathrm{E}_{smooth}(p_i, p_j, l_i, l_j) \qquad (7.1)$$

The data term $\mathrm{E}_{data}(p_i, l_i)$ is the cost of placing the template part p_i at the location l_i of the shape S that controls local similarity: it is small if local shape features between points on the shape and corresponding points on the template match. The smoothing term $\mathrm{E}_{smooth}(p_i, p_j, l_i, l_j)$ is the deformation cost between the parts p_i, p_j placed at the locations l_i, l_j that controls configuration consistency: it is small if the parts configuration (p_i, p_j) is aligned with the geometry of shape S at the locations l_i, l_j. Given a collection of 3D shapes, minimizing the energy produces the set of templates that best capture variability in data.

This model will allow us to visualize the 3D attributes by highlighting the discriminative parts and part configurations of the DPB templates that support the presence of the attribute. Please, refer here to our illustrative Figure 7.2 (middle object) of the 3D template for attribute *elegant*. Parts that are common to elegant chairs (in the collection) and discriminate them from non-elegant chairs are highlighted with the blue mask.

There is always the possibility that certain goals are difficult to achieve, because of: (i) limited amount of data to learn reliably the attributes, (ii) unbalanced data when there are only few positive samples and many negatives, (iii) lack of texture or color information in the 3D models that could bias certain attributes, for example, an eye-catching building because of red colored doors, etc. Alternatively, we could utilize a vast amount of image data (that is easy to collect

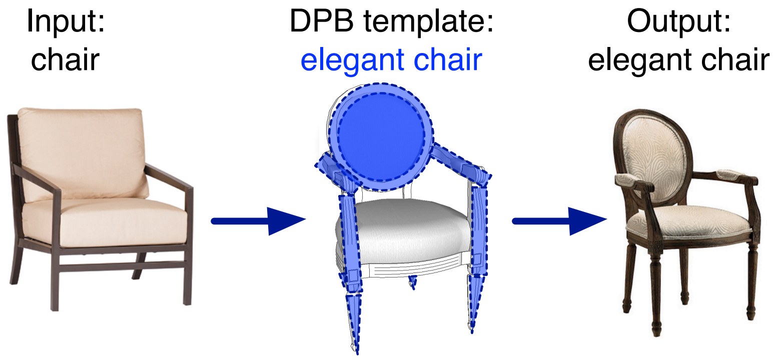


Figure 7.2: Application of creating 3D object model by attribute description.

and to label) in addition to available 3D models and learn 3D attribute models from heterogeneous domains: 2D images and 3D shape models. This opens a whole new perspective on future explorations including: (i) how to build the correspondence between images and 3D shapes (as, for example, Su et al. (2014)); (ii) how to transfer the information about color and appearance from images to 3D shape models as Xu et al. (2011); (iii) how exactly to combine these steps in order to learn the 3D attributes from images and 3D shapes jointly.

Synthesizing 3D models by attribute description Imagine the following scenario depicted in our Figure 7.2: the user picks up one of the objects in the collection on display, for example, a chair (on the left), and chooses to make it *elegant* (on the right). The task of the system is to know what makes an object elegant and to modify the 3D model of the input object with respect to this request. From the perspective of 3D attribute-based model, this application includes: (i) learning the DPB template of the semantic 3D attribute elegant (middle picture), (ii) transforming the input chair model into the elegant chair model with respect to the DPB template of the elegant attribute model. Apart from this, the 3D semantic attributes would be very interesting to explore for a variety of other applications including attribute-based categorization of 3D shapes, smart 3D shape editing and intelligent search for objects in 3D.

