

CONDITIONAL RANDOM FIELD MODELS FOR STRUCTURED
VISUAL OBJECT RECOGNITION

Kun Duan

Submitted to the faculty of the University Graduate School

in partial fulfillment of the requirements for the degree

Doctor of Philosophy

in the School of Informatics and Computing, Indiana University

August, 2014

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the
requirements for the degree of Doctor of Philosophy.

Doctoral Committee

David Crandall, PhD

Kris Hauser, PhD

David Leake, PhD

Russell Lyons, PhD

Devi Parikh, PhD

July 21st, 2014

Copyright © 2014

Kun Duan

ACKNOWLEDGMENTS

First and foremost, I would like to express my heartfelt gratitude to my thesis advisor Prof. David Crandall, who is more than an excellent advisor but also a lovely friend. I still remember the first day of taking his computer vision class, which immediately impressed me by his excellent teaching skills and deep understanding in every computer vision problem. He played the most important role during my five year journey at IU. What I learned from him is not just how to conduct research, how to write research paper, or how to present one's work, but more importantly, how to think.

Second, I want to thank Prof. Devi Parikh and Prof. Dhruv Batra. I was an intern at Toyota Technological Institute at Chicago (TTIC) working with Devi and Dhruv in the summer of 2011, and I view this as the start of my computer vision research career. Devi kindly guided me into the very cool local attribute project, which turns out to become my first CVPR paper, and gave me confidence to keep working hard. Dhruv gave me a lot of help in the human pose estimation project and the multimodal image modeling project. Without the help from Devi and Dhruv, I can hardly imagine if I could finish my PhD or not. I am very fortunate to have Devi in my advisory committee to give me continuous support during my PhD career.

I also want to thank my other committee members. Prof. Russell Lyons is a prestigious Math professor at IU. I was fortunate to take two Math graduate courses

(*Statistics and Stochastic Process*) with Prof. Lyons, and was strongly impressed by his deep, accurate, and clear knowledge in the field. These experiences are later proved to be quite helpful for me to understand computer vision problems using statistical approaches. I also want to thank Prof. Kris Hauser and Prof. David Leake, from whom I acquired many useful skills both inside and outside the classrooms.

I would like to thank Dr. Luca Marchesotti, who was my research mentor when I visited Xerox Research Center Europe (XRCE) in France. I got all kinds of help and support from Luca since the first day I arrive in France. His solid technical skills and Italian humor made my 4-month short stay at XRCE a really wonderful experience. I also want to thank my colleges at eBay Inc. and A9.com Inc., including my mentors Dr. Douglas Gray, Dr. Wei-Hong Chuang and Dr. Robinson Piramuthu, for their kind support during my internships.

And thanks to all my labmates Haipeng Zhang, Sven Bambach, Stefan Lee, Mohammed Korayem and Jingya Wang at IU Computer Vision Lab. I miss the days hanging out with them, and could not forget the day when I met Sven at “Neumarkt” in Cologne.

Finally, I want to give thanks to my parents. Even if we live on different halves of the earth, I can always feel your support in my mind every day. You are the best gift in my life.

Kun Duan

Conditional Random Field Models for Structured Visual Object Recognition

Image classification, object detection, and object description are classic problems in computer vision that have gained renewed attention due to the rapidly-growing collections of online imagery. Online images provide a free and nearly infinite source of data for training and testing vision algorithms, but the scale and heterogeneity of real-world photo collections require new scalable techniques that can handle substantial noise. In all of these recognition problems, structure is a recurring theme, albeit in different forms; geometric structure among the parts of an object, regularity of parts across different object instances, or patterns among images and image metadata of a collection. These may seem to be dramatically different types of (weak) information, but the conditional random field (CRF) is a powerful framework that can handle all of them. In this thesis, we propose and design novel CRF models to solve three distinct types of challenging object recognition problems, each incorporating different types of structured information while explicitly modeling uncertainty. We first study the monocular human pose estimation problem, introducing a fully-supervised multi-layer CRF to model the human body. A key challenge here is the huge label space, and we show how to achieve state-of-the art performance efficiently using dual decomposition. We then study fine-grained object recognition where the goal is to discriminate among

very similar categories (e.g. different bird species, or different vehicle models). We propose a CRF to automatically discover discriminative attributes of objects, using human interaction to infer attribute names. Finally, we study loosely-supervised clustering and classification of web images having noisy, sparse, multimodal metadata (GPS, timestamps, etc.), by generalizing the traditional K-means algorithm into a latent CRF. For all three problems, we conduct extensive experiments to compare our proposed approaches with the state-of-the-art on challenging benchmark datasets. We show that carefully designed and trained CRF models are able to achieve better recognition performance than competitive baselines. The major contribution of this thesis is to show how to design and train CRF models for different structured object recognition tasks at different levels of supervision.

CONTENTS

1	Introduction	1
1.1	Overview of Thesis	1
1.1.1	Three Motivating Problems	3
1.1.2	Learning Paradigms	4
1.1.3	Challenges	5
1.2	Conditional Random Fields	7
1.3	Three Motivating Problems	10
1.3.1	Conditional Random Field for Human Pose Estimation	11
1.3.2	Conditional Random Field for Discovering Localized Attributes	14
1.3.3	Conditional Random Field for Large-scale Multi-Modal Recog- nition	18
1.4	Summary	22
2	Conditional Random Fields	23
2.1	Overview	23
2.2	Model	25
2.3	CRF Parameter Estimation	28

2.4	CRF Inference	33
2.5	CRF with Latent Variables	37
2.6	Applications of CRFs to Computer Vision	39
2.7	Summary	40
3	Multi-layer Models for Human Pose Estimation	42
3.1	Related Work	42
3.1.1	Pictorial Structures	42
3.1.2	Hierarchical Models	44
3.1.3	Multi-scale Models	45
3.1.4	Mixture Models	46
3.1.5	Dual Decomposition	47
3.1.6	Other Relevant Work	47
3.1.7	Summary	48
3.2	Multi-layer Composite Models	48
3.2.1	Proposed Generalization	50
3.2.2	Dual Decomposition for Efficient Inference	51
3.2.3	Learning with Structural SVMs	54
3.3	Experiments	55
3.3.1	Datasets	55
3.3.2	Implementation	56
3.3.3	Results	58
3.4	Summary	63
4	Segmentation-based Local Attribute Discovery	67

4.1	Related Work	67
4.1.1	Visual Attribute Discovery	68
4.1.2	Modeling Local Attributes	69
4.1.3	Part Discovery for Object Models	70
4.1.4	Automatic Object Discovery	70
4.2	Modeling Localized Attributes via Latent CRF	71
4.2.1	Latent CRF Model Formulation	72
4.2.2	Training	75
4.2.3	Attribute Detection	77
4.2.4	Active Attribute Discovery	78
4.2.5	Identifying Semantic Attributes	78
4.3	Experiments	80
4.3.1	Attribute-based Image Classification	82
4.3.2	Image-to-text Generation	83
4.4	Summary	84
5	Detection-based Local Attribute Discovery	90
5.1	Related Work	91
5.2	Modeling Localized Attributes via	
	Multiple Instance SVM with Constraints	92
5.2.1	MI-SVMs for Attribute Discovery	93
5.2.2	MI-SVMs with Constraints	94
5.2.3	Recovering Viewpoint Angles	97
5.3	Experiments	98

5.3.1	Single Attributes	98
5.3.2	Multiple Attributes	100
5.4	Summary	103
6	Multimodal Image Modeling	105
6.1	Related Work	106
6.1.1	Multimodal Modeling	106
6.1.2	Constrained Clustering	107
6.2	Loosely Supervised Multimodal Learning	108
6.2.1	Constrained Clustering Framework	109
6.2.2	Learning Pairwise Potentials	112
6.3	Experiments	114
6.3.1	Applications and Datasets	114
6.3.2	Features	116
6.3.3	Results	117
6.4	Summary	120
7	Conclusion	125
7.1	Practices of Using CRF Models	127
7.2	Future Work	130
Bibliography		135
Curriculum Vitae		

LIST OF FIGURES

1.1	Illustration of image classification, object detection, and object description.	2
1.2	Different learning paradigms for the vehicle detection problem. Top: fully supervised, middle: weakly supervised, bottom: unsupervised.	6
1.3	Graphical structure of a CRF chain model for named-entity recognition task.	9
1.4	Illustration of multi-layer composite model for human pose estimation.	12
1.5	Sample local and semantically meaningful attributes automatically discovered by our approach.	16
1.6	Illustration of the latent Conditional Random Field for modeling multimodal images	20
2.1	A grid-structured conditional random field.	26
3.1	Dual-decomposition on our multi-layer composite pose model.	53
3.2	Primal objective and dual objective (left) and primal-dual gap (right) as a function of number of iterations during subgradient descent.	57
3.3	Part-based models used in our multi-layer composite model.	59
3.4	Illustration of two different measurements of PCP (Percentage of Correct Poses).	60

3.5 Qualitative results of the multi-layer composite model compared against the baseline approach.	64
4.1 Illustration of the latent CRF model for discovering local attributes given one active split.	74
4.2 Discovering local attributes: sample latent region evolution on an active split.	76
4.3 Illustration of the recommender system for prioritizing the order of discovered local attribute candidates based on the likelihood of being semantically meaningful.	86
4.4 Image classification performance using local attributes on four datasets.	87
4.5 Classification performance of the Proposed system with and without using the recommender.	87
4.6 Examples of automatic text generation.	88
4.7 Some local attributes discovered by our approach, along with the semantic attribute names provided by human users.	89
5.1 Visualization of different SVM models (standard SVM, standard MI-SVM, and constrained MI-SVM).	95
5.2 Relationship between region size and image classification performances.	99
5.3 Classification accuracy with different numbers of discovered attributes and different techniques for handling viewpoints on two benchmark datasets.	101
5.4 Examples of automatically generated local attributes for the Stanford cars dataset.	103

5.5 Examples of vehicle annotation results on new images.	104
6.1 Illustration of our constrained clustering framework compared with standard K -means algorithm.	122
6.2 Clustering performance as a function of number of images with different types of features.	123
6.3 Classification performance comparisons with loose supervision on training sets of increasing sizes.	123
6.4 Sample landmark clusters discovered automatically by our algorithm. .	124
6.5 Some activities discovered by our algorithm.	124

LIST OF TABLES

3.1	Pose estimation results (PCP) on Parse, UIUC Sport, and Leeds Sport datasets.	65
3.2	Evaluation results on the Parse dataset under different definitions of Percentage of Correct Poses (PCP) using different variants.	66
6.1	Purity and Inverse Purity scores on three Flickr datasets.	116
7.1	List of software tools for CRF training.	131
7.2	List of software tools for CRF inference.	132

CHAPTER 1

Introduction

1.1 Overview of Thesis

The goal of computer vision is to develop theories and methods to build computational systems that can perceive and process visual information. There are many examples of computer vision techniques applied in various applications, including industrial robots for controlling processes, autonomous vehicles, visual surveillance for detecting events, medical image analysis, and computer-human interactions, among many others. The most classical and fundamental task beneath these applications is *object recognition*.

Broadly speaking, object recognition tasks include *classifying* images, *detecting* object instances and *describing* semantic visual properties of object categories. In image classification, a collection of static images is given, then visual features are extracted and predictions are made for each image individually (e.g. bag of visual words approach) [22,37,38]; while these images are usually assumed to be independent and identically distributed (*i.i.d.*), there are relations among the images that could help to label them jointly [84,144]. In object detection, traditional approaches usually build a “template” for the object category of interest, and look for the closest match on a new image [10,51]. However, it is difficult for a simple model such as a template

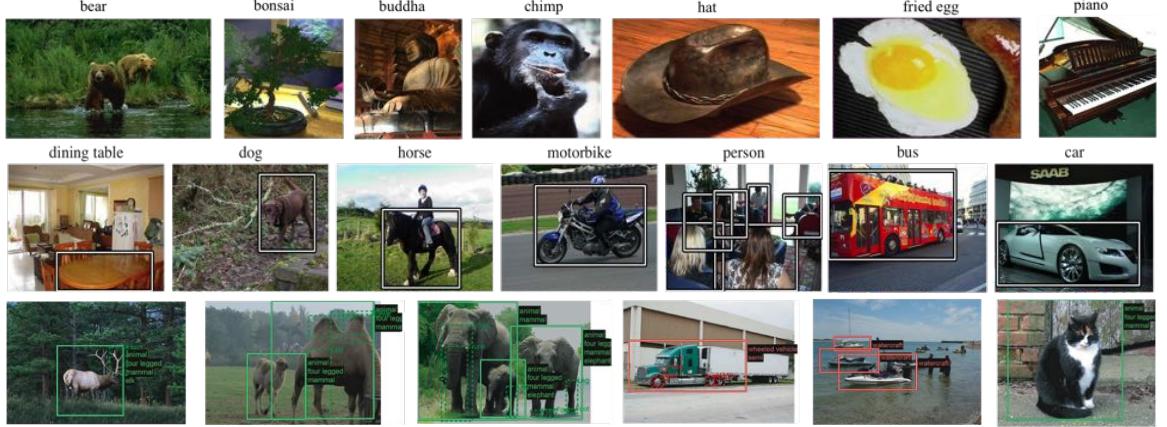


Figure 1.1: Illustration of the three key tasks in object recognition. Top row: Image classification and categorization on Caltech-256 dataset [47]. Middle row: Object detection for PASCAL image categories [35]. Bottom row: Object description using visual attributes [36].

to capture object appearance variations, since many objects are deformable in nature. For example, the human body can be considered as a combination of different parts (head, trunk, limbs, etc) with geometric constraints among them, which generate thousands of possible human poses [100, 138]. Even relatively rigid objects are still structured in some way or another; e.g. cars are decomposed into windows, wheels, head lights, etc. Thus, part-based methods for object detection become a natural solution based on these assumptions [20, 40].

Other recent work focuses on object description, where the intuition is that detecting certain object categories directly is difficult, but modeling their *visual attributes* is significantly easier. Visual attributes (e.g. hair and eye colors for face recognition, etc.) are discriminative and semantically meaningful, and describe the intermediate level features of object instances. These attributes are usually structured under geometry or co-occurrence constraints (e.g. co-occurrence of hair colors and eye colors

of human beings has been studied [122]), thus it is possible to utilize this structured information to model visual attributes [32, 136].

For all the above recognition problems, how to explore and correctly model the complex structures is key to improving state-of-art recognition performance. Our long term goal is to devise algorithms that can handle object recognition to both detect and describe object instances, and work for visual data at a large scale (Figure 1.1).

Here we show how to design novel conditional random fields (CRF) [70] to model structured information in these object recognition problems. CRFs are graphical models that provide a highly flexible way for modeling structures: many objects (e.g. human body, vehicles, animals, furnitures, etc) are decomposable into lower-level elements (defined as *parts*), and the geometry or kinematic constraints among these parts make them able to form an entire object model as a graph structure. Relations among images can also be captured by CRFs through modeling their pairwise similarities based on distance metrics calculated on different modality channels.

1.1.1 Three Motivating Problems

In this thesis, we will present our work on developing CRF models for three applications: 1) estimating human poses in static images, 2) discovering localized visual attributes and 3) organizing web images with multi-modal features. In all these problems, structure is a recurring theme, either at classification time (part-based techniques that encode geometry structure) or training time (weakly supervised learning of visual attributes). The structure can take on different forms, e.g. sometimes it is the structure among parts of the same object, sometimes it is the structure among images of a collection. These seem like dramatically different problems at first sight,

but conditional random fields are a framework that can handle all of them.

1.1.2 Learning Paradigms

Computer vision researchers used to hand-design algorithms based on intuition, but much better results have been achieved from learning models automatically using machine learning techniques. Then instead of having to hand-craft algorithms and heuristics, one instead needs to collect representative training data. There is typically substantial effort still involved in doing this, so different supervision levels have emerged depending on how much information is available (and how much human labor and money people are willing to invest to collect it).

Fully supervised methods rely on manually annotated training exemplars for each category of interest. For example, these annotations can be bounding boxes over object instances and their key points (for an object detection task), or a category name for each training image (for an image classification task). In this case, discriminative properties of certain object categories can be learned with a high recognition performance on the test data. Example applications include face recognition, image retrieval, etc. However fully annotated training data are expensive to obtain in practice, and they might also suffer from the curse of dataset bias [60].

Weakly supervised methods try to learn with less training information, and introduce *latent variables*. Unlike *observed variables* whose training labels can be collected in a less expensive way, latent variables are not observed in the training process, and try to model more information given the same amount of supervisions on the training exemplars. However, these methods require effective inference over the latent label space; thus iterative methods are usually taken to estimate the model parameters,

where in each iteration the confidence of latent labels for each training exemplar is updated. For example, part-based object detectors can be trained if there are no bounding box annotations but the class label of each training image is given [21]. The training process can be slower than the fully supervised case, but it keeps comparable performance against supervised methods, and requires much less annotations.

In the case of large scale learning, *e.g.* looking at images at a web scale, training labels are often difficult to obtain. Thus unsupervised or loosely-supervised (a special case of weakly-supervised learning, *i.e.* very small scale training data, with noise and missing labels) methods have to be used in order to learn the statistical model for object recognition. Interesting applications include large scale image classification, large scale image tag estimation, automatic photo album organization, etc. Figure 1.2 compares the three different learning paradigms.

1.1.3 Challenges

While CRFs are a flexible and convenient framework (which we outline in the next section and discuss in detail in Chapter 2), posing object recognition problems in terms of CRFs is non-trivial work, both from theoretical and implementation points of view. In our proposed framework, different graphical models are designed so that the learning and inference complexity can be scalable for the application of interest, while obtaining state-of-art recognition performance. We summarize the key challenges that we want to address in current object recognition tasks:

- How to design conditional random field models to capture and utilize complex structures in object recognition problems.

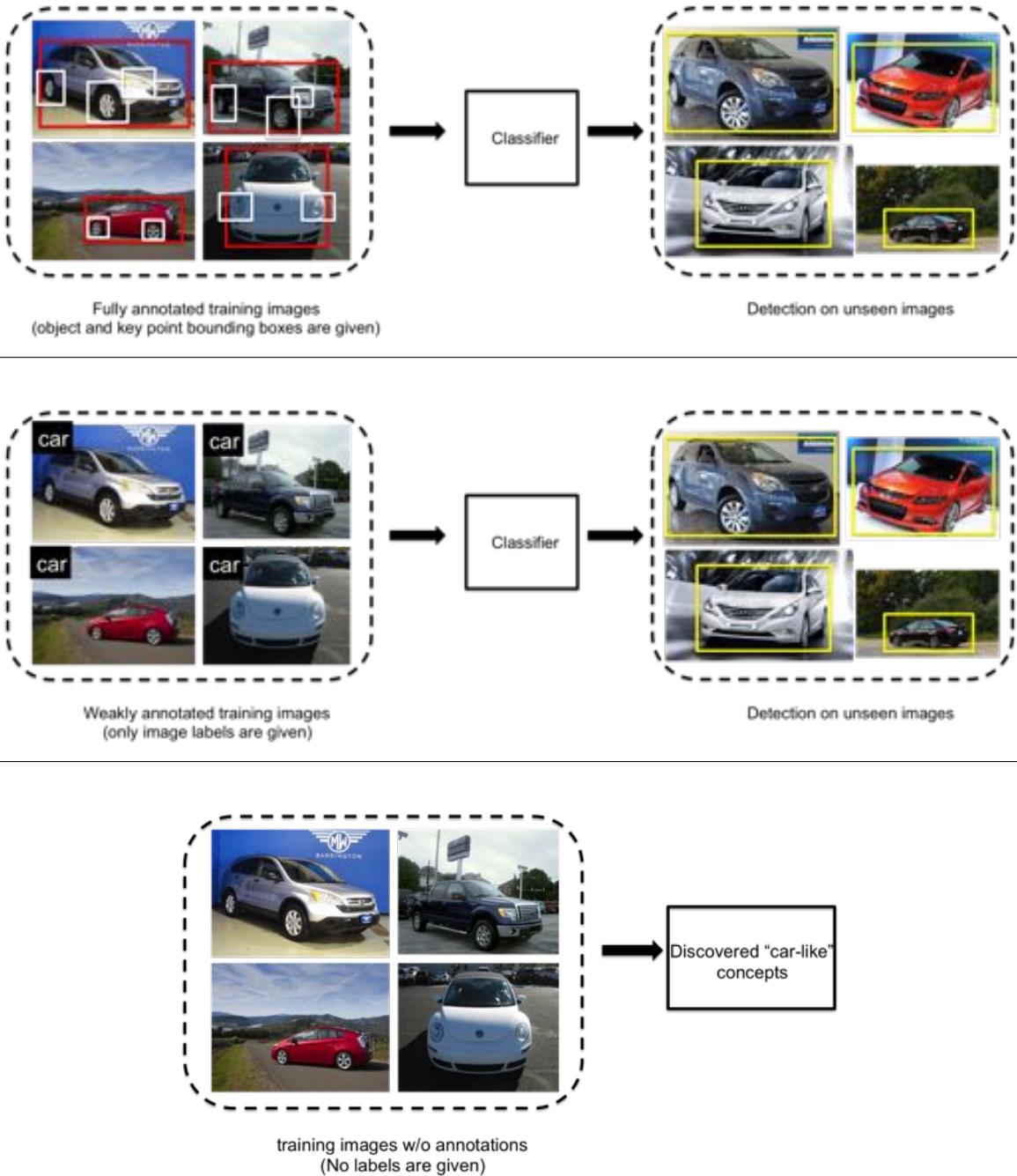


Figure 1.2: Different learning paradigms for the vehicle detection problem. Top: fully supervised, middle: weakly supervised, bottom: unsupervised.

- How to model latent information and use it to improve object recognition performance.
- How to model structured information for images at a large scale.
- How to train conditional random field models under different levels of supervision.

1.2 Conditional Random Fields

A Conditional random field (CRF) [70] is a form of discriminative undirected probabilistic graphical model that encodes relationships between different variables. Some of these variables can be directly observed whereas others cannot be, and the structure encoded in the CRF helps to estimate the unobserved ones given the observed ones. We briefly summarize CRFs in this section, and then explain them in much more detail in Chapter 2.

Let $G = (V, E)$ be an undirected graph, where V is the set of nodes in the graph, and E is the set of edges. Let $n = |V|$ denote the number of nodes in the graph. Define X as the set of input random variables, $Y = \{y_v\}_{v \in V}$ as the set of output random variables, where $V = X \cup Y$ and each y_v ($v \in V$) takes a value from a range of possible discrete labels. In a conditional random field, we assume each random variable y_v obeys the *Markov property* when conditioned on X , such that the conditional probability distribution of y_v given its adjacent nodes is independent of the rest of the nodes in the graph. That is, if G is such a graphical model that

$$P(y_v|X, y_w, w \neq v) = p(y_v|X, y_w, w \in N(v))$$

where $N(v)$ is the set of adjacent nodes of v , then (Y, X) is a conditional random field (CRF). In object recognition problems, the observations X are often the image data themselves, or extracted visual features; and Y correspond to the outputs of the vision system, e.g. possible locations of pedestrians in the image to be detected, or possible category labels of the image to be classified.

The structure of graph G may be a chain, a tree, a grid, or any arbitrary structure. Two nodes are connected to each other if they are constrained by the assumptions of the specific problem to be solved.

Given a graphical model, the most fundamental (yet highly non-trivial) task is to compute the marginal distribution of one or a few node variables in Y . This task is usually referred to as *inference*. In other words, the inference task for a CRF is to find a best label for each node, such that it maximizes the conditional probability $P(Y|X)$. A second fundamental task is *training* a CRF model, which requires a collection of annotated training exemplars whose output labels Y are observed in the training process. The structure of a CRF model is usually assumed to have a parametric form. Then an optimization problem (e.g. maximize the likelihood of a conditional random field on all training instances) is solved to obtain the parameters.

A brief example may be helpful to illustrate the use of CRFs in practice. *Named entity recognition* [86] in natural language processing applications which is the task of recognizing and classifying all proper nouns into pre-defined classes such as persons, locations, organizations and others. For example, in “Bill bought a cat in Texas,” we would like to assign “person” to “Bill,” “animal” to “cat,” and “location” to “Texas.” Stop words like “a” or “in” are removed in the preprocessing step. In this case, a chain-like CRF can be used since the words are sequential, and it is reasonable to

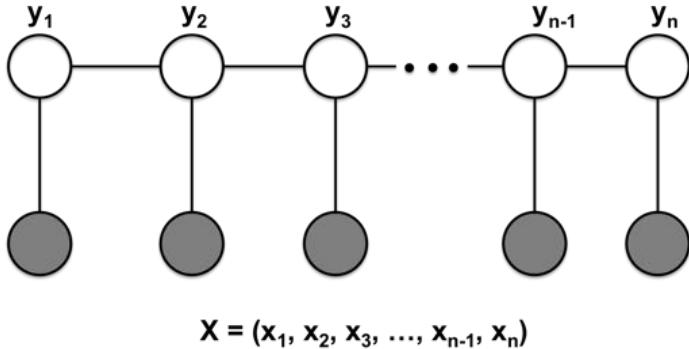


Figure 1.3: Graphical structure of a CRF chain model for named-entity recognition task.

assume that words are related to their immediate neighbors. Word neighbors are constrained by grammatical rules, punctuation context or co-occurrence relations. The probability of a word being assigned with a specific class label is defined using word level features (*e.g.* language or genre specific cues) and dictionary look-ups. A graphical representation of such a CRF is shown in Figure 1.3, where X corresponds to observed words, and Y corresponds to a collection of name variables representing the semantic labels assigned to each word.

For more sophisticated problems, a CRF can be defined as a tree graph or even a loopy graph, instead of a chain model; it can also be defined using high-order cliques, which are subgraphs with more than two nodes.

The inference on a conditional random field is *exact* if it leads to a global optimal solution, otherwise the inference is *approximate*. For chain-like or tree-like CRF models where there are no loops, exact inference can be performed in polynomial time. We will describe the details of different inference algorithms in the thesis. We also discuss the details of learning conditional random field using discriminative

methods.

CRFs can also contain *latent variables*. Latent variables are not observed in the CRF training process. They are used to model useful information (which are not annotated in the training data) in the problem structure. For example, in a vehicle detection task, a commonly provided annotation is a bounding box for the car in each training image. We can treat this as a fully supervised task, i.e. training a vehicle template using bounding box annotations. However, the problem structure is better defined if we assume a vehicle is composed of a set of rigid parts. The state-of-art *deformable part-based models* [39] consider different components of a vehicle as “parts” (wheels, car front, head lights, windows, car back, etc). Their approach learns the appearance for each part of the object and models the geometric constraints among these parts by treating them as latent variables, given only bounding box annotations in the training process. These graphical models with latent variables are called *latent conditional random fields* (latent CRFs). We will describe the details of latent CRF in Section 2.5.

1.3 Three Motivating Problems

In the subsequent sections, we showcase how to use CRF models to solve different object recognition problems. First, a multi-layer hierarchical CRF model is developed for modeling human pose structures, and a fully supervised approach is applied to detect the human body and estimate its pose on a new image. Second, a latent CRF model is formulated for modeling *localized attributes*, where these visual attributes are modeled as latent variables; then an iterative process for discovering local attributes is proposed. Finally, we extend the traditional K -means algorithm using a

latent CRF model, then describe how to use a loosely supervised approach to learn the multi-modal concepts at a large scale, and also demonstrate our approach on an unsupervised clustering task. Our choice of applications also correspond to different learning paradigms: supervised learning for pose estimation, weakly supervised learning for local attribute discovery, and loosely supervised / unsupervised learning for multimodal modeling.

1.3.1 Conditional Random Field for Human Pose

Estimation

Detecting humans and identifying body pose are key problems in understanding natural images, since people are the focus of many (if not most) consumer photographs. Pose recognition is a challenging problem due not only to the usual complications of object recognition—cluttered backgrounds, scale changes, illumination variations, etc.—but also to the highly flexible nature of the human body. To deal with this flexibility, deformable part-based models [39, 40] have emerged as a dominant approach in recognizing people and other articulated objects [19, 73, 124, 137, 138, 147]. These part-based models decompose an object into a set of parts, each of which is represented with a local appearance model, and a geometric model that constrains relative configurations of the parts. Recognition is then cast as an inference problem on a conditional random field (CRF) model, in which the parts are represented by vertices and the constraints between parts are represented as edges.

Many of these part-based models assume a tree structure [39, 40, 138], capturing the kinematic constraints between parts of the body—*e.g.* that the lower arm is connected to the upper arm, which is connected to the torso, etc. Such tree structures

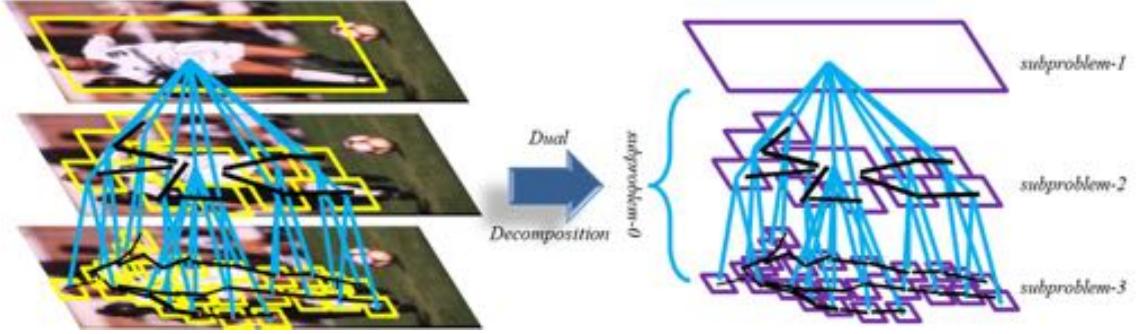


Figure 1.4: Illustration of our multi-layer composite part-based model.

allow exact inference to be performed efficiently on the underlying CRF model via dynamic programming. However, the tree structure makes conditional independence assumptions between unconnected parts, which can lead to pose estimates that obey kinematic constraints but are still not sensible; for example, a single image region might be recognized as two different body parts, or a pose might be estimated that violates constraints of gravity and human balance.

