Bag of Machine Learning Concepts for Visual Concept Recognition in Images

vorgelegt vom
Diplom-Mathematiker
Alexander Binder
aus Berlin

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Olaf Hellwich
1. Gutachter:	Prof. Dr. Klaus-Robert Müller
2. Gutachter:	Prof. Dr. Volker Tresp
3. Gutachter:	Prof. Dr. Marc Toussaint

Verteidigung geschehen am 27.02.2013

Berlin, 2013
D83

Abstract

My thesis deals with the recognition of visual concepts on images using statistical machine learning. Recognition is treated here as classification task with continuous predictions. The continuous predictions can be used to generate a ranking of images and thus will be often evaluated in a ranking setting. Ranking means that for a given visual concept the set of all test images will be sorted according to the prediction in a descending order and evaluated using a ranking measure. This dissertation treats the general case of visual concepts in which concepts are defined explicitly by a set of images. The aim is multi-label classification in which for one image all present concepts are to be predicted. The challenge compared to highly specialized tasks such as face recognition is the ability to deal with a generic set of visual concepts which are defined by the training data.

Classification is based on kernel methods such as extensions of support vector machines. The features are predominantly bag of visual words (BoW) which yield superior results for visual concept recognition on images with generic concepts as demonstrated constantly over the last years by the results of international benchmark competitions such as Pascal VOC classification and ImageCLEF Photo annotation. The problem of classification and ranking of a generic set of visual concepts can be divided into three subtasks: *Formulation of the problem and design or choice of a corresponding loss function*, the *Learning of feature combinations given a loss function* and the *Design of Features*. My publication record contains co-authored work on all subtasks. This dissertation contains contributions for the first two subtasks.

In the first part of the dissertation I consider (for the aspect of *Formulation of the problem and design or choice of a corresponding loss function*) models which

are capable of minimizing hierarchical loss functions which are induced by taxonomies over the set of all visual concepts. The idea is that a taxonomy defines a prioritization of classification and ranking errors. The goal is to avoid errors which originate from confusing concepts which are distant under the given taxonomy. One example is a system which annotates images such that it returns for a request of dogs in case of absence of dogs or in case of error rather images of cats than images of cars.

In contrast to preceding publications the focus lies not on speed during testing time but on improved classification and ranking performance under the hierarchical loss. The developed model aggregates the votes of all edges in the taxonomy, not only those of the locally best or shortest path. Furthermore the hierarchical models are generalized such that they can be predict multiple labels for multi-label ranking problems in which each image can have more than one visual concept. Previous approaches based on greedy walks along the edges of the hierarchy are able to predict only the most likely concept. In the context of multi-label ranking we define also a ranking measure which incorporates taxonomical information. The developed model is compared against one-versus-all and structured prediction baselines.

In the second part of the dissertation I analyze (for the aspect of *Learning of feature combinations given a loss function*) the non-sparse multiple kernel learning (MKL) for multi-label ranking of images. It is compared against average kernel support vector machines (SVMs) and sparse ℓ_1-norm MKL. For the empirical part I evaluate the performance of these methods on the Pascal VOC2009 Classification and ImageCLEF2010 Photo Annotation datasets. It is shown that when using model selection in a practical setup, non-sparse MKL yields equal or better results compared to the average kernel SVM which does not learn feature combinations, in contrast to sparse ℓ_1-norm MKL which yields worse results. For the theoretical part we identify limiting and promoting factors for the performance gains of non-sparse MKL when compared to the other methods.

The dissertation is closed by an outlook section.

Abstract

Meine Dissertation behandelt Probleme der Erkennung visueller Konzepte auf Bildern mit Hilfe von Methoden des statistischen maschinellen Lernens. Ziel der Erkennung im Rahmen meiner Dissertation ist es, einem Bild für jedes visuelle Konzept einen reellen Wert zuzuweisen, dessen Grösse einer (nicht probabilistischen) Konfidenz in das Vorhandensein des Konzeptes in diesem Bild entspricht. Derartige reellwertige Vorhersagen können für Klassifikation von Bildern und für die Rangsortierung benutzt werden. Unter Rangsortierung wird in dieser Arbeit die Anordnung der Bilder entsprechend der Konfidenzen für ein vorgegebenes Konzept verstanden, welche zum Beispiel als Ausgabe einer Suchmaschine genutzt werden könnte.

Diese Dissertation behandelt den allgemeinen Fall, bei dem im Kontext der Klassifikation ein visuelles Konzept implizit definiert werden kann durch die Vorgabe einer Menge von Bildern, die ein solches Konzept aufweisen. Ziel ist die sogenannte multi-label Klassifikation, bei der zu einem Bild alle dort vorhandenen visuellen Konzepte aus der vorgebenenen Menge aller visuellen Konzepte vorhergesagt werden sollen. Die Herausforderung im Unterschied zu hochspezifischen Aufgaben wie der Gesichtserkennung liegt darin, dass die Menge der visuellen Konzepte durch die Trainingsdaten frei vorgegeben werden kann und daher generisch ist.

Zur Klassifikation werden kern-basierte Methoden aufbauend auf support vektor Maschinen verwendet. Als Merkmale werden überwiegend sogenannte Histogramme über visuellen Wörtern verwendet (bag of words). Die Kombination von Histogramme über visuellen Wörtern und nichtlinearen repräsentiert den Stand der Technik im Bereich der Klassifikation von generischen visuellen Konzepten,

was durch internationale Wettbewerbe wie Pascal VOC Classification und ImageCLEF Photo Annotation alljährlich demonstriert wird. Das Klassifikationsproblem in seiner Gesamtheit kann in drei Teilprobleme unterteilt werden: die *Formulierung des Problems sowie die Auswahl der Verlustfunktion*, das *Lernen einer Kombination von Merkmalen* mit dem Ziel eine Verlustfunktion zu minimieren und die *Merkmalsextraktion*. Die Liste der von mir mitverfassten Publikationen weist Arbeiten zu allen Teilproblemen auf. Diese Dissertation leistet Beiträge zu den ersten zwei Teilproblemen.

Im ersten Teil der Dissertation werden im Rahmen des Entwurfs von Verlustfunktionen Modelle betrachtet, die hierarchische Verlustfunktionen minimieren können, welche durch Taxonomien auf über der Menge der visuellen Konzepte definiert werden. Die Idee besteht in der Nutzung einer Taxonomie als Priorisierung von Klassifikations- oder Rangsortierungsfehlern. Ziel ist es dabei, dass das Modell Vorhersagefehler vermeidet, die durch Verwechselung von in der Taxonomie weit voneinander entfernten Konzepte verursacht werden. Sollen z.B. Bilder von Hunden gefunden werden, kann dieses Ziel erreicht werden, indem im Falle statistischer Unsicherheit eher Bilder von verwandten Tieren, wie z.B. Katzen, anstelle von Autos oder Fernsehern als Ergebnisse präsentiert werden.

Im Unterschied zu vorangegangenen Publikationen liegt der Schwerpunkt nicht auf Geschwindigkeit zum Zeitpunkt der Evaluation eines Bildes, sondern auf verbesserter Rangsortierungs- und Klassifikationsgenauigkeit. Dazu werden die Vorhersagen aller Kanten im Taxonomie-graphen mit Hilfe von sogenannten p-means kombiniert anstelle wie bei vorangegangenen Arbeiten nur die lokal optimalen Kanten. Des weiteren werden die hierarchischen Modelle derart verallgemeinert, dass sie für Multilabel Probleme, bei denen jedes Bild mehrere visuelle Konzepte aufweisen kann, alle vorhandenen visuellen Konzepte vorhersagen können. Bisherige Ansätze, welche nur dem lokal optimalen (kürzesten) Pfad entlang der Kanten der Taxonomie folgen, können pro Bild nur ein visuelles Konzept erkennen. In diesem Zusammenhang wird auch ein taxonomie-basiertes Rangsortierungsmass definiert, welches Information aus der Taxonomie zur Berechnung der Genauigkeit

der Rangsortierung verwendet. Die entwickelten Verfahren werden gegen strukturierte Vorhersagemodelle und einer-gegen-alle Klassifikationsmodelle verglichen.

Im zweiten Teil der Dissertation werden im Rahmen des Lernens der Kombination von Merkmalen das non-sparse multiple kernel learning (MKL) auf dem Rangsortierungsproblem auf Bildern untersucht und gegen support vektor maschinen mit einem gemittelten Kern, welche keine Kombination von Merkmalen lernen, und dem ℓ_1-Norm multiple kernel learning, welches nur eine sehr kleines Anzahl von Merkmalen auswählt, verglichen. In empirischer Hinsicht wird dies auf den Datensätzen der PASCAL VOC 2009 Classification and ImageCLEF2010 Photo Annotation Wettbewerbe durchgeführt. Es wird gezeigt, dass das non-sparse MKL unter Praxisbedingungen bei Durchführung von Modellselektion gleich gute oder bessere Ergebnisse als support vektor maschinen mit einem gemittelten Kern liefert, im Unterschied zu ℓ_1-Norm MKL, welches oft schlechtere Ergebnisse liefert als die support vektor Maschinen mit einem gemittelten Kern, welche keine Kombination von Merkmalen lernen.

In theoretischer Hinsicht werden Faktoren identifiziert, die dazu führen, dass support vektor Maschinen mit einem gemittelten Kern gute Ergebnisse liefern, und untersucht, welche Faktoren potentielle Verbesserungen durch das Lernen der Kombination von Merkmalen begrenzen und welche Faktoren dazu führen, dass das non-sparse MKL im Schnitt etwas bessere Ergebnisse liefert.

Die Arbeit wird durch einen Ausblick abgeschlossen.

Acknowledgements

To some people whom I like to thank. My son Malte who can make me smile with his astonishing emotional intelligence and cuteness! Maybe he will dislike that sentence when he grows older. I quote him: ”I know I should not hit the tree with the ball ... but I will do that now anyway!”. My close friends from school times, particularly Giso and Falko. My supervisor Klaus, who was more than one time crucial in keeping my spirits up, and who encouraged me to continue in critical moments, my longterm boss Motoaki. Wojciech Samek and Marius Kloft, who were my main computer vision collaborators. Frederick Klauschen, one of the very few medical doctors who does not fear to use a (Linux) shell. Babette Neumann who kept my back free. Marco Feiler from study times. Gunnar Kedenburg from work. The admins Dominik Kuehne, Roger Holst and Rolf Schulz who suffered from my computational efforts and were crucial in my successes. Tammo Krüger, Daniel Bartz and Sebastian Bach. Many collegues with whom I could chat from time to time. Finally I would like to thank a non-human in the end. No, it is not a manga girl or a first person shooter character as one could assume for a guy doing a PhD thesis in an IT-related field. It is not even a pizza brand (I like asian food more) or a beer company (I don’t like to drink beer). It is the THESEUS project funded by BMWi which had funded my position for five years, gave me opportunity to gain experience with project management, gave me the opportunity to participate in established international benchmark challenges such as Pascal VOC and ImageCLEF PhotoAnnotation and allowed me together with the THESEUS administrators to travel to top-level computer vision and medical IT conferences such as ICCV, ACCV and MICCAI.

Contents

List of Figures

LIST OF FIGURES

List of Tables

1

Introduction

1.1 Problem Description of Semantic Concept Recognition in Images

At first I will define the problem which I have been working on.

1.1.1 What defines a Semantic Concept

Formally a semantic concept can be represented by an indicator function $\mathbb{I}_C$ on the space of all images $\mathcal{X}$ such that $\mathbb{I}_C(x) = 1$ denotes the presence of concept C in an image $x \in \mathcal{X}$.

$$\mathbb{I}_C : \mathcal{X} \longrightarrow \{0, 1\} \tag{1.1}$$

For ambiguous semantic concepts this definition can be extended by assigning an image x a score $l_C(x)$ in a bounded interval (e.g. $[0, 1]$) which represents a numerical value for the strength of the presence of a semantic concept in an image:

$$l_C : \mathcal{X} \longrightarrow [0, 1] \,. \tag{1.2}$$

This numerical value can be interpreted in a probabilistic manner as the agreement of a set of human annotators with respect to the question whether an image belongs to a semantic concept or not. In the context of classification this is known as label noise. In a probabilistic model of classification with $\mathcal{X}$ being the space of all images and $\mathcal{Y} = \{0, 1\}$ being the label for a semantic concept C this setting can be modeled by a joint distribution $P_C : \mathcal{X} \times \{0, 1\} \to [0, 1]$. The label noise is related to the prediction certainty $P_C(Y = 1 \mid X = x) = P(\mathbb{I}_C(x) = 1)$ which can be used to define the score $l_C(x)$ in Equation 1.2. Such ambiguities arise naturally for concepts

denoting the emotional impression of an image such as the concepts *scary*, *euphoric* or *calm* in the ImageCLEF2011 Photo Annotation dataset (1) or concepts related to aesthetic quality. The label noise plays an important role in the question why image annotation is inherently difficult and its impact on model selection be treated in more detail in section 1.2.2.

1.1.2 Two Modes of Semantic Concept Recognition

Semantic Concept Classification Given a semantic concept C a binary prediction function f_C acting on the set of all images $\mathcal{X}$ can be employed for semantic concept classification:

$$f_C : \mathcal{X} \longrightarrow \{0, 1\} \tag{1.3}$$

One application derived from it is automatic tagging of image collections based on pre-defined semantic concepts.