Bibliography

Akata, Zeynep, Perronnin, Florent, Harchaoui, Zaid, and Schmid, Cordelia. Label-embedding for attribute-based classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 49

Amari, Shun-Ichi. Natural gradient works efficiently in learning. *Neural Computation*, 1998. 30

Andrieu, Christophe, De Freitas, Nando, Doucet, Arnaud, and Jordan, Michael I. An introduction to mcmc for machine learning. *Machine Learning*, pp. 5–43, 2003. 64

Austerweil, Joseph and Griffiths, Tom. Learning invariant features using the transformed Indian Buffet Process. In *Conference on Neural Information Processing Systems (NIPS)*, 2010. 68, 75

Bay, Herbert, Ess, Andreas, Tuytelaars, Tinne, and Van Gool, Luc. Speeded-up robust features (SURF). *Computer Vision Image Understanding (CVIU)*, 2008. 26, 49, 117

Bekkerman, Ron and Jeon, Jiwoon. Multi-modal clustering for multimedia collections. In *Computer Vision and Pattern Recognition (CVPR)*, 2007. 98

Berg, Tamara L., Berg, Alexander C., and Shih, Jonathan. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision (ECCV)*, 2010. 9, 48, 98, 117

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006. 48

Blaschko, Matthew B. and Lampert, Christoph H. Correlational spectral clustering. In *Computer Vision and Pattern Recognition (CVPR)*, 2008. 85

Boureau, Y-Lan, Bach, Francis, LeCun, Yann, and Ponce, Jean. Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 21

Branson, Steve, Wah, Catherine, Babenko, Boris, Schroff, Florian, Welinder, Peter, Perona, Pietro, and Belongie, Serge. Visual recognition with humans in the loop. In *European Conference on Computer Vision (ECCV)*, 2010. 8, 21

Bratieres, Sébastien, Quadrianto, Novi, and Ghahramani, Zoubin. GPstruct: Bayesian structured prediction using Gaussian Process. In *International Conference on Machine Learing (ICML)*, 2014. 122

Chapelle, Olivier. Training a support vector machine in the primal. *Neural Computation*, pp. 1155–1178, 2007. 90

Chen, Chao-Yeh and Grauman, Kristen. Clues from the beaten path: Location estimation with bursty sequences of tourist photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2011. 85

Chen, Jixu, Liu, Xiaoming, and Lyu, Siwei. Boosting with side information. In *Asian Conference on Computer Vision (ACCV)*, 2013. 82

Chen, Ke and Kämäräinen, Joni-Kristian. Learning to count with back-propagated information. In *International Conference on Pattern Recognition (ICPR)*, 2014. 82

Christoudias, C Mario, Urtasun, Raquel, and Darrell, Trevor. Multi-view learning in the presence of view disagreement. In *Uncertainty in Artificial Intelligence (UAI)*, 2008. 85

Csurka, Gabriella, Dance, Christopher R., Fan, Lixin, Willamowski, Jutta, and Bray, Cedric. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004. 27

Demšar, Janez. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research (JMLR)*, pp. 1–30, 2006. 121

Donahue, Jeff and Grauman, Kristen. Annotator rationales for visual recognition. In *International Conference on Computer Vision (ICCV)*, 2011. 85

Donahue, Jeff, Jia, Yangqing, Vinyals, Oriol, Hoffman, Judy, Zhang, Ning, Tzeng, Eric, and Darrell, Trevor. DeCAF: A Deep Convolutional Activation Feature for generic visual recognition. In *International Conference on Machine Learing (ICML)*, 2014. 23, 34, 119

Doshi-Velez, Finale and Ghahramani, Zoubin. Correlated non-parametric latent feature models. In *Uncertainty in Artificial Intelligence (UAI)*, 2009. 68

Ebert, Sandra, Larlus, Diane, and Schiele, Bernt. Extracting structures in image collections for object recognition. In *European Conference on Computer Vision (ECCV)*, 2010. 55

Everingham, Mark, Gool, Luc J. Van, Williams, Christopher K.I., Winn, John M., and Zisserman, Andrew. The Pascal VOC challenge. *International Journal of Computer Vision (IJCV)*, pp. 303–338, 2010. 84

Everingham, Mark, Eslami, S. M. Ali, Gool, Luc J. Van, Williams, Christopher K.I., Winn, John M., and Zisserman, Andrew. The Pascal visual object classes challenge - a retrospective. *International Journal of Computer Vision (IJCV)*, 2014. 22

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research (JMLR)*, 2008. 90

Farhadi, Ali, Endres, Ian, Hoiem, Derek, and Forsyth, David A. Describing objects by their attributes. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 8, 20, 21