A variety of approaches have been proposed for dealing with these problems, including introducing a few cycles into a tree-structured graphical model [137, 147], adding common factor variables [73], or using a fully-connected graphical model [124] to capture more spatial constraints among the parts. Although effective, these approaches introduce cycles into the graphical model which generally makes exact inference intractable, leading to approximate solutions and increased computational complexity. How to model richer spatial constraints that still permit efficient inference is an important open question.

In order to address these problems from a different perspective, we propose a novel conditional random field model. Instead of adding cycles to the original model, we build a multi-level model consisting of multiple tree-structured models with different

resolution scales and numbers of parts, allowing different degrees of structural flexibility at different levels, and connect these models through hierarchical decomposition links between body parts in adjacent levels. A visualization of our model with three layers is shown in Figure 1.4 (left). Even though the composite model is a loopy graph, it can be naturally decomposed into tree-structured sub-problems within each level and the cross-model constraint sub-problem across levels (which is also tree-structured as shown in Figure 1.4 (right)). These tree-structured sub-problems are amenable to exact inference, and thus joint inference on the composite model can be performed via dual-decomposition [12].

The main idea behind dual-decomposition is to decompose the original optimization problem into smaller and easily solvable subproblems, then calculate a solution for the original problem by combining the solutions from these subproblems. At detection time, dual-decomposition methods modify a vector of dual variables [67] iteratively, and perform individual inference over the decomposable subgraphs. In our problem, the inference for each sub-task is separable from others, and can be done in polynomial time through dynamic programming; thus we can update the dual variables in parallel. This makes the dual-decomposition step very efficient in practice.

We train these models jointly, and show that the composite models outperform state-of-the-art techniques on two challenging pose recognition datasets. We believe these composite models provide a principled way to trade off the competing goals of model expressiveness and ease of inference, by “stitching” together multiple tree-structured models into a richer composite model while keeping the complexity of joint inference in check. Our preliminary work has been published [29].

1.3.2 Conditional Random Field for Discovering Localized Attributes

Human pose is so flexible that strong supervision at the body key points level has to be used to achieve reasonable performance. Also, it is not very expensive to label key points of a human body (e.g. head, shoulder, arm and leg joints, hands, ankles, etc) since these locations are well-known to ordinary people; just a few clicks will complete the annotation task for an image. However, there exist many other problems where obtaining detailed annotations is expensive. Here we consider modeling *visual attributes*, which are intermediate-level features that are both machine-detectable and semantically meaningful. For many objects, the definition of visual attributes usually requires strong domain knowledge, thus the labor cost for annotating these images is much higher. We propose to use latent conditional random field model to discover visual attribute candidates, and combine a *human-in-the-loop* process that intelligently interacts with human subjects through iterations to select candidates that are both machine-detectable and semantically meaningful.

Most image classification and object recognition approaches learn statistical models of low-level visual features like SIFT [78] and HOG [23]. While these approaches give state-of-the-art results in many settings, such low-level features and statistical classification models are meaningless to humans, thus limiting the ability of humans to understand object models or to easily contribute domain knowledge to recognition systems. Recent work has introduced visual attributes (e.g. [11, 17, 36, 42, 52, 69, 141]) that help to expose the details of an object model in a way that is accessible to humans: in bird species recognition, for example, they can explicitly model that

a cardinal has a “red-orange beak,” “red body,” “sharp crown,” “black face,” etc. Attributes are particularly attractive for fine-grained domains like animal species where the categories are closely related, so that a common attribute vocabulary exists across categories. Attributes also enable innovative applications like zero-shot learning [71, 91] and image-to-text generation [36, 91].

So where do these attributes come from? Most existing work uses hand-generated sets of attributes (*e.g.* [17, 69]), but creating these vocabularies is time-consuming and often requires a domain expert (*e.g.* an ornithologist familiar with the salient parts of a bird). Moreover, while these attributes are guaranteed to be human-understandable (which suffices for human-in-the-loop classification applications [17]), they may not be machine-detectable and hence may not work well in automatic systems. Some recent work has discovered image-level attributes (*e.g.* “outdoors” or “urban”) automatically [90], but such global attributes are of limited use for fine-grained object classification in which subtle differences between object appearances are important.

Discovering *local* attributes (like those illustrated in Figure 1.5) is significantly harder because a local attribute might correspond to features at different unknown positions and scales across images. Automatic techniques to do this have generally either found attributes that are discriminative or that are meaningful to humans, but not both. Finding discriminative local regions (*e.g.* [141]) works well for attaining good image classification performance, but the regions may not be semantically meaningful and thus not useful for applications like zero-shot learning and automatic image description. On the other hand, mining text can produce attribute vocabularies that are meaningful (*e.g.* [11]) but not necessarily complete, discriminative, or

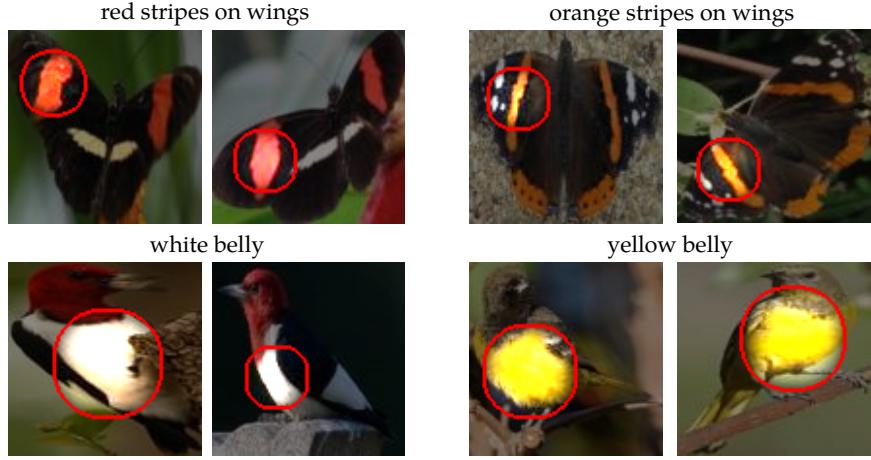


Figure 1.5: Sample local and semantically meaningful attributes automatically discovered by our approach. The names of the attributes are provided by the user-in-the-loop.

detectable.

In this thesis, we propose to discover local attributes for biological objects (e.g. birds, butterflies) and man-made objects (e.g. vehicles) for recognition tasks. These two kinds of objects have significant differences that lead to different learning paradigms. Vehicles are much more rigid than animals, and vehicle photos are often taken from relatively fixed viewpoint angles. Thus for vehicle recognition, such viewpoint information is very useful to build correspondence between image regions from two photos taken under different viewpoint angles. We use a *detection-based* method to generate region candidates for attribute discovery on vehicle images. The detection results of the object parts give us a probability distribution of the locations of possible region candidates. We also learn a viewpoint-dependent multiple instance SVMs where the attributes discovered on different images are constrained by geometric relations.

Biological objects are highly deformable, and it is difficult to train a part-based

object model for such categories. Thus, we use a *segmentation-based* method to generate region candidates. We use a hierarchical segmentation approach, which generates a rich set of region candidates, and preserves their geometric properties as much as possible. We design an interactive system that discovers discriminative local attributes that are both machine-detectable and human-understandable from an image dataset annotated with fine-grained category labels and object bounding boxes. At each iteration in the discovery process, we identify two categories that are most confusable given the attributes that have been discovered so far; we call these two categories an *active split*. We use a latent CRF model to automatically discover candidate local attributes that separate these two classes. For these candidates, we use a recommender system to identify those that are likely to be semantically meaningful to a human, and then present them to a human user to collect attribute names. Candidates for which the user can give a name are added to the pool of attributes, while unnamed ones are ignored. In either case, the recommender system’s model of semantic meaningfulness is updated using the user’s response. Once the discovery process has built a vocabulary of local attributes, these attributes are detected in new images and used for classification.

To the best of our knowledge, ours is the first system to discover vocabularies of local attributes that are both machine-detectable and human-understandable, and that yield good discriminative power on fine-grained recognition tasks (i.e. classification, image-to-text annotation, and object detection). We published preliminary work on discovering localized attributes in [31, 32]. We demonstrate our approach through systematic experiments on two challenging biological datasets: Caltech-UCSD Birds-200-2011 [130] and Leeds Butterflies [134], and two vehicle image datasets: Stanford

car dataset [115] and INRIA vehicle dataset [68]. We find on these datasets that our discovered local attributes outperform those generated by human experts and by other strong baselines, on fine-grained image classification tasks.

1.3.3 Conditional Random Field for Large-scale Multi-Modal Recognition

The above two applications using conditional random fields are based on small scale, hand-collected image datasets. However in the real world, online photo-sharing has become very popular, which generates huge collections of images on sites like Flickr, Picasa, and Instagram. As these datasets grow ever larger, a key challenge is how to organize them to allow for efficient navigation and browsing. For instance, we may want to discover the structure of photo collections by clustering images into coherent groups with similar objects, scenes, events, etc. in an automatic or semi-automatic way.

While image clustering has been studied extensively (e.g. [9, 140, 146] among many others), photo collections on modern photo-sharing sites introduce new opportunities and challenges. In addition to the images themselves, photos on these sites often include rich metadata that provide additional cues to the semantic content of the images, including text tags, timestamps, camera EXIF data, GPS coordinates, captions, and comments from other users. This metadata allows us to find connections between photos that are not obviously similar: a photo of the crowd at a candidate’s political rally is clearly related to a photo of his or her campaign logo, but these photos exhibit almost no visual similarity. In such cases, similarities in the non-visual metadata may help: image tags and captions often contain useful keywords related to the content,

activities, and context of the scene, while GPS coordinates and timestamps can be used to find photos taken nearby in space and time.

Of course, metadata alone is not enough: two random photos tagged *canon d50* are probably not related, while photos tagged with identical GPS and timestamps may be unrelated if taken on different floors of a large building. Moreover, metadata is typically not well constrained, and thus often missing, incomplete, ambiguous, or erroneous. For instance, some photos include detailed text tags, while others are tagged with unhelpful or noisy labels or are not tagged at all; even the most fastidious of photographers cannot list *all* possible tags that are relevant to an image. GPS coordinates are only collected by select devices like smartphones and are often hidden due to privacy concerns, so geo-tags typically appear on a small subset of images.

In this thesis, we present an approach for clustering large datasets with multi-modal, incomplete, and noisy features, and apply it to clustering social photo collections. Our method can be used in a fully unsupervised setting, or can use labeled training data if available, in contrast to supervised methods like SVMs that require significantly more training data. Our method is designed for cases where obtaining large labeled datasets is not feasible, but annotating a small amount of training data is still feasible. For example, in a large scale photo collection with millions of images, if the categories of interest are known in advance, one can manually annotate a few hundred instances, and apply our approach using this loosely-supervised information for organizing the rest.

As in traditional clustering (like K -means), we wish to assign each instance to a cluster, but the cluster identities (*e.g.* centroids) are themselves unknown and must

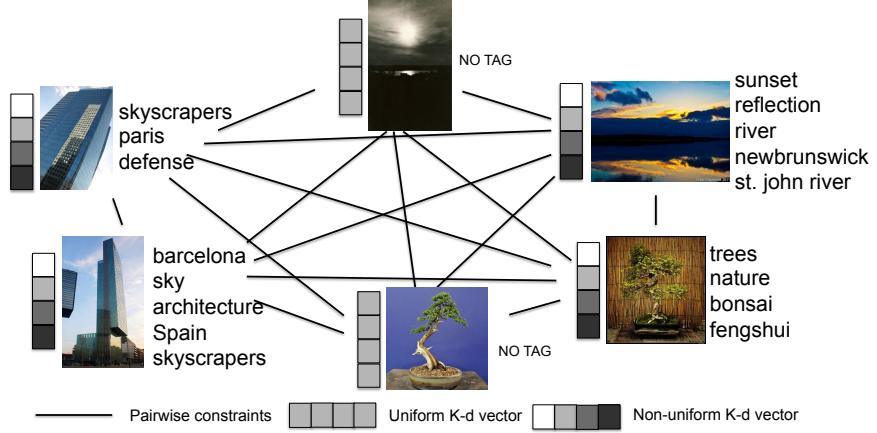


Figure 1.6: Illustration of the latent Conditional Random Field model for two feature types. The primary features here are text tags, which are encoded as unary potentials, while visual features are the constraints (encoded in the pairwise potentials). Missing text tags yield uniform unary potentials.

also be inferred. We pose this problem using a Latent Conditional Random Field, in which each node in the graph corresponds to an image, and our goal is to mark each node with a cluster label. We pick one type of feature to be the *primary feature* and use it to define the CRF’s unary potentials, which are functions of the distances from an image’s primary feature to each latent cluster center. The other feature channels are considered to be *constraints* and appear as pairwise potentials in the CRF. These constraints tie together images with similar secondary features, encouraging them to be assigned to the same cluster. Incomplete, noisy, and heterogeneous features can thus be naturally incorporated into this model through these soft constraints. To perform clustering, we alternately solve for cluster assignments and cluster centers in a manner similar to K -means and EM, except that the E-step is much more involved, requiring inference on a CRF.

A challenge in clustering with noisy, multi-modal features is how to define sensible distance metrics for the heterogeneous feature types, and how to weight them relative to one another. We address this problem by learning the distance and potential functions on a small amount of labeled training data we obtain from each category. In particular, we use Information Theoretic Metric Learning (ITML) [25] to learn the parameters of the distance metrics for constraint features, and use structural SVMs with the same training exemplars to learn the potential functions of the CRF. Our approach can still work for unsupervised cases, when obtaining labeled images is not feasible or no prior knowledge about the categories of interest is known; in this case, we can use a standard metric like L_2 distance, or use a distance functions trained on a different but similar dataset.

Finally, we evaluate our approach on three datasets from Flickr, with labeled ground truth and different types of features including visual, text, and GPS tags, and compare against strong baseline methods. We also test on a larger unlabeled image dataset to show how we can organize photo collections around coherent events and activities in a completely unsupervised manner.

In summary, we propose a general loosely supervised clustering framework for multi-modal data with missing features. We also apply metric learning and formulate a structural SVM problem for learning the structure of the latent CRF. In addition, we show that the approach can be used for unsupervised clustering on large-scale online image datasets. A preliminary version of this multimodal image modeling work has been published in [30].

1.4 Summary

In this thesis, we will study the important role of structured information in object recognition tasks. We choose three relevant applications (human pose estimation, local attribute discovery, and multi-modal data learning at web scale), and showcase how to develop different types of conditional random fields to model these structures. These applications are representative of many structured object recognition problems, and require different levels of supervisions (fully supervised, weakly supervised, or unsupervised). We perform systematic evaluations of the proposed coherent CRF framework, and compare against strong baseline methods on challenging datasets.

Thesis Statement. CRFs provide a unifying framework to help understanding and utilizing the implicit structure in a wide range of recognition problems, and yield state of the art performance on benchmark datasets.

CHAPTER 2

Conditional Random Fields

2.1 Overview

A graphical model [18, 54, 59] is a probabilistic framework where a graph is used to represent the dependency structures among different variables. Combining the use of probability theory and graph theory, graphical models become a principled and effective approach for modeling uncertainty. They allow us to represent a distribution over a collection of random variables, using the product of *potential functions* that are defined on small subsets of random variables. A graphical model can be either a *directed graph* (*i.e.* *Bayesian Network*) if the edges in the graph have directions, or an *undirected graph* (*i.e.* *Markov random field*) if no edges have directions.

Bayesian networks or *belief networks* must be both directed and acyclic, and model causal relations. Applications of Bayesian networks include *e.g.* medical diagnosis systems, criminal risk analysis, insurance policy modeling, *etc.* Markov random fields (MRFs), on the other hand, are a set of random variables having the *Markov property* described by an *undirected* graph. Unlike a Bayesian network, an MRF can have cycles in the graph, and thus can be used to model more complex relational structures.

A Markov random field models the joint probability distribution $p(\mathbf{x}, \mathbf{y})$, where \mathbf{y}

denotes the unobserved random variables whose value we want to predict, and the observed random variables \mathbf{x} whose values are known in the graphical model. However, modeling the joint distribution requires the calculation of the prior term $p(\mathbf{x})$, which contains complex dependencies when rich features are used in the relational structure. In this case, modeling these dependencies incurs intractable computations, but ignoring or simplifying them (*e.g.* assuming uniform or Gaussian distribution) can lead to inaccurate explanation of the model structure.

One solution to this problem is to model the conditional distribution $p(\mathbf{y}|\mathbf{x})$ directly, which is sufficient in many situations. *Conditional random fields* (CRFs) were proposed [70] as a variant of Markov random fields to solve this problem. Markov random field and conditional random field are examples of *generative models* and *discriminative models* in a probabilistic framework, respectively. Generative models make assumptions on the prior probability distribution $p(\mathbf{x})$ of the training data, and need to explicitly specify the joint probability distribution $p(\mathbf{y}, \mathbf{x})$. The training instances are assumed to be *sampled* from the joint probability distribution. Thus generative models describe a full probabilistic model of all variables, and are able to “hallucinate” the values of any variable in the model.

Generative approaches work best when we have prior knowledge of the given problem, and are able to handle small scale training data since they rely strongly on the prior probability distribution. When we have sufficient training data (so that modeling the prior term is not necessary), or when we do not have any prior knowledge, we can use *discriminative models*. In a probabilistic framework, discriminative models directly model the conditional probability $p(\mathbf{y}|\mathbf{x})$. They often have much better performance than generative models in *classification* or *regression* tasks since a full

joint probabilistic distribution is not necessary in these cases.

In this part of the thesis, we will review the definitions of conditional random field models (Section 2.2), and discuss the common learning and inference algorithms (Section 2.3 and Section 2.4). We will also describe a variant of conditional random fields by adding latent, unobserved information in the training data (Section 2.5).

2.2 Model

Let $G = (V, E)$ be an undirected graph, where V is the set of nodes in the graph, and E is the set of edges each of which connects two nodes. Define X as the set of *input variables* whose values are observed in the graph, and Y as the set of *output variables* that are *not* observed whose values needs to be estimated. We have $V = X \cup Y$, and use \mathbf{x} , \mathbf{y} to denote the values assigned to variables X , Y respectively.

To make the context clear, we only consider the case when each variable in V takes a value from a range of possible discrete labels, although they can be either continuous or discrete in a more general case.

Probabilistic interpretation. Given the set of all *maximal cliques* (*i.e.* maximal subgraphs of G that are fully-connected) \mathcal{A} of G , the conditional probability distribution of a CRF can be written as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{A \in \mathcal{A}} \Psi_A(\mathbf{x}_A, \mathbf{y}_A) \quad (2.1)$$

where $\Psi_A : A \rightarrow \mathcal{R}^+$ is called a *potential function* or *compatibility function*, and A is a maximal clique in G . Z is a normalization factor (also called *partition function*)

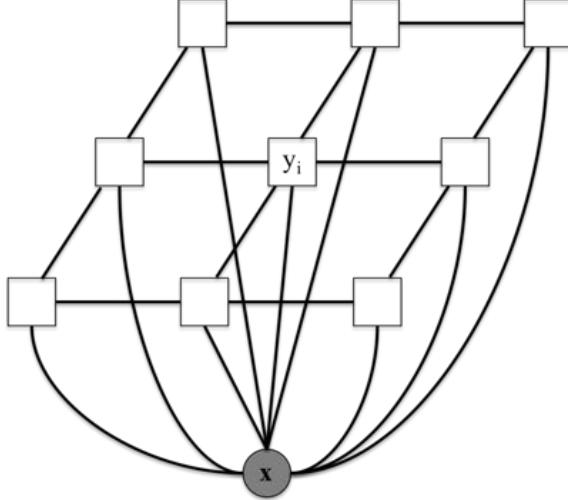


Figure 2.1: A grid-structured conditional random field. \mathbf{x} represents the (observed) input variables, and \mathbf{y} represents the (unobserved) output variables.

depending on the observed values of input variables \mathbf{x} , and is defined as:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{A \in \mathcal{A}} \Psi_A(\mathbf{x}_A, \mathbf{y}_A)$$

We also assume the conditional distribution over the graph G is an *exponential family* [3], thus we require each potential function Ψ_A to have the form

$$\Psi_A = \exp \left\{ \sum_k w_{A_k} f_{A_k}(\mathbf{x}_A, \mathbf{y}_A) \right\} \quad (2.2)$$

where w_A is a real-valued parameter vector, and $\{f_{A_k}\}$ is a set of *feature functions* defined on the potential Ψ_A .

Energy function interpretation. The above *probabilistic interpretation* of conditional random field models has a natural connection with another form of interpretation, *i.e.* energy function interpretation.

Mathematically, one can take the negative *logarithm* of the left hand side and right hand side of equation 2.1, and the problem of maximizing the conditional probability

becomes an energy minimization problem. In practice, we usually model structures using *pairwise* constraints, since inference is easier in this case and the model parameters are easy to learn. For example, in computer vision problems we often see CRFs with maximal cliques of size 2. In this case, we can write down the energy function for a grid structured CRF model (Figure 2.1) as

$$E(\mathbf{y}|\mathbf{x}) = \sum_i D(y_i|\mathbf{x}) + \sum_{i,j} V(y_i, y_j|\mathbf{x}) \quad (2.3)$$

where we call D the *unary potential*, and V the *pairwise potential*. Occasionally we also use high-order cliques (*i.e.* number of nodes involved in a potential function ≥ 3), and there are special types of high-order clique potentials (*e.g.* P^n *Potts Model* [62], *Cardinality Potentials* [119]) that are useful in a few applications.

Probabilistic models need to be normalized properly, and in many cases require evaluating intractable integrals over the space of all possible variable configurations. Energy functions have no such normalization requirement, thus providing more flexibility in designing the architecture of the underlying graphical model.

Applications in computer vision. Many computer vision tasks can be naturally described by conditional random fields. We now summarize a few basic low level computer vision problems with structures that can be captured by using conditional random field models.

- *Image Denoising.* Given an image I with (possibly noisy) observed pixel values, find I' so each pixel in I' has the corrected value [7]. We consider a smoothing term $V(y_i, y_j)$ that encourage neighboring pixels i and j to have the same pixel value, and a data term $D(y_i|x_i)$ that places a penalty if y_i is different with the observation x_i .

- *Image Segmentation.* Given an image I , group the pixels so that each pixel is assigned a label indicating which region or object it belongs to [89], *e.g.* labeling each pixel with 0 (background) or 1 (foreground). We define the data term $D(y_i|x_i)$ that encodes an appearance model (*e.g.* a GMM model trained on color features). We also place a cost $V(y_i, y_j)$ for neighboring pixels i and j if *a*) their labels are different and *b*) the difference between their pixel values is less than some threshold.
- *Stereo Matching.* Given a image pair of a scene, find the disparity value of each pixel (*i.e.* the distance that the pixel "moved" across images, which is proportional to the depth of the scene point imaged by the pixel) [116]. These disparities are assumed to be smooth, thus we add pairwise constraints for neighboring pixels, and place a cost $V(y_i, y_j)$ (either constant or linear with the disparity difference) if their disparity values are different. The data term $D(y_i|x_i)$ is defined using the matching cost of a small neighborhood around pixel i .

In the above examples, the structures are image grids, and the models are intuitive to understand. In next chapters, we will describe more difficult scenarios where conditional random field models are used to model more complex structures.

2.3 CRF Parameter Estimation

In order for CRFs to be applied to problems, one needs to design the potential functions. Typically this is done by assuming they have some parametric form and then we need to estimate the parameters. We discuss three parameter estimation

approaches for conditional random field models, *i.e.* *maximum likelihood learning*, *maximum a posterior learning* and *structured SVM learning*. We briefly review the first two methods and then focus on the structured support vector machine, as it plays an important role in later chapters.

Maximum likelihood learning. Given *i.i.d.* training data $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}$, maximum likelihood (ML) learning involves maximizing the following objective function:

$$\ell(\mathbf{w}) = \prod_{n=1}^N p(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}; \mathbf{w})$$

where N is the number of instances in the training data. For a CRF in Equation 2.1, we can re-write the above objective function using *conditional log likelihood* as:

$$\begin{aligned} \ell(\mathbf{w}) &= \sum_{n=1}^N \log(p(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}; \mathbf{w})) \\ &= \sum_{n=1}^N \left(\sum_{A \in \mathcal{A}} \log \Psi_A(\mathbf{x}_A^{(n)}, \mathbf{y}_A^{(n)}) - \log Z(\mathbf{x}^{(n)}; \mathbf{w}) \right), \end{aligned}$$

And then from Equation 2.2,

$$\ell(\mathbf{w}) = \sum_{n=1}^N \left(\sum_{A \in \mathcal{A}} \sum_k w_{A_k} f_{A_k}(\mathbf{x}_A^{(n)}, \mathbf{y}_A^{(n)}) - \log Z(\mathbf{x}^{(n)}; \mathbf{w}) \right)$$

In general, $\ell(\mathbf{w})$ cannot be solved in closed form. But we can compute the partial derivatives of $\ell(\mathbf{w})$ and apply numerical optimization to maximize $\ell(\mathbf{w})$.

Maximum a posterior learning. Maximum a posterior (MAP) learning simply adds a regularization term (*e.g.* ℓ_2 norm) on w in the above ML learning objective function. The posterior here comes from Bayesian view and the regularization term is like the log prior term:

$$\ell(w) = \|w\|^2 + C \cdot \sum_{n=1}^N \left(\sum_{A \in \mathcal{A}} \sum_k w_{A_k} f_{A_k}(\mathbf{x}_A^{(n)}, \mathbf{y}_A^{(n)}) - \log Z(\mathbf{x}^{(n)}; \mathbf{w}) \right)$$

where C is a parameter (usually tuned on a validation set) that controls how much regularization is enforced in the optimization objective. MAP learning can also be interpreted as *regularized empirical risk minimization*:

$$\ell(\mathbf{w}) = \|\mathbf{w}\|^2 + C \cdot \sum_{n=1}^N \Delta(\hat{\mathbf{y}}^{(n)}, \mathbf{y}^{(n)})$$

where we define $\hat{\mathbf{y}}^{(n)} = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{y} | \mathbf{x}^{(n)}; \mathbf{w})$, and define the *loss function* as:

$$\Delta(\hat{\mathbf{y}}^{(n)}, \mathbf{y}^{(n)}) = \sum_{A \in \mathcal{A}} \sum_k w_{A_k} f_{A_k}(\mathbf{x}_A^{(n)}, \mathbf{y}_A^{(n)}) - \log Z(\mathbf{x}^{(n)}; \mathbf{w}) \quad (2.4)$$

Note that ML and MAP learning both require computing the partition function $Z(\mathbf{x})$, thus these two methods are feasible only when the graph structure is a tree or chain, where the partition function can be computed efficiently and exactly (*e.g.* using sum-product algorithm [14] or other inference algorithms discussed in Chapter 2.4).

Structured SVM learning. Structured support vector machines (structured SVMs) [125] are a max-margin approach where features are extracted jointly from the input space \mathcal{X} and output space \mathcal{Y} . Let $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}) \dots (\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \in \mathcal{X} \times \mathcal{Y}$ be the training data instances. In traditional binary or multi-class classification problems, \mathcal{Y} consists of orderless integer class labels. Structured SVMs are a generalization where each output label $\mathbf{y} \in \mathcal{Y}$ is a structure, *e.g.* a sequence, string, tree or even an arbitrary graph. In the linear case of structured SVMs, we wish to learn a *linear discriminant function* $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$ such that

$$f(\mathbf{x}^{(n)}, \mathbf{y}) = \mathbf{w}^T \phi(\mathbf{x}^{(n)}, \mathbf{y}), \quad \forall \mathbf{y} \in \mathcal{Y}$$

where $\phi(\mathbf{x}, \mathbf{y})$ defines a mapping between input/output pair (\mathbf{x}, \mathbf{y}) and the corresponding features. The prediction $\hat{\mathbf{y}}^{(n)}$ under f is given by

$$\hat{\mathbf{y}}^{(n)} = \operatorname{argmax}_{\mathbf{y}} f(\mathbf{x}^{(n)}, \mathbf{y}) \quad (2.5)$$

where \mathbf{w} is the parameter vector to be learned from the training data. Letting $\mathbf{y}^{(n)}$ be the ground truth output label for training instance $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \in \mathcal{X} \times \mathcal{Y}$, we can define the *margin* as the gap between the scores $f(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ (using the ground truth output label) and $f(\mathbf{x}^{(n)}, \hat{\mathbf{y}}^{(n)})$ (using the predicted output label under f).