Semantic Concept Ranking Given a semantic concept C a continuously-valued prediction function f_C acting on the set of all images $\mathcal{X}$ can be employed for semantic concept ranking. The importance of semantic concept ranking lies in its application to the most relevant images for a semantic concept from a large set of images. This is the classical search engine paradigm and the aim of many search engines.

1.2 What makes semantic concept classification and ranking of images a challenging task?

One may ask why common internet search engines employ image search based on filenames as the default tool while search based on visual content appears to be in the beta phase at best. In this section we discuss issues and challenges of semantic concept classification for general semantic concepts.

We are interested in predicting a large set of generic semantic concepts in contrast to a small set of highly specialized concepts as it is the aim of face recognition as an example. One image may show multiple concepts. Figure 1.1 shows an example image from the ImageCLEF2011 Photo annotation dataset and all of its annotated visual concept labels. Note that this kind of annotation is far away from multi-class classification scenarios in which each image has at most one visual concept present in it, this images was labeled with 13 visual concepts. The prediction output is desired to be a continuous score usable for ranking purposes. The

continuous score allows to provide information about uncertainty of the classification. Such information is highly useful for the common search scenario in which a user is interested to find the K most likely images for a selected concept.

Figure 1.1: An example image from the ImageCLEF2011 Photo annotation dataset and its set of visual concept labels: *Outdoor, Plants, Day, Still Life, Neutral Illumination, Partly Blurred, No Persons, Park Garden, Toy, Natural, Cute, Funny, Calm*

1.2.1 Variability in the Structure of Semantic Concepts

The question "What defines a semantic concept" raised in the title of Section 1.1.1 can be interpreted in an alternative way as the an attempt to give an overview of the constituting elements of a semantic concept in a less mathematical sense, more driven by visual content. What kind of semantic concepts do we expect to observe and what kind would we like to be able to deal with?

One well known type are semantic concepts defined by the presence of a member of class of objects, e.g. *Porsche*, *Car* or *four-wheeled vehicle*. This is classic object recognition as proposed by the seminal Caltech101 dataset (2). In order to define the term object recognition we may say an object is a physical object of limited extent for which we can put a bounding box in a photo around large parts of it.

Another type of semantic concepts are more abstract ones defined by the presence of several visual cues in the image. The difference to object recognition is that the visual cues may vary highly and may not be classified into one object class in the above sense. Consider the concept *Concert*. Photos showing a small group of people known to be famous music artists on stage

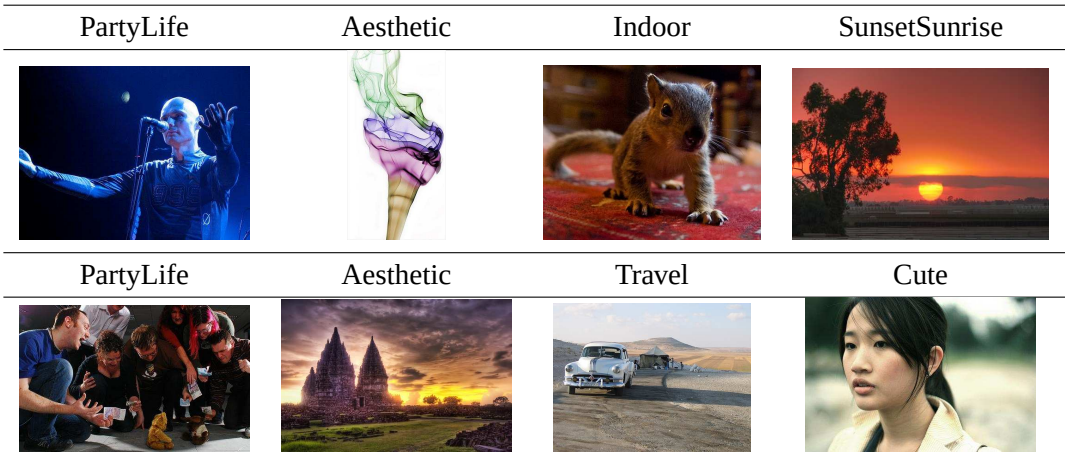


Figure 1.2: Some Concepts from the ImageCLEF 2011 Photo Annotation Challenge and example images.

are likely to belong to such a concept. At the same time a large group of hobby artists playing in an orchestra also defines a *Concert*.

Composition of cues beyond mere presence may play an important role: A person holding a guitar in a certain pose may contribute to the classification as a *Concert*. However another pose with a guitar on his back may depict rather a travelling person not involved in concert activities. Two people with a guitar in a different pose can have the meaning that some guy is smashing a paparazzo with a guitar unrelated to a concert scene. Similarly, music at a funeral scene is less likely called a *Concert*. One can think of many setups of musical instruments and people which are more or less likely to be a *Concert*.

One can extend this to abstract concepts which require the presence of several varying cues and the *absence* of certain cues. Consider the semantic concept *PartyLife*. Three people sticking together do not make a party – if they show faces full of grief or anger human annotators would hardly rate it to be a *PartyLife* scene. Similarly a lonely guy playing guitar at a campfire in the woods might not be a *Concert*.

This reveals that general semantic concepts are more difficult to recognize compared to classic single object recognition. Another reason besides the wide range of possible cues is that cues contribute in a non-deterministic way to the rating for belonging to a semantic concept. Consider the concept *StreetScene*: the presence of roads and buildings are cues for such a concept however the density and height of buildings, density of roads and the density of parked cars are important for judging whether this is a *StreetScene* or just a lonely road outside a town

with some buildings. If a probabilistic model contains only binary variables for the presence of roads or buildings, then these variables will likely contribute in a non-deterministic manner to the concept of a *StreetScene*. This probabilistic contribution of cues and their composition becomes obvious for concepts related to aesthetic quality or emotional impact such as *Funny* or *Scary*.

Figure 1.3: Left: Macro of a fly; Middle: *Not* a macro of an elephant; Right: Macro of an Elephant. Images by courtesy of wikimedia users nachu168, Fruggo and Alexander Klink.

Finally, some concepts require to have prior knowledge about properties of depicted cues which cannot be extracted ad hoc from the single image. Figure 1.3 gives an example. The concept *MacroShot* of an elephant looks different from the *MacroShot* of a fly. A macro image of a fly usually shows large parts of a fly while a macro image of an elephant can never show the whole elephant due to its elephantous size. The objects of interest fill roughly the same area in the left and middle images of Figure 1.3, however the middle image is not a macro shot. A macro of an elephant will rather show only a smaller piece of elephant skin like the right image in Figure 1.3. At least, there exists a theoretical replacement for prior knowledge in the framework of statistical learning: increasing numbers of training samples may overcome the lack of information in the single image.

The reader may note that this discussion starts to get messy because we left the domain of mathematical description and definition which yielded clear results in Section 1.1.1.

The conclusion from this confusion is that we observe a large variability in the *semantic* structure of semantic concepts. This presents a challenge for algorithms designed to predict semantic concepts and rank images according to them. The variability of a semantic concept can be defined in mathematical terms as a statistical variance over the set of images belonging to this concept computed by any kind of function which takes the pixels of a single image as an input. Key factors for the variance in the semantic structure of a semantic concept are the presence and absence of a wide range of visual cues, their composition and their contribution

to the classification of an image in a non-deterministic manner. This is what makes search for images based on filenames a task which is easier to solve than image search by visual cues.

We can identify some special cases of the variability of cues which we will be explain briefly in the next subsections.

Figure 1.4: Bottles in varying positions and sizes. Images from the PASCAL VOC 2009 challenge dataset.

Varying positions and sizes of Regions in an image relevant for a semantic concept When limited to objects one will note that an object can fill a large fraction of the image or a very small region. An smaller object may have a highly varying position within the image as shown in Figure 1.4 for the semantic concept *Bottle*. Similarly the appearance of an object may vary with its viewpoint. The same holds for cues contributing to a semantic concept.

Occlusion of Regions in an image relevant for a semantic concept Regions of an image relevant for the recognition of a semantic concept can be occluded. This is easy to understand for occluded objects shown in Figure 1.5.

Figure 1.5: Occluded objects. From left to right: airplane, bus, car and car. Images from the PASCAL VOC 2009 challenge dataset.

Clutter and Complex Scene Compositions Images can have large areas which are at least in part irrelevant for the classification of a semantic concept. The leftmost three images in Figure 1.4 may serve as an example, the bottles are embedded in complex sceneries which are not necessarily related to bottles.

1.2.2 The Impact of Label Noise on Model Selection

The points discussed above may have two effects on increasing the difficulty of the semantic concept classification problem. The first effect in a probabilistic classification setting is, given a fixed feature space, an increased complexity of the Bayes boundary[1]. The second effect is increased label noise.

Label noise can be measured as the uncertainty of human annotators in assigning an image to belong to a semantic concept. Mathematically it can be modelled as the probability of an image to belong to a concept $P(\mathfrak{I}_C(x) = 1)$.

Note that the notion of label noise is not disjoint from the preceding discussions. From a semantic viewpoint label noise can arise from occlusions of an object or transformations such that some human annotators will tend to reject the presence of a semantic concept based on their own definition, judgement or in case of concepts related to emotions or artistic quality, their perception.

We expect less ambiguity and label noise for object-based concepts such as *bicycle* than for concepts defined by a sentiment such as *Sad* or a very abstract notion like *technical*, *travel* or *work*.

Label noise has an obvious deteriorating impact on classification accuracy, and more importantly on model selection. Learning a support vector machine (3, 4, 5) by solving its optimization problem corresponds to the selection of a function from a class of functions by selecting support vectors, their weights and the bias when solving the SVM optimization problem. The selection of a function from a class of hypotheses by minimizing a regularized loss over a finite set of training samples can be treated in the framework of empirical risk minimization.

Theorem 6 in (6) provides lower bounds for the expected risk in empirical risk minimization depending on a uniform bound for the label noise.

Theorem 1 (Theorem 6 from (6))**.** *Let μ be a probability measure on $\mathcal{X}$ and S be some class of classifiers on $\mathcal{X}$ such that for some positive constants K_1,K_2,ϵ_0 and r*

$$K_2\epsilon^{-r} \leq H_1(\epsilon, S, \mu) \leq K_1\epsilon^{-r}$$

for all $0 < \epsilon \leq \epsilon_0$, where $H_1(\epsilon, S, \mu)$ denotes the $\ell_1(\mu)$-metric entropy of S. Furthermore let h be a bound on the label noise:

$$\forall x \; |P(Y = 1|X = x) - 0.5| \geq h/2$$

[1] the Bayes boundary is the optimal decision boundary for classification when the generating distribution of the data is assumed to be known.

Then, there exists a positive constant K depending on K_1,K_2,ϵ_0 and r such that the following bound holds

$$\begin{aligned} R_n(h,S,\mu) &= \inf_{\hat{s}\in S} \sup_{P\in\mathcal{P}(h,S,\mu)} \mathbb{E}[P(Y\neq\hat{s}(X)) - P(Y\neq s^*(X))] \\ &\geq K(1-h)^{\frac{1}{1+r}} \max(h^{-\frac{1-r}{1+r}} n^{-\frac{1}{1+r}}, n^{-\frac{1}{2}}) \end{aligned} \tag{1.4}$$

whenever $n \geq 2$.

The work in (7) contains examples how to establish the validity of the imposed condition on $H_1(\epsilon, S, \mu)$ for smoothly differentiable Bayes boundaries. This allows to apply it to support vector machines with Gaussian kernels and otherwise smooth settings like bounded domains and distributions with sufficiently smoothly differentiable Bayes boundaries[1]. For the understanding of the theorem note that $\mathcal{P}(h, S, \mu)$ is the set of distributions on the input-label product space $\mathcal{X} \times \mathcal{Y}$ such that the input space distribution is μ. Furthermore the label noise is bounded in each point of $\mathcal{X}$ by $1/2 - h/2$ due to $|P(Y = 1|X = x) - 0.5| \geq h/2$. Finally, s^* is the Bayes classifier. $\mathbb{E}[P(Y \neq \hat{s}(X)) - P(Y \neq s^*(X))]$ is the deviation between the expected errors of the classifier s and the a posteriori optimal Bayes classifier s^*. The supremum is taken over a class of distributions followed by selection of the optimal empirical classifier $\hat{s}$ given knowledge of the distribution. Since the distribution is unknown this implies that the lower bound has an optimistic formulation compared to practice.

An increase in the overall label noise corresponds to a decrease of the value of h which yields an increased lower bound in Theorem 1 for the expected deviation between the expected error of an optimistically selected classifier and the best possible classifier within a function class. The qualitative message is that label noise does have a deteriorating influence on model selection.

1.3 State of the art in Semantic Concept Recognition in Images

Image Annotation as a tool for content-based image retrieval is a field of ongoing research since decades. The reader is referred to the overview paper (8) for the numerous research efforts undertaken in the last century alone.

Image annotation follows two big lines, generative approaches based on a probabilistic model and discriminative approaches aiming at minimizing a loss function.

[1] for a brief introduction to support vector machines see Section 1.3.2

Among the discriminative approaches kernel-based methods such as support vector machines (3, 4) or kernel discriminant analysis (9) based on BoW (bag of words) features (10) have been proven particularly successful in the field of image annotation and ranking. Kernels computed over BoW features are constantly dominating international competitions on image annotation and ranking in terms of performance measures such as the PASCAL Visual Object Categorization (11) and the ImageCLEF PhotoAnnotation challenges (1, 12) over the last years. Thus they will be the fundament of the work described in this thesis. The following sections 1.3.1 and 1.3.2 will give a short introduction into BoW features and support vector machines (SVMs).