Fei-Fei, Li, Fergus, Robert, and Perona, Pietro. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, pp. 594–611, 2006. 49

Feng, Jiashi, Jegelka, Stefanie, Yan, Shuicheng, and Darrell, Trevor. Learning scalable discriminative dictionary with sample relatedness. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 69, 76

Ferrari, Vittorio and Zisserman, Andrew. Learning visual attributes. In *Conference on Neural Information Processing Systems (NIPS)*, 2008. 8, 9

Feyereisl, Jan and Aickelin, Uwe. Privileged information for data clustering. *Information Sciences*, pp. 4–23, 2012. 82

Feyereisl, Jan, Kwak, Suha, Son, Jeany, and Han, Bohyung. Object localization based on structural svm using privileged information. In *Conference on Neural Information Processing Systems (NIPS)*, 2014. 122

Fish, Noa, Averkiou, Melinos, van Kaick, Oliver, Sorkine-Hornung, Olga, Cohen-Or, Daniel, and Mitra, Niloy J. Meta-representation of shape families. *ACM Transactions on Graphics (SIGGRAPH)*, 2014. 128

Fouad, Shereen, Tino, Peter, Raychaudhury, Somak, and Schneider, Petra. Incorporating privileged information through metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1086–1098, 2013. 82

Gehler, Peter and Nowozin, Sebastian. On feature combination for multiclass object classification. In *International Conference on Computer Vision (ICCV)*, 2009. 84

Gershman, Samuel J., Frazier, Peter I., and Blei, David M. Distance dependent infinite latent feature models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, pp. 334–345, 2015. 68, 69, 74

Gilks, Walter R. and Wild, Pascal. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 1992. 66

Goldberg, Paul W., Williams, Christopher K.I., and Bishop, Christopher M. Regression with input-dependent noise: A Gaussian Process treatment. In *Conference on Neural Information Processing Systems (NIPS)*, 1998. 112

Goldberger, Jacob, Roweis, Sam T., Hinton, Geoffrey E., and Salakhutdinov, Ruslan. Neighbourhood components analysis. In *Conference on Neural Information Processing Systems (NIPS)*, 2004. 59

Görür, Dilan, Jäkel, Frank, and Rasmussen, Carl Edward. A choice model with infinitely many latent features. In *International Conference on Machine Learing (ICML)*, 2006. 60, 68

Gregor, Karo and LeCun, Yann. Emergence of complex-like cells in a temporal product network with local receptive fields. *arXiv:1006.0448*, 2010. 48

Griffiths, Thomas L. and Ghahramani, Zoubin. Infinite latent feature models and the Indian Buffet Process. In *Conference on Neural Information Processing Systems (NIPS)*, 2005. 58, 60, 61, 65, 68, 74

Griffiths, Thomas L. and Ghahramani, Zoubin. The Indian Buffet Process: An introduction and review. *Journal of Machine Learning Research (JMLR)*, 2011. 68

Hinton, Geoffrey and Salakhutdinov, Ruslan. Reducing the dimensionality of data with neural networks. *Science*, pp. 504 – 507, 2006. 40, 48

Hinton, Geoffrey E., Krizhevsky, Alex, and Wang, Sida D. Transforming auto-encoders. In *International Conference on Artificial Neural Networks (ICANN)*, 2011. 48

Jaakkola, Tommi and Haussler, David. Exploiting generative models in discriminative classifiers. In *Conference on Neural Information Processing Systems (NIPS)*, 1998. 30

Jayaraman, Dinesh and Grauman, Kristen. Zero-shot recognition with unreliable attributes. In *Conference on Neural Information Processing Systems (NIPS)*, 2014. 16, 49

Kalal, Zdenek, Matas, Jiri, and Mikolajczyk, Krystian. Online learning of robust object detectors during unstable tracking. In *ICCV Workshop On-line learning for Computer Vision*, 2009. 85

Khamis, Sameh and Lampert, Christoph H. CoConut: Co-classification with output space regularization. In *British Machine Vision Conference (BMVC)*, 2014. 85