We use loss functions to measure the disagreement between the prediction and the ground truth. For example, the loss function in Equation 2.4 is called a *log loss*. From a risk minimization point of view, MAP learning minimizes the log loss to learn the CRF parameters, and it requires the computation of $Z(\mathbf{x})$. In structured SVM, we assume that there exists a task-dependent loss function $\Delta(\mathbf{y}^{(n)}, \mathbf{y})$, such that

$$\Delta(\mathbf{y}^{(n)}, \mathbf{y}) > 0 \quad \forall \mathbf{y} \neq \mathbf{y}^{(n)} \quad \text{and} \quad \Delta(\mathbf{y}^{(n)}, \mathbf{y}^{(n)}) = 0$$

The definition of $\Delta(\mathbf{y}^{(n)}, \mathbf{y})$ can be very flexible compared with the standard zero-one loss function for classification task, and it quantifies how much the prediction $\hat{\mathbf{y}}^{(n)}$ is different from the ground truth output label $\mathbf{y}^{(n)}$. For example, in natural language parsing tasks, the F_1 score [56] can be used as the loss function to measure the correctness of predictions $\hat{\mathbf{y}}^{(n)}$ given ground truth output labels $\mathbf{y}^{(n)}$. In object detection tasks, we can use *intersection-over-union* (IOU) scores computed on the detection and ground truth bounding boxes as the loss function [15]. The choice of a particular loss function usually depends on the task, and is often related to the evaluation metric. However, such $\Delta(\mathbf{y}^{(n)}, \mathbf{y})$ is usually non-convex, which makes it difficult to incorporate in an optimization framework. Thus we consider minimizing

a piece-wise linear convex upper bound of the loss function:

$$\begin{aligned}
\Delta(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}) &= \Delta(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}) + \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) - \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \\
&\leq \Delta(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}) + \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \hat{\mathbf{y}}^{(n)}) - \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \quad (\text{via Eq. 2.5}) \\
&\leq \max_{\mathbf{y}} (\Delta(\mathbf{y}^{(n)}, \mathbf{y}) + \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y})) - \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \quad (2.6)
\end{aligned}$$

Equation 2.6 introduces the *margin rescaling* method, and the structured SVM can be formulated as the convex optimization problem below:

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \left(\max_{\mathbf{y}} (\Delta(\mathbf{y}^{(n)}, \mathbf{y}) + \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y})) - \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \right), \quad C > 0 \quad (2.7)$$

The above optimization problem can be solved by *cutting-plane* [55] or *stochastic gradient descent* [81, 102, 120] methods. Note that the number of possible assignments for $\mathbf{y} \in \mathcal{Y}$ is often exponential. Thus a common approach is to consider only a subset or working set of constraints at a time. In order to minimize the size of the working set, the training of a structured SVM is iterative, considering the subset of most violated constraints. Finding these constraints involves an important step called *loss-augmented inference*:

$$\tilde{\mathbf{y}}^{(n)} = \operatorname{argmax}_{\mathbf{y}} (\Delta(\mathbf{y}^{(n)}, \mathbf{y}) + \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y})) \quad (2.8)$$

For fixed \mathbf{w} that is estimated in the previous iteration, a new training instance $(\mathbf{x}^{(n)}, \tilde{\mathbf{y}}^{(n)})$ is added to the current working set of constraints according to Equation 2.8. Then the estimation of \mathbf{w} is updated, and the iteration stops when \mathbf{w} does not change.

2.4 CRF Inference

Given the observations \mathbf{x} and the model parameters \mathbf{w} , there are two common tasks in CRF inference problems: (1) compute the marginal distribution of the unknown variable(s) in \mathbf{y} ; (2) estimate the most likely configuration for each unknown variable in \mathbf{y} . Note that any CRF can be solved exactly using sum product for (1) and max product for (2). But the running time is exponential in size of the maximum clique. For chains and trees, dynamic programming (*i.e.* forward-backward for (1) and Viterbi algorithm [96] for (2)) are efficient. Here we review several major inference algorithms for conditional random field models with general graph structures. Inference on arbitrary graphs is NP hard, and these following approaches are approximate algorithms.

Iterated conditional modes. Iterated conditional modes (ICM) [13] is a deterministic algorithm that uses a “greedy” strategy to sequentially maximize the local conditional probabilities and find an approximate solution. It starts with an initial estimate of the output labels, and then for each unknown variable y_s in the graphical model, the label that gives largest increase of $p(\mathbf{y}|\mathbf{x})$ is chosen for y_s . The process is repeated until convergence. ICM results are quite sensitive to initialization, and the algorithm can be extremely inefficient.

Monte Carlo methods. Monte Carlo methods [14] are a series of sampling-based approaches for approximate inference in graphical models. The main idea of Monte Carlo methods is to express the given task as an expectation of a random variable $g(\mu)$ with respect to some distribution \mathcal{P} . Then we can estimate the expected value

from T samples generated by \mathcal{P} as follows:

$$\hat{g} = \frac{1}{T} \sum_{t=1}^T g(\mu^{(t)})$$

where $\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(T)}$ are T *i.i.d.* samples generated by \mathcal{P} . Monte Carlo estimates are guaranteed to converge by the law of large numbers, and the variance of the estimator can be reduced by increasing the number of samples.

Basic sampling-based approaches [14] include *rejection sampling*, *likelihood weighting*, and *importance sampling*, etc.; while in the case of high dimensional spaces, people usually use *Markov Chain Monte Carlo* (MCMC) methods [14,87] (e.g. *Metropolis-Hastings*, *Gibbs sampling*).

Graph-cuts methods. In the case of two labels, exact inference can be done using graph-cuts on *submodular* potential functions. For multi-label cases, Boykov *et al.* [16] propose two graph-cuts algorithm for approximate inference, *i.e.* *swap move* and *expansion move*, which involve solving min-cut / max-flow problems on specially constructed graphs. These algorithms are able to solve energy minimization problems with only unary and pairwise potentials efficiently. For high-order clique potentials, there exist efficient energy minimization methods [53, 64, 98] if the graph potential functions are still submodular.

The swap move algorithm randomly initializes a labeling \mathbf{f} for \mathbf{y} in the graph. Then in each iteration, a swap move operation is performed for each label pair $\langle \alpha, \beta \rangle$ and a new \mathbf{f}' is obtained. If the overall energy decreases, we update $\mathbf{f} = \mathbf{f}'$, and this process is repeated until no energy drop happens. The algorithm produces the final labeling \mathbf{f}^* as the inference results.

Expansion move is similarly defined for each label α such that the set of nodes

assigned with α increases, and a new labeling \mathbf{f}' is obtained if this operation leads to lower energy than original labeling \mathbf{f} . It finds a local minimum where no expansion move for any label can give another labeling with lower energy. An important property of the expansion move algorithm is that it produces a solution with a provable factor of the global minimum of the energy function [127]. In both swap move and expansion move algorithms, evaluating the swap move or expansion move involves inference on a 2-label subproblem.

Message passing methods. Message passing algorithms iteratively update the states of the unknown variables in \mathbf{y} by sending messages from a variable y_s to each of its neighbors y_t in the graph. Intuitively, the message represents the belief or confidence of which state y_s thinks y_t should be. Combining the confidence from its neighbors, y_t updates its own local belief and sends new messages to its neighbors. The process is iterative and stops when the overall energy of the graphical model does not change.

The most common message passing algorithm is called *belief propagation* (BP). BP is exact if the underlying graph structure is a tree, and is only approximate if the graph has loops. It has a *sum-product* version (used for estimating marginals) and a *max-product* version (used for estimate state configurations with maximum probability).

Let's take the CRF structure defined by Equation 2.3 as an example. In max-product BP, the algorithm iteratively computes a message $\mathbf{M}_{s \rightarrow t}$ from y_s to y_t , where $\mathbf{M}_{s \rightarrow t}$ is a vector of $|\mathcal{Y}_t|$ dimensions, and \mathcal{Y}_t is the set of possible labels for the unknown variable y_t . When passing a message from y_s to y_t at iteration t , it uses the following

message update rule to compute the message:

$$\mathbf{M}_{s \rightarrow t}^{(t)}(y_t) = \min_{y_s} \left(D(y_s) + V(y_s, y_t) + \sum_{u \in \mathcal{N}(s) \setminus t} \mathbf{M}_{u \rightarrow s}^{(t)}(y_s) \right)$$

Note that the name “max-product” comes from the probabilistic formulation. We typically use the log formulation, which is sometimes called *min-sum*. BP keeps passing messages between any two neighboring nodes in some order until convergence (*e.g.* when all messages do not change). For tree structures, BP is applied by computing the messages in a forward pass and then a backward pass. For general graph structures, there exist different choices for scheduling the messages (*e.g. random ordering, static schedule* [34], *dynamic schedule* [117]) since BP is not guaranteed to converge if the graph has loops. These different schedules not just affect the running time of BP, but also affect its convergence behavior.

As mentioned, belief propagation is not guaranteed to converge on loopy graph structures, and since it often finds a local minimum, it’s not clear how much gap exists between the CRF energy using the inference results and the global minimum. *Tree-Reweighted Message Passing* [67] (TRW), on the other hand, decomposes a general graph structure into multiple trees, and proves an upper bound for the original graph energy using these multiple tree structures via linear program relaxation. The inference on the original graph structure is called the *master problem*, and the inference on each tree graph is called as a *slave problem*. The constraints encode the equality relation between duplicated variables in different slave problems, and are combined all together to iteratively find the solution of the master problem.

To summarize, we have briefly introduced several common inference algorithms for conditional random fields. Among these algorithms, graph-cuts and message passing

algorithms (*e.g.* belief propagation) won the most academic popularity and have been applied to a number of problems recently. For highly connected graph structures, graph-cuts outperforms BP and TRW in terms of lower error rates and lower energy, and TRW is able to obtain lower energy than BP [66]. The challenges of TRW includes how to decompose the graph into different trees, and how to schedule the message updates. In addition, the convergence speed of TRW decreases as the graph connectivity increase. All these problems are open questions to solve, and thus the performance of TRW could be improved.

2.5 CRF with Latent Variables

In many structured prediction tasks, latent variables capture useful information that are not observed in the training process [95, 142]. For example, in object detection [39], locations of the components or parts of an object are very useful labels, but they are usually not available in the training stage. Object instances are labeled only with bounding boxes, since obtaining detailed part-level annotations from human users is usually time-consuming and tedious. Another example is protein function prediction [97], where we are given genomic sequences in the training data, and the goal of the algorithm is to generate a protein function name for each new genomic sequence. Here the protein structure is crucial and useful structured information that are unobserved for training instances.

CRFs with latent variables (or *latent CRFs*) are important tools to describe structures in these applications while capturing the latent information. We use \mathcal{X} to represent the label space of input variables. We also assume a label space \mathcal{Y} for the output variables \mathbf{y} , and a label space \mathcal{H} for the latent variables \mathbf{h} . The inference task

for structured models with latent variables involves finding labels that best explain the latent variables and assigning output labels to test instances at the same time. In other words, a desired form of the prediction rule is:

$$(\hat{\mathbf{y}}^{(n)}, \hat{\mathbf{h}}^{(n)}) = \underset{\mathbf{y} \times \mathbf{h} \in \mathcal{Y} \times \mathcal{H}}{\operatorname{argmax}} f(\mathbf{x}^{(n)}, \mathbf{y}, \mathbf{h}) \quad (2.9)$$

where $\hat{\mathbf{y}}^{(n)}$ is the predicted output label for $\mathbf{x}^{(n)}$, and $\hat{\mathbf{h}}^{(n)}$ is the inferred latent label.

To learn such a prediction rule from training data, we assume there is a loss function $\Delta(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}, \hat{\mathbf{h}}^{(n)})$ associated with the ground truth output label $\mathbf{y}^{(n)}$, predicted output label $\hat{\mathbf{y}}^{(n)}$ and the inferred latent label $\hat{\mathbf{h}}^{(n)}$ defined on the training instances. Again, the definition of the loss function depends on the particular task. Sometimes the performance evaluation is only relevant with $\mathbf{y}^{(n)}$ and $\hat{\mathbf{y}}^{(n)}$, and is independent of $\hat{\mathbf{h}}^{(n)}$ (*e.g.* object detection). To simplify the problem, the loss function in these cases can be written as $\Delta(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)})$.

We train the model using the *latent structural SVM* [142]. Similar to structural SVMs, we find an upper bound of the loss function:

$$\begin{aligned} \Delta(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}, \hat{\mathbf{h}}^{(n)}) &\leq \max_{\mathbf{y} \times \mathbf{h} \in \mathcal{Y} \times \mathcal{H}} \left[\mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \hat{\mathbf{y}}^{(n)}, \hat{\mathbf{h}}^{(n)}) + \Delta(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}, \hat{\mathbf{h}}^{(n)}) \right] \\ &\quad - \max_{\mathbf{h} \in \mathcal{H}} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{h}) \end{aligned} \quad (2.10)$$

We use the right hand side as the surrogate loss function, and the latent structural SVM can be formulated as:

$$\begin{aligned} \min_{\mathbf{w}} \|\mathbf{w}\|^2 + C \sum_{n=1}^N & \left(\max_{\mathbf{y} \times \mathbf{h} \in \mathcal{Y} \times \mathcal{H}} \left[\mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \hat{\mathbf{y}}^{(n)}, \hat{\mathbf{h}}^{(n)}) + \Delta(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}, \hat{\mathbf{h}}^{(n)}) \right] \right) \\ & - C \sum_{n=1}^N \left(\max_{\mathbf{h} \in \mathcal{H}} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{h}) \right) \end{aligned} \quad (2.11)$$

where $C > 0$ is a parameter that controls how much regularization is enforced on \mathbf{w} .

The training procedure of a latent structural SVM is still iterative, similar to structural SVM. Each iteration involves two inference operations: (1) loss-augmented inference on $\mathbf{y} \times \mathbf{h}$, and (2) inference on the latent variable \mathbf{h} given the ground truth output label $\mathbf{y}^{(n)}$. We can apply the inference algorithms in Section 2.4 for solving these problems. However, if the structure in \mathbf{y} is a general graph, then approximate inference algorithms without convergence guarantees will harm the SVM performance, since the constraint found in (1) is not necessarily the most difficult one. An interesting work [110] by Schwing *et al.* proposes an efficient message passing algorithm for *structured loss minimization with latent variables* to solve this problem. Their method is guaranteed to converge, and generalizes two different loss minimizations (MAP learning and SVM learning) for CRF into a single framework.

2.6 Applications of CRFs to Computer Vision

Besides the low level computer vision applications, CRFs can also be used in a variety of high level computer tasks, *e.g. image scene understanding, object detection, object tracking, and action recognition, etc.*

The goal of image scene understanding is to detect and recognize all objects and stuffs in an image, as well as estimate the scene structure and type. Shotton *et al.* [112] proposes TextronBoost for total scene understanding, where a CRF is built to capture local image evidence at pixel level and the interactions between neighboring pixels jointly in the same framework. In the object detection task, a CRF is often used to represent an object as a set of parts decomposed from the object, plus the geometric relations among the parts [21, 39, 40]. These models are able to combine evidence from each part detector as well as the constraints among the parts, and estimate the

object’s location until having collected all information. CRFs can also be used to capture temporal relations in the object tracking task via a chain-like structure that put constraints on corresponding parts from adjacent video frames [108, 121]. In the action recognition task, Lan *et al.* [72] proposes a novel CRF model that captures group activity, individual actions, and their interactions in the single framework, and proposes a discriminative learning method to learn all parameters together.

In this thesis, we apply CRFs in three different object recognition problems. We first design a novel hierarchical CRF that models the kinematic structure of a human body to estimate human poses in 2D static images. Then we use a latent multi-layer CRF to model local image regions (*i.e.* what we call *local attributes*) that are both discriminative and semantically meaningful. Lastly, we apply a latent CRF to generalize the classical K -means clustering framework for multimodal image modeling. Although different in terms of specific applications, their structured properties allow conditional random fields to be a useful common framework for solving these problems.

2.7 Summary

We have reviewed the definition of conditional random fields, and have discussed major learning and inference algorithms on such graphical models. We focus on using structural SVMs and its variant (*e.g.* latent structural SVM) to train model parameters, and using message passing algorithms for solving inference problem in CRFs with general graph structures. The advantages of conditional random fields and their variants include that (1) they have flexible model structures, thus can be applied to many structured prediction problems; and (2) they directly model the

conditional probability, and thus have strong discriminative power. We also briefly introduced several high level computer vision tasks where CRFs play an important role on capturing the structured information.

In the following chapters, we will discuss the details of using CRFs to solve different object recognition problems at different supervision levels (full supervision for pose estimation, semi-supervision for attribute discovery, and loose supervision / no supervision for multimodal image modeling).

CHAPTER 3

Multi-layer Models for Human Pose Estimation

In this chapter, we introduce our novel multi-layer CRF models for human pose estimation problems. We begin in Section 3.1 by discussing related existing work, and then describe our baseline model in Section 3.2, which is very similar to [138]. We show how to generalize this to a multiscale model in Section 3.2.1, and then discuss how to perform inference efficiently using dual-decomposition in Section 3.2.2. We then describe how to jointly learn the parameters of the model from labeled training data in Section 3.2.3.

3.1 Related Work

Given a static image of human bodies, the human pose estimation task not only requires estimating a bounding box over the person, but also requires locating the limbs and body parts. We now review important literature related to human pose estimation.

3.1.1 Pictorial Structures

Felzenszwalb and Huttenlocher [40] introduced part-based tree-structured deformable models to the problem of human pose recognition, and called these models “pictorial

structures.” In a part-based model, an object is represented using a collection of “parts” with some model of the spatial relationship between them. Each part has an appearance model capturing local appearance, while the spatial models typically prefer some configurations but allow deformation due to human flexibility and image transformations. In the case of human pose recognition, the spatial model encodes the kinematic constraints, *e.g.* the arms are connected to shoulders, *etc.* Pictorial structures can naturally be modeled in a probabilistic graphical model framework, in which there is a latent variable for each part denoting that individual part’s pose, and there are connections between some of the part variables that encode constraints on relative part position. Each variable also observes the image data. Pose recognition can then be considered as an inference problem, where the goal is typically to find the most-likely values for the latent pose variables given the image data. Felzenszwalb and Huttenlocher [40] used 4-d pose variables that parameterize part pose as 2-d position, orientation, and scale (foreshortening), and encode the pairwise configuration priors as normal distributions. They show that exact inference on tree-structured graphical models can be performed efficiently via dynamic programming and distance transforms, requiring only $O(ph)$ time, where p is the number of parts in the model and h is the number of possible pose configurations for each part. They used very simple part appearance models, essentially looking for rectangular blobs in binary segmentations produced by background subtraction, but later work built on their technique to create state-of-the-art pose recognition systems.

For instance, Ramanan *et al.* [100] used the same framework but improved the part appearance models and adopted an iterative inference approach. An edge-based deformable model is first applied on the image to obtain a soft (and noisy) estimate of

body part locations, then a region-based model is used to look for body parts (torso, legs, arms, *etc.*) based on learned “part specific” appearance models. The resulting soft estimates from the region-based model are further updated in an iterative process as new region-based models are repeatedly built with better and better part appearance models. Their later work [99] extended this to tracking pose across time in video, where the appearance models become customized to a particular person (e.g. based on clothing color). Andriluka *et al.* [5] achieved significantly better results by learning appearance models in a discriminative Adaboost-based framework. The appearance representation is based on dense shape context descriptors; the kinematic pose prior is modeled as a tree structure, and is learned separately on a multi-view and multi-articulation dataset.

More recent work has explored various strategies for enhancing the same basic part-based framework, and we borrow several of these innovations in our proposed approach. We discuss three specific innovations in three sections: hierarchical models, multi-scale feature representations, and mixture models.

3.1.2 Hierarchical Models

There have been various techniques for relaxing the part independence assumptions have been proposed (*e.g.* [73, 124]). Particularly relevant to our work are approaches using hierarchical models, such as [147] and [137]. Zhu *et al.* [147] proposed a Max Margin AND/OR graph for parsing human body into parts. The model is a multi-level mixture of Markov Random Fields where each node represents a human body part at a certain level in the hierarchy. However the appearance models are defined on sub-parts of body segments, but image segmentation is not always accurate so

the results are not always reliable. Wang *et al.* [137] proposed a hierarchical human parsing model based on the notion of “hierarchical poselets”, *i.e.* multi-scale body parts with various sizes. The body parts are defined in a hierarchy at different scales, and are able to cover human poses at various levels of granularity. Our proposed composite models are also hierarchical, but differ in the structure of the hierarchy. In our ensemble, each submodel is a separate and complete tree-structured model of human pose, as opposed to simply being “larger” parts as in [147] and [137]. This distinction is crucial since this unique graphical structure allows the use of principled and efficient inference based on dual-decomposition, while reusing existing algorithms developed for tree-structured models.

3.1.3 Multi-scale Models

Recent work in pose recognition has shown that capturing visual features at multiple image scales is important. Sapp *et al.* [107] use cascaded models at different resolutions for estimating articulated human poses. However the major benefit is to speed up inference, rather than to improve the estimation accuracy. Park *et al.* [92] use multi-resolution models to detect objects at different scales. They introduce a binary variable for each object to encode two states for the object size, corresponding to visual features at two scales (coarse scale and fine scale). At detection time, the binary variable is jointly inferred with multi-resolution detector output (*i.e.* object locations). Yang *et al.* [138] incorporate visual cues at multiple resolutions by building *Histogram-of-Gradients* (HoG) feature pyramids, a technique which we also use here.

3.1.4 Mixture Models

To accurately model the highly flexible human form, mixture models for both appearance and geometry have been proposed. Singh *et al.* [113] use a linear weighted combination of heterogeneous part detectors, fusing evidence from different feature types. A branch-and-bound approach is used at test time to make inference on the graphical model faster. Wang *et al.* [135] use mixtures of tree models to capture richer spatial constraints and explicitly model part occlusions at the same resolution level. A boosting procedure is used to combine different tree structures when inference needs to be done on test images. However, the design of the structures for different tree models is hand-crafted, and it is not clear whether a tree model is stronger or weaker than another. Thus, how to select the best set of tree structures is a difficult and unsolved problem.

Johnson *et al.* [57] cluster human poses and partition the space of human poses into a mixture of components, then build mixtures of pictorial structure models using these clusters. The appearance models are “pose specific” and capture the correlation between part appearances. Yang *et al.* [138] introduce mixture models for each human body part. Specifically, their model assigns a latent “type” variable to each part, allowing parts to select between several appearance models, and they jointly learn the parameters in a discriminative structured learning framework. We use a similar approach based on latent part types, but in a framework featuring hierarchical, multi-scale models.

3.1.5 Dual Decomposition

In this thesis, we apply an inference technique called dual decomposition to the problem of pose recognition. Dual decomposition, also called Lagrangian relaxation, has been shown to be a useful tool for solving optimization problems with discrete variables. It decomposes the original optimization objective into small, independent, and easy-to-solve subproblems, and then the solutions to the subproblems are combined together into a global solution to the original problem. Some recent work has also applied dual decomposition to pose recognition, but on different models and applications than ours. Wang *et al.* [133] model pose estimation and segmentation jointly, and apply dual-decomposition for efficient inference. The human pose estimation guides the foreground pose segmentation at a high level, and the segmentation cues also help to improve pose estimation results by explaining low-level pixels. Sapp *et al.* [108] use dual-decomposition for articulated motion parsing in video using motion features. A tree model is used to capture the human pose structures, and the motion cues connect joints across consecutive frames. Dual-decomposition decouples the problems into a handful of small subproblems where only one joint is being tracked and is used to connect two tree structures, so all subproblems are still tree structures and one can perform exact and efficient inference on them. However, that paper does not consider hierarchical models as we do here.

3.1.6 Other Relevant Work

More recent papers have tried to address pose estimation problems from different perspectives. Hara *et al.* [50] propose to use a dependency graph for modeling relations between body parts, and use independently trained discriminative regressors as

part appearance models. Danstone *et al.* [24] jointly learn non-linear part regressors using a two-layered random forests for more discriminative part templates and show impressive improvements over the start-of-art methods. Sapp *et al.* [106] capture multiple pose models for half and full human bodies at a large granularity. Each pose mode is defined as one of the clusters of the human body joint configurations, and is trained using a discriminative structured linear model.

3.1.7 Summary

Our proposed model is based on pictorial structures, and is relevant with all the above variants of deformable models. Similar with other approaches, we use hierarchical parts with different scales and resolutions in our model; however in our composite model, parts with same scale and resolution are in the same layer, and they form an individual pictorial structure model as well. The entire composite model can be considered as a mixture of multiple tree submodels, while the inference on these submodels is performed jointly in the same framework. The “composite” nature of our model allows the use of dual-decomposition method for efficient inference.

3.2 Multi-layer Composite Models

We now describe our technique for pose recognition using multi-layer composite models. Given an image I and a model of the human body, the goal of pose recognition is to find high-likelihood model configurations in the image. Our approach builds on the work of Yang and Ramanan [138] which has demonstrated state-of-art performance. The key innovation in their deformable part-based model is the use of a mixture of parts, which allows the appearance of each part to change discretely between differ-

ent “part types.” One part type is one configuration of the pose for a specific part. For example, a leg part might have part types like “lying down” or “standing up,” corresponding to different orientations and articulations of the leg. These are latent variables so they must be estimated along with pose, but they allow the model to switch between different spatial models in order handle larger variations in pose than would otherwise be possible. For example, even though the individual appearance and spatial models for each part type are relatively simple, the composition of the mixture of parts can approximate complex transformations, such as both in-plane and out-of-plane image warps. Also, instead of using parts that correspond with natural body (arms, torso, hands, *etc.*), authors in [138] use small square part templates for each *joint* of the human body (*e.g.* ankles, elbows, chin, top of head, *etc.*). This also gives greater invariance to pose changes, since the appearance of a joint varies less dramatically than the part appearances themselves.

More formally, their model consists of a set \mathcal{P} of parts in a tree-structured model having edges $\mathcal{E} \subseteq \binom{\mathcal{P}}{2}$, such that \mathcal{E} is a tree. Let \mathbf{y} be a vector that represents a particular configuration of the parts, *i.e.* the location and type of each part. They define a function $S(I, \mathbf{y})$ that scores the likelihood that a given configuration \mathbf{y} corresponds to a person in the image. Moreover, $S(I, \mathbf{y})$ decomposes along the nodes and edges of the tree:

$$S(I, \mathbf{y}) = \sum_{p \in \mathcal{P}} D(I, \mathbf{y}_p) + \sum_{(p,q) \in \mathcal{E}} \left(L(\mathbf{y}_p, \mathbf{y}_q) + T(\mathbf{y}_p, \mathbf{y}_q) \right), \quad (3.1)$$

where $D(I, \mathbf{y}_p)$ is the score for part p being in configuration \mathbf{y}_p given local image data (the data term), $L(\mathbf{y}_p, \mathbf{y}_q)$ is the relative location term measuring agreement between locations of two connected parts, and $T(\mathbf{y}_p, \mathbf{y}_q)$ measures the likelihood of observing

this pair of part-types. Specifically, $D(I, \mathbf{y}_p)$ is the template matching score for part p at location \mathbf{y}_p , $L(\mathbf{y}_p, \mathbf{y}_q)$ is defined as the negative Mahalanobis distance between part locations, and $T(\mathbf{y}_p, \mathbf{y}_q) = \vec{\mathbf{B}}^{t(\mathbf{y}_p), t(\mathbf{y}_q)}$ is a part co-occurrence table that is learned discriminatively in the training stage, where $t(\mathbf{y}_p)$ gives the part type of part p .

3.2.1 Proposed Generalization

We generalize this model to include multiple layers, with each layer like the base model but with a different number of parts and a different tree structure. In particular, let $\mathcal{M} = \{(\mathcal{P}_1, \mathcal{E}_1), \dots, (\mathcal{P}_K, \mathcal{E}_K)\}$ be a set of K tree-structured models, let \mathbf{y}^k denote the configuration of the parts in the k -th model, and let $\mathbf{Y} = (\mathbf{y}^1, \dots, \mathbf{y}^K)$ be the configuration of the entire multi-layer composite model. We now define a joint scoring function,

$$\hat{S}(I, \mathbf{Y}) = \sum_{k=1}^K S_k(I, \mathbf{y}^k) + \sum_{k=1}^{K-1} \chi(\mathbf{y}^k, \mathbf{y}^{k+1}), \quad (3.2)$$

where $S_k(\cdot, \cdot)$ is the single-layer scoring function of equation (3.1) under the model $(\mathcal{P}_k, \mathcal{E}_k)$, and $\chi(\mathbf{y}^k, \mathbf{y}^{k+1})$ is the cross-model scoring function that measures the compatibility of the estimated configurations between adjacent layers of the model.

As Figure 1.4 shows, we impose a hierarchical structure on the composite model, such that each part at level k is decomposed into multiple parts at level $k+1$. We call these decomposed parts the child nodes. For a part $p \in \mathcal{P}_k$, let $C(p) \subseteq \mathcal{P}_{k+1}$ be the set of child nodes of p in layer $k+1$. The cross-model scoring function χ scores the relative location and part types of a node in one layer with respect to its children in the layer below,

$$\chi(\mathbf{y}^k, \mathbf{y}^{k+1}) = \sum_{p \in \mathcal{P}_k} \sum_{q \in C(p)} B(\mathbf{y}_p^k, \mathbf{y}_q^{k+1}), \quad (3.3)$$

where $B(\mathbf{y}_p^k, \mathbf{y}_q^{k+1})$ is also a look-up table, and measures the likelihood of the relative configuration of a part and its child across the two submodels. Next, we describe inference in this composite model and then discuss how to learn parameters of the composite model in Chapter 3.2.3.

3.2.2 Dual Decomposition for Efficient Inference

We have defined our multi-layer composite model as a collection of pose estimation models and a cross-model scoring function. As Figure 1.4 illustrates, each layer of the hierarchy is tree-structured, so exact inference within each layer can be performed efficiently via dynamic programming. The constraints between layers (blue lines in the figure) also form a tree-structured model, so they are also amenable to exact efficient inference. The overall graphical model has cycles, however, and thus exact inference on this model is not tractable. Fortunately, we can exploit the natural decomposition of this composite model into tree-structured subproblems to perform inference using dual-decomposition. Dual-decomposition is a classical technique [12] that has recently been introduced to the vision literature [67] for solving inference problems in loopy graphical models. The idea is to decompose a joint inference problem into easy sub-problems, solve each sub-problem, and then iteratively have the sub-problems communicate with each other until they agree on variable values.