The state of the art for Semantic Concept Recognition in Images is based on computing many features for each image. When considering a larger set of many different semantic concepts it may be very difficult to construct the one ultimate feature for classifying them all reliably. The basic idea is to counter the high variability and complexity of general semantic concepts described in Section 1.2.1 by computing many different features per image and if necessary learning combinations of them adapted to the semantic concept to be classified. This is the main reason to compute many features per image.

It is worth to remark about a very recent development. While it was known before that neural nets are very suitable for object classes with rigid structure such as the CIFAR datasets (13) which do not have a high scale variance and are centered, recent results using neural nets with additional regularization ideas yielded excellent performance on problems with much more diverse visual concepts such as the Imagenet Challenge (14, 15). From that we may expect a revival of neural networks for general visual concept recognition in the next years.

1.3.1 Bag of Word Features

The Bag of Word (BoW) feature is a framework rather than a fixed feature computation algorithm useful for computing a vector-valued representation for one image which can be used for subsequent classification and ranking. Intuitively speaking it looks at many parts of the image, each of them represented by a local feature and aggregates the local features into one global representation for the image which is the final bag of Word feature. The most notable property of the BoW framework is the fact that the spatial relations between local features are ignored.

Figure 1.6 shows the stages of computing a BoW feature.

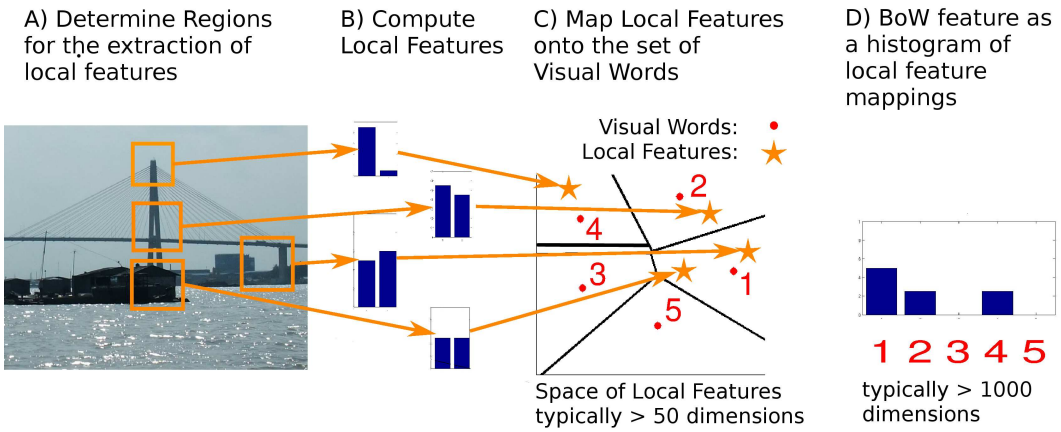


Figure 1.6: Bag of Word Feature Computation pipeline.

First Stage: Local Features In the first stage (left part of figure 1.6) a set of local features is computed from an image. Formally, a local feature is a vector computed over a region of the image by some fixed algorithm. In Figure 1.6 the local feature is for the sake of demonstration merely composed of the gradient norms along the horizontal and vertical axes which results in two dimensions. For real applications the SIFT descriptor (16) is the most famous choice for general multimedia images. Besides the choice of the local feature, regions for its computation have to be chosen. Typically, local features are computed on small overlapping regions distributed across the whole image. Apart from grid sampling as the simplest method, biased random sampling (17, 18, 19) may serve for the computation of the corresponding descriptor regions. The number of local features may vary across images, for example by adaptation to image size. The work in (20, 21) shows that a sufficiently dense sampling is required for good classification performance which is the reason why for image classification, in contrast to object matching across images, classic keypoint based detectors yielded somewhat lesser performance as demonstrated in the Pascal VOC 2007 Challenge (22). This is consistent to the author's own experience.

For improvement of performance local features are often computed over a set of different color channels and concatenated (23). This allows to incorporate color information and correlations between various color channels. We assume in the following that the images are available as digital RGB-images with color channels red, green and blue with values lying in $[0, 1]$. Examples for such sets of color channels are the basic set of red, green, and blue (RGB),

the set (OPP) composed of the three channels grey (1.5), opponent color 1 (1.6) and opponent color 2 (1.7), the normalized RGB set (1.8) (nRGB) or the normalized opponent colors set (nOPP) (1.9). The latter color channels are given in Equations (1.5),(1.6),(1.7),(1.8) and (1.9) as functions of RGB-values (r, g, b) lying in $[0, 1]$.

$$gr(r,g,b) = (r+g+b)/3 \tag{1.5}$$

$$o1(r,g,b) = (r-g+1)/2 \tag{1.6}$$

$$o2(r,g,b) = (r+g-2b+2)/4 \tag{1.7}$$

$$nrngnb(r,g,b) = \begin{cases} \left(\frac{r}{r+g+b}, \frac{g}{r+g+b}, \frac{b}{r+g+b}\right) & \text{if } r+g+b>0 \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

$$nopp(r,g,b) = \begin{cases} \left(gr(r,g,b), \frac{o1(r,g,b)}{gr(r,g,b)}, \frac{o2(r,g,b)}{gr(r,g,b)}\right) & \text{if } r+g+b>0 \\ 0 & \text{otherwise} \end{cases} \tag{1.9}$$

The idea of computing features over sets of color channels and subsequently concatenating them is applied also to other feature extraction algorithms as well.

Second Stage: Visual Words The second stage, the computation of the set of visual words, which is not shown in figure 1.6, is done once during training time for each BoW feature to be computed.

It is important to understand that BoW features cannot be computed in a classic paradigm in which a feature is a function of an image alone, because the BoW histograms are defined relative to the set of visual words which must be obtained in some way, usually from training images. The BoW features are a function of the image *and* the visual words. After having computed visual words from training images, BoW features can be computed for training and testing data using the same fixed set of visual words for both datasets. A change in the visual words requires to recompute the BoW histograms for all images.

Formally, a visual word is merely a point in the space of the local features. Figure 1.6 depicts exemplarily the two-dimensional local feature space with red dots as the five visual words. One possibility to compute the visual words is discretization of the empirical local feature density using k-means. Practically proven alternatives are radius-based clustering (20), Bayesian methods like pLSA (24) and more commonly Fisher vectors based on Gaussian mixture models (25), sparse coding (26). It is an open question for what kind of data a density-based method like k-means is preferable over a radius-based method like radius-based clustering (20).

Third Stage: Mapping of Local Features onto Visual Words The third stage is the mapping of local features onto the visual words, usually by computing weights based on the distances between the local feature and all the prototypes. This step, depicted in the middle of figure 1.6 yields for each local feature a vector of weights with its dimensionality being equal to the number of prototypes in the visual codebook. Examples are soft codebooks (27) and fast local linear coding (28).

There has been considerable research on improvements for the two steps of visual word generation and mapping, such as hierarchical clustering (29), class-wise clustering (30), random forests (31), hybrid semi-supervised clustering (32) or optimization of information-theoretic criteria (33). Note that many of these works have been very recently developed during the author's work for this thesis. Hierarchical clustering and random forests aim at improved speed of feature computation, class-wise and hybrid semi-supervised clustering intend to interpolate between improved speed and improved precision while local coordinate coding (33) focuses on improvement of precision at the cost of higher dimensional features.

Some particular mapping functions are given in the following. Let l be a local feature, m the mapping function, and finally m_d the projection of the mapping function on the d-th output dimension corresponding to the d-th visual word v_d. Hard zero-one mapping is the simplest procedure. Each local feature is mapped onto its nearest visual word resulting in a unit vector as in equation (1.10).

$$m_d(l) = \begin{cases} 1 & \text{if } d = \operatorname{argmin}_e \|l - v_e\|^2 \\ 0 & \text{otherwise} \end{cases} \tag{1.10}$$

The norm $\|\cdot\|^2$ in equation (1.10) is usually the euclidean norm however it might be interesting to try out other norms such as ℓ_p-norms with $p < 1$, or more generally distance functions like the χ^2-distance between two vectors x and y: $\chi^2(x, y) = \sum_d (x_d - y_d)^2/(x_d + y_d)$. Both alternative distance functions would put more emphasis on dimensions d with small values of the vectors x and y.

Soft mapping as in equation (1.11) was introduced in (27) and became popular in the context of competitions in image annotation and ranking

$$m_d(l) = \frac{\exp\left(-\sigma \|l - v_d\|^2\right)}{\sum_e \exp\left(-\sigma \|l - v_e\|^2\right)} \tag{1.11}$$

Soft mapping acts a smoothed version of hard mapping because it distributes the mapping for a local feature to a set of its neighboring visual words.

It was found however in (34), and by the author's own experiments during the ImageCLEF2011 PhotoAnnotation Challenge (1) that for good ranking performance it is necessary to achieve a sufficiently fast decay of assignments as a function of distances from a local feature to neighboring visual words. A revised version of soft-assignment (34) in equation (1.12) assigns votes only to the k nearest neighbors $N_k(l)$ for local feature l in the set of visual words.

$$m_d(l) = \begin{cases} \frac{\exp\left(-\sigma_d \|l-v_d\|^2\right)}{\sum_e \exp(-\sigma_e \|l-v_e\|^2)} & \text{if } d \in N_k(l) \\ 0 & \text{otherwise} \end{cases} \tag{1.12}$$

The author used another form of localized mapping successfully for submissions of the ImageCLEF2011 PhotoAnnotation Challenge (1), rank mapping as in equation (1.13). Let $Rank(z)$ be the rank of the value $z \in \{\|l - v_d\|^2, d = 1, \ldots, B\}$ within the set of distances $\|l - v_d\|^2$ sorted in ascending order.

$$m_d(l) = \begin{cases} 2.4^{-Rank\left(\|l-v_d\|^2\right)} & \text{if } d \in N_k(l) \\ 0 & \text{otherwise} \end{cases} \tag{1.13}$$

While the revised soft mapping from equation (1.12) showed slightly better performance on the ImageCLEF2011 PhotoAnnotation corpus in a post-challenge evaluation, the advantage of rank-mapping is its explicit modelling of decay of mappings as a function of the number of nearest neighbors. The author used in his submissions (17, 18) for the ImageCLEF2011 PhotoAnnotation challenge rank mapping (equation (1.13)) with parameter $k = 8$ having in mind that $2.4^{-8} \approx 1000$. For revised soft mapping from equation (1.12) it is still necessary to fit the constants σ_d appropriately for each visual word. The author's solution for the post-challenge evaluation [1] was to set

$$\sigma_d = \sigma s_d \tag{1.14}$$

where s_d is the inverse of the median of squared distances $\|l-v_d\|^2$ from all local features l such that the visual word v_d is their nearest word within the set of all visual words. This reduces the number of parameters for that mapping to be estimated to one global parameter σ and allows the width parameters σ_d in equation (1.12) to scale according to robust local distance statistics. The need for such scaling comes from the fact that k-means clustering for visual

[1] The author tried the revised soft mapping (equation (1.12)) during the ImageCLEF2011 PhotoAnnotation challenge before learning of the work in (34), noticed slightly better results via cross-validation compared to rank-mapping (equation (1.13)) and still decided to submit solutions based on rank-mapping due to its simpler and thus potentially more robust structure compared to the revised soft mapping.

word generation results in clusters with neighboorhoods of varying diameters as it is a density-sensitive clustering method. This implies that the neighborhoods for different visual words v_d have different distance statistics of the local features which lie in the respective neighborhoods.

Further notable coding methods which yield good results in published work (35) are sparse coding as in equation (1.15) and local linear coding (28) as in equation (1.16)

$$m(l) = \underset{z}{\operatorname{argmin}} \|l - Vz\|^2 + c\|z\|_1 \tag{1.15}$$

where V is the matrix of visual words of format $L \times B$, l is the local feature of formal $L \times 1$, and the mapping vector has format $B \times 1$.

$$m(l) = \underset{z}{\operatorname{argmin}} \|l - Vz\|^2 + c \left\| \sum_{d=1}^{B} z_d \exp\left(\sigma \|l - v_d\|^2\right) \right\|_2^2 \tag{1.16}$$

The missing minus in equation (1.16) is intended. The idea behind local linear coding is that locality is able to induce sparsity such that weights z_d for distant visual words v_d are set to zero or very small values. Finally, the author likes to point out again that Fisher vectors (25) also perform well on large-scale image classification tasks like the ImageNet dataset (15). An overview of the performance of different coding methods is given in (35).

Fourth Stage: Aggregation of Local Feature Mappings Finally, the mapping weight vectors, one from each local feature, will be aggregated into one global feature, which is the final BoW feature, as depicted on the right side of figure 1.6. The usual aggregation step consists of summing the mapping weight vectors and normalizing the resulting vector to adjust for varying numbers of local features.

The combination of a mapping function $m : \mathbb{R}^L \longrightarrow \mathbb{R}^B$ and sum aggregation yields a representation of a BoW feature x as

$$x = \sum_{l} m(l) \in \mathbb{R}^B \tag{1.17}$$

Maximum pooling (34) where the sum in equation (1.17) is replaced by a maximum operator has also been applied as a biologically-inspired alternative.