Kim, Vladimir G., Li, Wilmot, Mitra, Niloy J., Chaudhuri, Siddhartha, DiVerdi, Stephen, and Funkhouser, Thomas. Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (SIGGRAPH)*, 2013. 128

Knowles, David and Ghahramani, Zoubin. Infinite sparse factor analysis and infinite independent components analysis. In *Independent Component Analysis and Signal Separation*, 2007. 68

Kovashka, Adriana and Grauman, Kristen. Attribute adaptation for personalized image search. In *International Conference on Computer Vision (ICCV)*, 2013. 20

Kovashka, Adriana, Parikh, Devi, and Grauman, Kristen. WhittleSearch: image search with relative attribute feedback. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. 8, 18

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Conference on Neural Information Processing Systems (NIPS)*, 2012. 23, 32, 33

Kuettel, Daniel, Guillaumin, Matthieu, and Ferrari, Vittorio. Segmentation propagation in ImageNet. In *European Conference on Computer Vision (ECCV)*, 2012. 84

Kulkarni, Girish, Premraj, Visruth, Dhar, Sagnik, Li, Siming, Choi, Yejin, Berg, Alexander C, and Berg, Tamara L. Baby talk: Understanding and generating simple image descriptions. In *Computer Vision and Pattern Recognition (CVPR)*, 2011. 20

Kumar, Neeraj, Belhumeur, Peter, and Nayar, Shree. FaceTracer: A search engine for large collections of images with faces. In *European Conference on Computer Vision (ECCV)*, 2008. 8, 18

Kuss, Malte and Rasmussen, Carl Edward. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research (JMLR)*, pp. 1679–1704, 2005. 113

Lampert, Christoph H., Nickisch, Hannes, and Harmeling, Stefan. Learning to detect unseen object classes by betweenclass attribute transfer. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 8, 13, 16, 40, 41, 48, 49, 50, 51, 54, 55, 72, 92, 93, 119

Lampert, Christoph H., Nickisch, Hannes, and Harmeling, Stefan. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2013. 41, 48, 49, 50, 72, 92, 93, 119

Lapin, Maksim, Hein, Matthias, and Schiele, Bernt. Learning using privileged information: SVM+ and weighted SVM. *Neural Networks*, 2014. 90

Lázaro-Gredilla, Miguel and Titsias, Michalis K. Variational heteroscedastic Gaussian Process regression. In *International Conference on Machine Learing (ICML)*, 2011. 112

Le, Quoc V and Mikolov, Tomas. Distributed representations of sentences and documents. In *International Conference on Machine Learing (ICML)*, 2014. 118

Lowe, David G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, pp. 91–110, 2004. 22, 23, 24

Mahajan, Dhruv, Sellamanickam, Sundararajan, and Nair, Vinod. A joint learning framework for attribute models and object descriptions. In *International Conference on Computer Vision (ICCV)*, 2011. 49

Maron, Oded and Ratan, Aparna Lakshmi. Multiple-instance learning for natural scene classification. In *International Conference on Machine Learing (ICML)*, 1998. 84

Meeds, Edward, Ghahramani, Zoubin, Neal, Radford M., and Roweis, Sam T. Modeling dyadic data with binary latent factors. In *Conference on Neural Information Processing Systems (NIPS)*, 2007. 65

Mensink, Thomas, Gavves, Efstratios, and Snoek, Cees GM. COSTA: Co-Occurrence Statistics for zero-shot classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 16, 49

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In *Conference on Neural Information Processing Systems (NIPS)*, 2013. 98, 118

Miller, Kurt, Griffiths, Thomas, and Jordan, Michael. Nonparametric latent feature models for link prediction. In *Conference on Neural Information Processing Systems (NIPS)*, 2009. 68

Miller, Kurt T., Griffiths, Thomas L., and Jordan, Michael I. The phylogenetic Indian Buffet Process: A non-exchangeable nonparametric prior for latent features. In *Uncertainty in Artificial Intelligence (UAI)*, 2008. 68

Minka, Thomas. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, 2001. 77, 113, 115

Mittelman, Roni, Lee, Honglak, Kuipers, Benjamin, and Savarese, Silvio. Weakly supervised learning of mid-level features with Beta-Bernoulli Process Restricted Boltzmann Machines. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 21, 49