The following steps are a straightforward adaptation from [67]. Let \mathcal{C}_k denote the set of all feasible (discrete) values for \mathbf{y}^k for each layer of the model. We make a copy of \mathbf{Y} , which we call $\mathbf{X} = (\mathbf{x}^1, \dots, \mathbf{x}^K)$, and enforce equality constraints that require

$\mathbf{Y} = \mathbf{X}$. With this notation, we can rewrite equation (3.2) as:

$$\begin{aligned} & \max_{\mathbf{Y}, \mathbf{X}} \sum_{k=1}^K S(I, \mathbf{y}^k) + \sum_{k=1}^{K-1} \chi(\mathbf{x}^k, \mathbf{x}^{k+1}), \\ & \text{s.t. } \mathbf{y}^k = \mathbf{x}^k, \mathbf{y}^k \in C_k, \mathbf{x}^k \in \mathcal{C}_k, \quad \forall k. \end{aligned} \quad (3.4)$$

We then *dualize* the equality constraints, replacing the hard equality constraints between \mathbf{Y} and \mathbf{X} with a soft penalty term,

$$\begin{aligned} g(\lambda) &= \max_{\mathbf{Y}, \mathbf{X}} \sum_{k=1}^K S(I, \mathbf{y}^k) + \sum_{k=1}^{K-1} \chi(\mathbf{x}^k, \mathbf{x}^{k+1}) + \sum_{k=1}^K \lambda_k \cdot (\mathbf{y}^k - \mathbf{x}^k), \\ & \text{s.t. } \mathbf{y}^k \in \mathcal{C}_k, \mathbf{x}^k \in \mathcal{C}_k, \end{aligned} \quad (3.5)$$

where λ_k is the Lagrangian multiplier that specifies the strength of the penalty, and \cdot denotes inner product between two vectors. The effect of relaxing the hard equality constraint is that the maximization can now be decoupled into independent terms,

$$g(\lambda) = \sum_{k=1}^K \max_{\mathbf{y}^k} (S(I, \mathbf{y}^k) + \lambda_k^T \cdot \mathbf{y}^k) + \max_{\mathbf{X}} \left(\sum_{k=1}^{K-1} \chi(\mathbf{x}^k, \mathbf{x}^{k+1}) - \sum_{k=1}^K \lambda_k^T \cdot \mathbf{x}^k \right), \quad (3.6)$$

again with $\mathbf{y}^k \in \mathcal{C}_k$, $\mathbf{x}^k \in \mathcal{C}_k$. In this form, it is clear that $g(\lambda)$ can be evaluated for a given λ by solving a series of simpler sub-problems. The optimal \mathbf{Y} is found by maximizing each term of the first summation, *i.e.* by performing inference on each individual layer of our composite model, which is efficient because each layer is a tree-structured model. We can find the optimal \mathbf{X} by solving the maximization in the second term of equation (3.6), which is also tree-structured and allows the use of dynamic programming.

Figure 3.1 provides an intuitive illustration of performing dual-decomposition on our model. Each of the slave problems are tree-structured inference tasks and can be performed efficiently with dynamic programming. The master's job is to encourage agreement between the solutions found by the slave tasks.

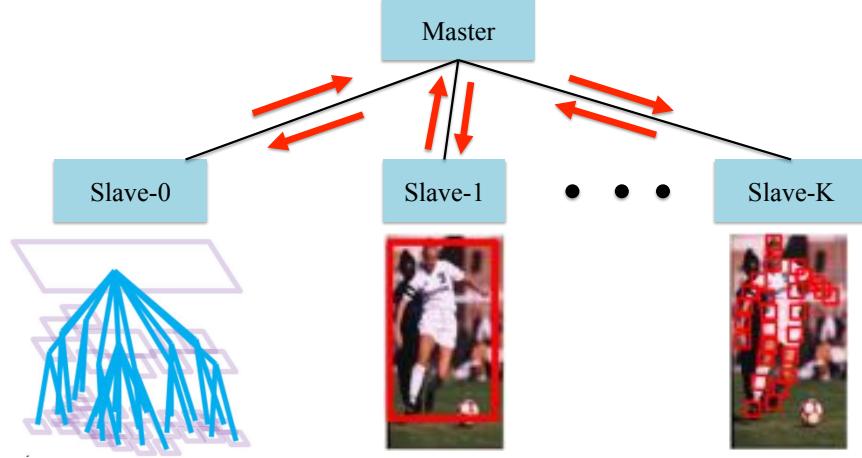


Figure 3.1: Dual-decomposition on our multi-layer composite pose model. Messages are passed between Master problem (primal objective) as slave problems (decomposed dual objectives) iteratively.

It can be shown [12] that for each value of λ , the function $g(\lambda)$ provides an upper-bound on the original (constrained) maximization. Thus, we can set up a dual problem that achieves the tightest upper-bound as $\min_{\lambda} g(\lambda)$. This dual problem is convex but non-smooth [12], so we use subgradient descent to perform the minimization. Subgradient descent is an iterative algorithm that updates the current setting of $\lambda_k^{(t)}$ at iteration t ,

$$\lambda_k^{(t+1)} \leftarrow \lambda_k^{(t)} - \alpha^{(t)} \left(\mathbf{y}^k(\lambda_k^{(t)}) - \mathbf{x}^k(\lambda_k^{(t)}) \right), \quad (3.7)$$

where $\mathbf{y}^k(\lambda_k^{(t)})$, $\mathbf{x}^k(\lambda_k^{(t)})$ are the optimal solutions in equation (3.6) for the current setting of $\lambda_k^{(t)}$, and $\alpha^{(t)}$ is the step size at iteration t . For a good choice of step size, subgradient descent is guaranteed to converge to the optimum of the dual problem [12]. We discuss implementation details like the step size and stopping criteria in Chapter 5.3.

3.2.3 Learning with Structural SVMs

We now address the issue of learning the parameters of our composite model, including the submodel parameters for each layer and the parameters for the cross-model scoring function.

Features. We use four kinds of features: the part appearance features that help learn what each part “looks like” based on local image evidence, deformation features which capture spatial relationships between parts, the part type co-occurrence features within layers, and the part type co-occurrence features across layers. We combine the first three of these into a feature vector called $f(I_m, \mathbf{y}^k)$, which denotes the vector for image I_m under submodel k , and define in the same way as [138]. In particular, $f(I_m, \mathbf{y}^k)$ consists of HOG features for each part filter, part type co-occurrence features, and deformation features (dx, dx^2, dy, dy^2) , where (dx, dy) is the displacement between two parts. We denote the fourth feature type, the cross-model part type co-occurrence feature as $f^\chi(\mathbf{Y})$ by converting the 2D look up table $\delta_{t(\mathbf{y}_p), t(\mathbf{y}_q)}$ to a 1D vector, where $\delta_{t(\mathbf{y}_p), t(\mathbf{y}_q)} = 1$ if $t(\mathbf{y}_p) = t(\mathbf{y}_q)$, and otherwise $\delta_{t(\mathbf{y}_p), t(\mathbf{y}_q)} = 0$.

Parameters. To perform joint training for the entire composite model, we stack all features of all of the layers along with the cross-model features into a single feature vector $\Phi(I_m, \mathbf{Y})$,

$$\Phi(I_m, \mathbf{Y}) = [f(I_m, \mathbf{y}^1), f(I_m, \mathbf{y}^2), \dots, f(I_m, \mathbf{y}^K), f^\chi(\mathbf{Y})], \quad (3.8)$$

and parameters of the model are also placed into a single vector,

$$\beta = (\beta^1, \dots, \beta^K, \beta^\chi).$$

The score of the entire composite model on a given image and configuration can then

be written as a dot product between parameters and features,

$$\hat{S}(I, \mathbf{Y}) = \beta \cdot \Phi(I_m, \mathbf{Y}).$$

Training. Given training data with labeled positive instances, *i.e.* images containing people with annotated part locations $\{\{I_m, \mathbf{Y}_m\} \mid m \in \text{pos}\}$, and negative instances, *i.e.* images not containing people $\{\{I_m, \emptyset\} \mid m \in \text{neg}\}$, we learn β with a structured SVM formulation [125],

$$\begin{aligned} \min_{\beta} \quad & \frac{1}{2} \|\beta\|^2 + C \sum_m \xi_m \\ \text{s.t.} \quad & \beta \cdot \Phi(I_m, \mathbf{Y}_m) \geq 1 - \xi_m \quad \forall m \in \text{pos} \\ & \beta \cdot \Phi(I_m, \mathbf{Y}) \leq -1 + \xi_m \quad \forall m \in \text{neg}, \forall \mathbf{Y}. \end{aligned} \tag{3.9}$$

We optimize this objective function using the dual coordinate descent method of [138]. Note that this formulation forces all of the exponentially many configurations of negative instances to score lower than -1 . In practice, we perform dual decomposition with our multi-layer composite model on each negative image to search for hard negative training examples. Implementation details are explained in Chapter 3.3.2.

3.3 Experiments

3.3.1 Datasets

We evaluate our composite models on three challenging datasets: Image Parse [100], UIUC Sport [137] and Leeds Sport Pose [57]. Parse contains 100 training and 205 test images, while UIUC Sport contains 649 training and 650 test images. Leeds Sport Pose is much larger, with 1000 training and 1000 test images. All three datasets have

one person per image annotated with 14 body joints. We follow [138] and draw our negative images from the INRIA person dataset [23].

3.3.2 Implementation

We implemented our inference and learning methods described in Chapter 3.2. Here we give some implementation details that are important in practice.

Inference

For the part appearance models, we follow [138] and others by using HOG features [23] computed at multiple resolutions, yielding a feature pyramid for each image. We perform dual decomposition on each level of the feature pyramid independently, collect detections from all of the levels, and remove overlapping detections via non-maximal suppression. In our current implementation, we restrict our cross-modeling scoring function $B(\cdot, \cdot)$ in equation (3.3) to capture only part type co-occurrence relations. This gives a relatively small label space, which allows efficient inference while obtaining good performance (although modeling relative location between parts across layers is an interesting direction for future work).

The subgradient descent step size in equation (3.7) is important in making inference work well in practice. We experimented with various strategies, finding that a modification of Polyak's step size rule [94],

$$\alpha_k^{(t)} = \frac{1+m}{\tau^{(t)}+m} \cdot \frac{(dual^{(t)} - primal_{best}^{(t)})}{\|\nabla g_t\|},$$

worked best, where $dual^{(t)}$ is the objective value of the dual problem in equation (3.6) in iteration t , $primal_{best}^{(t)}$ is the *best* primal objective value in equation (3.4) observed

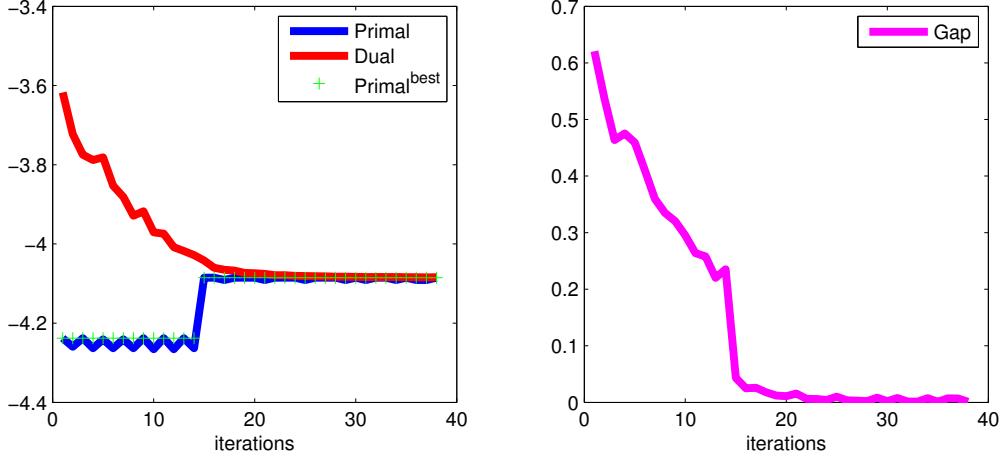


Figure 3.2: Primal objective and dual objective (left) and primal-dual gap (right) as a function of number of iterations during subgradient descent.

so far in iterations up to t , $\|\nabla g_t\|$ is the norm of the subgradient at t , m is a scalar constant (we use $m = 10$), and $\tau^{(t)}$ is the number of times that the dual-objective has increased up to t . Using this step size rule, dual decomposition converges to a very small gap (< 0.001) quickly, as shown in Figure 3.2 (for a sample image). The entire inference process takes about 20 seconds per Parse image on a 3.0GHz machine.

Learning

For each dataset, we trained several variants of our composite models: 1) a two-layer model consisting of a 1-part model and a 26-part model; 2) a two-layer model consisting of a 10-part model and a 26-part model; 3) a three-layer model consisting of 1-part, 10-part, and 26-part models.

In all of these models, the 26-part model is the same defined in [138], consisting of both body parts and joints. The 10-part model is defined using new body parts (head, torso, upper arms, lower arms, upper legs, lower legs), and the 1-part model

is a simple whole-body template mixture model. The annotations for the 10 and 1 part models were derived from the existing annotations in the datasets. As in [138], the mixture types of each body part were obtained by k -means clustering over joint locations. For the 26-part model, we use the same number of part types per body part as in [138], *i.e.* a variable number of 5 or 6 mixtures for each part, while for the 10-part model we use 5 torso types, 5 head types, 5 arm types and 6 leg types. The 1-part model uses 9 types. To learn each composite model, we first train a separate model for each layer using the publicly-available code of [138], and then use these models as initialization for learning our composite model.

In practice, there are many more negative (non-person) instances available than positive instances. To reduce the set of negative exemplars that must be considered in equation (3.9), we select hard negative exemplars for the next iteration of learning by looking for high-scoring non-person instances under the current multi-layer composite model. To construct negative training instances efficiently, we run the composite model on each negative image, select all detected poses having score above a threshold, sort the detections from each layer, and construct joint exemplars by matching them in the order of detection scores. To speed up training, we stopped subgradient descent after 50 iterations, since in practice the optimization algorithm has typically converged by that point (as in the example in Figure 3.2). A visualization of a sample multi-layer composite model learned using our technique is shown in Figure 3.3.

3.3.3 Results

Evaluation Criteria. We evaluate our results using the Percentage of Correct Parts (PCP) metric, which counts the fraction of body parts that are correctly localized

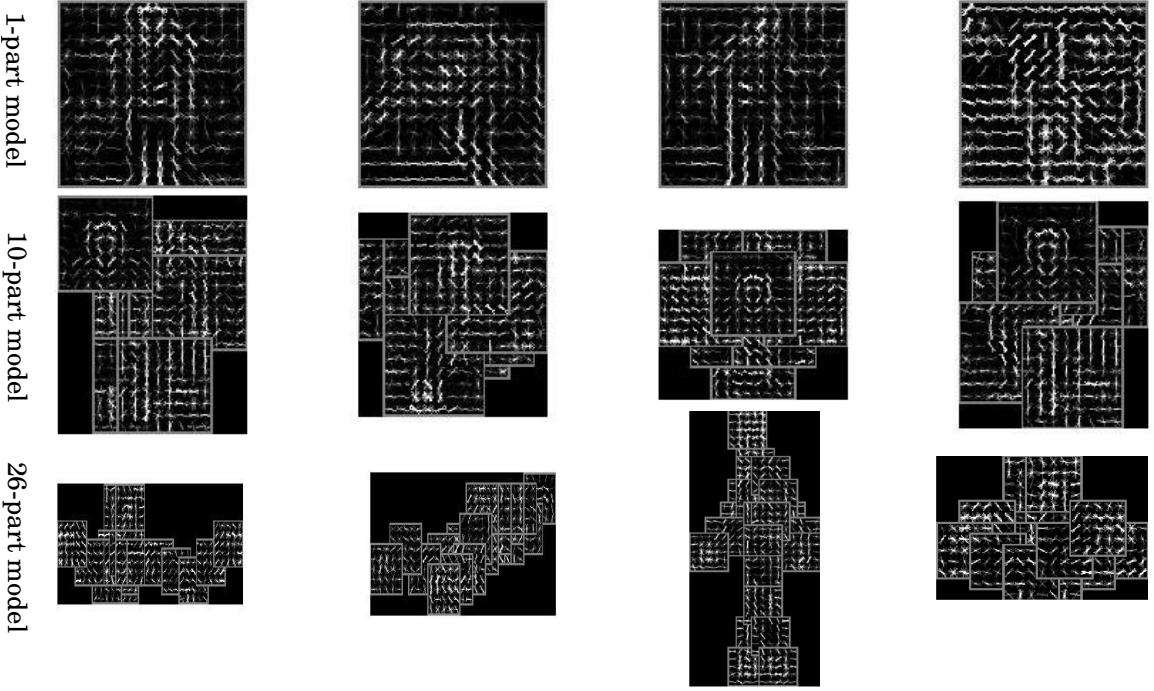


Figure 3.3: Part-based models used in our multi-layer composite model. For each layer (row) of the composite model, we show four randomly-chosen mixture components.

according to the ground-truth. One needs to define what a correct localization is, since small discrepancies in part pose are probably not noticeable in most applications. Unfortunately, as pointed out in [93], the PCP scoring metric has been implemented in slightly different ways in different papers, which has led to some confusion in the literature. These differences fall along two different dimensions. First, there are two subtly-different definitions of a correct part localization: 1) Part is correctly localized if the distance of *both* its endpoints from respective ground truth endpoints is less than a fraction of the part length; or 2) Part is correctly localized if the *mean* distance between estimated and ground truth endpoints is less than a fraction of the part length.

This difference is illustrated in Figure 3.4.



Figure 3.4: Illustration of distances $D1$ and $D2$, the two measurements involved in evaluating body part localizations (see text). The first measure of PCP considers a part correctly localized if both distances are below a threshold, whereas the second measure considers a part correctly localized if the *mean* of $D1$ and $D2$ is below a threshold.

Second, there are two ways to compute the final aggregate PCP score across the dataset: A) PCP is calculated for every image, and averaged across *all* images to produce an aggregate score; or B) PCP is calculated *only* for images in which the human is correctly localized according to a ground truth bounding box, these scores are averaged together, and then multiplied by the detection rate.

The cross-product of these two possibilities yields four possible evaluation criteria. According to our understanding, Eichner *et al.* [33] proposed variant 1B, but their publicly-released software toolkit implemented 2B which yields higher scores. Yang *et al.* [138] also used 2B, while both Pischulin *et al.* [93] and Wang *et al.* [137] used 1A. Unfortunately, these seemingly subtle variations lead to significant differences. We follow the two latter papers and also use 1A, which we hope will become the standard

definition, but also report results under the other variants to illustrate the significant differences they create. Note that [93] do not report PCP numbers for individual parts, but rather combine right and left parts together. We do the same, and also average the PCP of the left and right limbs reported by [137] to convert their results into this metric as well. Recently, Yang *et al.* [139] propose a novel evaluation metric called *Percentage of Correct Keypoints* (PCK) and *Average Precision of Keypoints* (APK), which are based on a comparison between predicted and ground truth bounding boxes surrounding each keypoint of the human body. Since no previous work has reported the performance under such criteria, it is difficult for us to do systematic comparisons with this metric, so we stick with the better-known PCP metric here.

Results. Results on Parse, UIUC Sport and Leeds Sport Pose datasets are shown in Table 3.1 for our technique and several other recent methods. To make all of the numbers compatible, we converted all numbers (both our own and those in the literature) to use criteria 1A. To do this, we re-computed the results from [138] to use this criterion, and for [100] we use the re-computed statistics reported in [137]. We see that our composite models outperform state-of-the-art methods on all three datasets, beating [138] by about 2 percentage points for Parse and by $1.0 - 1.5$ percentage point for the other two datasets. We also show results from two recent techniques, Pishchulin *et al.* [93] and [58], that are not directly comparable to our technique or the other baselines because they used additional training data with richer annotations.¹ Our results do not outperform these techniques, but again we cannot compare

¹In particular, Johnson *et al.* [58] annotated a training dataset of 10,800 images downloaded from Flickr using Amazon Mechanical Turk. Pishchulin *et al.* [93] fit a 3D human body shape model to each training image with annotated 2D body keypoints, and then vary the 3D shape parameters

them directly because they use vastly more training data, and our results are still numerically competitive given that we use much less information during training.

Table 3.2 presents experimental results under alternative definitions of PCP. For PCP criterion 1A, we present scores for different values of the part localization threshold (which specifies the percentage of body part length that part endpoints can be from the positions given in ground truth). The table also shows PCP results computed under two alternative definitions that have been used in the literature (1B and 2B). We see that seemingly subtle differences in PCP definition can yield very different conclusions. Our composite models beat [138] under all of the criteria, but which composite model performs best depends on the PCP metric. For instance, the 2-layer model (26+1) achieves the best performance under 1A, but the 3-layer model performs best under 1B and 2B. Moreover, variant 2B yields much higher absolute PCP scores, illustrating the importance of adopting a consistent metric to avoid further confusion in the literature.

Some qualitative pose recognition results are presented in Figure 3.5, showing cases in which our method correctly estimated pose while [138] failed for one or more limbs. We also show some images on which our technique failed.

We also evaluated our techniques in terms of person detection rate, with 79.0%, 81.9%, and 82.4% for our 26+10, 26+1, and 26+10+1 models, respectively, compared to 76.6% for [138]. This suggests that much of our increase in PCP is due to more accurate detections. This is an intuitive result because our 1-part model (consisting of a mixture of large HOG templates) can be considered a person detector (essentially the same as [23]). Our composite models featuring models at multiple scales combine in order to create new 2D poses for additional training data.

the advantages of single-part models for person detection, with the highly flexible multi-part models needed for accurate part localization.

3.4 Summary

In this chapter, we presented a multi-layer composite model for human pose estimation problems. By combining different cues from different submodels, our composite model outperforms state-of-the-art pose estimation methods on challenging datasets. These results show that hierarchical structures and mixture models for parsing human body is important, and dual decomposition technique for such composite model is effective in practice. Our model is a general framework for combining different pose estimation models. In future work, we plan to study any improvements to our approach, *e.g.* capture richer cross-model constraints by defining spatial constraints between adjacent submodels, or learn the composite model in a weakly supervised mode when annotations are not available for all key points on the training images. Our model is also related to interesting tasks like human action recognition, and has potential application in surveillance problems.

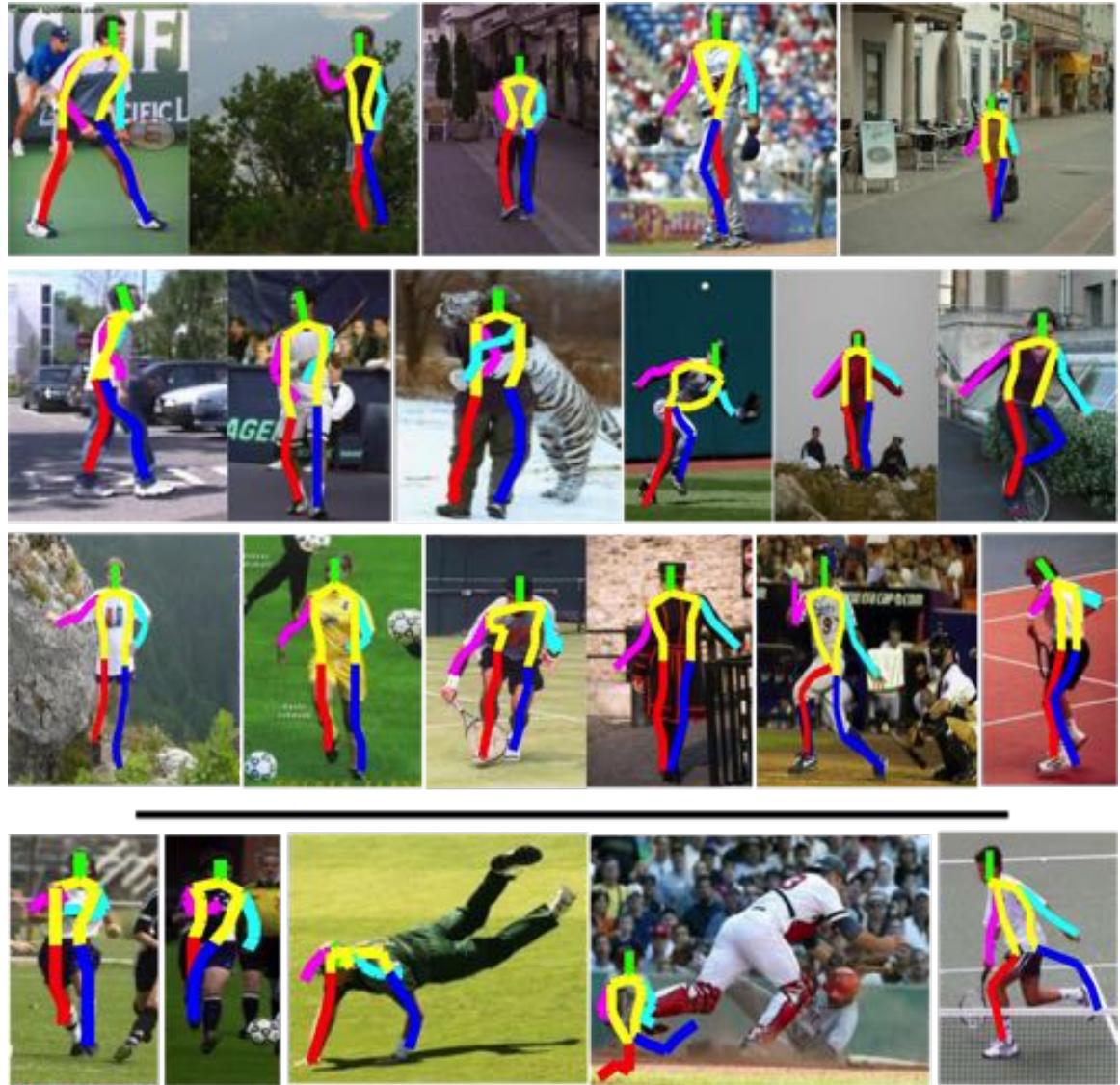


Figure 3.5: Sample results. **(Top)**: Examples in which [138] failed, but our 3-level model estimated poses correctly. **(Bottom)**: Some failure cases of our model.

Parse dataset

	Torso	Upper legs	Lower legs	Upper arms	Lower arms	Head	Total
Ramanan <i>et al.</i>	52.1	37.5	31.0	29.0	17.5	13.6	27.2
Yang <i>et al.</i>	82.9	69.0	63.9	55.1	35.4	77.6	60.7
Ours (26+10)	82.0	72.4	67.8	55.6	36.6	79.0	62.6
Ours (26+1)	85.6	71.7	65.6	57.1	36.6	80.4	62.8
Ours (26+10+1)	81.0	71.7	67.6	55.9	36.3	79.5	62.3
Pishchulin <i>et al.</i> *	88.8	77.3	67.1	53.7	36.1	73.7	63.1
Johnson <i>et al.</i> (2011)*	87.6	74.7	67.1	67.3	45.8	76.8	67.4

UIUC Sport dataset

Ramanan <i>et al.</i>	28.7	7.3	19.2	7.5	20.6	12.9	15.1
Wang <i>et al.</i>	75.3	49.2	39.5	25.2	11.2	47.5	37.3
Yang <i>et al.</i>	85.3	61.3	55.5	49.7	35.5	73.5	56.3
Ours (26+10)	85.4	61.6	57.9	49.1	34.8	72.9	56.4
Ours (26+1)	86.0	62.2	57.5	51.0	36.3	73.7	57.3
Ours (26+10+1)	86.2	61.2	55.7	49.9	35.9	73.8	56.5

Leeds Sport Pose dataset

Johnson <i>et al.</i> (2010)	78.1	65.75	58.8	47.4	32.85	62.9	55.1
Yang <i>et al.</i>	86.0	62.4	55.25	48.6	31.65	80.0	56.2
Ours (26+10)	86.2	64.0	57.3	47.6	31.9	79.8	56.6
Ours (26+1)	86.9	65.3	58.3	48.9	32.3	80.5	57.7
Ours (26+10+1)	86.2	64.1	57.4	47.9	31.9	80.0	56.9
Johnson <i>et al.</i> (2011)*	88.1	74.5	66.5	53.7	38.9	74.6	62.7

Table 3.1: Pose estimation results (PCP) on Parse (top), UIUC Sport (middle), and Leeds Sport (bottom) datasets. PCP scores are shown for each of six body parts and the combined score for all parts. All PCP scores here use criterion 1A (see text for details); for consistency, we re-computed the results from [138] to use this criterion, and for [100] we use the re-computed statistics reported in [137]. *Note that [93]

	Threshold	Yang <i>et al.</i>	Ours (26+10)	Ours (26+1)	Ours (26+10+1)
PCP (variant 1A)	0.2	33.4	34.5	34.5	34.3
	0.3	47.2	49.2	48.3	48.9
	0.4	56.0	57.6	56.5	57.3
	0.5	60.7	62.6	62.8	57.3
	0.6	64.4	65.9	66.9	65.7
	0.7	67.2	68.7	70.0	68.6
	0.8	69.7	71.3	72.0	70.9
	0.9	71.5	73.0	73.6	72.7
PCP 1B	0.5	56.0	58.5	59.3	59.5
PCP 2B	0.5	74.9	75.0	75.8	75.9

Table 3.2: Evaluation results on the Parse dataset under different definitions of Percentage of Correct Poses (PCP), using variants 1A, 1B and 2B which have all been used by different papers in the literature (see text for details). For variant 1A, we show results under different evaluation thresholds, where larger thresholds are more lenient in scoring part localizations.

CHAPTER 4

Segmentation-based Local Attribute Discovery

The previous chapter introduced *fully supervised* CRF models for modeling the human body as a hierarchical structure. In this chapter, we discuss our proposed method for modeling localized attributes for biological categories (*e.g.* birds, butterflies). Our approach treats the attribute discovery problem as a “region selection” problem, and use latent variables to model the region selection in each training image. Thus, the problem can be naturally described using a latent conditional random field, and we propose a *semi-supervised* training approach to learn the model parameters and infer the region selections at the same time. We first review existing literature in Section 4.1, and then introduce our method for modeling local attributes using latent CRFs in Section 4.2. We show that our discovered local attributes can be used in different fine-grained problems by conducting well-designed experiments (Section 4.3).