Finally, one frequently used modification of the Bag of word features are spatial tilings. Originally they were introduced as spatial pyramids in (36). The idea of a spatial tiling is to split each image into a set of regularly shaped spatial tiles, to compute one BoW feature for each tile separately and finally to concatenate the BoW features over all tiles into one BoW feature.

Examples are the spatial tiling 3×1 which decomposes each image into three horizontal stripes of equal height and 2×2 which cut an image into 4 regular squares. Spatial tilings allow to incorporate a low degree of spatial information into BoW features in a robust manner.

Further Remarks The strength of the bag of word feature lies in its robustness which comes from the following factors:

- the absence of modelling of spatial relations between parts unlike earlier approaches which are susceptible to noise in images with complex sceneries.
- the aggregation of local features into a global feature which implies denoising via averaging of contributions of many local features. Equation (1.17) can be interpreted as a sum of many noisy parts which are nonlinear mappings of local features onto the set of visual words. For an alternative interpretation see (37). Apart from normalization of the BoW feature to unit ℓ_1- or ℓ_2-norm, other pooling methods than the sum can be employed like max pooling in which the sum is replaced by a maximum over all mappings $m_d(l_i)$, or generalized p-means $m_p(x) = N^{-1}(\sum_{i=1}^{N} x_i^p)^{1/p}$ which allows to interpolate between the minimum, the maximum, harmonic, geometric and arithmetic means as special cases.
- the choice of robust local features such as SIFT (16) or SURF (38) which are known to be invariant against many changes in lighting conditions. See (23) for an overview of invariance against lighting variations from a color theoretic point of view.

Another advantage of bag of words features is their computational scalability. This is an advantage over intuitively more appealing Bayesian approaches which often need to rely on restricted probability models or inference approximations in practice. Computation of bag of words features in real-time is demonstrated in (39) while (40) demonstrates their efficient computation on GPUs.

The most critical choices in the BoW feature is the local feature, the BoW feature dimensionality and the way of mapping (m in Equation (1.17)) of local features onto the BoW dimensions.

The work (41) shows by comparing against human performance that Bag of word features yield a similar performance to humans on so-called *jumbled images* which were cut into square parts and then piecewise randomly permutated and rejoined. The human advantage is our

ability to extract spatial relations between parts which requires us, however, to spend years of training and learning in childhood from millions of examples and some hundred thousand years of brain evolution before our base learning system became operational. Compared to that BoW models enjoy the advantage of algorithmic simplicity.

Notably, (42) and (43) but also (44) propose methods which avoid the discretization step implied by the usage of visual words. These works go beyond the limits of classical BoW models. (42) uses a boosting type formulation on sets of local features while (43) learns a set kernel metric for pairs of local features under incorporation of local context. A potential drawback is the loss of computational scalability which comes with the original Bag of words model.

The BoW method is also applied with superior results in competitions in related domains such as semantic indexing for videos in *TRECVID* (45) or the winning entry in *ILSVRC2011* large scale object detection challenge (46).

Despite their robustness for domains with highly variable images, Bag of word features are also applied to narrow domains such as concept recognition for medical images (47, 48, 49).

1.3.2 Support Vector Machines in a Nutshell

We will give a short introduction to support vector machines (SVM). For more details the reader is referred to (4). I refrain from reciting all the known facts about SVMs except for what is necessary to understand their usage.

A support vector machine learns a linear predictor

$$f(x) = \boldsymbol{w} \cdot \boldsymbol{x} + b \tag{1.18}$$

for an input sample x by minimization of a loss function l together with a quadratic regularizer for the parameters $\boldsymbol{w}$ of the predictor.

Let $\{(x_i, y_i) \mid i = 1, \ldots, N\}$ be the training data: a set of input features x_i and their binary labels $y_i \in \{-1, +1\}$. Then the support vector machine can be defined as the following optimization problem for learning the parameters $(\boldsymbol{w}, b)$ of the classifier given in equation 1.18:

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2} w \cdot w + C \sum_{i=1}^{N} l\left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b y_i\right) \tag{1.19}$$

The loss function l can be chosen to maximize the margin $f(x_i)y_i$ of samples (x_i, y_i). Examples are the hinge loss

$$l(z, y) = \max(0, 1 - zy) \tag{1.20}$$

and the logistic loss

$$l(z, y) = \ln(1 + \exp(-zy)) \,. \tag{1.21}$$

This approach has two principled advantages. Firstly, from a theoretical point of view the solution of support vector machine is known to be parametrized such that it is based on the span of the training samples x_i. Differentiation of $0 = \frac{1}{2} w \cdot w + C \sum_{i=1}^{N} l\left((\boldsymbol{w} \cdot \boldsymbol{x}_i + b)\, y_i\right)$ based on Formula 1.19 for the variable component $w^{(d)}$ in dimension d proves this claim.

Secondly, from a practical point of view the support vector machine allows for certain losses like the hinge loss and the quadratic loss to incorporate non-linear similarities between data points in the form of Mercer kernels. The nonlinear version of Formula 1.19 is given by replacing x_i with its mapped value $\phi(x_i)$ for some mapping $\phi : \mathcal{X} \longrightarrow \mathcal{H}$ into a Hilbert space $\mathcal{H}$.

The non-linear similarities can be specified implicitly via the choice of a Mercer kernel $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$. The dual formulation of the support vector machine can be written for appropriate loss functions to depend merely on Mercer kernel similarities

$$k(x_i, x_j) = \phi(x_i) \cdot_{\mathcal{H}} \phi(x_i) \tag{1.22}$$

without explicit references to the mappings ϕ into a feature space.

For the sake of self-containedness we give a formal definition of a mercer kernel. A mercer kernel is a symmetric function $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ on a compact subset $\mathcal{X} \subset \mathbb{R}^d$ such that with respect to the Lebesgue measure λ on $\mathbb{R}^d$ the operator

$$T[k](f)(y) = \int_{\mathcal{X}} k(x, y) f(x) d\lambda(x) \tag{1.23}$$

does result always in a function $T[k](f)$ lying in $L_2(\mathcal{X})$ when $f \in L_2(\mathcal{X})$ and all the eigenvalues of the operator $T[k] : L_2(\mathcal{X}) \to L_2(\mathcal{X})$ are non-negative. The eigenvalues are defined by the L_2-Hilbert space $L_2(\mathcal{X})$ of real-valued functions on $\mathcal{X}$ induced from the Lebesgue measure λ:

$$f \cdot g = \int_{\mathcal{X}} f(x) g(x) d\lambda(x) \tag{1.24}$$

$$L_2(\mathcal{X}) = \left\{ f : \mathcal{X} \to \mathbb{R} \,\middle|\, f \text{ is measurable for } \lambda \text{ and } f \cdot f = \|f\|^2 < \infty \right\} \tag{1.25}$$

This result can be generalized to compact Hausdorff spaces with a finite and countably additive measure λ operating on the Borel-σ-Algebra of it. For practical purposes in the context

of SVMs, however, it is sufficient that the matrix $k(x_i, x_j)$ defined over a set of samples $\{x_i\}$ is always non-negative definite for all sets of samples in the sense of common linear algebra.

Back to the formulation of a support vector machine, its essential parameter is the regularization constant C in equation 1.19. High values put more emphasis on minimizing the loss while low values emphasize the quadratic regularization. Appropriate normalization of kernel matrices balances the loss and the regularizer term to be on the same scale and thus allows in practice to choose a regularization constant on a grid around the value $C = 1$.

1.3.3 Kernels Related to this Dissertation

The kernel mostly used in this dissertation is the χ^2-Kernel which is an established kernel for capturing histogram features (50, 51). Let $x^{(d)}$ be the d-th component of vector x.

$$k(x_1, x_2) = \exp\left(-\frac{1}{\sigma} \sum_{d|x_1^{(d)}+x_2^{(d)}>0} \frac{(x_1^{(d)} - x_2^{(d)})^2}{x_1^{(d)} + x_2^{(d)}}\right) \tag{1.26}$$

The bandwidth σ of the χ^2 kernel in (1.26) is thereby heuristically chosen as the mean χ^2 distance (1.27) over all pairs of training examples (x_1, x_2), as done, for example, in (52).

$$\chi^2(x_1, x_2) = \sum_{d|x_1^{(d)}+x_2^{(d)}>0} \frac{(x_1^{(d)} - x_2^{(d)})^2}{x_1^{(d)} + x_2^{(d)}} \tag{1.27}$$

It shares with the gaussian kernel (equation (1.28)) the structure of being an exponential of a negative function of a distance. For the gaussian kernel it is the squared ℓ_2-distance while for the χ^2-kernel it is the χ^2-distance given in equation (1.27). Compared to the gaussian kernel, differences in histogram bins d with low counts $x_1^{(d)} + x_2^{(d)} \approx 0$ are upscaled in the χ^2-kernel. We remark that there exists also another non-exponential formulation of a χ^2-kernel which is not guaranteed to be positive definite (53).

$$k(x_1, x_2) = \exp\left(-\frac{1}{\sigma} \sum_{d} (x_1^{(d)} - x_2^{(d)})^2\right) \tag{1.28}$$

Another established kernel for histograms is the histogram intersection kernel (eq. (1.29)).

$$k(x_1, x_2) = \sum_{d} \min(x_1^{(d)}, x_2^{(d)}) \tag{1.29}$$

All kernels in this study are normalized to have standard deviation 1 in Hilbert space. This amounts to compute

$$K \longmapsto \frac{K}{\frac{1}{n}\mathrm{tr}(K) - \frac{1}{n^2}\mathbf{1}^\top K \mathbf{1}} \tag{1.30}$$

which was proposed in (54, 55) and entitled *multiplicative normalization* in (56). This avoids situations in which a kernel with low variance is dominated by a kernel with high variance when both are combined.

For large scale applications many of those kernels can be approximated well by explicit feature maps (53, 57, 58) which are then used as higher-dimensional features for a linear kernel. This allows to use primal support vector machines with approximations of non-linear kernels.

1.3.4 Kernel Alignment

The kernel alignment introduced by (59) measures the similarity of two matrices as a cosine angle in a Hilbert space defined by the Frobenius product of matrices

$$\mathcal{A}(k_1, k_2) := \frac{\langle k_1, k_2 \rangle_F}{\|k_1\|_F \|k_2\|_F}, \tag{1.31}$$

We will use kernel alignment in two variants in Chapters 2 and 3 for the analysis of kernel properties.

The first variant computes the cosine angle between two kernels computed from image features. We call this kernel-kernel alignment (KKA).

The second variant, kernel target alignment (KTA) measures the similarity between one kernel from features and an optimally discriminative kernel computed from the labels for a given visual concept. The centered kernel which achieves a perfect separation of two classes can be derived from the labels and is proportional to $\widetilde{\boldsymbol{y}}\widetilde{\boldsymbol{y}}^\top$, where

$$\widetilde{\boldsymbol{y}} = (\widetilde{y}_i), \quad \widetilde{y}_i := \begin{cases} \frac{1}{n_+} & y_i = +1 \\ -\frac{1}{n_-} & y_i = -1 \end{cases} \tag{1.32}$$

and n_+ and n_- are the sizes of the positive and negative classes, respectively.

It was argued in (60) that centering (61) is required in order to correctly reflect the test errors from SVMs via kernel alignment. Centering in the corresponding feature spaces is the replacement of $k(x_i, x_j) = \langle \phi(x_i), \phi(x_i) \rangle$ by

$$\left\langle \phi(x_i) - N^{-1} \sum_{k=1}^{N} \phi(x_k), \phi(x_i) - N^{-1} \sum_{k=1}^{N} \phi(x_k) \right\rangle \tag{1.33}$$

Note that support vector machines using a bias term are invariant against centering, which can be shown using the condition $\sum_i \alpha_i y_i = 0$ from the optimization problem given by equation (3.2). To see the influence of centering on kernel alignment consider that the normalized kernel alignment with an added bias z and non-negative kernels $\langle z_1, z_2 \rangle \geq 0$ will be dominated by the bias z when $\|z\| \to \infty$:

$$\frac{\langle \phi(x_1) + z, \phi(x_2) + z \rangle}{\|\phi(x_1) + z\| \|\phi(x_2) + z\|} \geq \frac{\|z\|^2}{\|\phi(x_1) + z\| \|\phi(x_2) + z\|} \overset{\|z\| \to \infty}{\longrightarrow} 1 \,. \tag{1.34}$$

Centering can be achieved by taking the product HKH, with

$$H := I - \frac{1}{n} \mathbf{1}\mathbf{1}^\top, \tag{1.35}$$

I is the identity matrix of size n and $\mathbf{1}$ is the column vector with all ones.

1.4 Overview of this dissertation

This thesis is not method driven, it is problem driven. This means, I did not develop one single method which I apply to various kinds of datasets and compare where it works better than existing baselines. Neither did I perform a theoretical analysis for one class of algorithms. Instead I have worked on one larger problem, namely that of image annotation and ranking, which required me to tackle several aspects of that problem ranging from feature design to loss function design and optimization. This problem can be divided for discriminative approaches which aim at minimizing a loss or maximizing a score into three big topics.

- Formulation of the problem and design or choice of a corresponding loss function
- Learning of feature combinations given a loss function
- Design of Features

This is not a strict hierarchy, since the design of features and their properties may have influence on the method to learn the feature combination. The simplest example for this argument is the case when one makes the assumption that only a small but a priori unknown subset of the given features will be useful. In that case one would rely on sparse algorithms to learn the feature combination.