Mu, Yadong, Shen, Jialie, and Yan, Shuicheng. Weakly-supervised hashing in kernel space. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 69

Murray, Iain, Adams, Ryan Prescott, and MacKay, David J. C. Elliptical slice sampling. *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, 2010. 66

Neal, Radford M. Slice sampling. *Annals of Statistics*, 2003. 65

Nickisch, Hannes and Rasmussen, Carl Edward. Approximations for binary Gaussian Process classification. *Journal of Machine Learning Research (JMLR)*, pp. 2035–2078, 2008. 113

Norouzi, Mohammad, Fleet, David J., and Salakhutdinov, Ruslan. Hamming distance metric learning. In *Conference on Neural Information Processing Systems (NIPS)*, 2012. 69

Osherson, Daniel N., Stern, Joshua, Wilkie, Ormond, Stob, Michael, and Smith, Edward E. Default probability. *Cognitive Science*, pp. 251–269, 1991. 13, 49

Palatucci, Mark, Pomerleau, Dean, Hinton, Geoffrey E, and Mitchell, Tom M. Zero-shot learning with semantic output codes. In *Conference on Neural Information Processing Systems (NIPS)*, 2009. 13, 49

Parikh, Devi and Grauman, Kristen. Interactively building a discriminative vocabulary of nameable attributes. In *Computer Vision and Pattern Recognition (CVPR)*, 2011a. 9, 48

Parikh, Devi and Grauman, Kristen. Relative attributes. In *International Conference on Computer Vision (ICCV)*, 2011b. 8, 17

Parikh, Devi and Grauman, Kristen. Implied feedback: Learning nuances of user behavior in image search. In *International Conference on Computer Vision (ICCV)*, 2013. 20

Patterson, Genevieve and Hays, James. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. 8, 22, 49

Pechyony, Dmitry and Vapnik, Vladimir. On the theory of learning with privileged information. In *Conference on Neural Information Processing Systems (NIPS)*, 2010. 86

Pechyony, Dmitry and Vapnik, Vladimir. Fast optimization algorithms for solving SVM+. In *Statistical Learning and Data Science*, 2011. 88, 92, 116

Perronnin, Florent and Dance, Christopher R. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition (CVPR)*, 2007. 23, 28, 31

Perronnin, Florent, Sánchez, Jorge, and Mensink, Thomas. Improving the Fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*, 2010. 23, 28, 95, 98

Platt, J. Probabilities for SV Machines. In *Advances in Large Margin Classifiers*. MIT Press, 2000. 14

Quadrianto, Novi and Lampert, Christoph H. Learning multi-view neighborhood preserving projections. In *International Conference on Machine Learing (ICML)*, 2011. 44, 59, 85

Quadrianto, Novi, Kersting, Kristian, Reid, Mark D., Caetano, Tibério S., and Buntine, Wray L. Kernel conditional quantile estimation via reduction revisited. In *International Conference on Data Mining (ICDM)*, 2009. 112

Quadrianto, Novi, Kersting, Kristian, and Xu, Zhao. Gaussian Process. In *Encyclopedia of Machine Learning and Data Mining*. Springer, 2010. 113

Quadrianto, Novi, Sharmanska, Viktoriia, Knowles, David, and Ghahramani, Zoubin. The supervised IBP: Neighbourhood preserving infinite latent feature models. In *Uncertainty in Artificial Intelligence (UAI)*, 2013. 21

Rai, Piyush and Daume III, Hal. Multi-label prediction via sparse infinite CCA. In *Conference on Neural Information Processing Systems (NIPS)*, 2009. 68

Ranzato, Marc'Aurelio, Huang, Fu Jie, Boureau, Y-Lan, and LeCun, Yann. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2007. 48

Rasmussen, Carl Edward and Williams, Christopher K.I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006. 108, 109, 112

Rastegari, Mohammad, Farhadi, Ali, and Forsyth, David A. Attribute discovery via predictable discriminative binary codes. In *European Conference on Computer Vision (ECCV)*, 2012. 21, 49