4.1 Related Work

Visual attributes describe characteristics of an object. For example, a fish has fins, gills, but no limbs; it lives in the water and is vertebrate. All these visual characteristics are called visual attributes. Visual attributes are both discriminative and semantically meaningful, providing semantic representations for objects, and can be

generalized across different categories. They are useful for fine-grained recognition, and support *zero-shot learning* [88]. Attributes can be either *global* or *local*. Global attributes measure visual characteristics (*e.g.* open, congest, indoor, cluttered, *etc.*) based on global features. Local attributes, on the other hand, describe local visual characteristics (*e.g.* red strips on the wing of a bird), which are especially useful when images are similar in terms of global features (*e.g.* fine-grained categories). Here we propose to use CRFs to model *localized attributes*, and design novel approaches for discovering local attributes automatically from training images.

4.1.1 Visual Attribute Discovery

Visual attributes for classification and recognition have received significant attention over the last few years. Much of this work assumes that the attribute vocabulary is defined ahead of time by a human expert [17, 69, 71, 136]. For example, Branson *et al.* [17] aim at improving existing object recognition techniques on fine-grained categories by using a human-computer interactive method. It repeatedly requests human users to answer yes/no type questions on predefined visual attributes of the unknown object instance, and refine the recognition result by incorporating human feedbacks. Kumar *et al.* [69] trains a list of classifiers for facial attributes (defined by domain experts and labeled by Amazon Mechanical Turk), and then treats the classifier scores on training face images as middle-level image representations.

An exception is the work of Parikh and Grauman [90] which proposes a system that discovers the vocabulary of attributes. Their system iteratively selects discriminative hyperplanes between two sets of images (corresponding to two different subsets of image classes) using global image features (*e.g.* color, GIST); it would be difficult

to apply this approach to find local attributes because of the exponential number of possible local regions in each image.

4.1.2 Modeling Local Attributes

We define local attributes to be local image regions that are discriminative for object recognition tasks (*e.g.* image categorization, image annotation, and zero-shot learning, *etc.*) and are semantically meaningful so that they can be shared across different object categories. A few papers have studied how to discover local attributes. Berg et al. [11] identify attributes by mining text and images from the web. Their approach is able to localize attributes and rank them based on visual characteristics, but these attributes are not necessarily discriminative and thus may not perform well for image classification; they also require a corpus of text and images, while we just need images. Wang and Forsyth [131] present a multiple instance learning framework for both local attributes and object classes, but they assume attribute labels for each image are given. In contrast, our approach does not require attribute labels; we discover these attributes automatically.

Work outside the context of attribute discovery has explored local discriminative regions for image classification. For example, Yao *et al.* [141] use a random forest with dense sampling to discover discriminative regions. The random forest combines thousands of region classifiers together, thus improving classification compared with using only low-level image features. In contrast, our approach treats each image as a bag of “regions” and applies multiple instance learning to find the most discriminative ones. We propose to enforce pairwise constraints on object geometry, and thus is more likely to find image regions that are both discriminative and semantically meaningful.

4.1.3 Part Discovery for Object Models

Related to our work on local attribute selection is the extensive literature on learning part-based object models for recognition (e.g. [39, 41, 109, 138]). These learning techniques usually look for highly distinctive parts – regions that are common within an object category but rare outside of it – and they make no attempt to ensure that the parts of the model actually correspond to meaningful semantic parts of an object. Local attribute discovery is similar in that we too seek distinctive image regions, but we would like these regions to be shared across categories and to have semantic meaning. Note that while most semantically meaningful local attributes are likely to correspond to semantic parts of objects [109], we view attributes as more general: an attribute is potentially any visual property that humans can precisely communicate or understand, even if it does not correspond to a traditionally-defined object part. For example “red-dot in center of wings” is a valid local attribute, even though there is not a single butterfly part that corresponds to it.

Maji and Shakhnarovich [82] propose an approach for “part discovery” on landmark images, by collecting pairs of user click annotations. They use exemplar SVMs [83] to find salient regions, while using click pair information to jointly infer object parts. Their method does not optimize classification accuracy, while our proposed approach learns a set of regions by maximizing the classification performance through a multiple instance learning framework.

4.1.4 Automatic Object Discovery

Our work is also related to the literature on automatic object discovery and unsupervised learning of object models [27, 61, 74]. Ferrari *et al.* [27] use a conditional random

field model for alternately localizing objects in images and learning object appearance models. Gunhee *et al.* [61] extract superpixels on each image in the training dataset to generate region hypothesis, and then identifies the statistically significant regions through iterative link analysis. Lee and Grauman [74] computes local feature descriptors (*e.g.* SIFT), and cluster images while localizing object foreground through feature correspondences between pairs of images. However, these methods aim to find objects that are *common* across images, while we are interested in finding *discriminative* local regions that will maximize classification performance.

Summary. We propose local attributes as a novel image representation for fine-grained object recognition. In this chapter, we design approaches for modeling and discovering local attributes on biological objects. We use a segmentation-based approach to generate region candidates. Region candidates are then treated as input to latent CRF framework, and human feedback is incorporated in order to choose candidates that are both discriminative and semantically meaningful. Our CRF model consists of multiple layers, where each layer uses a fully connected graph to model similarity or distances between image region appearances. Adjacent layers are connected using spatial overlap constraints. We use local attributes for fine-grained image classification and image annotation tasks.

4.2 Modeling Localized Attributes via Latent CRF

We first consider the problem of finding discriminative and machine-detectable visual attributes in a set of training images. We then describe a recommender system that finds candidates that are likely to be human-understandable and presents them to users for human verification and naming.

4.2.1 Latent CRF Model Formulation

We assume that each image in the training set has been annotated with a class label (e.g. species of bird) and object bounding box similar to [134, 141],¹ but that the set of possible attributes and the attribute labels for each image are unknown. We run a hierarchical segmentation algorithm on the images to produce regions at different scales, and assume that any attribute of interest corresponds to *at most* one region in each image. This assumption reduces the computational complexity and is reasonable because the hierarchical segmentation gives regions at many scales.

Formally, we are given a set of annotated training images $\mathcal{I} = (\mathcal{I}_1, \dots, \mathcal{I}_M)$, with each exemplar $\mathcal{I}_i = (I_i, y_i)$ consisting of an image I_i and a corresponding class label y_i . For now we assume a binary class label, $y_i \in \{+1, -1\}$; we will generalize this to multiple classes in Chapter 4.2.4. Each image I_i consists of a set of overlapping multi-scale regions produced by the hierarchical segmentation. To find a discriminative local attribute for these images, we look for regions in positive images, one per image, that are similar to one another (in terms of appearance, scale and location) but not similar to regions in negative images. We formulate this task as an inference problem on a latent conditional random field (L-CRF) [95], the parameters of which we learn via a discriminative max-margin framework in the next section.

First consider finding a single attribute k for the training set \mathcal{I} . For each image we want to select a single region $l_i^k \in I_i$ such that the selected regions in the positive images have similar appearances to one another, but are different from those on the negative side. We denote the labeling for the entire training set as $L_k = (l_1^k, \dots, l_M^k)$,

¹ [134] in fact requires the user to interactively segment the object out.

and then formulate this task in terms of minimizing an energy function [26],

$$E(L_k|\mathcal{I}) = \sum_{i=1}^M \phi_k(l_i^k|\mathcal{I}_i) + \sum_{i=1}^M \sum_{j=1}^M \psi_k(l_i^k, l_j^k|\mathcal{I}_i, \mathcal{I}_j), \quad (4.1)$$

where $\phi_k(l_i^k|\mathcal{I}_i)$ measures the preference of a discriminative classifier trained on the selected regions to predict the category labels, while $\psi_k(l_i^k, l_j^k|\mathcal{I}_i, \mathcal{I}_j)$ measures pairwise similarities and differences between the selected regions. In particular, we define the unary term as,

$$\phi_k(l_i^k|\mathcal{I}_i) = -y_i w_k^T \cdot f(l_i^k) \quad (4.2)$$

where $f(l_i^k)$ denotes a vector of visual features for region l_i^k and w_k is a weight vector. We will use several different types of visual features, as discussed in Section 4.3; for now we just assume that there are d feature types that are concatenated together into a single n -dimensional vector. The weights are learned as an SVM on the latent regions from positive and negative images (discussed in Chapter 4.2.2).

The pairwise consistency term is given by,

$$\psi_k(l_i^k, l_j^k|\mathcal{I}_i, \mathcal{I}_j) = \begin{cases} \vec{\alpha}_k^+ \cdot D(f(l_i^k), f(l_j^k)) + \beta_k^+ & \text{if } y_i, y_j = +1 \\ \beta_k^0 & \text{if } y_i, y_j = -1 \\ \vec{\alpha}_k^- \cdot D(f(l_i^k), f(l_j^k)) + \beta_k^- & \text{otherwise,} \end{cases} \quad (4.3)$$

where $D(\cdot, \cdot)$ is a function $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^d$ that given two feature vectors computes a distance for each feature type, $\vec{\alpha}_k^-$ and $\vec{\alpha}_k^+$ are weight vectors, and $\vec{\beta}_k = (\beta_k^-, \beta_k^+, \beta_k^0)$ are constant bias terms (all learned in Chapter 4.2.2). This pairwise energy function encourages similarity among regions in positive images and dissimilarity between positive and negative regions. We allow negative regions to be different from one another since they serve only as negative exemplars; thus we use a constant β_k^0 as the edge potential between negative images in lieu of a similarity constraint.

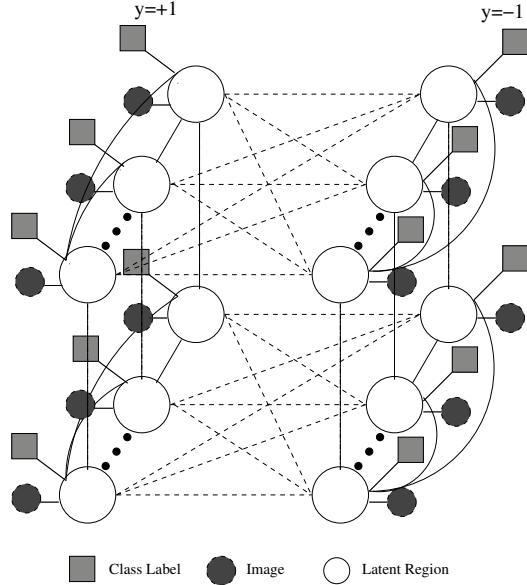


Figure 4.1: Our L-CRF model for one active split with $K = 2$ attributes, where white circles represent latent region variables (l_i^k), shaded circles represent observed image features (I_i), and squares represent observed image class labels (y_i).

The energy function presented in equation (4.1) defines a first-order Conditional Random Field (CRF) graphical model. Each vertex of the model corresponds to an image, and the inference problem involves choosing one of the regions of each image to be part of the attribute. Edges between nodes reflect pairwise constraints across images, where here we use a fully-connected graphical model such that there is a constraint between every image pair.

The single attribute candidate identified by the L-CRF may not necessarily be semantically meaningful, but there may be other candidates that can discriminate between the two categories that are semantically meaningful. To increase the chances of finding these, we wish to identify multiple candidate attributes. We generalize the above approach to select $K \geq 2$ attributes for a given split by introducing an energy function that sums equation (4.1) over all K attributes. We encourage the CRF to

find a set of *diverse* attributes by adding an additional term that discourages spatial overlap among selected regions,

$$E(\mathcal{L}|\mathcal{I}) = \sum_{k=1}^K E(L_k|\mathcal{I}) + \sum_{i=1}^M \sum_{k,k'} \delta(l_i^k, l_i^{k'}|\mathcal{I}_i), \quad (4.4)$$

where $\mathcal{L} = (L_1, \dots, L_K)$ denotes the latent region variables, δ measures spatial overlap between two regions,

$$\delta(l_i^k, l_i^{k'}|\mathcal{I}_i) = \sigma \cdot \frac{\text{area}(l_i^k \cap l_i^{k'})}{\text{area}(l_i^k \cup l_i^{k'})}, \quad (4.5)$$

and $\sigma \geq 0$ is a scalar which is also learned in the next section. This term is needed because we want a diverse set of candidates; without this constraint, the CRF may find a set of very similar candidates because those are most discriminative. Intuitively, $\delta(\cdot)$ penalizes the total amount of overlap between regions selected as attributes. Minimizing the energy in equation (4.4) also corresponds to an inference problem on a CRF; one can visualize the CRF as a three-dimensional graph with K layers, each corresponding to a single attribute, with the edges in each layer enforcing the pairwise consistency constraints ψ among training images and the edges between layers enforcing the anti-overlap constraints δ . The vertices within each layer form a fully-connected subgraph, as do the vertices across layers corresponding to the same image. Figure 4.1 illustrates the CRF for the case of two attributes.

4.2.2 Training

There are two sets of model parameters that must be learned: the weight vectors w_k in the unary potential of the CRF, and the parameters of the pairwise potentials ψ and δ which we can concatenate into a single vector $\vec{\alpha}$,

$$\vec{\alpha} = (\vec{\alpha}_1^-, \dots, \vec{\alpha}_K^-, \vec{\alpha}_1^+, \dots, \vec{\alpha}_K^+, \vec{\beta}_1, \dots, \vec{\beta}_K, \sigma).$$

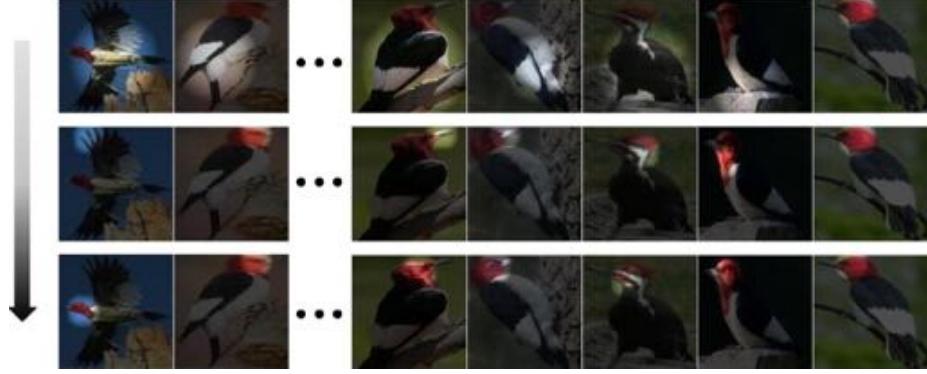


Figure 4.2: Sample latent region evolution on an active split, across three iterations (top to bottom). The latent region selected by the CRF on each positive image in each iteration is shown. These variables converged after three iterations to roughly correspond to the bird’s red head. Best viewed on-screen and in color.

We could easily learn these parameters if we knew the correct values for the latent variables \mathcal{L} , and we could perform CRF inference to estimate the values of the latent variables if we knew the parameters. To solve for both, we take an iterative approach in the style of Expectation-Maximization. We initialize the latent variables \mathcal{L} to random values. We then estimate w_k in equation (4.2) for each k by learning a linear SVM on the regions in L_k , using regions in positive images as positive exemplars and regions in negative images as negative exemplars. Holding w_k fixed, we then estimate the pairwise parameters $\vec{\alpha}$ via a standard latent structural SVM (LSSVM) framework,

$$\min_{\vec{\alpha}} \lambda \|\vec{\alpha}\|^2 + \xi, \text{ such that } \forall \tilde{l}_i \in I_i, \forall \tilde{y}_i \in \{+1, -1\}, \quad (4.6)$$

$$E(\{\tilde{l}_i\} | \{(I_i, \tilde{y}_i)\}) - \min_{\mathcal{L}^*} E(\mathcal{L}^* | \mathcal{I}) \geq \Delta(\{\tilde{y}_i\}, \{y_i\}) - \xi$$

where $\xi \geq 0$ is a slack variable and the loss function is defined as the number of mislabeled images,

$$\Delta(\{\tilde{y}_i\}, \{y_i\}) = \sum_i \mathbb{1}_{\tilde{y}_i \neq y_i}.$$

We solve this quadratic programming problem using CVX [46]. Since there are an exponential number of constraints in equation (4.6), we follow existing work on structured SVMs [125] and find the most violated constraints, in this case using tree-reweighted message passing (TRW) [65] on the CRF. Once the CRF parameters have been learned, we hold them fixed and estimate new values for the latent variables \mathcal{L} by performing inference using TRW. This process of alternating between estimating CRF parameters and latent variable values usually takes 3 to 5 iterations to converge (Figure 4.2). In our experiments we use $K = 5$. This takes about 3-5 minutes on a 3.0GHz server.

The above formulation was inspired by Multiple Instance CRFs [26, 27], but with some important differences (besides application domain). Our formulation is a standard latent structural SVM in which we minimize classification error, whereas the loss function in [26] is based on incorrect instance selections. Their unary potential is pre-trained instead of being updated iteratively. Finally, our model simultaneously discovers multiple discriminative candidate attributes (instances).

4.2.3 Attribute Detection

To detect attributes in a new image I_t , we simply add I_t to the L-CRF as an additional node, fixing the values of the latent variables for the training image nodes. We perform CRF inference on this new graph to estimate both the class label \hat{y}_t and its corresponding region label $\hat{l}_t \in I_t$. If $\hat{y}_t = 1$, then we report a successful detection and return \hat{l}_t , and otherwise report that I_t does not have this attribute. Note that this inference is exact and can be done in linear time.

4.2.4 Active Attribute Discovery

Having shown how to automatically discover attributes for images labeled with one of two classes (positive or negative), we now describe how to discover attributes in a dataset with multiple category labels, $y_i \in \{1, \dots, N\}$. We would like to discover an attribute vocabulary that collectively discriminates well among all categories. It is intractable to consider all $O(N^2)$ possible binary splits of the labels, so we use an iterative approach with a greedy heuristic to try to actively prioritize the order in which splits are considered. At each iteration, we identify the two categories that are most similar in terms of the presence and absence of attributes discovered so far. We use these two categories to define an active split, and find a set of discriminative attributes for this split using the procedure above. We then add these to our attribute set, and repeat the process.

4.2.5 Identifying Semantic Attributes

The approach we described in previous sections is able to discover K candidate discriminative local attributes for each active split, but not all of these will be meaningful at a semantic level. We now describe how to introduce a minimal amount of human feedback at each iteration of the discovery process in order to identify candidates that are discriminative *and* meaningful. Of the K candidates, we first identify the candidate that is most discriminative – *i.e.* that increases the performance of a nearest neighbor classifier the most on held out validation data. We present this candidate to a human user by displaying a subset of the positive training images from the corresponding active split marked with the hypothesized attribute regions determined by the L-CRF (see Figure 4.7). If the user finds the candidate meaningful (and thus

provides it with a name), it is added to our vocabulary of attributes. If not, that candidate is rejected, and we select the second most discriminative candidate in our pool of K candidates. If none of the candidates is judged to be meaningful, no attribute is added to our pool, and we identify the second most confusing pair of categories as our next active split.

In order to reduce user response time we propose an attribute recommender system that automatically prioritizes candidates before presenting them to a user. It uses past user feedback to predict whether the user is likely to find a new candidate attribute meaningful. Our recommender system is based on the hypothesis that users judge the meaningfulness of an attribute by whether it is located on consistent parts of the object across the positive instances (e.g. if the regions in the images correspond to the same nameable part of a bird).

We use a simple approach to measure the spatial consistency of an attribute with respect to the object (illustrated in Figure 4.3). At each active split, we train our attribute recommendation system using all attribute candidates that have been presented to human users so far, with accepted ones as positive exemplars and rejected ones as negative exemplars. Note that the L-CRF model (Section 4.2.1) can also encourage spatial consistency among training images (as we will see in Section 4.3); however those constraints are only pairwise, whereas the features here are higher-order statistics capturing the set of regions as a whole. Our recommender system is related to the nameability model of [90], but that model was restricted to global image-level attributes, whereas we model whether a group of local regions is likely to be deemed consistent and hence meaningful by a human.

4.3 Experiments

We now test our proposed approach to local attribute discovery. We use data from two recent datasets with fine-grained category labels: a subset of the Caltech-UCSD Birds-200-2011 (CUB) [130] dataset containing about 1,500 images of 25 categories of birds, and the Leeds Butterfly (LB) [134] dataset, which contains 832 images from 10 categories of butterflies. We apply a hierarchical segmentation algorithm [6] on each image to generate regions, and filter out background regions by applying GrabCut [104] using the ground truth bounding boxes provided by the datasets (for LB, using a bounding box around the GT segmentation mask in order to be consistent with CUB). Most images contain about 100 – 150 such regions.

For the region appearance features $f(\cdot)$ in equations (4.2) and (4.3), we combine a color feature (color histogram with 8 bins per RGB channel), a contour feature (gPb [6]), a size feature (region area and boundary length), a shape feature (an 8×8 subsampled binary mask), and spatial location (absolute pixel location of the centroid). For the distance function $D(\cdot, \cdot)$ in equation (4.3), we compute χ^2 distances for the color, contour, size, and shape features, and Euclidean distance for the spatial location feature. During learning, we constrain the weights of $\vec{\alpha}_k^+$ and $\vec{\alpha}_k^-$ corresponding to the spatial location feature to be positive to encourage candidates to appear at consistent locations. The weights in $\vec{\alpha}_k^+$ and $\vec{\alpha}_k^-$ corresponding to other feature types are constrained to be nonnegative and nonpositive, respectively, to encourage visual similarity among regions on the positive side of an active split and dissimilarity for regions on opposite sides. The bias terms $\vec{\beta}_k$ are not constrained.

Exhaustive data collection for all 200 categories in the CUB birds dataset is not

feasible because it would require about a million user responses. So we conduct systematic experiments on three subsets of CUB: ten randomly-selected categories, the ten hardest categories (defined as the 10 categories for which a linear SVM classifier using global color and gist features exhibits the worst classification performance), and five categories consisting of different species of warblers (to test performance on very fine-grained category differences). Each dataset is split into training, validation, and testing subsets. For CUB these subsets are one-half, one-quarter, and one-quarter of the images, respectively, while for LB each subset is one-third.

We use Amazon’s Mechanical Turk to run our human interaction experiments. For each dataset, we generate an exhaustive list of all possible active splits, and use an “offline” collection approach [90] to conduct systematic experiments using data from real users without needing a live user-in-the-loop. We present attribute visualizations by superimposing on each latent region a “spotlight” consisting of a 2-D Gaussian whose mean is the region centroid and whose standard deviation is proportional to its size (and including a red outline for the butterfly images to enhance contrast). We do this to “blur” the precise boundaries of the selected regions, since they are an artifact of the choice of segmentation algorithm and are not important. We present each candidate attribute to 10 subjects, each of whom is asked to name the highlighted region (*e.g.* belly) and give a descriptive word (*e.g.* white). See Figure 4.7. We also ask the subjects to rate their confidence on a scale from 1 (“no idea”) to 4 (“very confident”); candidates with mean score above 3 across users are declared to be semantically meaningful.

4.3.1 Attribute-based Image Classification

We now use our discovered attributes for image classification. We detect attributes in validation images and learn linear SVM and nearest-neighbor classifiers, and then detect attributes and measure performance on the test subset. We represent each image as a binary feature vector indicating which attributes were detected. Each category is represented as the average feature vector of its training images. The nearest-neighbor classifier works by assigning the test image to the category with the closest feature vector (similar to [71]). The SVM classifier is trained directly on the above binary features using cross-validation to choose parameters.

Figure 4.4 presents classification results on CUB birds and LB butterflies, comparing the attribute vocabularies produced by our **Proposed** technique with two baselines that are representative of existing approaches in the literature. These results do not include the recommender system; we evaluate that separately. **Hand-listed** uses the expert-generated attributes provided with the datasets. These are guaranteed to be semantically meaningful but may not be discriminative. **Discriminative only**, at the other extreme, greedily finds the most discriminative candidates and hopes for them to be semantic. At each iteration (i.e. active split) among K candidates, it picks the one that provides the biggest boost in classification performance on a held-out validation set. Candidates that are not semantic (and hence not attributes) are dropped in a post-process. As reference, we also show performance if *all* discriminative candidates are used (semantic or not). This **Upper bound** performance depicts the sacrifice in performance one makes in return for semantically meaningful attributes.

We see in Figure 4.4 that our proposed method performs significantly better than either the hand-listed attributes or discriminative only baselines. These conclusions are stable across all of the datasets and across both SVM and nearest neighbor classifiers. Hand-listed can be viewed as a semantic-only baseline (since the human experts likely ignored machine-detectability while selecting the attributes) and discriminative only can be thought of as focusing only on discriminative power and then addressing semantics after-the-fact. Our proposed approach that balances both these aspects performs better than either one.

We also evaluate our recommender system as shown in Figure 4.5. We see that using the recommender allows us to gather more attributes and achieve higher accuracy for a fixed number of user iterations. The recommender thus allows our system to reduce the human effort involved in the discovery process, without sacrificing discriminability.

4.3.2 Image-to-text Generation

Through the interaction with users, our process generates names for each of our discovered attributes; Figure 4.7 shows some examples. We can use these names to produce textual annotations for unseen images. We list the name of the attribute with maximum detection score among all candidates detected on the detected region. Sample annotation results are shown in Figure 5.5 using the system trained on the 10 random categories subset of the CUB birds dataset. Note that some of these images belong to categories that our system has never seen before and were not part of the discovery process at all. Being able to meaningfully annotate unseen images demonstrates the ability of our system to find human-understandable and machine-

detectable attributes that can be shared across categories.

We can use the fact that several of the attribute names provided by our users match the hand-selected attribute names given in the CUB dataset to evaluate the detection accuracy of our attributes.² Some attributes that have high accuracy include blue wing (71.2%), red eye (83.3%), yellow belly (72.5%), red forehead (75.7%), and white nape (71.7%). Others are less accurate: spotted wing (67.2%), orange leg (60.3%), white crown (61.7%). In computing these accuracies, we use all positive examples that have the attribute, and randomly sample the same number of negative examples. We also observe that our approach is able to discover some interesting attributes that were not provided in the hand-selected annotations, including “sharp bill”, and “long/thin leg.”

4.4 Summary

We have presented a novel approach for discovering localized attributes for fine-grained recognition tasks. Our system generates local attributes that are both discriminative and human understandable, while keeping human effort to a minimum. Our approach intelligently selects active splits among training images, looking for the most discriminative local information. Involving a human in the loop, it identifies semantically meaningful attributes. We propose a recommender system that prioritizes likely to be meaningful candidate attributes, thus saving user time. Results on different datasets show the advantages of our novel local attribute discovery model as compared to existing approaches to determining an attribute vocabulary. In fu-

²The hand-selected annotations are not used in our discovery process; we use them only as ground-truth for measuring detection accuracy.

ture work, we would like to find links between local attributes and object models, in order to bring object detection into the loop of discovering localized attributes, such that both tasks benefit from each other. We would also like to study how to better incorporate human interactions into recognition techniques.

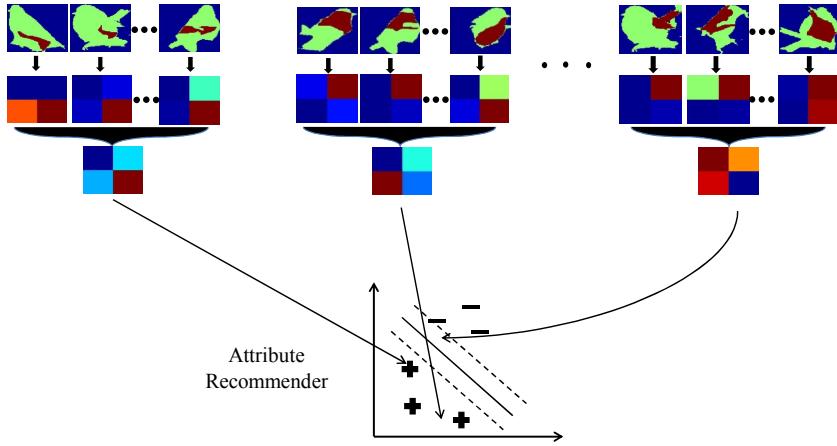


Figure 4.3: Illustration of the recommender system. A background mask is estimated for each image (top row, blue) using GrabCut [104]. The foreground mask is divided into a 2×2 grid. For each attribute region in the positive images (top row, dark red), we measure its spatial overlap with each grid cell shown in the second row, where degree of overlap is represented by colors ranging from dark blue (no overlap) to dark red (high overlap). Averaging these features across all positive images in the split (third row) gives a representation for the candidate attribute. We add two extra dimensions containing the mean and standard deviation of the areas of the selected regions, creating a 6-D feature vector to train a classifier. This is a positive exemplar if the candidate is deemed meaningful by the user, and negative otherwise.

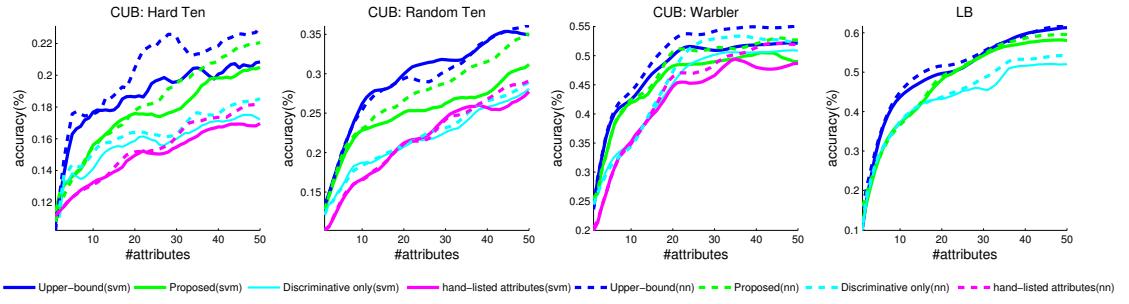


Figure 4.4: Image classification performance on four datasets with SVM and nearest neighbor (nn) classifiers, and using four different attribute discovery strategies: attributes selected by a purely discriminative criterion (**Upper bound**), a purely discriminative criterion from which non-meaningful candidates are removed by post-processing (**Discriminative only**), attributes produced by a human expert (**Hand-listed**), and our proposed approach which includes human interaction (**Proposed**). Classification statistics are averaged over 10 trials. The LB dataset does not include ground truth attributes so we do not evaluate hand-listed attributes on this dataset.