Figure 1.7 depicts these three big topics. The decomposition into three topics is the reason why subsequent chapters have their own related work and conclusion subsections. Essentially,

the following chapters tackle different topics of the same grand problem. Furthermore, the field of computer vision is sufficiently developed and diversified such that each part deserves its own specific set of references.

For the aspect of *Design of Features* I have analyzed the impact of biased random sampling using novel sampling methods for BoW (Bag of words) features (17). This methodology was part of the author's submission on out of sample testing data for the ImageCLEF2011 Photo Annotation Challenge which yielded the winning entries in this competition for multi-modal and pure visual categories (18).

For the same aspect I also worked on hybrid algorithms which combine the ability for fast feature computation due to tree structures together with supervised learning of splits based on support vector machines (32).

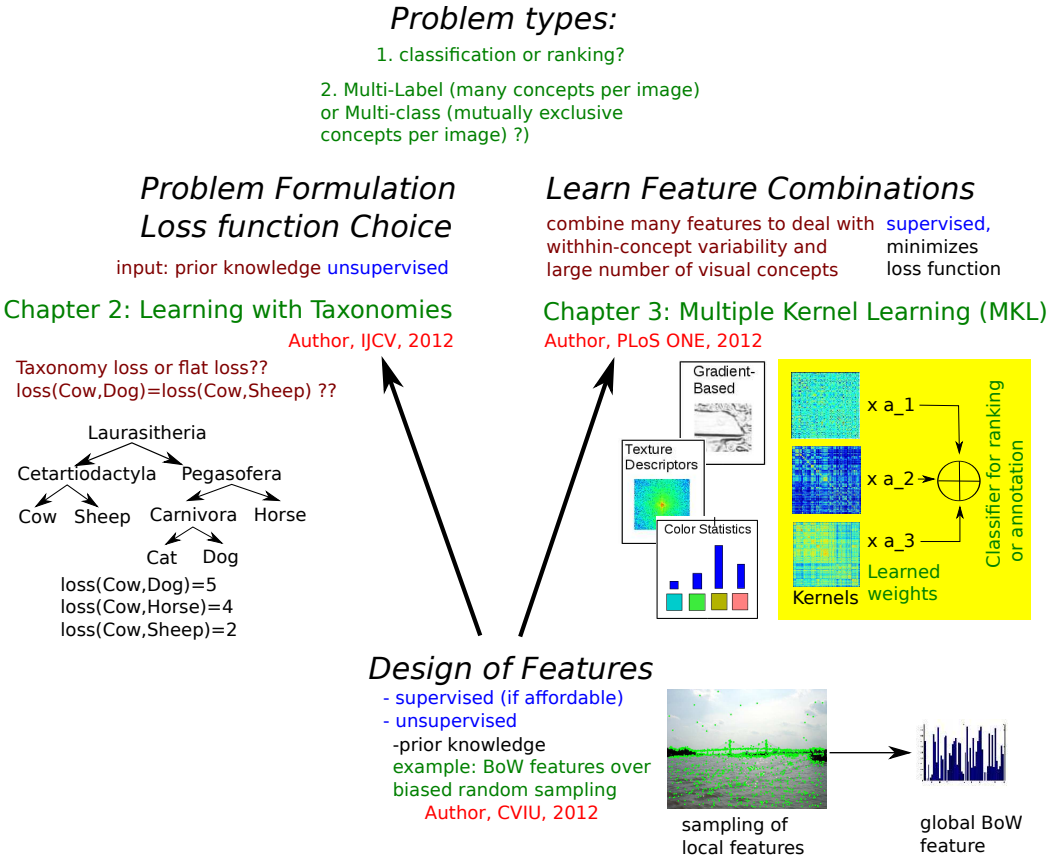


Figure 1.7: Three big topics of the image annotation and ranking problem. Blue shows the type of supervision. Green colors examples. Brown colors ideas.

A brief overview over the state of the art of feature design for BoW features is given in section 1.3.1. The following two contributions will be shown in this thesis in more detail.

For the aspect of *Formulation of the problem and design or choice of a corresponding loss function* I proposed a novel algorithm capable of optimizing taxonomy-induced loss functions for multi-class data in a computationally efficient manner and taxonomy-based ranking scores for multi-label data (62). This will be discussed in Chapter 2.

For the aspect of *Learning of feature combinations given a loss function* I analyzed the behavior of the existing non-sparse multiple kernel learning (MKL) algorithm (56) specific to properties of features commonly used in image annotation due to their state of the art performance (63). I will give novel explanations on its limits and benefits based on experiments on real-world data. This will be discussed in Chapter 3.

The two aspects on which I will focus subsequently, namely learning with taxonomy-induced loss functions and ranking scores and an analysis of non-sparse multiple kernel learning in image ranking, can be treated separately or in a combined manner. Given the complexity of these topics and the authors' impression that both of them contain many problems which are not understood sufficiently, I will treat them independently in two separate chapters.

An overview over publications coauthored by me is given in Section 1.4.2.

The annotation system was tested in three international benchmark competitions which were evaluated on image collections with undisclosed ground truth, namely Pascal VOC 2009 Classification (64), ImageCLEF2009 Photo Annotation (65) and ImageCLEF2011 Photo Annotation (1). It yielded in these competitions top-five placements and winning entries in two categories of the most recent of these competitions, ImageCLEF2011 Photo Annotation (1).

1.4.1 Why do we not learn anything at once but divide the problem into parts?

One may ask here why I did decompose the problem into parts and did not follow the way to learn everything simultaneously. It might be indeed a desirable long term goal to learn all possible parameters from data in a unified framework. Still, elegant theory is not always practical when real data has to be processed. For example full-scale cross-validation over all hyperparameters is limited in practice to a low number of parameters because the number of grid points may grow exponentially with the number of parameters. In practice sequential cross-validation or alternative heuristics like genetic algorithms may yield the best results as demonstrated in (66). The alternative to cross-validation are likelihood based models. Discriminative models in computer vision like SVMs may overfit in practice strongly on the training data when being

at their optimum with respect to their performance on test data – see for example the necessity to use cross-validation for generating SVM outputs which are used for learning of subsequent models in (67, 68). This effect makes the usage of cross-validation preferable for discriminative methods over direct likelihood based models acting on the whole training data directly. Problem decomposition allows to include prior knowledge easily yielding better recognition performance or saving time even when the problems are solved only approximately. The tables 2.10 and 2.11 in Chapter 2 provide an example, where structured prediction algorithms with all their mathematical elegance do not provide significant performance gains over simpler and much faster approximate models. Problem decomposition as the alternative can make problems to be solved more efficiently and in less time which is an argument against monolithic unified frameworks. For these reasons I will approach the problem of image annotation and ranking by decompising it into three levels mentioned in Section 1.4.

The three levels of the problem can be also classified by their relation to supervision. Feature design is a part which can be performed efficiently in an unsupervised or merely weakly supervised manner. It may include prior knowledge about the problem. The weak supervision can be used to ensure that certain statistical properties of the dataset are reflected in the features. One example would be for the case of Bag of word features the question which images are used for computing visual words. The visual words will be computed from a set of local features which have been extracted from the images in question. In problems with many visual concepts it maybe helpful to ensure that images from visual concepts with low abundance in the training data do appear in the set used for computation of visual words. This matter has been investigated in (30) where it was shown that learning a separate visual vocabulary for each visual concept and fusing all these vocabularies into one big set of visual words may help to improve ranking performance. Further examples of introducing supervision to feature design are (31, 32). Using more supervision in feature design has the potential to improve recognition performance at the price of slower algorithms.

The feature combination part relies on supervision for learning a useful combination of unsupervised or weakly supervised features as it is based on minimization of a given loss function. For that part an empirical analysis of multiple kernel learning will be discussed in Chapter 3.

The last part, the choice of a loss function, relies on incorporation of prior knowledge in one or another way. The usage of supervision for the choice of a loss function requires some kind of regularization because the criterion used for supervision itself is defined at this

level. Introducing regularization can be interpreted as a way to incorporate prior knowledge. Regularization implies that hypotheses which receive stronger regularization are only chosen if the data supports them particularly well. This is a way to express the prior knowledge that these hypotheses are expected to be chosen less likely. In summary, the incorporation of prior knowledge is necessary for choosing a loss function.

As an extreme example why usage of supervision may not be always helpful at the level where the loss is designed consider a loss function which is learnt from data in a way such that it places no or low penalties for misclassifying images showing visual concepts which are hard to recognize. It might be not always in the interest of users to ignore misclassification of hard cases. On the contrary, in some cases it might be useful to improve the recognition performance of badly recognized visual concepts at the cost of reducing recognition performance of easier recognized visual concepts.

In this dissertation I did not attempt to learn loss functions for this reason but instead chose the simpler way in Chapter 2 to learn models based on hierarchical losses which were derived from prior knowledge about the problem. The following section 1.4.2 lists work published by the author.

1.4.2 The Author's Contributions

- *Choice of Loss Function:* Classification with Hierarchical Structure
  - A. Binder, K. R. Müller, M. Kawanabe, **On Taxonomies for Multi-class Image Categorization**, International Journal of Computer Vision 99(3), 281-301, 2012, accepted January 2011 (62)
- *Feature Combination for a given loss:* Learning Kernel Combinations
  - A. Binder, S. Nakajima, M. Kloft, C. Müller, W. Samek, U. Brefeld, K.-R. Müller, M. Kawanabe: **Insights from Classifying Visual Concepts with Multiple Kernel Learning** PLoS ONE 7(8), 2012, doi:10.1371/journal.pone.0038897 (63)
  - S. Nakajima, A. Binder, C. Müller, W. Wojcikiewicz, M. Kloft, U. Brefeld, K.-R. Müller, M. Kawanabe: **Multiple Kernel Learning for Object Classification**, IBIS2009 Workshop, Fukuoka, Japan (69)
  - M. Kawanabe, S. Nakajima, A. Binder:**A procedure of adaptive kernel combination with kernel-target alignment for object classification**, CIVR2009 (70)

- *Feature Combination for a given loss:* Learning Relations between Semantic Concepts
  - A. Binder, W. Samek, K.-R. Müller, M. Kawanabe: **Enhanced Representation and Multi-Task Learning for Image Annotation**, *Computer Vision and Image Understanding, accepted, DOI: 10.1016/j.cviu.2012.09.006* (17)
  - W. Samek, A. Binder, M. Kawanabe: **Multi-task Learning via Non-sparse Multiple Kernel Learning**, CAIP 2011(1): 335-342 (67)
- *Feature Combination for a given loss:* Multi-Modal Classification of Images
  - M. Kawanabe, A. Binder, C. Müller, W. Wojcikiewicz: **Multi-modal visual concept classification of images via Markov random walk over tags**, IEEE WACV 2011: 396-401 (71)
- *Feature Design:* Vocabulary Optimization for Bag of Word Features
  - A. Binder, W. Wojcikiewicz, C. Müller, M. Kawanabe: **A Hybrid Supervised-Unsupervised Vocabulary Generation Algorithm for Visual Concept Recognition**, ACCV 2010 (3): 95-108 (32)
  - W. Wojcikiewicz, A. Binder, M. Kawanabe: **Shrinking large visual vocabularies using multi-label agglomerative information bottleneck**, ICIP 2010: 3849-3852 (72)
  - W. Wojcikiewicz, A. Binder, M. Kawanabe: **Enhancing Image Classification with Class-wise Clustered Vocabularies**, ICPR 2010: 1060-1063 (30)
- *Feature Design:* Analysis of biased random sampling and Learning of Relations between Semantic Concepts for the ImageCLEF 2011 Photo Annotation dataset.
  - A. Binder, W. Samek, K.-R. Müller, M. Kawanabe: **Enhanced Representation and Multi-Task Learning for Image Annotation**, *Computer Vision and Image Understanding, accepted, DOI: 10.1016/j.cviu.2012.09.006* (17)
- Overview Chapters in Books

  - A. Binder, F.C. Meinecke, F. Biessmann, M. Kawanabe, K.-R. Müller:**Maschinelles Lernen und Mustererkennung in der Bildverarbeitung**, *Grundlagen der praktischen Information und Dokumentation*, editors: R. Kuhlen, T. Seeger, D. Strauch, submitted
  - A. Binder, W. Samek, K.-R. Müller, M. Kawanabe: **Machine Learning for Visual Concept Recognition and Ranking for Images**, published in: *Towards the Internet of Services: The Theseus Project*, editors: W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich, T. Widenka, accepted

- Challenge Results

  - A. Binder, W. Samek, M. Kloft, C. Müller, K.-R. Müller, M. Kawanabe: **The Joint Submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF2011 Photo Annotation Task**, CLEF(Notebook Papers/Labs/Workshop) 2011, https://doc.ml.tu-berlin.de/publications/data/ABinder/imageclef2011workingnote.pdf (18)
  - A. Binder, M. Kawanabe: **Enhancing Recognition of Visual Concepts with Primitive Color Histograms via Non-sparse Multiple Kernel Learning**, CLEF Postproceedings 2009: 269-276, Springer LNCS 6242 (73)

- Open Source Software

  - S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. De Bona, A. Binder, C. Gehl, V. Franc: **The SHOGUN Machine Learning Toolbox**, Journal of Machine Learning Research 11: 1799-1802 (2010) (74)

2

Semantic Concept Recognition with a Tree Structure over Concepts

2.1 Motivation for this aspect of Semantic Concept Recognition in Images

Given image data with an additional structure between semantic concepts which can be represented by a tree, the problem considered here is to classify images into semantic concepts such that a loss function which incorporates the tree structure is minimized.