Rastegari, Mohammad, Diba, Ali, Parikh, Devi, and Farhadi, Ali. Multi-attribute queries: to merge or not to merge? In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 19

Restle, Frank. *Psychology of judgment and choice: A theoretical essay*. John Wiley & Sons, 1961. 60

Riihimäki, Jaakko, Jylänki, Pasi, and Vehtari, Aki. Nested expectation propagation for Gaussian Process classification with a multinomial probit likelihood. *Journal of Machine Learning Research (JMLR)*, pp. 75–109, 2013. 122

Rohrbach, Marcus, Stark, Michael, Szarvas, György, Gurevych, Iryna, and Schiele, Bernt. What helps where - and why? semantic relatedness for knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 9, 49

Rohrbach, Marcus, Stark, Michael, and Schiele, Bernt. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR)*, 2011. 49

Russakovsky, Olga and Fei-Fei, Li. Attribute learning in large-scale datasets. In *ECCV Workshop on Parts and Attributes*, 2010. 49

Russakovsky, Olga, Lin, Yuanqing, Yu, Kai, and Fei-Fei, Li. Object-centric spatial pooling for image classification. In *European Conference on Computer Vision (ECCV)*, 2012. 85

Sadovnik, Amir, Gallagher, Andrew, Parikh, Devi, and Chen, Tsuhan. Spoken attributes: mixing binary and relative attributes to say the right thing. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 19

Salakhutdinov, Ruslan and Hinton, Geoffrey E. Semantic hashing. In *SIGIR workshop on Information Retrieval and applications of Graphical Models*, 2007a. 69

Salakhutdinov, Ruslan and Hinton, Geoffrey E. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, 2007b. 48

Saleh, Babak, Farhadi, Ali, and Elgammal, Ahmed M. Object-centric anomaly detection by attribute-based reasoning. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 20

Sánchez, Jorge and Perronnin, Florent. High-dimensional signature compression for large-scale image classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2011. 32

Sánchez, Jorge, Perronnin, Florent, Mensink, Thomas, and Verbeek, Jakob. Image classification with the Fisher vector: Theory and practice. *International Journal of Computer Vision (IJCV)*, 2013. 28, 29, 31

Seeger, Matthias. Expectation propagation for exponential families. Technical report, Department of EECS, University of California, Berkeley, 2006. 114, 116

Shalev-Shwartz, Shai, Singer, Yoram, and Srebro, Nathan. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *International Conference on Machine Learing (ICML)*, 2007. 90

Sharmanska, Viktoriia, Quadrianto, Novi, and Lampert, Christoph H. Augmented attribute representations. In *European Conference on Computer Vision (ECCV)*, 2012. 21

Sharmanska, Viktoriia, Quadrianto, Novi, and Lampert, Christoph H. Learning to rank using privileged information. In *International Conference on Computer Vision (ICCV)*, 2013. 82, 83, 91

Shi, Zhiyuan, Yang, Yongxin, Hospedales, Timothy M, and Xiang, Tao. Weakly supervised learning of objects, attributes and their associations. In *European Conference on Computer Vision (ECCV)*, 2014. 69

Singh, Saurabh, Gupta, Abhinav, and Efros, Alexei A. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision (ECCV)*, 2012. 21

Sivic, Josef and Zisserman, Andrew. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision (ICCV)*, 2003. 27

Snoek, Cees GM, Worring, Marcel, and Smeulders, Arnold WM. Early versus late fusion in semantic video analysis. In *ACM Multimedia (ACM MM)*, 2005. 84

Su, Hao, Huang, Qixing, Mitra, Niloy J., Li, Yangyan, and Guibas, Leonidas. Estimating image depth using shape collections. *ACM Transactions on Graphics (SIGGRAPH)*, 2014. 128, 130

Sydorov, Vladyslav, Sakurada, Mayu, and Lampert, Christoph H. Deep Fisher kernels – end to end learning of the Fisher kernel GMM parameters. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 31

Tang, Kevin D., Tappen, Marshall F., Sukthankar, Rahul, and Lampert, Christoph H. Optimizing one-shot recognition with micro-set learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 48, 55