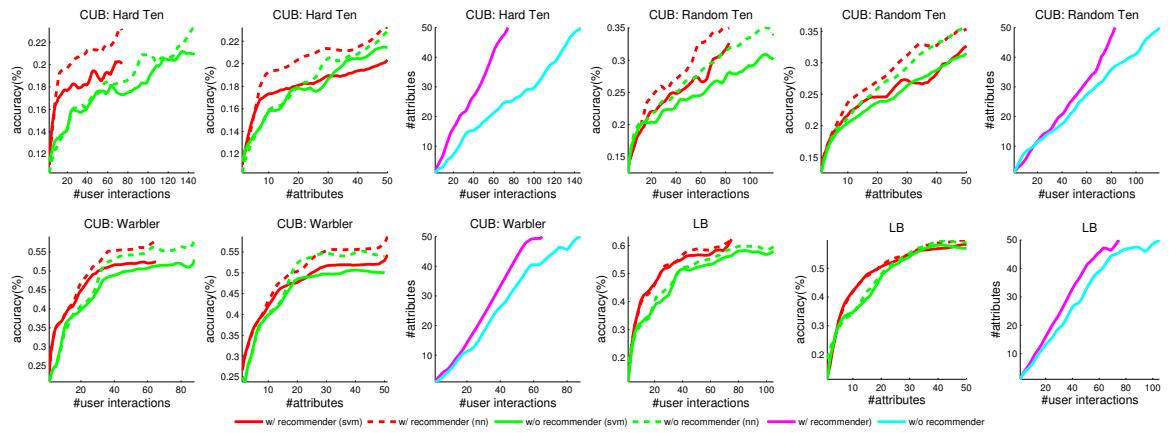


Figure 4.5: Classification performance of the Proposed system with and without using the recommender.



Figure 4.6: Examples of automatic text generation.

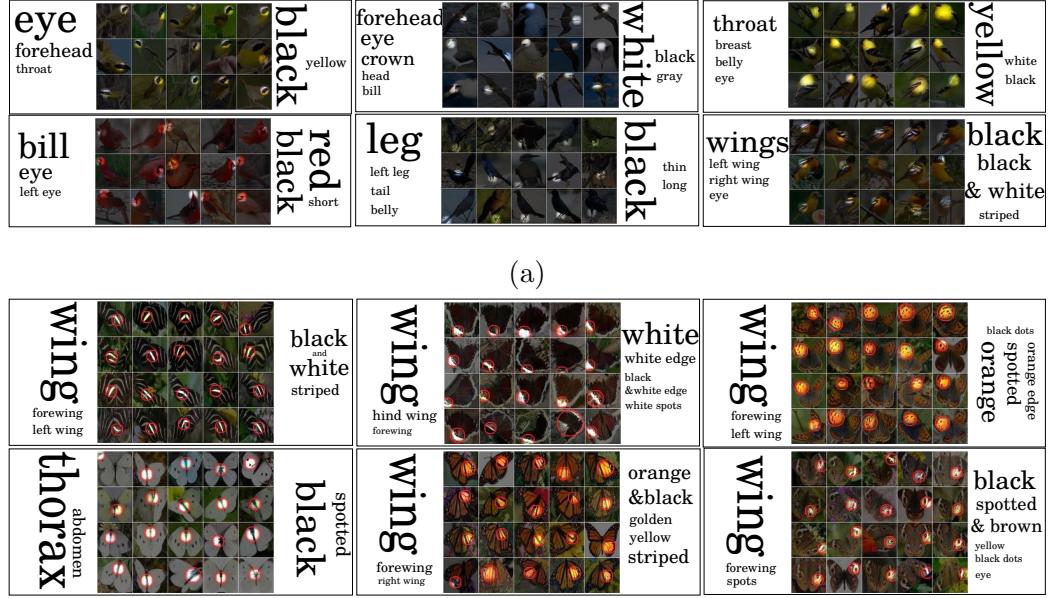


Figure 4.7: Some local attributes discovered by our approach, along with the semantic attribute names provided by users (where font size is proportional to number of users reporting that name), for (a) CUB birds, and (b) LB butterflies. Best viewed on-screen and in color.

CHAPTER 5

Detection-based Local Attribute Discovery

In this chapter, we look at man-made fine-grained categories like vehicles, and try to model local attributes for the given vehicle training images. Segmentation methods (*e.g.* GrabCut) often fail to find the correct object boundaries on vehicle images due to appearance inconsistency (*e.g.* completely different appearance between the vehicle body and the wheels). For example, on a blue car, segmentation-based method fails because the entire car except the wheels is just one big blue region, so important local details are lost. Also, GrabCut algorithm often misses the wheels in the foreground mask, which prevents us from removing background noisy regions. Vehicles are much more rigid than certain biological objects like birds or butterflies, and vehicle photos are often taken from canonical viewpoint angles. Taking into account the rigidity of such object categories helps modeling the visual appearance, and explicitly modeling the viewpoint angles helps finding robust correspondence between image regions even if the photos are taken from different viewpoints.

We propose to use multiple instance SVM to model localized attributes for vehicles by incorporating viewpoint information, and instead of using segmentation-based methods, we use detection-based approaches to generate image region hypotheses. We discuss related work in Section 5.1, and then describe our method in Section 5.2,

and report results on two vehicle image datasets in Section 5.3.

5.1 Related Work

We have conducted an extensive literature review of visual attribute discovery in Section 4.1. Here we focus on discussing related work on modeling viewpoints together with local attributes for man-made objects. Our proposed approach uses latent CRFs to discover local attributes of vehicles by extending multiple instance SVM models with pairwise constraints on viewpoint angles.

As with modeling local attributes for biological objects, we treat the attribute discovery problem as an image region selection problem for man-made objects. These two different types of objects exhibit different properties in their visual appearances. For example, man-made objects (like vehicles) are typically more rigid, while biological objects are deformable. The visual patterns on man-made objects might not be coherent (*e.g.* wheels look completely different with other parts of a vehicle), meaning that segmentation-based approach for generating local attribute candidates might not work well on this type of object categories. Thus, different methods for modeling local attributes must be designed separately. For example, we apply hierarchical segmentation to generate region candidates for birds and butterflies when modeling local attributes. Here we pick vehicles as a representative man-made object category in our experiments, and use a detection-based method to generate region candidates for vehicle categories. We design an approach for modeling both the local attributes and the viewpoint angles at the same time, and show that modeling viewpoint angles explicitly improves the performance of attribute-based recognition methods.

A number of recent papers on attribute discovery are relevant to our proposed

approach, but all have important differences. Gu and Ren [48] learn viewpoint angles and vehicle classifiers at the same time, but they do not consider requiring these models to be semantically-meaningful (at either global or local levels). Our approach in Chapter 4 learns local attributes in the context of animal species recognition, but it does not consider multiple viewpoints, and our method relies on multiple features (contour, shape, color, etc.) with carefully learned weights for each feature channel. In contrast, we model local attributes and viewpoint angles together in a single framework, such that the local attribute discovery helps to model the viewpoint angles, and vice versa. Perhaps the most relevant work to ours is that of Sharma *et al.* [111], which automatically mines a collection of parts and corresponding templates for recognizing human attributes and actions. However this method assumes that the attribute labels for training images are given, while we assume only category labels are available, and we want to model local attributes and viewpoint angles at the same time.

5.2 Modeling Localized Attributes via Multiple Instance SVM with Constraints

We propose a method for automatically discovering discriminative local attributes for vehicle categories. The discovered collection of local attributes serves as a new image representation, which improves vehicle classification performance when fused together with low-level features using the method in [28]. Meanwhile, the discovered attributes can be assigned semantic meanings, allowing novel cross-modal applications such as querying vehicles using textual descriptions. We first describe a technique based on the classic Multiple Instance SVM (MI-SVM) model (Section 5.2.1), and then we

extend it by introducing pairwise constraints (Section 5.2.2). Finally we describe how to learn the latent viewpoint angles when these annotations are not available (Section 5.2.3).

5.2.1 MI-SVMs for Attribute Discovery

Multiple Instance Learning (MIL) is a form of semi-supervised learning in which training instances are grouped into *bags*. The ground-truth labels of the individual instances are unknown, but each bag has a label that is positive if *at least* one instance in the bag is positive, and negative if *all* its instances are negative. Suppose we have a set of bags $\{x_I\}$. The standard Multiple Instance SVM (MI-SVM) [4] is formulated as an optimization,

$$\begin{aligned} & \min_{\mathbf{w}, \xi, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_I \xi_I \\ & \text{s.t. } Y_I \cdot \max_i (\mathbf{w} \cdot x_I^i + b) \geq 1 - \xi_I, \end{aligned} \quad (5.1)$$

where \mathbf{w} is a feature weight vector, b is a scalar bias, ξ_I is a slack variable corresponding to training bag x_I , x_I^i is the i th training instance of bag x_I , and Y_I is the ground truth label (+1 or -1) of x_I . Intuitively, this is the classic SVM max-margin framework with an additional (soft) constraint that all instances in the negative bags should be classified as negative, and at least one instance in each positive bag should be classified as positive.

Our goal is to find local image regions across the training set that are discriminative — that occur often in one vehicle category but not in another. We can apply the MI-SVM framework to this problem in the following way. Choose a pair of vehicle categories, calling one positive and one negative. We think of each image as

a bag with a positive or negative label depending on its category, and then sample many patches from each image to produce instances for each bag. We then solve equation (5.2), which produces a weight vector but also implicitly chooses positive instances, and these can be viewed as the set of discriminative regions that we are interested in. We can repeat this process for many pairs of categories to produce a set of candidate attributes.

5.2.2 MI-SVMs with Constraints

A problem with the above approach is that discovered regions may not correspond to the same part of the vehicle, and thus may not have semantic meaning, and also that more than one region may be selected in each positive image. To address these problems, we add constraints to encourage spatial consistency, requiring regions to occur in roughly the same position on the vehicle by adding pairwise spatial constraints among instances in the positive bag. But since viewpoints vary across images, we must explicitly model viewpoint in order to compare spatial positions.

Our model. Let $v_I \in \mathcal{V}$ denote the viewpoint label of image (bag) I , where we assume that \mathcal{V} is a small set of possible discrete viewpoints. For now we assume the viewpoint labels are given; we discuss how to handle unknown viewpoint labels in Section 5.2.3. We formulate the attribute discovery problem using MI-SVMs, with additional pairwise spatial constraints among positive instances that encourage the spatial consistency property, as illustrated in Figure 5.1. Suppose that we knew which instance in each positive bag should be part of the attribute, and denote this region x_I^* for bag I . Then we could solve a separate MI-SVM problem for each individual

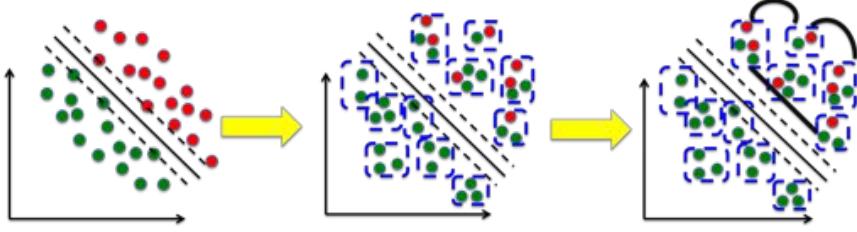


Figure 5.1: Visualization of SVM models: standard SVM (*left*), standard MI-SVM (*middle*), and our MI-SVM with constraints (*right*) between instances in each positive bag. For recognizing vehicles given their viewpoint angles, we define the constraints such that two selected region candidates must come from consistent locations on the vehicles.

viewpoint $v \in \mathcal{V}$,

$$\begin{aligned} & \min_{\{\mathbf{w}^{(v)}, \xi, b^{(v)}\}} \frac{1}{2} \|\mathbf{w}^{(v)}\|^2 + C^{(v)} \sum_{I \in \mathcal{I}^{(v)}} \xi_I \\ & \text{s.t. } \forall I \in \mathcal{I}^{(v)}, \quad Y_I \cdot (\mathbf{w}^{(v)} \cdot x_I^* + b^{(v)}) \geq 1 - \xi_I, \end{aligned} \quad (5.2)$$

where $\mathcal{I}^{(v)}$ is the set of images having viewpoint label v , i.e. $\mathcal{I}^{(v)} = \{I | v_I = v\}$.

Now suppose the weight vectors and biases for each viewpoint were already known, so that we need to estimate the x_I^* for each bag I . We want to do this in a way that encourages spatial consistency. We pose this problem as inference on a Conditional Random Field (CRF) [70]. Let l_I be a scalar variable which takes a value from the region indices in image I . We define an energy function to measure the compatibility of a given assignment of variables to l_I ,

$$E(\{l_I\} | \{v_I\}) = \sum_I \phi(l_I | v_I) + \sum_{I,J} \delta(l_I, l_J | v_I, v_J), \quad (5.3)$$

where the first set of terms in the summation measures how well the selected regions

are modeled by the MI-SVM,

$$\phi(l_I|v_I) = -(\mathbf{w}^{(v_I)} \cdot x_I^{l_I} + b^{(v_I)}),$$

and the pairwise terms encourage positive regions to be at about the same spatial position on the car. If the viewpoint labels between two images are the same, then measuring this distance is a simple matter of comparing image coordinates. If the labels are different, then we need to apply a transformation so that the two coordinate systems are comparable. In particular, our pairwise function is ¹,

$$\delta(l_I, l_J|v_I, v_J) = \begin{cases} \|\mu(l_I) - \mu(l_J)\|^2, & \text{if } v_I = v_J \\ \|H_{v_I}^{v_J} \mu(l_I) - \mu(l_J)\|^2, & \text{if } v_I \neq v_J, \end{cases}$$

where $\mu(l_I)$ denotes the spatial position of region l_I relative to the vehicle center, and $H_{v_I}^{v_J}$ is a homography matrix. We estimate the homography between two viewpoints by extracting SIFT features [79] from the training images having each viewpoint and running RANSAC [43] on feature correspondences. Finally, to estimate the best region x_I^* for each image I , we minimize equation (5.3) through CRF inference,

$$\{x_I^*\} = \arg \min_{\{l_I\}} E(\{l_I\}|\{v_I\}). \quad (5.4)$$

Of course, in our problem we know neither the SVM parameters or the region selections. We thus solve these iteratively, first finding the weights and biases in equation (5.2) by holding the region variables fixed, and then solve for the region variables in equation (5.4) while holding the SVM parameters fixed. The result is a collection of region selections for all positive training images.

¹In order to minimize the computational cost, our distance function is not defined to be symmetric when viewpoint angles of two regions are different. One possible solution is to compute two transformation matrices $H_{v_I}^{v_J}$ and $H_{v_J}^{v_I}$, and take the average of two distances from both directions.

Generating regions. We have not yet addressed how to generate the instances within each bag. Although we could randomly sample patches, in practice this creates many irrelevant regions. We thus use an approach similar to [82], applying a pre-trained deformable part-based model car detector [39] on the training images to produce multiple detections with part locations. We then sample from the part detection bounding boxes to generate region candidates. This is faster than the hierarchical segmentation in [32] and produces regions that are more likely to be on the vehicles.

Generating multiple attributes. The above procedure can be used to find the best attribute for a given pair of categories, but in practice we want to generate multiple attribute hypotheses. To do this, we first find the best attribute by solving for $\{x_I^*\}$ using the iterative procedure described above. To find a second attribute, we modify the unary term of equation (5.3) so that a large constant penalty is paid for selecting an l_I that was chosen as part of the earlier attribute. In our experiments, we repeat this procedure 5 times to produce 5 attribute candidates per pair of categories.

5.2.3 Recovering Viewpoint Angles

We now consider the case in which the viewpoint labels $\{v_I\}$ are not available ahead of time, so we need to estimate the viewpoint label of each image in addition to the local attributes. We first initialize the viewpoint labels with K -means clustering using global image gradient features (*e.g.* dense SIFT [79]) with $K = |\mathcal{V}|$. Then, after each new attribute is discovered, we update the viewpoint label of each image. To do this, we apply the attribute detectors that have been found so far across all viewpoint angles on the discovered region, choose the best detector, and assign that viewpoint to the region. For all of the discovered regions in an image, we collect all

such viewpoint predictions, and use these to vote for the viewpoint of the image.

5.3 Experiments

We consider two datasets in our experiments: **Stanford cars** [115] with 14 car categories (and 68 training and 34 test images in each category); and **INRIA vehicles** [68] with 29 categories and a total of 10,000 images equally split into training and test sets. There is viewpoint angle bias in both datasets (*e.g.* images in Stanford cars are mostly from 45° and 135°). In Stanford cars, each category consists of car images of the same make, model and year, and bounding box annotations and 8 discrete viewpoint labels are also provided. The INRIA dataset does not have viewpoint labels, and the images in a category are only guaranteed to be the same make and model, not necessarily from the same year.

On both datasets, we extract dense SIFT and color histogram features for each region candidate and compute corresponding Fisher vectors using 32D Gaussian mixture models. We also extract these features on the whole image with a three-layer spatial pyramid as a baseline. Note that [32] uses hierarchical segmentation to generate image region candidates, and different types of features (shape, contour, color, gradient, etc) are extracted from the segments. In our case, since the sampled regions are all rectangles, we only use gradient and color features.

5.3.1 Single Attributes

To validate the region pooling parameters and test how our sampling strategy is related to accuracy, we test *single region performance*, where we train multi-class linear SVM classifiers on *all* image region features, using category labels as training labels

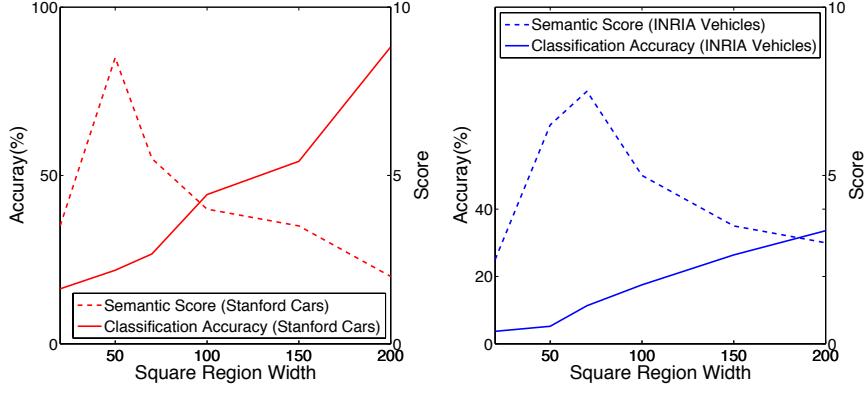


Figure 5.2: Relationship between region size and (*solid line*) the performance of image classification using single regions, and (*dashed line*) the semantic meaningfulness as judged by humans, for Stanford (*left*) and INRIA (*right*).

for classifying vehicle categories. We observe that the performance for classifying single regions decreases as region size decreases (Figure 5.2). This make intuitive sense because discriminative information is lost when the image is broken into small pieces. For example, it is difficult to tell the difference between two vehicle categories if only parts of the wheels are given.

We also wanted to measure the relationship between region size and whether or not a region is semantically meaningful. To do this, we conducted a simple experiment on Mechanical Turk where image regions of varying sizes were shown, and users were asked to rate (on a scale of 1-10) whether the region corresponded to a meaningful part of the vehicle or not. Results are also shown in Figure 5.2. We found that semantic meaning suffers if regions are too big or too small: too small cannot capture useful image content, while too big loses interpretability and locality of attributes. Based on these results, we set the region size for the remainder of the experiments in order to maximize the semantic meaningfulness of our image region candidates,

generating 50×50 regions for Stanford and 70×70 regions for INRIA.

5.3.2 Multiple Attributes

To use multiple attributes for classification, we aggregate the discovered attributes and use them to build a new representation for each training image. To do this, we apply each attribute classifier on a held-out validation set, and collect all attribute detection scores. We build a $T = (K \times A)$ table, where K is the number of categories and A is the number of attributes. If more than half of the images in a category k have attribute a , then we set $T(k, a) = 1$, otherwise to 0. We use T for nearest neighbor classification.

Our framework can be used to generate multiple attributes by learning MI-SVM on different category pairs and by forcing candidate regions not to overlap (Chapter 5.2). However not all such attribute candidates are beneficial to the overall classification performance, so we use an attribute selection method similar to [32] to select the subset of best ones for classification. When a new attribute is generated, we keep it if it improves the overall classification accuracy on a held-out validation set; otherwise it is dropped.

Semantic filtering and naming. We post process these attribute candidates to name them using human feedback from Amazon Mechanical Turk. We present the attribute visualizations generated from each viewpoint to human subjects with their cropped bounding boxes, and put them in a single image gallery if they correspond to the same attribute. Specifically, we asked each subject for the part name, a descriptive word, and their confidence score (on a 1-5 scale) as well. We remove the non-semantic ones if the average confidence score is lower than 3. Every candidate was shown to

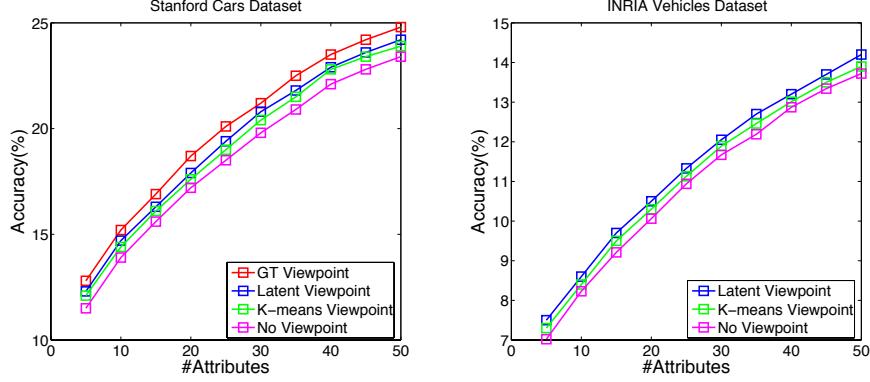


Figure 5.3: Classification accuracy with different numbers of discovered attributes and different techniques for handling viewpoints, for Stanford (*left*) and INRIA (*right*) datasets.

5 human users, and the names of the attributes were determined by the majority of the feedbacks.

Category classification results. We studied classification accuracy according to number of detected attributes, as shown in Figure 5.3. We also compare several attribute selection methods requiring different degrees of viewpoint supervision. **GT Viewpoint** uses the ground truth viewpoint labels in the training set using our technique of Chapter 5.2.2. **No Viewpoint** completely ignores viewpoint information in the vehicle discovery process (*i.e.* all images are assumed to have the same viewpoint label). **K-means Viewpoint** runs K -means using global image features to assign initial viewpoint labels without any further update (*i.e.* performs only the initialization phase of Chapter 5.2.3). Finally, **Latent Viewpoint** uses our full model, treating viewpoint labels as unknown latent variables and applying the method in Chapter 5.2.3. From the figure, we see that incorporating viewpoints into the model helps classification accuracy across any number of attributes. The best results are

achieved when viewpoint is available in ground truth, but our technique that can infer viewpoints automatically performs better than either of the simpler baselines. Note that we use 8 viewpoints in these experiments and the INRIA vehicles dataset does not have ground truth viewpoint annotations, so we only report results for the other three methods.

Combining with low level features. We achieve better results by combining the attribute features with low-level Fisher vector features [28]. We use a simple blending scheme on the normalized scores of each test image as $S = \alpha \cdot S_{low} + (1 - \alpha) \cdot S_{attr}$, where S_{attr} is the classification score from attributes and S_{low} is the score from the Fisher vectors. We choose the best α using a held-out validation set. On both datasets, we find that combining attributes and low-level features improves classification accuracy compared with just using the low-level features, with an increase from 88.2% to 89.57% on Stanford cars and 33.58% to 34.54% on INRIA, both using 50 attributes. (Note that low-level results on the INRIA set reported in [68] are higher, but they use a much larger mixture model to compute Fisher vectors, so the numbers are not directly comparable.)

Qualitative Results. Figure 5.4 shows sample local attributes learned using our technique applied on the Stanford cars dataset. These semantic and discriminative visual attributes can be used for automatic image annotation on new images. Figure 5.5 shows sample tags produced for test images on the Stanford cars dataset.

(a) head light, fender, red, red and blue	(b) wheel, silver wheel cup, black tire	(c) back wheel, tire, white, right headlight	(d) window, trunk, rear, black	(e) headlight, front light, square
(f) side door, sil- ver wheel cup, black tire	(g) front wheel, black, silver, right headlight	(h) hood, wind- shield, bumper, silver, blue	(i) front wheel, fender, red, white	(j) rear head- light, back, black window

Figure 5.4: Examples of automatically generated local attributes for the Stanford cars dataset. Each panel represents one discovered local attribute for a particular viewpoint of the vehicle category, with names coming from Mechanical Turk users.

5.4 Summary

We have presented a novel approach for discovering local visual attributes for vehicle categories and for modeling viewpoint classes at the same time. We have performed systematic experimental evaluations to demonstrate our discovered attributes help to improve baseline classification methods. We showed that our discovered attributes are both discriminative and semantically meaningful, leveraging user feedback on the machine-generated attribute candidates. In future work, we will explore more useful applications of local attributes (*e.g.* image retrieval, automatically caption generation, etc.) and will study incorporating local attributes into vehicle detectors.

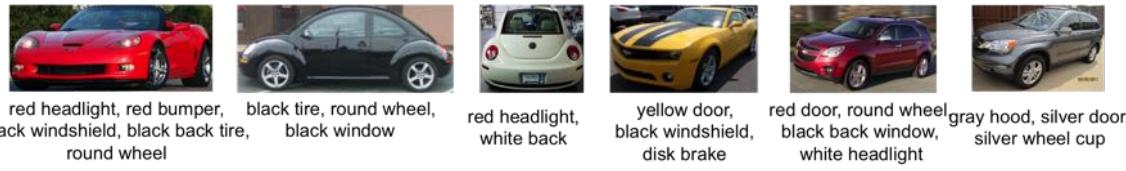


Figure 5.5: Examples of vehicle annotation results on new images.

CHAPTER 6

Multimodal Image Modeling

So far we have applied CRF models to capture different structured information (*e.g.* human pose, image local regions). However, the scale of these problems is usually constrained to small training and test datasets. In this chapter, we investigate how to model multimodal information (*e.g.* visual, text, GPS locations, *etc.*) on the web using CRF models. We propose multimodal latent CRF for organizing image collections. We review and discuss existing work in Section 6.1, and then we describe our approach in Section 6.2. We show experimental results in Section 6.3.

There are billions of images on the web. For example, the daily photo uploads on Flickr.com are more than one million, and on Facebook, this number is more than 350 million [1]. To help users organize and browse photos at such a large scale, people usually use clustering techniques. Web photos do not appear alone: metadata such as image captions, text tags, GPS tags are all useful information that can be used to improve clustering performance. However, these metadata are usually sparse and noisy. For example, only about 80% of randomly downloaded Flickr images have been annotated with tags, and only 5% to 10% Flickr images have GPS tags. How to incorporate these sparse and noisy metadata information into the clustering framework is still an open question.

6.1 Related Work

We consider using conditional random fields for organizing multimodal web images automatically or semi-automatically. There is a vast literature on unsupervised and semi-supervised learning in the data mining community, and these techniques have been applied to organizing photos in a variety of contexts [44, 45, 77, 80, 140, 143, 146]. Two research threads are most closely related to this thesis: multimodal modeling in image collections, and constrained clustering.

6.1.1 Multimodal Modeling

McAuley and Leskovec [85] use relational image metadata (social connections between photographers) to model pairwise relations between images, and they apply a structural learning framework to solve the resulting labeling problem. While similar to our work in spirit, their formulation does not allow for missing metadata, and does not incorporate multimodal features (and does not use visual features at all). Rohrbach *et al* [103] propose a framework to recognize human activities in videos using both visual and detailed textual descriptions. Guillaumin *et al* [49] use a semi-supervised classifier on visual and text features for image classification; they allow missing class labels on training images, but do not allow for sparse features (they assume that all training images have text tags). In contrast, our model allows missing features in any modality channel, and learns the concepts in a *loosely supervised* manner (using just a small labeled training dataset to learn the parameters of our CRF).

Bekkerman and Jeon [9] perform unsupervised multimodal learning for visual features and text, but similarly do not attempt to handle sparse or missing features.

Perhaps most relevant to our work is that of Srivastava and Salakhutdinov [114], who propose a multimodal learning method using deep belief networks. Their work allows for missing modalities on training instances by a sampling approach, but their technique can be expensive because it requires many different layers and also a lot of parameters. On the other hand, we propose a lightweight unsupervised learning framework which discovers clusters automatically, but that can still be used to build discriminative classifiers to predict missing modalities on new unseen images.