In computer vision, one of the most difficult challenges is to bridge the semantic gap between appearances of image contents and high-level semantic concepts (8). While systems for image annotation and content-based image retrieval are continuously progressing, they are still far from resembling the recognition abilities of humans that have closed this gap. Humans are known to exploit taxonomical hierarchies in order to recognize general semantic contents accurately and efficiently. Therefore, it remains important for artificial systems to incorporate extra sources of information, such as user tags (75, 76, 77) or prior knowledge such as taxonomical relations between visual concepts.

Most work on hierarchies focused on speed gains at testing time based on the idea to achieve a logarithmic number of SVM evaluations when traversing the hierarchy during classification. The second observation is that it is apparent in the preceding work that the losses used to measure classification performance are flat in that sense that the losses ignore the same hierarchic structure employed for classification. This usually resulted in speed gains at testing time at the cost of higher flat zero-one loss. The third observation is that many publications

focus on multi-class settings, in which each image shows at most one semantic concept. This is a too restrictive assumption – for many real-world annotation problems on internet photo collections one has to deal with complex images and larger sets of visual concepts. In such settings overlap of semantic concepts becomes unavoidable.

2.1.1 Contributions

We are interested here in optimizing a loss function for multi-class classification setting and a ranking score for the multi-label ranking setting which is non-flat in the sense that it incorporates the hierarchical structure. Non-flat implies for multi-class classification, that confusions between two semantic concept classes are penalized depending on the given hierarchy. Classes which are more distant in the hierarchy yield a higher penalization when the prediction function confuses them. One example is given for the multi-label ranking setting in figure 2.1 where mistaking a cat image to show a car is intended to give a lower ranking score than confusing a cat with a dog. In the multi-label ranking setting we have no notion of confusion, because multiple semantic concepts can be present in one image. However when ranking images for the cat category, a sequence which shows images with dogs in high ranks should receive higher scores than a sequence in which the images showing dogs are replaced with images showing cars as the closest concept to cats in the hierarchy. This is based on the assumption that dogs are closer in a hierarchy to cats than cars.

Figure 2.1: Two sequences for concept *cat* in a multi-label setting with mistakes which affect ranking performance, upper: a dog image, lower: a car image. Under a taxonomy-induced measure the lower sequence should receive a lower ranking score because the difference between the closest visual concept and *cat* is larger compared to the upper sequence. Images from Wikimedia Commons.

We will see that for the multi-class setting for certain loss functions there exists a natural solution in the framework of structured prediction. This permits the usage of methods from structured prediction as baselines for comparison with our novel method.

The contributions of this chapter are [1]

- a novel method to optimize certain loss functions derived from a hierarchical structure based on combination of scores of support vector machines which correspond to local paths in the hierarchy. Unlike greedy walk-down schemes in this work the scores from all paths to semantic concepts and all local SVMs are taken into account for improved classification performance. The main advantages of this novel method are improved speed and scalability relative to structured prediction and improved classification performance with respect to hierarchic loss compared to the established one versus all classification baseline and greedy walk-down schemes.

- an extension of hierarchical classification approaches to the multi-label setting which allows to predict multiple semantic concepts in one image while relying on hierarchical structures.

- an extension of average precision ranking scores to the multi-label setting which incorporates the hierarchical structure. This extension is general because any structured loss function can be plugged in as a replacement for the average precision ranking measure, not just loss functions derived hierarchical learning models.

- we compare the novel local SVM method against various baselines such as one versus all classification and structured prediction methods and discuss insights in the way it works.

The author regards the discussion in subsection *Generalization Ability for Learning of Superclasses in Taxonomies* of section 2.4.8 important for the understanding why classification with taxonomies is a challenging problem and why results obtained by using it may be different from an intuitive view of human abilities.

Why do we need another algorithm for hierarchic classification?

Our work focuses on the question whether we may improve classification losses or scores rather than speed using hierarchies. As a preliminary step to optimizing losses we like to revisit the question what kind of loss or score functions we intend to optimize when using

[1] The content of this chapter is based on the author's own peer-reviewed work in (62).

hierarchical models for classification and ranking. We felt that this question was not sufficiently considered in many of the preceding works. Furthermore we extend hierarchical approaches to multilabel datasets which we think to be a more realistic assumption for image data sets with many concepts defined over them.

In this work, we contribute a tractable alternative to the structure learning framework which can solve our task in a sophisticated way, but is less time consuming. We propose its efficient decomposition into an ensemble of local support vector machines (SVMs) that can be trained efficiently. Since the primal goal of this chapter is to discuss how much and why predetermined taxonomies improve classification performance, we consider any techniques for speed-up which degrade performance to be out of the scope of this chapter. [1]

Our work is similar in spirit to (78) who deployed user-determined taxonomies and showed that classifiers for super-classes defined at parent and grand-parent nodes can enhance leaf-edge classifiers by controlling the bias-variance trade-off. However in (78) the discrimination of images was performed against a small set of common backgrounds, and thus, all classifiers at all edges share the same negative samples, i.e. the background images. Performance was measured for object versus background scenarios. In contrast to (78), we will study a more difficult problem, namely, multi-class or multi-label classification between object categories. Since our problem does not contain uniform sets of background, it is an interesting question whether an averaging along the leaves of a taxonomy integrating everything from super-class classifiers until the lower leaf-edges can still help to improve the object recognition result, in particular as the negative samples can not be shared among all classifiers as in (78).

We remark furthermore that we observe from our experiments that greedy strategies as e.g. (79) are inferior by prediction accuracies to our novel taxonomy based methods that we propose in this chapter.

In contrast to this work the approaches mentioned in Section 2.1.2 have one aspect common in their methodology: they restrict performance measurement to flat loss measures which do not distinguish between different types of misclassification. In contrast to that humans tend to perceive some confusions like cat versus fridge to be more unnatural than others like cat versus dog which can be reflected by a taxonomy. The hierarchy in (79) *learned* from features reflects feature similarities and is as a consequence in part not biologically plausible: the gorilla

[1] For instance, we use all images for SVM training at every edge, which is of course more costly than the greedy strategy. It may be possible reducing the large number of negative examples which are inferred irrelevant to current and future decisions with high probability without decreasing classification accuracy.

is closer to a raccoon than to a chimpanzee, the grasshopper is closest to penguin, and more distant to other insect lifeforms. Such problems can arise generally when the hierarchy is learned from image contents.

This prompts the question whether it is useful to employ a taxonomy which is based merely on information already present in the images and which is thus implicitly already in use through the extracted feature sets that feed the learning machine. Furthermore basic information derived from the images only, may not always be coherent with the user's rich body of experience and implicit or explicit knowledge.

An example is the discrimination of several Protostomia, sea cucumbers and fish (see Figure 2.2). While sea cucumbers look definitely more similar to many Protostomia, they are much closer to fish sharing the property of belonging to Deuterostomia according to phylogenetic systematics. Equally, horseshoe crabs look more similar to crabs as both have a shell and live on the coast, but the horseshoe-crab as a member of Chelicerata is closer to spiders than to crabs. Therefore, this work is focused on *pre-determined* taxonomies constructed independently from basic image features as a way for providing such additional information rsp. knowledge. This task fits well into the popular structured learning framework (80, 81) which has recently seen many applications among them in particular document classification with taxonomies (82). Note furthermore that a given taxonomy permits to deduce a *taxonomy* loss function which – in contrast to the common 0/1 loss – allows to weight misclassification unevenly according to their mismatch when measured in the taxonomy. Thus, it is rather natural to evaluate classification results according to the taxonomy losses instead of the flat 0/1 loss, in this sense imposing a more human-like error measure.

The remainder of this chapter is organized as follows. Section 2.1.2 gives an overview of algorithms using hierarchical classification in image annotation tasks besides the paper which have been mentioned already. In Section 2.2 we will explain our novel local procedures with scoring deduced from generalized p-means, along with structure learning approaches. We discuss in Section 2.3 when and why our procedures can improve the one-vs-all baseline. The empirical comparisons between our local approach and other taxonomical algorithms and taxonomy-free baselines are presented in Section 2.4. For the present work, we have constructed multi-class classification datasets with taxonomy trees between object categories based on the benchmarks Caltech256 (83) and VOC2006 (84) as explained in Section 2.4.1. In this Section we discuss why our local approach can improve the one-vs-all baseline from the viewpoint of averaging processes. Section 2.6 gives concluding remarks and a discussion.

Figure 2.2: Mismatch between taxonomy and visual similarity: the first column are Protostomia, the second (sea cucumbers) and third row are Deuterostomia. The difference is based on embryonal development. Images from Wikimedia Commons.

2.1.2 Related Work

There have been a number of studies considering *learning* class-hierarchies, for instance on the basis of delayed decisions (85), dependency graphs and co-occurrences (52, 86), greedy margin-trees (87), by hierarchical clustering (79, 88), and by incorporating additional information (89). Unfortunately, few could so far report significant performance gains in the final object classification (even though they contributed to other aspects, for instance, computational efficiency).

When a taxonomy is available, a standard way of using the hierarchy is sequential greedy decision (79). Starting from the root node, the strategy selects only the most probable edge rooted at each node and ignores other possibilities until reaching a leaf node. Therefore, for classifying an unseen image only the classifiers on one path of the hierarchy need to be evaluated. Furthermore, since each node takes only relevant images for current and future decisions during the training phase, such greedy methods are computationally very attractive. The work in (79) focuses on learning hierarchies and demonstrates speed gains by the greedy classification schemes compared to one versus all classifiers (e.g. 5-fold speed gain at the cost of 10% performance drop). Another greedy walk approach over a learned hierarchy (85) shows small improvements on the Caltech256 dataset. A similar result using a non-convex formu-

lation for learning a relaxed hierarchy is presented in (90). It achieves computation speedups and even small recognition performance improvements on Caltech256 with respect to zero-one loss. [1] The later work in (91) develops general structured prediction for multi-label datasets and applies it also to hierarchical classification. It finds the one-versus-all classification baseline which we also considered here hard to beat. These findings are consistent with our experiments which structured prediction algorithms below.

2.2 Methods

2.2.1 Problem Formulation

We consider the following problem setting: given are n pairs $\{(x^{(i)}, y^{(i)})\}$, $1 \leq i \leq n$, where $x^{(i)} \in \Re^d$ denotes the vectorial representation of the i-th image which can be represented in higher dimensions by a possibly non-linear mapping $\phi(x^{(i)})$. The latter gives also rise to a kernel function on images, given by $K_X(x, x') = \langle \phi(x), \phi(x') \rangle$. The set of labels is denoted by $Y = \{c_1, c_2, \ldots, c_k\}$. We focus initially on multi-class classification tasks, where every image is annotated by exactly one element of Y. Some image databases fall into the multi-label setting, where an image can be annotated with several class labels which will be dealt with later on.

In addition, we are given a taxonomy T in form of an arbitrary directed graph (V, E) where $V = (v_1, \ldots, v_{|V|})$ and $Y \subset V$ such that classes are identified with leaf nodes (see Figure 2.3 for an example). We assume the existence of one unique root node. The set of edges on the path from the root node to a leaf node y is defined as $\pi(y)$. Alternatively, the set $\pi(y)$ can be represented by a vector $\kappa(y)$ where the j-th element is given by

$$\kappa_j(y) = \begin{cases} 1 : v_j \in \pi(y) \\ 0 : \text{otherwise}, \end{cases}$$

such that the category *sheep* in Figure 2.3 is represented by the vector

$$\kappa(\text{sheep}) = (1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0)'.$$

[1] For convention purposes please note that a classifier is rooted at each *edge*. For trees this is equivalent to the view that each node except for the root node has one classifier. For directed acyclic graphs, however, the first view is necessary because each node may have more than one directed edges pointing to it. We will speak about nodes when we refer to sets of classes or images and edges when we refer to classifiers itself. In this sense a classifier at a node refers to a classifier at the directed edges leading to that node.

The goal is to find a function f that minimizes the generalization error $R(f)$,

$$R(f) = \int_{\Re^d \times Y} \delta(y, f(x)) dP(x, y),$$

where $P(x, y)$ is the (unknown) distribution of images and annotations. The quality of f is measured by an appropriate, symmetric, non-negative loss function $\delta : Y \times Y \to \Re_0^+$ detailing the distance between the true class y and the prediction. For instance, δ may be the common 0/1 loss, given by

$$\delta_{0/1}(y, \hat{y}) = \begin{cases} 0 & : \quad y = \hat{y} \\ 1 & : \quad \text{otherwise.} \end{cases} \tag{2.1}$$

When learning with taxonomies, the distance of y and $\hat{y}$ with respect to the taxonomy is fundamental. For instance, confusing an *bus* with a *cat* is more severe than confusing the classes *cat* and *dog*. We will therefore also utilize a taxonomy-based loss function reflecting this intuition by counting the number of non-shared edges on the path between the true class y and the prediction $\hat{y}$,

$$\delta_T(y, \hat{y}) = \sum_{j=1}^{|V|} |\kappa_j(y) - \kappa_j(\hat{y})|. \tag{2.2}$$

This distance can be induced as Hilbert space norm by the kernel between labels defined as

$$\mathrm{K}_Y(y, \hat{y}) = \sum_{j=1}^{|V|} \kappa_j(y)\kappa_j(\hat{y}). \tag{2.3}$$

Note here that each node except for the root node can be identified with the path element in the hierarchy from its parent node to the current node. In that sense the usage of the notions of node in the hierarchy and of path element in the hierarchy is equivalent for hierarchies. For direct acyclic graphs, however, one has to resort to the notion of edges because a node may have multiple ancestors and edges leading to it.