Teh, Yee Whye, Görür, Dilan, and Ghahramani, Zoubin. Stick-breaking construction for the Indian Buffet Process. *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, 2007. 62, 65, 66

Tomasik, Brian, Thiha, Phyo, and Turnbull, Douglas. Tagging products using image classification. In *Special Interest Group On Information Retrieval (SIGIR)*, 2009. 28

Tommasi, Tatiana, Orabona, Francesco, and Caputo, Barbara. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 49

Torralba, Antonio B., Fergus, Robert, and Weiss, Yair. Small codes and large image databases for recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2008. 69

Tung, Hsiao-yu and Smola, Alex J. Spectral methods for Indian Buffet Process inference. In *Conference on Neural Information Processing Systems (NIPS)*, 2014. 77

Umetani, Nobuyuki, Igarashi, Takeo, and Mitra, Niloy J. Guided exploration of physically valid shapes for furniture design. *ACM Transactions on Graphics (SIGGRAPH)*, 2012. 128

Vapnik, Vladimir. *The nature of statistical learning theory.* Springer, 1999. 87

Vapnik, Vladimir and Vashist, Akshay. A new learning paradigm: Learning using privileged information. *Neural Networks*, pp. 544–557, 2009. 82, 83, 86, 110

Vedaldi, Andrea, Gulshan, Varun, Varma, Manik, and Zisserman, Andrew. Multiple kernels for object detection. In *International Conference on Computer Vision (ICCV)*, 2009. 84

Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research (JMLR)*, pp. 3371–3408, 2010. 44

Wang, Jun, Kumar, Sanjiv, and Chang, Shih-Fu. Semi-supervised hashing for large-scale search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2012. 69

Wang, Yang and Mori, Greg. A discriminative latent model of object classes and attributes. In *European Conference on Computer Vision (ECCV)*, 2010. 49

Weinberger, Kilian Q. and Saul, Lawrence K. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, pp. 207–244, 2009. 40, 44, 48, 59

Weiss, Yair, Torralba, Antonio, and Fergus, Rob. Spectral hashing. In *Conference on Neural Information Processing Systems (NIPS)*, 2009. 69

Welling, Max, Rosen-Zvi, Michal, and Hinton, Geoffrey E. Exponential family harmoniums with an application to information retrieval. In *Conference on Neural Information Processing Systems (NIPS)*, 2005. 48

Williamson, Sinead, Orbanz, Peter, and Ghahramani, Zoubin. Dependent Indian Buffet Process. *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, 2010. 62, 68

Winn, John, Criminisi, Antonio, and Minka, Thomas. Object categorization by learned universal visual dictionary. In *International Conference on Computer Vision (ICCV)*, 2005. 29

Wolf, Lior, Hassner, Tal, and Taigman, Yaniv. Descriptor based methods in the wild. In *ECCV Workshop on Faces in Real Life Images*, 2008. 49

Xu, Kai, Zheng, Hanlin, Zhang, Hao, Cohen-Or, Daniel, Liu, Ligang, and Xiong, Yueshan. Photo-inspired model-driven 3d object modeling. *ACM Transactions on Graphics (SIGGRAPH)*, 2011. 130

Yang, Heng and Patras, Ioannis. Privileged information-based conditional regression forest for facial feature detection. In *International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2013. 82

Yu, Felix X, Cao, Liangliang, Feris, Rogerio S, Smith, John R, and Chang, Shih-Fu. Designing category-level attributes for discriminative visual recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 16, 49

Zeiler, Matthew D. and Fergus, Rob. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014. 33

Zhai, Ke, Hu, Yuening, Boyd-Graber, Jordan L., and Williamson, Sinead. Modeling images using transformed Indian Buffet Process. In *International Conference on Machine Learing (ICML)*, 2012. 68, 75

Zhen, Yi, Rai, Piyush, Zha, Hongyuan, and Carin, Lawrence. Cross-modal similarity learning via pairs, preferences, and active supervision. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015. 68

Zheng, Youyi, Cohen-Or, Daniel, Averkiou, Melinos, and Mitra, Niloy J. Recurring part arrangements in shape collections. *Computer Graphics Forum (Eurographics)*, 2014. 128