6.1.2 Constrained Clustering

Several papers incorporate constraints into classical clustering algorithms like K -means. Our approach can be thought of as constrained clustering, similar to HMRF-Kmeans [8] and related work [77, 80, 128], but there are key differences in motivation and formulation. We explicitly deal with missing features (which are quite common in web images) while these existing methods do not consider this problem. Intuitively, our framework only performs K -means updates (the “M-step”) for one feature channel; when this type of feature is missing on some instances, K -means updates are calculated based on a subset of the network. Our work is related to Wagstaff *et al* [129] and Basu *et al* [8] who add “hard” constraints to the standard K -means framework, including “must-link” and “cannot-link” constraints between data points. In our application, where metadata is noisy and often inaccurate or ambiguous, such hard constraints are too strong; we instead use “soft” constraints that encourage instances to link together without introducing rigid requirements. Our models also allow different feature types in the pairwise constraints (e.g. some constraints may be defined in terms of tag relations, while others are defined using GPS, etc).

Summary. Multimodal image modeling is an important topic in both the computer vision and multimedia research communities. We propose a general framework using latent conditional random fields for learning semantic concepts from multimodal web image data in two supervision modes (*i.e.* weak supervision and loose supervision). Our approach directly generalizes the classical K -means method, and can incorporate arbitrary modality features altogether in a single framework. The proposed CRF framework models the relations between images through the similarity relations on different modality feature channels. A link is added between two image instances if they are available on one or more common feature types. Compared to the above relevant research literature, our model is lightweight, generic, and mathematically principled. Our model not only helps organizing and browsing web images, but could also enable interesting applications like image annotation, image recommendation, etc.

6.2 Loosely Supervised Multimodal Learning

We now present our approach for loosely supervised clustering in datasets with multimodal, sparse features. We assume that there are multiple feature types that are not comparable with one another, and observed values for some of these features on each instance in our dataset. For example, for online photos we may have visual features, text tags, and geotags, for a total of three feature modalities, and visual features are observable in all images but the others are available on just a subset. Our goal is to jointly consider all of this sparse and heterogeneous evidence when clustering.

6.2.1 Constrained Clustering Framework

We can think of our approach as a generalization of the classic K -means clustering algorithm. In K -means, we are given a dataset of instances $X = \{x_1, \dots, x_N\}$, where each instance is a point in a d -dimensional space, $x_i \in \mathcal{R}^d$. Our goal is to assign one of K labels to each instance, i.e. to choose $y_i \in [1, \dots, K]$ for each instance x_i , and to estimate K cluster centers $\mu = \{\mu_1, \dots, \mu_K\}$, so as to minimize an objective function measuring the total distance of points from assigned centroids,

$$\min_{\mu, \mathbf{y}} \sum_{i=1}^N \sum_{k=1}^K \mathbb{1}(y_i = k) \|x_i - \mu_k\|^2, \quad (6.1)$$

where $\mathbf{y} = (y_1, \dots, y_N)$ and $\mathbb{1}(\cdot)$ is an indicator function that is 1 if the given condition is true and 0 otherwise. Note that this formulation implicitly assumes that each instance can be represented by a point in a d -dimensional space, and that Euclidean distances in this space are meaningful.

In our approach, we assume that we have M different types of features, only a subset of which are observable in any given instance. Our dataset thus consists of a set of N instances, $\mathbf{X} = \{x_1, \dots, x_N\}$, where each $x_i = (x_i^1, \dots, x_i^M)$, and a given x_i^m is either a feature vector or \emptyset to indicate a missing value. We treat one of these as the *primary* feature (we discuss how to choose the primary feature below) and consider the others as soft constraints, which tie together instances having similar values. We assume without loss of generality that the primary features have index $m = 1$. Any of these feature types (including primary) may be missing on a given instance. An illustration of our approach is shown in Figure 6.1. Now we can generalize the K -means energy function in equation (6.1) as,

$$\min_{\mu, \mathbf{y}} E(\{y_i\} | \{x_i\}), \quad (6.2)$$

with

$$\begin{aligned} E(\{y_i\}|\{x_i\}) &= \sum_{i=1}^N \sum_{k=1}^K \mathbb{1}(y_i = k) \cdot \alpha(x_i^1, \mu_k) \\ &\quad + \sum_{m=2}^M \sum_{i=1}^N \sum_{j=1}^N \beta_m(x_i^m, x_j^m) \cdot \mathbb{1}(y_i \neq y_j), \end{aligned} \tag{6.3}$$

and where $\alpha(\cdot, \cdot)$ is a distance function that defaults to 0 if a primary feature is missing,

$$\alpha(x_i^1, \mu_k) = \mathbb{1}(x_i^1 \neq \emptyset) \cdot \|x_i^1 - \mu_k\|^2,$$

and $\beta_m(\cdot, \cdot)$ is a function that measures the similarity between the m -th (non-primary) feature of two instances (described below), or is 0 if one or both of the features are missing. Intuitively, the first summation of this objective function is identical to that of the objective function of K -means in equation (6.1), penalizing distance from the primary features to the cluster centroids. If a primary feature is missing in a given instance, it does not contribute to the objective function (since any assigned label has equal cost). In the special case that there is exactly one feature type and it is always observable, equation (6.3) is equivalent to simple K -means in equation (6.1). The non-primary features add soft constraints through the second set of summations in equation (6.3), penalizing pairs of instances from being assigned to different clusters if they have similar features.

The objective function in equation (6.3) is a Latent Conditional Random Field model. Each instance (image) is a node in the CRF, and the goal is to label each node with a cluster identifier. The primary features define unary potentials, which give a cost for assigning a given node to each centroid, or a uniform distribution if the primary feature is missing. As in K -means, the cluster centroids are latent variables that must be estimated from data. Edges connect together pairs of instances where

non-primary feature are available, with pairwise potentials given by the β functions. To perform clustering in this framework, we must perform inference on the latent CRF. This is an optimization problem with two sets of unknown variables: the cluster centers μ and the cluster assignments \mathbf{y} . We use an EM-like coordinate descent algorithm to solve this problem, iteratively applying the following steps:

1. In the **E-step**, we fix μ and (approximately) solve for \mathbf{y} by performing discrete optimization on the CRF using tree-reweighted message passing (TRW-S) [65].
2. In the **M-step**, we fix \mathbf{y} , and solve for each μ_k with simple maximum-likelihood estimation.

Note that these two steps are the familiar algorithm used in K -means, except that the E-step here involves jointly assigning cluster labels to the instances by performing inference on a CRF (instead of simply assigning each instance to the nearest cluster center as in K -means). The M-step is identical to that of K -means, except that here we ignore instances with missing primary features.

We can use this framework in different ways, depending on the amount of information available in a given application. In a *weakly supervised* setting, we assume that for some pairs of instances (in a held-out set), we know whether each pair belongs to the same class or a different class. We use these labels to learn the pairwise potentials as described in Chapter 6.2.2. We can learn a distance metric even when the constraint features are available but the primary feature is missing, or when the labeled set is in a different domain than the clustering application at hand. In a *loosely supervised* setting, we make the stronger assumption that a small subset of instances have ground-truth class labels, such that we can estimate the centroids us-

ing the small subset, and fix the centroid labels in that subset while solving for the rest.

6.2.2 Learning Pairwise Potentials

The clustering framework in Chapter 6.2.1 requires pairwise potential functions $\beta_m(\cdot, \cdot)$ to evaluate the similarity between two instances according to each feature type. These functions are critically important to clustering performance and thus we learn their parameters automatically. We define the pairwise potentials for each feature type m to have the following parametric form,

$$\beta_m(x_i^m, x_j^m) = \mathbb{1}(x_i^m \neq \emptyset \wedge x_j^m \neq \emptyset) \cdot (w_m \cdot d_m(x_i^m, x_j^m) + b), \quad (6.4)$$

where $d_m(\cdot, \cdot)$ is a (learned) distance function for the given feature type, w_m and b are scalar weight and bias terms, and the indicator function ensures $\beta_m(\cdot, \cdot)$ is clamped to 0 if either feature is missing. Learning the potential functions now involves estimating the distance function $d_m(\cdot, \cdot)$ for each feature type, and the weight and bias terms w_m and b ; we estimate these in two separate steps.

Learning the distance functions. We assume that the distance functions are Mahalanobis distances,

$$d_m(x_i^m, x_j^m) = (x_i^m - x_j^m)^T A_m (x_i^m - x_j^m),$$

and thus we need only to estimate the matrices A_m . To do this, we use Information Theoretic Metric Learning (ITML) [25] to learn these matrices from pairwise supervision on the small labeled training data. For increased robustness to noise, we used diagonal Mahalanobis matrices.

Learning the potential function parameters. We wish to jointly estimate the $M - 1$ feature weight parameters $\mathbf{w} = (w_2, \dots, w_M)$ and the bias term b in equation (6.4). We formulate this as a standard margin-rescaled structural SVM learning problem [126]. Let y_i and \tilde{y}_i be the ground truth and predicted label of x_i , $E(\{y_i\}|\{x_i\})$ be the energy when the labelings are $\{y_i\}$ (in equation (6.3)); we minimize,

$$\min_{\lambda, \mathbf{w}, b} \lambda \|\mathbf{w}\|^2 + \xi,$$

such that,

$$E(\{\tilde{y}_i\}|\{x_i\}) - E(\{y_i\}|\{x_i\}) \geq \Delta(\{\tilde{y}_i\}, \{y_i\}) - \xi,$$

$$\forall \{\tilde{y}_i\} \neq \{y_i\}, \mathbf{w} \geq 0, \xi \geq 0.$$

We define our loss function using *number of incorrect pairs*,

$$\Delta(\{\tilde{y}_i\}, \{y_i\}) = \sum_{i=1}^N \sum_{j=1}^N \mathbb{1}_{\tilde{y}_i = \tilde{y}_j \wedge y_i \neq y_j \vee \tilde{y}_i \neq \tilde{y}_j \wedge y_i = y_j};$$

in other words, for each pair of instances in the dataset, we count how many of them were incorrectly assigned to different clusters and how many were incorrectly assigned to the same cluster. This definition of loss is the *Rand Index* [101], a popular evaluation metric in the clustering literature. We chose to use this metric (as opposed to other popular metrics like purity) because it allows the loss function to decouple into independent optimizations over each data point. We can then perform *loss-augmented inference* using the TRW-S algorithm [65] at training time, allowing for efficient inference in the inner loop of structured SVM training.

6.3 Experiments

We demonstrate our clustering method on four datasets collected from Flickr, three of which have ground-truth to allow for quantitative evaluation. In the fourth dataset, we show how our technique can be used to discover structure in large collections of images for which no ground truth exists.

6.3.1 Applications and Datasets

We use four datasets of images from Flickr collected using the public API. To test the robustness of our approach in different settings, each of these datasets targets a different application of unsupervised clustering, and uses different feature types and ground truth collected in varying ways.

Landmarks. Our first dataset contains images from the ten most-photographed landmarks on Flickr, using the dataset from [76]. That paper clusters geo-tags to find highly-photographed places and learns discriminative classifiers for each of these landmarks. Here we test if our method can separate the landmarks in a less supervised manner, which could be useful in organizing large tourist photo collections around travel destinations. In this dataset we use only image features and text tags; we do not use GPS features because they were used to define the ground truth classes. We hide the ground truth, apply our clustering framework on image and tag features, and then compare the clustered results with the ideal clustering induced by the class labels. This **Landmarks** dataset includes 8,814 images.

Groups. Sites like Flickr let users contribute their photos to groups about user-defined topics. These groups have rich and varied themes, and the ability to categorize

photos into groups automatically could be useful to help users organize their photos. We collected 1,000 images from each of 10 Flickr groups related to the following topics: aquarium, boat, bonsai, cars, Christmas, fireworks, food, penguins, skyscrapers, and sunsets. (These are the topics shown in Fig. 1 of [132]; unfortunately those authors could not share their dataset, so we found Flickr groups corresponding to the same topics and gathered our own images). We use visual, text, and geo-tag features in this **Groups** dataset.

Activities. We are also interested in clustering images according to human activities like attending a game, going to a museum, taking a hike, etc. Since these activities correspond to higher level semantics than simple actions like walking, running, etc., they are difficult to classify using visual features alone. (For instance, a picture of cars could be “car racing” if the cars are moving or “car show” if they are stationary, but the difference in visual appearance is subtle.) We thus use our multimodal clustering algorithm to incorporate visual, textual, and GPS features into this organization process. We collected two activity-related datasets. **Sport** consists of 10,000 images related to sporting events, which we collected by crawling 10 types of Flickr groups (American football, baseball, basketball, hockey, horse racing, marathons, NASCAR, football (soccer), swimming, tennis). These group labels give ground truth for evaluation. **Activity** includes about 30,000 random images from Flickr, which we use to qualitatively test our approach’s ability to discover activities in unlabeled data. Here we use a large number of clusters ($K = 1000$) so that we can find coherent clusters despite the large number of outlier images.

In collecting the above datasets, we were careful to prevent “leaks” between class labels and the features used for clustering. For example, we did not use text features

Purity:

	Visual features	Text features	Visual+Text	Proposed (V+T)	Proposed (V+T+G)
Landmarks	0.1677 ± 0.0134	0.3224 ± 0.0335	0.3449 ± 0.0383	0.4060 ± 0.0279	—
Groups	0.2508 ± 0.0097	0.3696 ± 0.0263	0.3955 ± 0.0341	0.4395 ± 0.0389	0.4450 ± 0.0353
Sport	0.1483 ± 0.0101	0.3454 ± 0.0386	0.3524 ± 0.0387	0.3713 ± 0.0309	0.3965 ± 0.0182

Inverse purity:

Landmarks	0.3163 ± 0.0180	0.4907 ± 0.0344	0.5297 ± 0.0227	0.5611 ± 0.0210	—
Groups	0.4066 ± 0.0448	0.5893 ± 0.0275	0.5971 ± 0.0310	0.6010 ± 0.0322	0.6336 ± 0.0152
Sport	0.3707 ± 0.0411	0.6593 ± 0.0244	0.6789 ± 0.0175	0.6931 ± 0.0173	0.7062 ± 0.0190

Table 6.1: Purity (top) and Inverse Purity (bottom) on three datasets with $K = 10$ clusters. Means and standard deviations are over 5 trials. (GPS information is not available for **Landmarks**.) Our multimodal approach significantly outperforms single modality baselines and combined feature baselines, both in terms of purity and inverse purity.

to define class labels, instead relying on geo-tags and group assignments. We also prevented any single photographer from dominating the datasets by sampling at most 5 photos from any single user. In general, about 80% of images have at least one text tag and about 10% of images have a geo-tag.

6.3.2 Features

On **Landmarks**, **Groups**, and **Sport**, we represent each image using histograms of visual words (using SIFT descriptors and a visual vocabulary of 500 words built using K -means). For the text features, we apply PCA on the binary tag occurrence vectors to reduce the dimensionality to 200. We learn a Mahalanobis distance for the text features using the method in Chapter 6.2.2 on the lower-dimensional space.

For geo-tags, we use chord lengths on the sphere as the distance between two GPS coordinates. On the **Activity** dataset, we compute high-level features using object bank [75], and use image captions as the text features. Stop words are removed, the remaining words are stemmed, and we represent the text using binary occurrence vectors and again apply PCA to reduce the dimensionality to 200.

6.3.3 Results

As mentioned in Chapter 6.2.1, our framework can be applied in different ways depending on the type of ground truth available. We first evaluate under weak supervision, which assumes that we have pairs of exemplars which we know belong to either the same or different classes, and we use these to learn the pairwise distances and potential functions. We also evaluate under loose supervision, which makes the stronger assumption that we have some exemplars with ground-truth class labels, so that the primary feature centroids can also be initialized.

Weak supervision. Table 6.1 presents quantitative results for three datasets under weak supervision, using *purity* and *inverse purity* [2] as the evaluation metrics. For example, to compute purity, we calculate the percentage of instances within each estimated cluster that agree with the majority ground truth label of those instances. These numbers are averaged across all clusters to compute a final purity score. Mathematically, the purity score is defined as:

$$Purity = \sum_i \frac{|C_i|}{n} \max_j \frac{|C_i \cap L_j|}{|C_i|}$$

where C_i is a predicted cluster and L_j is a ground truth category. Similarly, the

inverse purity score is defined as

$$Inverse\ Purity = \sum_i \frac{|L_i|}{n} \max_j \frac{|L_i \cap C_j|}{|L_i|}$$

The table compares our method against several baselines: *Visual features* runs K -means on visual features only, *Text features* performs K -means using text features only, *Visual+Text* concatenates both features and performs K -means. Photos without tags are assigned random tags. *Proposed (V+T)* uses our approach with visual and text features, and *Proposed (V+T+G)* uses our approach with visual, text and GPS features. In each case we run 5 trials and report means and standard deviations, since results are non-deterministic due to the random initialization.

As shown in the table, our proposed method to incorporate (weak, sparse, noisy) multimodal data outperforms the baselines significantly. Visual features alone work relatively poorly (*e.g.* purity of about 0.17 for **Landmarks**), while text features are much more informative (0.32). Combining text and visual features together by simply appending the feature vectors and running K -means improves results slightly (0.34), while combining visual and text features in our framework significantly outperforms all of these baselines (0.41). Much of this improvement may come from our technique's ability to better handle photos that do not have text tags (about 20% of photos): when we exclude photos having no tags, the text-only K -means baseline increases to 0.3705 for **Landmarks** and 0.4567 for **Groups**. Finally, adding GPS features results in a modest additional gain.

We use text as the primary feature in the above experiments. We have found that the choice of primary feature is important, due to the different roles that the unary and pairwise potentials play in the constrained clustering framework. Intuitively, the

pairwise constraints only depend on whether the labelings of two neighbors are the same, while the unary potentials encourage each node to explicitly select one of the K labels. It is thus easier for a labeling of the nodes to minimize the pairwise cost than the unary cost. To understand this better, we tested each of the two feature types (visual and text) in isolation as unary or pairwise constraints. Results of using only a unary term were already presented above, in the first two columns of Table 6.1; we tested the pairwise potentials in isolation by fixing the unary potentials to be uninformative uniform distributions. On **Landmarks**, switching visual features from primary to pairwise features causes purity to change from 0.1677 to 0.1462, a drop of 13%, while switching text features from primary to pairwise drops the purity by 31% from 0.3224 to 0.2223. This result suggests that we should select the “strongest,” most informative feature as the primary.

Figure 6.2 studies how sparsity of primary and secondary and text and visual features affects results, by hiding features of varying numbers of images. For each dataset, the left plot compares results of using a subset of text features as the primary and no constraint features (red), with using all visual features as primary and subsets of text features as constraints (blue). The red line is thus the same as simple K -means, where images without text features are randomly assigned to a cluster. The right plot shows a similar comparison but with the roles of the text and visual features swapped. We see of course that more observations lead to better performance, with best results when using all available text as primary features and all visual features as constraints. But the results also highlight the flexibility of our approach, showing that multi-modal features (blue lines) significantly improve performance over a single feature type (red lines), even when only a small percentage of photos have the feature.

Loose supervision. We used small labeled subsets of different sizes to evaluate the loosely supervised paradigm, and evaluate using classification accuracy. We used linear SVMs trained on visual and text features as baseline methods, with the classifier parameters chosen according to 5-fold cross validation on the training data. Figure 6.3 shows that our proposed loosely supervised method outperforms SVM classifiers given the same amount of supervision, especially when the available training data is only a small percentage of the entire dataset. For instance, on **Landmarks**, our technique can achieve about 54% classification accuracy (relative to 10% random baseline) with 1,000 labeled exemplars, versus just 33% for a trained SVM using the same features and training set.

Qualitative results. Figure 6.4 presents sample clustering results for the **Landmarks**, where in each group we show the images closest to the cluster centroid and the most frequent tags in the cluster. Figure 6.5 presents sample clusters from our **Activity** dataset of 30,000 images, showing that the algorithm has discovered intuitively meaningful activity and event clusters like car shows, wildlife, festivals, beaches, etc. Since we do not have labeled ground truth for this dataset, we simply used the learned parameters from **Sport**.

6.4 Summary

We proposed a multimodal image clustering framework that incorporates both visual features and sparse, noisy metadata typical of web images. Our approach is loosely supervised, and is reminiscent of the standard K -means algorithm: one feature is used as the primary feature in K -means-style updates, while other features are incorporated as pairwise constraints. The proposed approach is flexible and can be applied

under different degrees of supervision, including when no training data is available at all, and when features are missing. In future work, we plan to incorporate other types of constraints in the graphical model, and to apply our approach to various applications (*e.g.* automatic image annotation and recommendation).

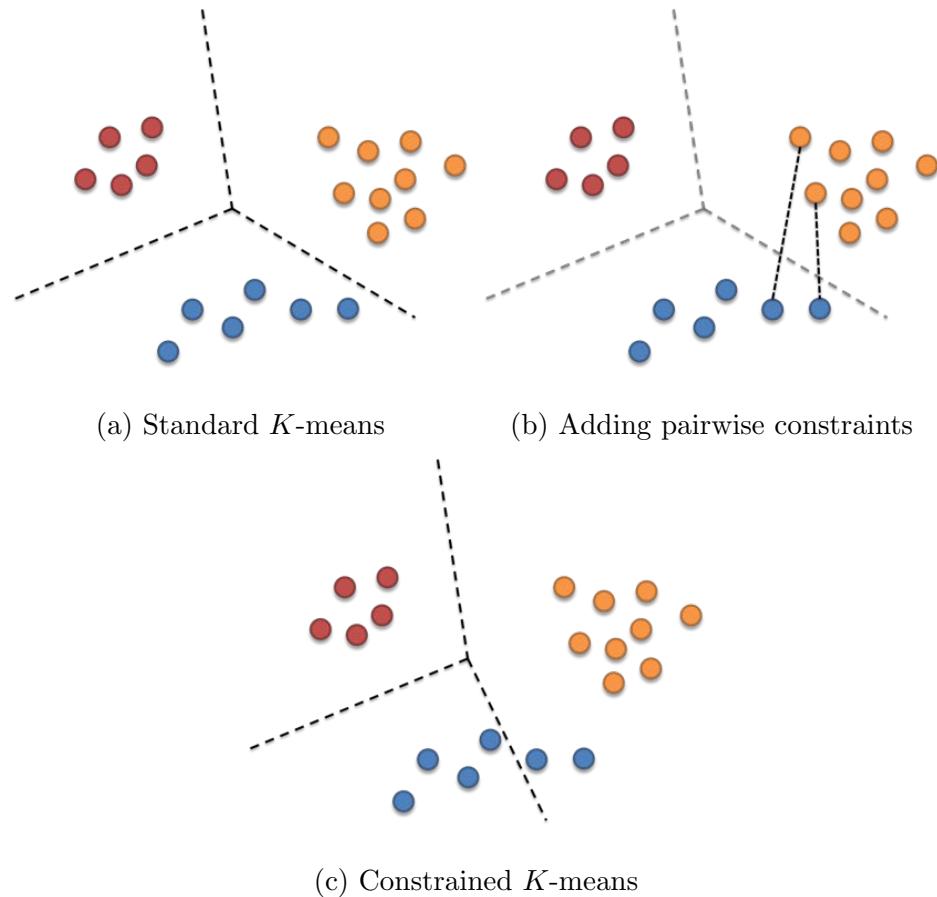


Figure 6.1: Illustration of our constrained clustering framework. (a) Standard K -means has only one feature type; (b) we add more feature types, which induce pairwise soft constraints between instances; (c) CRF inference balances evidence from all features in performing the clustering.

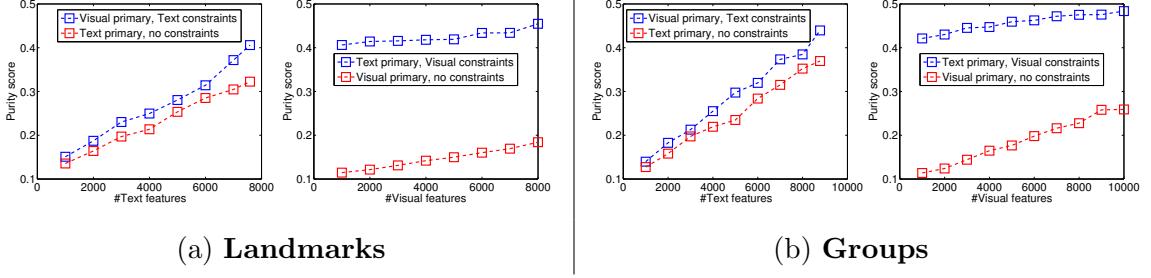


Figure 6.2: Clustering performance as a function of number of images with different types of features. Red lines use primary features for only a subset of images and do not use constraints (*i.e.* as in classic K -means). Blue lines use our multimodal clustering framework, incorporating primary features for all images and a subset of images with constraint features. For each dataset, purity in the left plot is calculated using all images, while in the right plot it is calculated using images having tags.

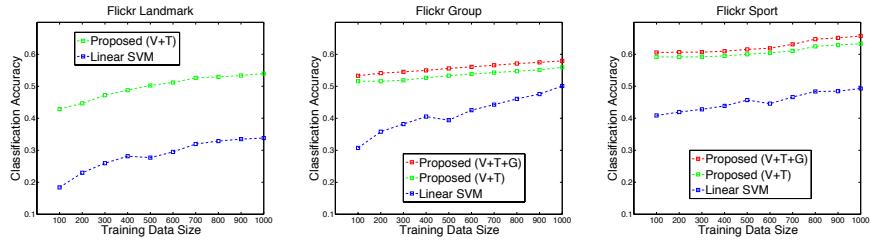


Figure 6.3: Classification performance comparisons with loose supervision on training sets of increasing sizes, using **Landmarks** (left), **Groups** (middle), and **Sport** (right). Linear SVM baseline is trained on concatenated visual and text features.



Figure 6.4: Sample landmark clusters discovered automatically by our algorithm.

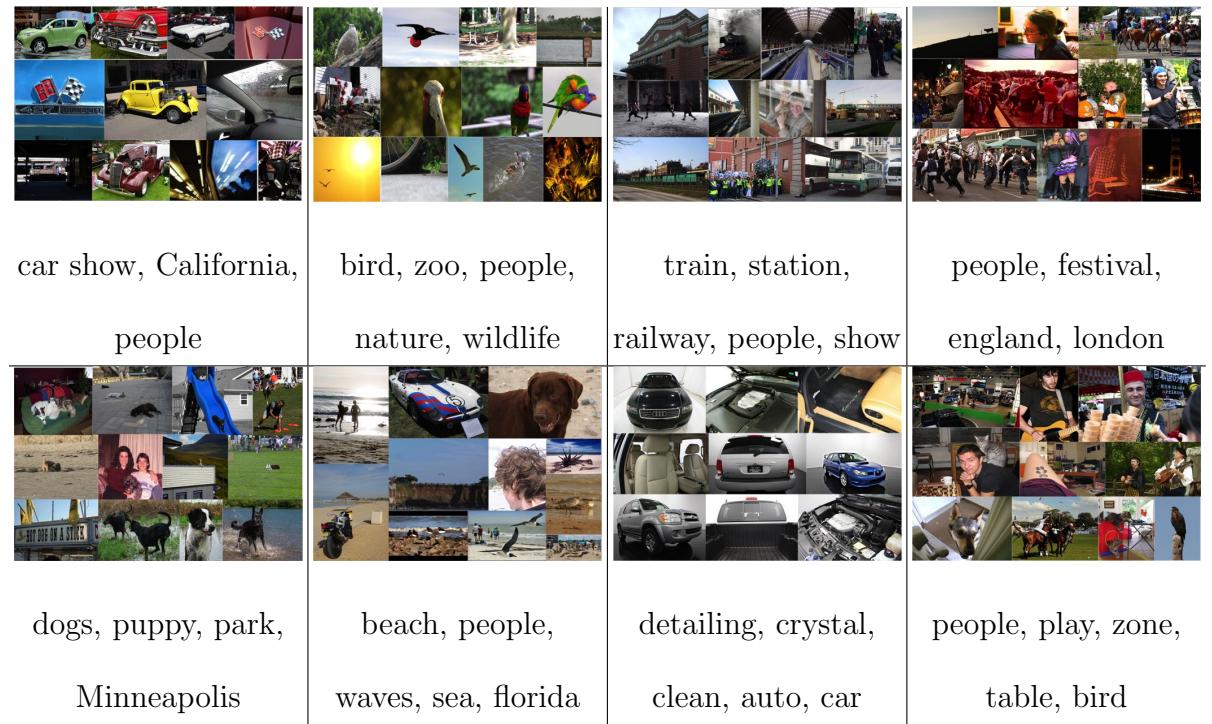


Figure 6.5: Some activities discovered by our algorithm.

CHAPTER 7

Conclusion

In this thesis, we have studied different types of object recognition problems, and used CRF as a common framework to abstract the problems and derive the formulations. We studied how to learn a CRF discriminatively, and how to perform efficient inference on different graph structures. Specifically, we looked at three object recognition applications (*i.e.* human pose estimation, fine-grained object recognition, and multimodal image modeling) in detail. Each of them comes with different supervision levels, and with different types of structures. We performed systematic evaluations on standard benchmark datasets and on real world data as well.

The major contribution of the thesis is that we proposed conditional random field models for different structured object recognition problems at different supervision levels. These applications are representative, in that they all have rich structured information that captures different aspects (*e.g.* object part structure, image region structure, and structure of an image collection) of the recognition problems. For example, in human pose estimation, we assumed each human body can be decomposed into a set of key joints (or parts), and model the geometric constraints among these parts using a tree structure. In fine-grained recognition, we proposed a novel image representation called “local attributes”, and proposed an attribute discovery workflow

based on a latent CRF that captures pairwise relations between image regions. We also used a latent CRF to model the structure of an image collection for automatic photo organization and browsing. The latent variables are the K -means centroids, and the graph structure of the CRF is defined by the relations between pairs of images on different modality channels.

In Chapter 3, we introduced a multi-layer composite representation of the human body structure for human pose estimation tasks. Our model is defined using a CRF, and generalizes a common collection of pose estimation models using single tree structures. We formulated the problem of learning the model parameters using a structural SVM, and train the multi-layer model in a fully supervised manner (*i.e.* all key body joint locations are known in the training process). We explored the nature of human pose structures, and explicitly used a tree structure for the submodel at each individual layer. We also used a tree-structured decomposition constraint to “link” these layers together. Thus, our multi-layer composite model can be naturally decomposed into multiple trees; at testing time, we used dual-decomposition as a linear program relaxation method to optimize the inference objective function.