For instance, the taxonomy-based loss between categories *horse* and *cow* in Figure 2.3 is $\delta_T(\text{horse}, \text{cow}) = 4$ because $\kappa(\text{horse})$ and $\kappa(\text{cow})$ differ at the edges pointing to nodes horse, pegasofera, cetartiodactyla and cow.

2.2.2 Structure Learning with Taxonomies

The taxonomy-based learning task can be framed as structured learning problem (80, 81) where a function

$$f(x) = \underset{y}{\operatorname{argmax}} \langle w, \Psi(x, y) \rangle \tag{2.4}$$

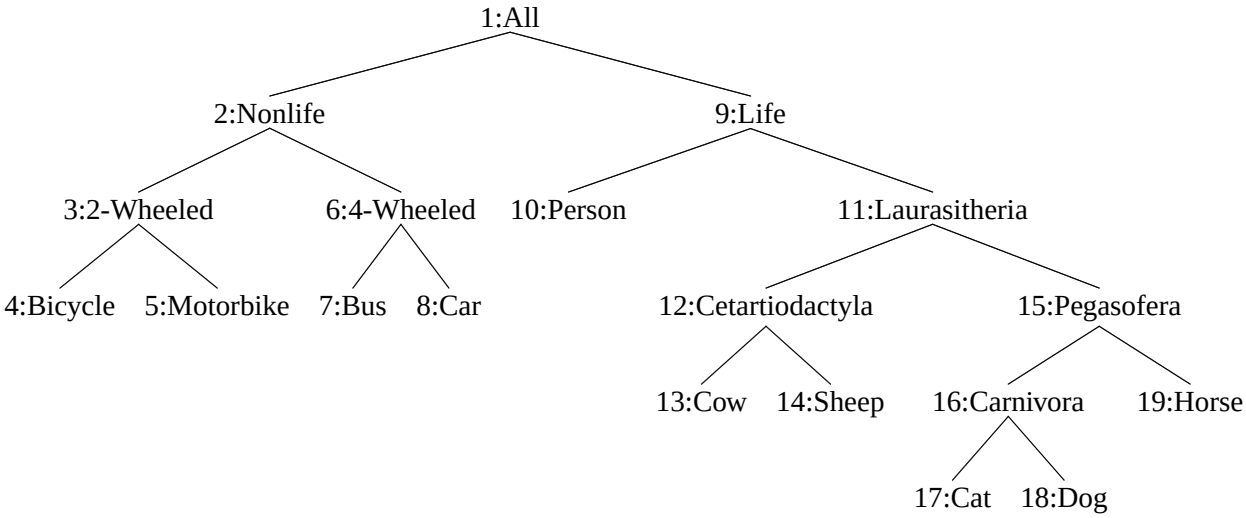


Figure 2.3: Taxonomy constructed from VOC2006 labels. The life subtree is based on biological systematics.

defined jointly on inputs and outputs is to be learned. The mapping $\Psi(x, y)$ is often called the joint feature representation and for learning taxonomies given by the tensor product (82) with indicator functions

$$\kappa_i(y) = [[v_i \in \pi(y)]] \tag{2.5}$$

and the input feature mapping $\phi(x)$

$$\Psi(x, y) = \phi(x) \otimes \kappa(y) = \begin{pmatrix} \phi(x)[[v_1 \in \pi(y)]] \\ \phi(x)[[v_2 \in \pi(y)]] \\ \vdots \\ \phi(x)[[v_{|V|} \in \pi(y)]] \end{pmatrix}. \tag{2.6}$$

Thus, the joint feature representation subsumes the structural information and explicitly encodes paths in the taxonomy. It leads to a joint kernel

$$K_{X,Y}((x_1, y_1), (x_2, y_2)) = K_X(x_1, x_2)K_Y(y_1, y_2), \tag{2.7}$$

where $K_X(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$ and the label kernel $K_Y(y_1, y_2)$ is defined according to the taxonomy T as in Equation (2.3).

The empirical risk can be optimized utilizing conditional random fields (CRFs) (92) or structural support vector machines (SVMs). We will follow structural learning in the formulation by (93, 94). There are two ways of incorporating a loss $\Delta(y, \bar{y})$ such as $\delta_{0/1}$ and δ_T in the

structural SVMs. The optimization problem with margin rescaling is given by

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi^{(i)} \\ \text{s.t. } & \forall i,\ \forall \bar{y} \neq y^{(i)} : \\ & \langle w, \Psi(x^{(i)}, y^{(i)}) - \Psi(x^{(i)}, \bar{y})\rangle \geq \Delta(y^{(i)}, \bar{y}) - \xi^{(i)} \\ & \forall i : \xi^{(i)} \geq 0. \end{aligned} \tag{2.8}$$

The above minimization problem has one constraint for each image. Every constraint is associated with a slack-variable $\xi^{(i)}$ that acts as an upper bound on the error Δ caused by annotating the i-th image with a wrong label. Once, optimal parameters w^* have been found, these are used as plug-in estimates to compute predictions for new and unseen examples using Equation (2.4). The computation of the argmax can be performed by explicit enumeration of all paths in the taxonomy.

An alternative formulation (81) uses slack rescaling instead of margin rescaling in the constraints:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi^{(i)} \\ \text{s.t. } & \forall i,\ \forall \bar{y} \neq y^{(i)} : \\ & \langle w, \Psi(x^{(i)}, y^{(i)}) - \Psi(x^{(i)}, \bar{y})\rangle \geq 1 - \frac{\xi^{(i)}}{\Delta(y^{(i)}, \bar{y})} \\ & \forall i : \xi^{(i)} \geq 0. \end{aligned} \tag{2.9}$$

In this multiplicative formulation based on a hinge loss (assume $\Delta(y, \hat{y}) \geq 0, \Delta(y, y) = 0\ \forall y$)

$$\max_{\bar{y}} \Delta(\bar{y}, y^{(i)})(1 + \langle w, \Psi(x^{(i)}, \bar{y}) - \Psi(x^{(i)}, y^{(i)})\rangle) \tag{2.10}$$

each sample receives the same margin of one. As a drawback finding the maximally violated label can be more complicated compared to margin rescaling due to the label $\bar{y}$ appearing in both factors of a product. Margin rescaling is also based on the hinge loss but uses an additive formulation in $\Delta(\bar{y}, y^{(i)})$

$$\max_{\bar{y}} \Delta(\bar{y}, y^{(i)}) + \langle w, \Psi(x^{(i)}, \bar{y}) - \Psi(x^{(i)}, y^{(i)})\rangle \tag{2.11}$$

where it might be easier to find the maximally violated constraint but on the other side here the loss function Δ might dominate the loss term (2.11) if it is badly scaled.

Although, equations (2.8) and (2.9) can be optimized with standard techniques, the number of categories in state-of-the-art object recognition tasks can easily exceed several hundreds which renders the structural approaches inherently slow.

2.2.3 Remark on Feasible Taxonomy Loss Functions

The factorization of the combined feature label kernel (cf. Eq. 2.6) in the structured prediction setup allows to insert more general label kernels beside the one which induces the canonical taxonomy distance given in Eq. 2.2. Any mapping $\kappa(y)$ may be chosen in Equations 2.5 and 2.6. One particular useful possibility in the connection with a given taxonomy is to use weighted taxonomy loss functions which assign non-negative weights to edges in the hierarchy from one node to its child node. This permits to emphasize the importance of certain confusions over others in an easily interpretable manner. To do this, replace $\kappa(y)$ from Eq. 2.5 by element-wise multiplication with the square-root of the desired edge weights $\boldsymbol{u}$:

$$\kappa[\boldsymbol{u}]_i(y) = \sqrt{u_i}[[v_i \in \pi(y)]].$$

This extends the original setup to taxonomy losses with weighted edges. One meaningful application is to weight each edge by the binary power 2^{-d} of its negative depth d in the hierarchy. Since $\sum_{i=1}^{s} 2^{-i} = 1 - 2^{-s} < 1$ this ensures that a classification error made at a higher level closer to the root node always counts more than confusions at lower levels of the hierarchy independent of the length of the path from root to the leaf node.

2.2.4 Assembling Local Binary SVMs

We propose here an efficient alternative to the structural approaches by decomposing the structural approach from Equation (2.8) into several local tasks. The idea is to learn a binary SVM (e.g. (3, 4)) using the original representation $\phi(x)$ for *each edge* $e_j \in E$ in the taxonomy instead of solving the *whole* problem at once with a structured learning approach. This will help to circumvent the high computational load typically encountered in structured learning. To preserve the predictive power, the final ensemble of binary SVMs from each edge need to be assembled in an intelligent manner, i.e. appropriately according to the taxonomy. We remark that this novel approach is different from greedy hierarchical classifiers where at each edge only categories (leaf nodes) lying below the edge are taken into account. On the contrary, we are considering *all* images and categories at each node: for example, we learn binary SVMs

such as 'Carnivora vs the others' and 'horse vs the others', while only 'Carnivora vs horse', 'cat vs dog' etc. would be used in the greedy hierarchical classification. As outlined in Section 2.4.7, the greedy approaches perform sub-optimally, because they may rely on erroneous decisions of upper internal edges without the possibility to recover by correct decisions in lower internal edges.

Thus essentially, our approach consists of training $|V|$ independent binary support vector machines (which can be done highly efficiently in parallel!) such that the score $f_j(x) = \langle \tilde{w}_j, \phi(x) \rangle + \tilde{b}_j$ of the j-th SVM centered at edge e_j serves as an estimate for the probability that e_j lies on the path y of instance x, i.e., $Pr(\kappa_j(y) = 1)$. An image $x^{(i)}$ is therefore treated as a positive example for edge e_j if this very edge lies on the path from the root to label $y^{(i)}$ and as a negative instance otherwise, which amounts to the sign of $2\kappa_j(y^{(i)}) - 1$.

We resolve our *local-SVM* optimization problem that can be split into $|V|$ independent optimization problems, effectively implementing a one-vs-all classifier for each edge.

$$\begin{aligned} \min_{\tilde{w}_j, \tilde{b}_j, \tilde{\xi}_j} \quad & \frac{1}{2} \sum_{j=1}^{|V|} \|\tilde{w}_j\|^2 + \sum_{j=1}^{|V|} \tilde{C}_j \sum_{i=1}^{n} \tilde{\xi}_j^{(i)} \\ \text{s.t.} \, \forall i, \, \forall j : \; & (2\kappa_j(y^{(i)}) - 1)(\langle \tilde{w}_j, \phi(x^{(i)}) \rangle + \tilde{b}_j) \geq 1 - \tilde{\xi}_j^{(i)} \\ \forall i, \, \forall j : \; & \tilde{\xi}_j^{(i)} \geq 0. \end{aligned} \tag{2.12}$$

At test phases, the prediction for new and unseen examples can be computed similarly to Equation (2.4). Denote the local-SVM for the j-th edge by f_j, then the score for class y is simply the sum of all edges lying on the path from the root to the leaf y,

$$f_y(x) = \frac{\sum_{j:\kappa_j(y)=1} f_j(x)}{\sum_j \kappa_j(y)}. \tag{2.13}$$

The normalization is required due to varying path lengths in our taxonomies which is a difference compared to the taxonomies considered in (82). The class y which has the maximum score f_y over all classes is selected as the final prediction.

Note that since the entire problem decomposes into $|V|$ binary classification tasks, parallelization becomes possible and thus, the training time of our approach is considerably shorter compared to the structural SVMs. Another advantage is that our local procedures can be directly extended to multi-label problems without taking the maximum operation at the end, but by setting thresholds only which determine whether object categories are included in images or not.

Although our initial motivation was to construct an efficient approximation of the structural SVMs, we would like to remark that there exists a fundamental difference between the structural SVMs and our local-SVM procedure with respect to their optimization target. The constraints of the structure learning in Equation (2.8) aim to order the *set of all class labels* correctly *for each image* in the sense that the SVM score for the correct class label is highest. For our local-SVM approach the SVM constraints aim at ordering the *set of all images* correctly *for each edge* with respect to the binarized learning problem whether an image belongs to a class lying on a path passing through this taxonomy node or not. We remark further that the constraints of the structural optimization problems do not imply necessarily that the set of all images is ordered correctly for the binary classification problem at each taxonomy edge. In order to foster a better intuitive understanding, the difference between both approaches are illustrated in Figure 2.4.