The above CRF models benefit from fully annotated training exemplars at training time. However, there might be useful latent information that could be used to solve the given task. In Chapter 4 and Chapter 5, we proposed to discover such latent information for fine-grained recognition tasks using latent conditional random field models. We noticed that image representation is an important aspect for solving fine-grained recognition problems. The existing “attribute-based” approaches mainly handle global attributes, and cannot capture the differences between fine-grained categories very well, in that the discriminative information between fine-grained categories

are mostly from local regions. Thus we proposed to treat these local, discriminative image regions as latent information, with the assumption that class label for the training categories are given.

In Chapter 6, we further explored the use of the latent conditional random field in the case of modeling multimodal web images. Modeling such online photo collections is a challenging problem because of their large scale and the existence of sparse, noisy meta data (*e.g.* text tags, captions, GPS tags, *etc.*). We proposed a novel framework called “multimodal latent CRF” as a generalization of the classical K -means algorithm with multimodal features. We used one primary feature channel to define the unary potentials in the CRF through calculating the distances of an instance to all the centroids, and used similarity functions defined on learned Mahalanobis distances to define the pairwise potentials in the photo collection.

Having studied these motivating problems that are representative of many others, we have drawn some conclusions about best practices for applying CRFs to various problems. In the rest of this chapter, we will first summarize the practices of using conditional random field models (*e.g.* how to design the graph structure, how to choose training and inference algorithms, *etc.*) for structured prediction problems in computer vision. Then we show and compare a list of popular software tools related to CRF training and inference.

7.1 Practices of Using CRF Models

CRF models have seen applications in many different research areas for modeling structured data, and have been used extensively in computer vision problems (*e.g.* image denoising, image restoration, 3D reconstruction, stereo matching, segmenta-

tion, recognition, *etc.*). A CRF can be used whenever there are dependencies among random variables, and efficient ways to propagate information or constraints are fundamental.

CRF models have strong descriptive power that can easily convert the observations and assumptions into principled mathematical formulations. On highly-connected graph structures, the performance of message passing inference algorithms decrease because of the difficulty to converge with a low gap between the energy under inferred labels and the lower bound energy. However, there have been evidence that message passing algorithms (*e.g.* TRW-S) could perform better than Graph-cuts approach [66] if the message passing speed can be improved.

There are trade-offs in designing the model structure: using simple structures v.s. using complicated structure. One naive solution to many structured problems would be to use a fully connected graph to model the relations, but since algorithms for exact and efficient inference are not known on such graphs, approximate inference must be used. To correctly build better graphical models for a given problem, we need to apply domain knowledge and explore the relations between random variables exclusively. In general, “minimalism” is a reasonable guideline for designing the CRF structure, meaning that we need to remove unnecessary edges in the graph as much as possible. For example, a human body structure obeys the kinematic constraints, thus a tree structure is more appropriate than a fully connected graph. We need to find good reasons for additional edges added into the graph, *e.g.* the decomposition links that connect adjacent layers in our multi-layer composite pose model.

We also need to think about the design of potential functions when designing the graph structure. Some special types of potential functions allow efficient inference

even if the graph has higher order clique potentials. For example, a cardinality potential function which is associated with a sum of binary variables allows efficient and exact inference in the graph [119]. Also, Robust P^n model [64] generalizes the P^n Potts model, and the inference problem can be solved using graph cuts methods based on move-making algorithms.

Pairwise graph structures are the most common choice in practice, since most relations can be modeled as pairwise relations, or can be decomposed into multiple pairwise relations. On the other hand, graphical models with high order potential functions are useful when “long-range” relations need to be captured, while local information such as pairwise relations between neighboring nodes are not enough. One example for this is the image segmentation task, where neighboring pixels are encouraged to have the same label via pairwise smoothing terms in the graphical model. However, this type of local constraints assumes pixels are independent, and is especially not sufficient in *semantic segmentation* tasks (*i.e.* automatically annotate each pixel with an object category label). Thus, global constraints are used as high order clique potentials in CRF models for capturing the interaction between superpixels (or regions) to obtain better results.

To make the CRF training more convenient, we usually define the potential functions to be linear scoring functions (*i.e.* dot product between a weight vector and the feature vector). In this way, we can re-write the entire energy function of the CRF model as a linear scoring function, and a linear structural SVM can be used for learning the model parameters discriminatively. Once the parameters are learned, we can perform inference by applying the learned CRF model on unseen instances using the methods introduced in Section 2.4.

CRF Software Comparisons. We make a brief comparison of popular software tools related to training and inference of conditional random field models in Table 7.1 and Table 7.2. Some of these software tools or packages were used in one or more experiments discussed in the previous chapters. For example, we have used CVX and TRW-S in the attribute discovery project and the multimodal image modeling project for CRF training and inference, respectively.

7.2 Future Work

In future work, we will study how to bring state-of-art ideas into the design of graphical models like CRFs. For example, deep learning methods have recently been very popular [118, 123, 145], and have been proved to be quite successful in many machine learning applications.

A deep network has multiple layers with millions of parameters. It has strong connections with feature learning, since the network input is usually directly taken from raw signals (*e.g.* image pixels). The input signals are passed through several convolution layers and hidden layers, and the parameters associated with these layers are iteratively estimated using gradient descent algorithms. The output of the last layer is learned, discriminative and high dimensional representation corresponding to the input signal, and is used to feed into a logistic regression node to generate the estimated label.

The success of deep learning methods in computer vision applications can be attributed in part to the emergence of *big data*. Traditional datasets are mostly constrained to hand-annotated images or video in research labs, and have shown strong bias in terms of data selection. Approaches that generate good results on one

CRF Training Tools

	Authors	Language	Notes
LIBSVM	Chih-Jen Lin (National Taiwan University)	C++, Java, Python, Matlab	very popular choice of SVM tools, lots of documentations, many functionality supports. Another tool LIBLINEAR from the same group has a optimized implementation of SVMs with linear or polynomial kernels.
SVM-Struct	Thorsten Joachims (Cornell University)	C++, Matlab, Python	original implementation has been tailored according to a set of common tasks, including SVM-muticlass, SVM-hmm, SVM-align, and SVM-rank, etc.
Latent SVM-Struct	Chun-Nam Yu (Cornell University)	C++, Matlab	requires user implementation of the loss function as well as the inference algorithm.
CVX	Michael C. Grant & Stephen P. Boyd (Stanford University)	Matlab	restricted to small scale optimization problems; easy to use and implement; suitable for linear or quadratic programming with less than a few hundred constraints.

Table 7.1: List of software tools for CRF training.

CRF Inference Tools

	Authors	Language	Notes
TRW-S	V. Kolmogorov (IST Austria)	C++	support general graph structures with pairwise potential functions; Matlab warper available at http://www.robots.ox.ac.uk/~ojw/files/imrender_v2.4.zip .
QPBO	V. Kolmogorov (IST Austria)	C++	support higher order clique potential functions.
UGM	Mark Schmidt (ENS)	Matlab	includes a lot of decoding (or inference) methods, e.g. ICM, GraphCut, LBP, Junction Tree, TRBP; also support sampling and parameter estimation.
libDAI	Joris Mooij (University of Amsterdam)	C++	winner of UAI approximate inference challenge; limited support to Matlab / Python interface.

Table 7.2: List of software tools for CRF inference.

dataset might not beat simple baselines on others. With datasets (*e.g.* ImageNet) of a much larger scale, however, we can assume the training data are representative enough (close to real world data distribution), and therefore data bias issue can be ignored. Sophisticated methods developed on smaller datasets are difficult to scale well on such large datasets, either because of the difficulty to tune the parameters or

because of instability of the model structure. Simple methods like linear SVM or linear regression can generate relatively good performance, but they do not take advantage of enough discriminative information that would be discovered in the large scale data. Deep networks, on the other hand, contain hundreds of millions of parameters, and thus can learn discriminative patterns even on datasets of very large scale.

One argument about deep networks is its possibility of overfitting. One small datasets, there is evidence that deep learning methods do strongly overfit the data, with classification accuracies close to 100% on the training set. Even for large scale datasets, there are still possible overfitting issues: *e.g.* the DeepFace [118] system trained a deep network with 120 millions of parameters on 4 millions of training instances. This looks counterintuitive with the classical machine learning theory that the training objective function should optimize the *generalization error* rather than the *training error*. However, the assumptions behind the classical theory is that the availability of training data is limited, thus the system should care whether or not the learned function can perform well on *unseen* data. In the case of big data, these assumptions do not hold anymore since the training exemplars become quite representative, and even simple nearest neighbor classifiers can perform quite well at such a scale.

It will be interesting to see how to design graphical models with “deep” hidden layers tailored for specific applications. The deep random field model [63] is an interesting start, where a Markov random field with multiple hidden layer structure is proposed for solving the image segmentation problem. The model structure shows a strong connection with deep networks such as Deep Boltzmann Machines (DBM) [105]. We are interested in studying how to design conditional random field

models with deep hidden layers for object recognition applications.

We are also interested in exploring the three applications discussed in the thesis into more depths. We are interested in fine-grained recognition, and especially interested in building explicit 3D models for fine-grained recognition tasks (*e.g.* food recognition, clothes recognition, *etc.*), since such 3D models together with the domain knowledge will generate more useful image representations. For example, in food recognition, we can use photos of a plate of food from multiple views instead of one, and then generate a simple 3D model so that more useful image features could be extracted. We are also interested in large scale computer vision problems, *e.g.* modeling multimodal information (including visual, text, *etc.*) on the web, and devise novel training and inference algorithms that work on datasets at such a large scale. We believe that the methodology developed in this thesis foresees many possible solutions to these various object recognition problems through the use of conditional random field models.

BIBLIOGRAPHY

- [1] 11 reasons why Flickr, not Facebook, is the place to put your photos, May 2013. <http://www.xconomy.com/national/2013/05/31/11-reasons-why-flickr-not-facebook-is-the-place-to-put-your-photos/>.
- [2] Enrique Amigo, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4), 2009.
- [3] Erling Bernhard Andersen. Sufficiency and exponential families for discrete sample spaces. *Journal of the American Statistical Association*, 65(331):1248–1255, 1970.
- [4] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Neural Information Processing Systems*, 2002.
- [5] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

- [6] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:898–916, 2011.
- [7] Adrian Barbu. Learning real-time mrf inference for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1574–1581. IEEE, 2009.
- [8] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 59–68, 2004.
- [9] Ron Bekkerman and Jiwoon Jeon. Multi-modal clustering for multimedia collections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [10] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.
- [11] Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision*, 2010.
- [12] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.
- [13] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48(3):259–302, 1986.

- [14] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.
- [15] Matthew B Blaschko and Christoph H Lampert. Learning to localize objects with structured output regression. In *European Conference on Computer Vision 2008*, pages 2–15. Springer, 2008.
- [16] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
- [17] Steve Branson, Catherine Wah, Boris Babenko, Florian Schroff, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision*, 2010.
- [18] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York, 1999.
- [19] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [20] David Crandall, Pedro Felzenszwalb, and Daniel Huttenlocher. Spatial priors for part-based recognition using statistical models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10–17. IEEE, 2005.
- [21] David J Crandall and Daniel P Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *European Conference on Computer Vision*, pages 16–29. Springer, 2006.

- [22] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on statistical learning in computer vision*, page 22, 2004.
- [23] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893. IEEE, 2005.
- [24] Matthias Dantone, Juergen Gall, Christian Leistner, and Luc J. Van Gool. Human pose estimation using body parts dependent joint regressors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3041–3048, 2013.
- [25] Jason Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit Dhillon. Information-theoretic metric learning. In *International Conference on Machine Learning*, 2007.
- [26] T. Deselaers and V. Ferrari. A conditional random field for multiple-instance learning. In *International Conference on Machine Learning*, 2010.
- [27] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *European Conference on Computer Vision*, 2010.
- [28] Matthijs Douze, Arnau Ramisa, and Cordelia Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [29] Kun Duan, Dhruv Batra, and David J Crandall. A multi-layer composite model for human pose estimation. In *British Machine Vision Conference*, 2012.

- [30] Kun Duan, David J Crandall, and Dhruv Batra. Multimodal learning in loosely-organized web images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [31] Kun Duan, Luca Marchesotti, and David J Crandall. Attribute-based vehicle recognition using viewpoint-aware multiple instance svms. In *IEEE Winter Conference on Applications of Computer Vision*, 2014.
- [32] Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. Discovering localized attributes for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3474–3481. IEEE, 2012.
- [33] Marcin Eichner, Manuel Marin-Jimenez, Andrew Zisserman, and Vittorio Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*, 99(2):190–214, 2012.
- [34] Gal Elidan, Ian McGraw, and Daphne Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *Uncertainty in Artificial Intelligence*, pages 165–173, Arlington, Virginia, 2006. AUAI Press.
- [35] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [36] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

- [37] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.
- [38] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531. IEEE, 2005.
- [39] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.
- [40] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.
- [41] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
- [42] V. Ferrari and A. Zisserman. Learning visual attributes. In *Neural Information Processing Systems*, 2007.
- [43] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

- [44] Bin Gao, Tie-Yan Liu, Tao Qin, Xin Zheng, QianSheng Cheng, and Wei-Ying Ma. Web image clustering by consistent utilization of visual features and surrounding texts. In *ACM Multimedia*, 2005.
- [45] Jacob Goldberger, Shiri Gordon, and Hayit Greenspan. Unsupervised image-set clustering using an information theoretic framework. *IEEE Transaction on Image Processing*, pages 449–458, 2006.
- [46] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, v1.21. <http://cvxr.com/cvx>.
- [47] Gregory Griffin, Alex Holub, and Pietro Perona. *Caltech-256 object category dataset*. California Institute of Technology, 2007.
- [48] Chunhui Gu and Xiaofeng Ren. Discriminative mixture-of-templates for viewpoint classification. In *European Conference on Computer Vision*, 2010.
- [49] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multimodal semi-supervised learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
- [50] Kota Hara and Rama Chellappa. Computationally efficient regression on a dependency graph for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3390–3397, 2013.
- [51] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

- [52] Sung Ju Hwang, Fei Sha, and Kristen Grauman. Sharing features between objects and their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [53] Hiroshi Ishikawa. Higher-order clique reduction in binary graph cut. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2993–3000. IEEE, 2009.
- [54] F. V. Jensen. Bayesian networks and decision graphs. 2001.
- [55] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.
- [56] Mark Johnson. Pcfg models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632, 1998.
- [57] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference*, 2010.
- [58] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [59] Michael I Jordan et al. Graphical models. *Statistical Science*, 19(1):140–155, 2004.
- [60] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, Florence, Italy, October 2012.

- [61] Gunhee Kim and Antonio Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *Neural Information Processing Systems*, 2009.
- [62] Pushmeet Kohli, M Pawan Kumar, and Philip HS Torr. P3 & beyond: Solving energies with higher order cliques. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [63] Pushmeet Kohli, Anton Osokin, and Stefanie Jegelka. A principled deep random field model for image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1971–1978. IEEE, 2013.
- [64] Pushmeet Kohli, Philip HS Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.
- [65] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1568–1583, 2006.
- [66] Vladimir Kolmogorov and Carsten Rother. Comparison of energy minimization algorithms for highly connected graphs. In *European Conference on Computer Vision*, pages 1–15. Springer, 2006.
- [67] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. Mrf energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):531–552, 2011.

- [68] Josip Krapac, Florent Perronnin, Teddy Furon, and Herve Jegou. Instance classification with prototype selection. In *ACM International Conference on Multimedia Retrieval*, 2014.
- [69] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *IEEE International Conference on Computer Vision*, 2009.
- [70] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [71] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [72] Tian Lan, Yang Wang, Weilong Yang, and Greg Mori. Beyond actions: Discriminative models for contextual group activities. In *Neural Information Processing Systems*, pages 1216–1224, 2010.
- [73] Xiangyang Lan and Daniel P. Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *International Conference on Computer Vision*, 2005.
- [74] Yong Jae Lee and Kristen Grauman. Foreground focus: Unsupervised learning from partially matching images. *International Journal of Computer Vision*, 85(2):143–166, 2009.

- [75] Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Neural Information Processing Systems*, 2010.
- [76] Yunpeng Li, David Crandall, and Daniel Huttenlocher. Landmark classification in large-scale image collections. In *International Conference on Computer Vision*, 2009.
- [77] Zhenguo Li, Jianzhuang Liu, and Xiaoou Tang. Constrained clustering via spectral regularization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [78] David G Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, pages 1150–1157. IEEE, 1999.
- [79] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.
- [80] Zhengdong Lu and Miguel Carreira-Perpiñán. Constrained spectral clustering through affinity propagation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [81] Aurelien Lucchi, Yunpeng Li, and Pascal Fua. Learning for structured prediction using approximate subgradient descent with working sets. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1987–1994. IEEE, 2013.

- [82] Subhransu Maji and Greg Shakhnarovich. Part discovery from partial correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [83] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svms for object detection and beyond. In *IEEE International Conference on Computer Vision*, 2011.
- [84] Julian McAuley and Jure Leskovec. Image labeling on a network: using social-network metadata for image classification. In *European Conference on Computer Vision*, pages 828–841. Springer, 2012.
- [85] Julian J. McAuley and Jure Leskovec. Image labeling on a network: Using social-network metadata for image classification. In *European Conference on Computer Vision*, 2012.
- [86] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, pages 188–191. Association for Computational Linguistics, 2003.
- [87] Radford M Neal. *Probabilistic inference using Markov chain Monte Carlo methods*. University of Toronto, 1993.
- [88] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Neural Information Processing Systems*, 2009.

- [89] Dileep Kumar Panjwani and Glenn Healey. Markov random field models for unsupervised segmentation of textured color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):939–954, 1995.
- [90] Devi Parikh and Kristen Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [91] Devi Parikh and Kristen Grauman. Relative attributes. In *IEEE International Conference on Computer Vision*, 2011.
- [92] Dennis Park, Deva Ramanan, and Charless Fowlkes. Multiresolution models for object detection. In *European Conference on Computer Vision*, 2010.
- [93] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. Articulated people detection and pose estimation: Reshaping the future. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3178–3185, 2012.
- [94] Boris T. Polyak. A general method for solving extremum problems. *Soviet Math*, 8(3), 1967.
- [95] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden-state conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1848–1852, 2007.
- [96] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

- [97] Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verpoor, Asa Ben-Hur, et al. A large-scale evaluation of computational protein function prediction. *Nature methods*, 10(3):221–227, 2013.
- [98] Srikumar Ramalingam, Pushmeet Kohli, Kartek Alahari, and Philip HS Torr. Exact inference in multi-label crfs with higher order cliques. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.
- [99] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.
- [100] Deva Ramanan. Learning to parse images of articulated bodies. In *Neural and Information Processing Systems*, 2006.
- [101] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of American Statistical Association*, 66(336):846–850, 1971.
- [102] Nathan D Ratliff, J Andrew Bagnell, and Martin Zinkevich. (approximate) subgradient methods for structured prediction. In *International Conference on Artificial Intelligence and Statistics*, pages 380–387, 2007.
- [103] Marcus Rohrbach, Michaela Regneri, Mykhaylo Andriluka, Sikandar Amin, Manfred Pinkal, and Bernt Schiele. Script data for attribute-based recognition of composite activities. In *European Conference on Computer Vision*, 2012.
- [104] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

- [105] Ruslan Salakhutdinov and Geoffrey E Hinton. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455, 2009.
- [106] Benjamin Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3681, 2013.
- [107] Benjamin Sapp, Alexander Toshev, and Ben Taskar. Cascaded models for articulated pose estimation. In *European Conference on Computer Vision*, 2010.
- [108] Benjamin Sapp, David Weiss, and Ben Taskar. Parsing human motion with stretchable models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [109] Paul Schnitzspan, Stefan Roth, and Bernt Schiele. Automatic discovery of meaningful object parts with latent crfs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
- [110] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient Structured Prediction with Latent Variables for General Graphical Models. In *International Conference on Machine Learning*, 2012.
- [111] Gaurav Sharma, Frederic Jurie, and Cordelia Schmid. Expanded parts model for human attribute and action recognition in still images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [112] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Texton-boost for image understanding: Multi-class object recognition and segmenta-

- tion by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.
- [113] Vivek Kumar Singh, Ram Nevatia, and Chang Huang. Efficient inference with multiple heterogeneous part detectors for human pose estimation. In *European Conference on Computer Vision*, 2010.
- [114] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep Boltzmann machines. In *Neural Information Processing Systems*, 2012.
- [115] Michael Stark, Jonathan Krause, Bojan Pepik, David Meger, James J. Little, Bernt Schiele, and Daphne Koller. Fine-grained categorization for 3d scene understanding. In *British Machine Vision Conference*, Surrey, UK, 2012.
- [116] Jian Sun, Heung-Yeung Shum, and Nan-Ning Zheng. Stereo matching using belief propagation. In *European Conference on Computer Vision*, pages 510–524. Springer, 2002.
- [117] Charles Sutton and Andrew McCallum. Improved dynamic schedules for belief propagation. In *Uncertainty in Artificial Intelligence*, 2007.
- [118] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [119] Daniel Tarlow, Kevin Swersky, Richard S. Zemel, Ryan Prescott Adams, and Brendan J. Frey. Fast exact inference for recursive cardinality models. In *Uncertainty in Artificial Intelligence*, pages 825–834, 2012.

- [120] Ben Taskar, Simon Lacoste-Julien, and Michael I Jordan. Structured prediction, dual extragradient and bregman projections. *The Journal of Machine Learning Research*, 7:1627–1653, 2006.
- [121] Leonid Taycher, David Demirdjian, Trevor Darrell, and Gregory Shakhnarovich. Conditional random people: Tracking humans with crfs and grid filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–229. IEEE, 2006.
- [122] Sandra C Tomany, Ronald Klein, and Barbara EK Klein. The relationship between iris color, hair color, and skin sun sensitivity and the 10-year incidence of age-related maculopathy: the beaver dam eye study. *Ophthalmology*, 110(8):1526–1533, 2003.
- [123] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [124] Duan Tran and David Forsyth. Improved human parsing with a full relational model. In *European Conference on Computer Vision*, 2010.
- [125] Ioannis Tschantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
- [126] Ioannis Tschantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

- [127] Olga Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision*. PhD thesis, Ithaca, NY, USA, 1999. AAI9939932.
- [128] Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. In *International Conference on Machine Learning*, 2000.
- [129] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, 2001.
- [130] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. In *CVPR Workshop on Fine-Grained Visual Categorization*, 2011.
- [131] Gang Wang and David Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *IEEE International Conference on Computer Vision*, 2009.
- [132] Gang Wang, Derek Hoiem, and David A. Forsyth. Learning image similarity from Flickr groups using fast kernel machines. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pages 2177–2188, 2012.
- [133] Huayan Wang and Daphne Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [134] Josiah Wang, Katja Markert, and Mark Everingham. Learning models for object recognition from natural language descriptions. In *British Machine Vision Conference*, 2009.

- [135] Yang Wang and Greg Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *European Conference on Computer Vision*, 2008.
- [136] Yang Wang and Greg Mori. A discriminative latent model of object classes and attributes. In *European Conference on Computer Vision*, pages 155–168. Springer, 2010.
- [137] Yang Wang, Duan Tran, and Zicheng Liao. Learning hierarchical poselets for human parsing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [138] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [139] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, 2013.
- [140] Yi Yang, Dong Xu, Feiping Nie, Shuicheng Yan, and Yuetong Zhuang. Image clustering using local discriminant models and global integration. *IEEE Transaction on Image Processing*, pages 2761–2773, 2010.
- [141] Bangpeng Yao, Aditya Khosla, and Li Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

- [142] Chun-Nam John Yu and Thorsten Joachims. Learning structural svms with latent variables. In *International Conference on Machine Learning*, pages 1169–1176. ACM, 2009.
- [143] Jun Yu, Meng Wang, and Dacheng Tao. Semisupervised multiview distance metric learning for cartoon synthesis. *IEEE Transaction on Image Processing*, pages 4636–4648, 2012.
- [144] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, and Zengfu Wang. Joint multi-label multi-instance learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [145] Ning Zhang, Manohar Paluri, Marc’Aurelio Ranzato, Trevor Darrell, and Lubomir Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [146] Xin Zheng, Deng Cai, Xiaofei He, Wei Ma, and Xueyin Lin. Locality preserving clustering for image database. In *ACM Multimedia*, 2004.
- [147] Long Zhu, Yuanhao Chen, Yifei Lu, Chenxi Lin, and Alan L. Yuille. Max margin AND/OR graph learning for parsing the human body. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

Kun Duan

Email: kduan@indiana.edu

EDUCATION *Doctor of Philosophy*, Indiana University, Bloomington, IN 2009 - 2014

Major: Computer Science

Minor: Mathematics

Thesis Advisor: Prof. David J. Crandall

Master of Science, Indiana University, Bloomington, IN 2009 - 2011

Major: Computer Science

Bachelor of Arts, Beijing University of Posts and Telecommunications, Beijing, China 2005 - 2009

Majors: English Language, Computer Science

EXPERIENCE Software Developer Intern 2013.06 - 2013.09

Amazon A9.com Inc., Visual Search Team, Palo Alto, CA

- Face detection and recognition in videos.

Research Intern 2013.02 - 2013.06

Xerox Research Centre Europe, Computer Vision Group, Grenoble, France

- Large scale semantic object recognition using visual attributes.

Research Intern 2012.05 - 2012.08

eBay Research Labs, Computer Vision Group, San Jose, CA

- Shape-based fashion object retrieval.

Research Intern

2011.06 - 2011.08

Toyota Technological Institute at Chicago, Chicago, IL

- Attribute-based fine-grained object recognition.
- Human pose estimation using part-based models.

PUBLICATIONS

- **Kun Duan**, Dhruv Batra, David Crandall. “Human Pose Estimation via Multi-layer Composite Models”, *Elsevier Signal Processing (Special Issue on Machine Learning and Signal Processing for Human Pose Recovery and Behavior Analysis)*, Impact Factor=1.851, accepted (to appear).
- **Kun Duan**, David Crandall, Dhruv Batra. “Multi-modal Learning in Loosely-organized Web Images”, *IEEE conference on Computer Vision and Pattern Recognition (CVPR) 2014*.
- **Kun Duan**, Luca Marchesotti, David Crandall. “Viewpoint-aware Multiple Instance SVMs for Vehicle Recognition”, *IEEE Winter Conference on Applications of Computer Vision (WACV) 2014*.
- **Kun Duan**, Dhruv Batra, David Crandall. “A Multi-layer Composite Model for Human Pose Estimation”, *British Machine Vision Conference (BMVC) 2012*.
- **Kun Duan**, Devi Parikh, David Crandall, Kristen Grauman. “Discovering Localized Attributes for Fine-grained Recognition”, *IEEE conference on Computer Vision and Pattern*

Recognition (CVPR) 2012.

- Lei Zhang, Xiang-wu Meng, Jun-liang Chen, Si-cheng Xiong, **Kun Duan**. “Alleviating Cold-Start Problem by Using Implicit Feedback”, *The 5th International Conference on Advanced Data Mining and Applications (ADMA) 2009, Lecture Notes in Computer Science, Springer*. 2009.
- Lei Zhang, Xiang-wu Meng, Jun-liang Chen, **Kun Duan**, Yong Peng. “Personalized service recommendation algorithm”, *The 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT) 2009, Lecture Notes in Computer Science, Springer*. 2009.
- Lei Zhang, Xiang-wu Meng, Jun-liang Chen, Xiaoyan Shen, **Kun Duan**. “BP Neural Network-based Collaborative Filtering Recommendation Algorithm”, *Journal of Beijing University of Posts and Telecommunications, Vol.6* 2009.

PATENTS

- **Kun Duan** and Luca Marchesotti. “Object Classification With Constrained Multiple Instance Support Vector Machine”, US Patent Filed (Serial No.: 14/14/186,337)

INVITED TALKS

- *Conditional Random Field Models for Structured Object Recognition*
 - *GE Global Research, Schenectady, NY, USA* 2014.05
 - *SRI International, Princeton, NJ, USA* 2014.04
 - *Computer Vision Group, Amazon Inc., Seattle, WA, USA* 2014.03

- *Samsung Advanced Institute of Technology, Beijing, China*
2014.01
 - *Intelligent Media Technology Lab, Xiamen University, Xiamen, China* 2014.01
 - *Microsoft Research Asia, Beijing, China* 2013.12
- Towards Semantic Visual Recognition
 - *Xerox Research Centre Europe (XRCE) Computer Vision Group, Grenoble, France*, 2013.05
 - *Indiana University Intelligent Systems Seminar, Bloomington, IN*, 2012.10

SERVICES

- Reviewer for BMVC 2014, ACCV 2014, CVIU, IET Computer Vision, GEOMM 2014, ICVGIP 2014
- IEEE Student Member
- Student Volunteer for CVPR 2012, CVPR 2014

AWARDS

- Best Project Award, Intelligent Robotics Open House, Indiana University 2014
- WACV Conference Travel Grant 2014
- XRCE ID&Patent Award 2013
- Undergraduate Scholarship, BUPT 2006 - 2008
- 2nd Prize in National High School Mathematical Contest 2004

REFERENCES

- Prof. David Crandall
 - Assistant Professor of Computer Science, School of Informatics and Computing, Indiana University
 - djcra@indiana.edu
- Dr. Luca Marchesotti
 - Research Scientist, Xerox Research Centre Europe
 - luca.marchesotti@gmail.com
- Prof. Devi Parikh
 - Assistant Professor, Bradley Department of Electrical and Computer Engineering, Virginia Tech
 - parikh@vt.edu