2.2.5 Scoring with Generalized p-means

When we combine the binary classification scores at the edges along a path, it is not necessary to take their arithmetic mean as in (2.13). Instead, our procedures permit more general scoring methods such as the generalized p-means of outputs

$$M_p(z_1, \ldots, z_m) = \left(\frac{1}{m} \sum_{i=1}^{m} z_i^p \right)^{1/p} . \tag{2.14}$$

after scaling to $[0, 1]$. This includes the geometric mean as the limit $p \to 0$ and the harmonic mean for $p = -1$ as well as the minimum as the limit $p \to -\infty$. Tuning of this extra degree of freedom p may improve classification performance. To see this note that the geometric mean and generalized means with negative norms of scores in $[0, 1]$ are upper bounded by a power of the smallest element.

$$s_i \in [0, 1] \Rightarrow \prod_{i=1}^{n} s_i^{1/n} \leq \min_i s_i^{1/n}$$

$$p < 0 \Rightarrow \left(\frac{1}{n} \sum_{i=1}^{n} s_i^p \right)^{1/p} \leq \frac{1}{n^{1/p}} \min_i s_i$$

For positive norms the generalized mean is upper bounded instead by a power of its largest element. In that sense generalized means with non-positive norms are more sensitive to negative outliers and more robust against strong positive outlier votes from edges than generalized

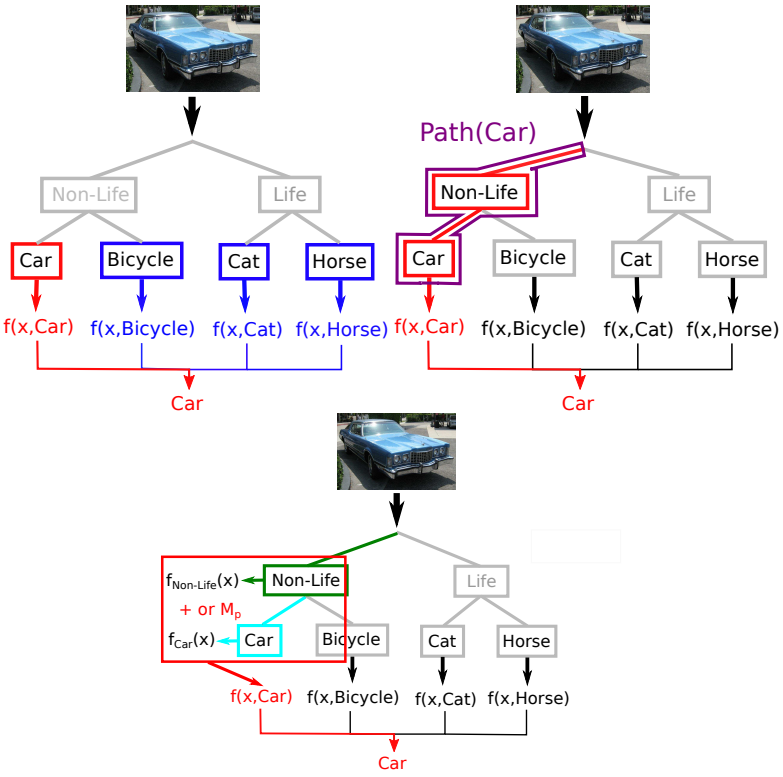


Figure 2.4: Differences between one vs all (top left), structure learning (top right) and local approach (bottom). The one vs all procedure ignores internal nodes of taxonomies and takes the maximum of the SVM outputs at leaf edges. The structured approach takes paths as a whole into account, maximizes the margin between correct and wrong paths in training and returns as a predictor the label of the path with the maximum score. The local procedures optimize each binary problem of passing through a path independently and then combine the outputs of the local SVMs into a score with generalized p-means.

means with positive norms where the distortion by strong positive outliers can be arbitrarily large. The selection of an optimal p-norm thus adjusts the sensitivities to very small votes close to 0 versus very large votes close to 1. The usage of generalized means with arbitrary norms requires the scores to be non-negative and SVM outputs to be scaled.[1]

In order to scale SVM outputs into $[0, 1]$, we deploy a logistic function with fixed parame-

[1]While there exist convex mappings of $\mathbb{R}^1$ to the interval $[0, \infty)$ we are not aware of the existence of a monotonous and continuous mapping of $\mathbb{R}^1$ onto a bounded nontrivial interval which is everywhere concave or convex. This implies that a model using scaling of unbounded inner products cannot be optimized by applying convex methods in the structured output framework.

ters

$$s(y) = \frac{1}{1 + \exp(-10y)}.$$

Experimentally we have seen that learning the logistic regression parameters from data (95) did not further improve performance of image categorization.

Scaling with logistic functions is closely linked to a probabilistic interpretation of a classification procedure. Our current approach does not immediately permit a probabilistic interpretation fitting to a taxonomy graph. This is because we so far have chosen to always consider classification between a part of the categories and all remaining others at each edge, instead of conditioning on its parent nodes.

2.2.6 Baselines

In our experiments, we will use additionally two kinds of classification methods. One is the standard one-vs-all classification: we train one binary SVM for each class which uses the samples of this class as positive labeled data and all the other class data as negative examples. The multi-class labeling is obtained by the class maximizing the scores of all binary SVMs. This is a completely taxonomy-free approach. The second is structured multi-class SVMs which uses the joint feature representation ignoring the taxonomy graph

$$\Psi(x, y) = \phi(x) \otimes \iota(y) = \begin{pmatrix} \phi(x)[[y = c_1]] \\ \phi(x)[[y = c_2]] \\ \vdots \\ \phi(x)[[y = c_k]] \end{pmatrix},$$

where $\iota(y)$ is the vector of the indicator functions $[[y = c_i]]$. This leads to the 0/1 loss from the label kernel

$$2 - 2K_Y(y_1, y_2) = \delta_{0/1}(y_1, y_2),$$

instead of the taxonomical one in the structured taxonomical SVMs. No taxonomy information is used, if the 0/1 loss is deployed as the loss function Δ in Equation (2.8) and (2.9), while it is incorporated indirectly into the learning process, when Δ is the taxonomy loss δ_T.

2.3 Insights from Synthetic Data

In this Section, we discuss when and why the taxonomical approaches might outperform the one-vs-all baseline. Furthermore we can observe differences in AUC scores between leaf and

internal edges which can be linked to flat losses in later experiments on real data. We remark that the one-vs-all baseline can be regarded as a classification procedure only with leaf edges, while the taxonomy-based learning combines classification results of leaf and internal edges, namely by generalized p-means in the local-SVM approach and by implicit arithmetic mean integrated in the structural SVMs.

2.3.1 Experimental Results

To illustrate our claim, we consider a 16 class example with the taxonomy being a binary balanced tree with 16 leaf nodes. Each class is generated from one Gaussian distribution in 15 dimensions. The variances are equal for all Gaussian and are varied to give seven datasets with $\sigma = 1, 0.5, 0.3725, 0.25, 0.1875, 0.125, 0.0625$. The means are distributed such that their Euclidean distance matrix equals the normalized taxonomy loss matrix which has values $i/4, i = 0, \ldots, 4$. Our intention is to illustrate that taxonomy-based learning reduces taxonomy loss, if the data is aligned to the taxonomy. For the sake of computation speed we compare the one-vs-all baseline versus a local algorithm with scoring based on the geometric mean of logistically scaled scores of 19200 data points each independently, where we use 200 samples per class for training and the remaining 1000 per class for testing. We deployed Gaussian kernels here, set the width to be the mean of squared distances and normalized all kernels to have standard deviation one in Hilbert space.

Table 2.1 shows the 0/1 and taxonomy losses of one-vs-all and our local SVM procedure with the scaled geometric mean over different noise levels. The standard deviations are computed between the 15 draws.

The local algorithm improved the one-vs-all baseline significantly under the taxonomy loss in all cases. The relative improvements are more than 2% with the maximum above 5% for $\sigma = 1/8$. We also conducted Wilcoxon's signed rank test, which showed that all performance gains are significant with p-values of orders 10^{-4} or 10^{-5}. Surprisingly, the local SVM procedure the taxonomy compares favorably with the baseline under the flat 0/1 loss as well.

There is an intuitive explanation why hierarchical approaches do improve losses consistent with the hierarchy compared to one versus all classifiers. One versus all classifiers attempt to rank the images belonging to positive class highest. Classifiers from superclasses in a hierarchy attempt to rank the images belonging to the positive class *and similar classes* to be highest. Averaging many versus all classifiers from superclasses with one versus all classifiers at the

leafs achieves a tradeoff between both aims. At the same time such an averaging can potentially harm the zero-one-loss which does not consider similarities encoded in a taxonomy.

Table 2.2 shows the AUC score (equation (2.15)) (96) at different levels in the hierarchy.

$$AUC(f, \{(x_i, y_i)\}) = \frac{\sum_{i:\ y_i=+1} \sum_{k:\ y_k=-1} \mathbb{I}\{f(x_i) > f(x_k)\}}{|\{i:\ y_i = +1\}| \cdot |\{k:\ y_k = -1\}|} \tag{2.15}$$

It allows to judge how difficult the learning problems are at the internal edges compared

Table 2.1: Synthetic data perfectly aligned to the taxonomy: Losses of the one-vs-all baseline (left) versus the local procedure with taxonomy (right) for different label noise levels. $\delta_{0/1}$ is the zero-one-loss. δ_T is the taxonomy loss. Lower losses are better.

	one-vs-all		local-SVM approach	
σ	$\delta_{0/1}$	δ_T	$\delta_{0/1}$	δ_T
1	89.10±0.32	67.09±0.34	**88.59**±0.34	**65.69**±0.35
1/2	78.24±0.32	51.37±0.31	**77.84**±0.39	**50.27**±0.35
3/8	69.30±0.38	41.29±0.28	**68.94**±0.39	**40.21**±0.29
1/4	51.61±0.52	25.05±0.26	**51.26**±0.52	**24.17**±0.22
3/16	37.32±0.46	14.94±0.23	**36.91**±0.48	**14.24**±0.23
1/8	19.49±0.39	6.05±0.11	**19.12**±0.41	**5.70**±0.12
1/16	2.41±0.13	0.61±0.03	**2.38**±0.13	**0.60**±0.03

Table 2.2: Synthetic data perfectly aligned to the taxonomy: AUC scores in the taxonomy for $\sigma = 1/4$ at different levels. Higher scores are better.

level in taxonomy	1	2	3	4 (leaf)
AUC	99.21	97.78	95.42	92.40

Table 2.3: Synthetic data perfectly aligned to the taxonomy: At which level does misclassification occur for $\sigma = 1/4$?

level in taxonomy	1	2	3	4 (leaf)
Differences of Error Rates	-1.55	-0.68	0.48	1.74

to leaf edges. Note that we observe on this synthetic dataset a higher AUC score on internal edges compared to leaf edges and a decrease in the flat zero-one-error compared to the one versus all baseline. This implies that the learning problems are easier on superclass level than at the leaf edges. This might explain why we observe here an improvement in the flat zero one loss as well. It is not straightforward in a statistical sense that optimizing for one loss improves another loss as well. As an explanation we propose that in this synthetic case the features allow a good generalization at superclass level because the given taxonomies are perfectly aligned to the similarities between classes at the feature level. The higher AUC score at internal edges compared to leaf edges supports this view. This good alignment might be also the case when learning similarities from visual features and explain results for flat losses in (78, 85) but it cannot be expected to hold in general when a taxonomy is provided independent of visual features. We will return to this observation in the forthcoming Section 2.4 on experiments on real data.

Table 2.3 shows another aspect of hierarchical averaging: given a pair consisting of true and predicted label we can ask where in the hierarchy the error did occur. This leads to two histograms, for the taxonomy-based and for the one versus all classifier. The Table shows the difference between both histograms. Negative values imply a reduction of errors at this level for the taxonomic method. We see that under our taxonomy based approach the classification errors are moved to lower levels in the hierarchy compared to a flat one versus all classification implying that confusions occur more often between taxonomically closer classes.

2.3.2 Robustness by p-means

The parameter p of the generalized controls robustness against outlying classifier outputs. Negative p's make the mean robust against upper extremes while in the opposite cases lower extremes are suppressed. To see this we conducted an experiment on controlled perturbation of SVM outputs over the toy data. We fixed a priori a set of 10% of the samples to be perturbed and for each sample one edge in the taxonomy to be perturbed. We applied these fixed sets to values of perturbation factors $\{+8, +4, -4, -8\}$. The perturbation is computed for a sample by adding to the SVM output of this sample the factor times the standard deviation of the outputs of the SVM corresponding to the taxonomy node. The negative factors allow to simulate large negative outliers, the positive factors large positive outliers. Table 2.4 shows the results.

We can see that for large positive distortions both positive means perform lower than geometric mean and a negative mean.

For large negative distortions the first ranks are held by the non-scaled arithmetic mean and a scaled positive mean. These two methods suffer less from negative outliers than negative means. Furthermore we observe in both settings that unscaled variants are less robust than scaled ones.

Finally the last part of the Table 2.4 shows a result where 80% of the perturbed samples are modified by a factor of $+4$ and 20% by -4. Here the geometric mean turns out to be the best choice which corresponds well to our empirical findings in Section 2.4.5. We conclude that the geometric mean is well suited to deal with SVM outputs which suffers from positive and negative outliers in taxonomy edges coming from noisy classification problems.

In summary, we would like to emphasize that classification techniques with taxonomies can improve the one-vs-all baselines, under the taxonomical loss and the flat zero one loss.

2.4 Experiments on Real World Multi-class Data

2.4.1 Datasets

For the present work, we constructed multi-class classification datasets with taxonomy trees between object categories by modifying the benchmarks Caltech256 (83) and VOC2006 (84).

Caltech256 all classes.
The Caltech256 dataset (83) contains 256 classes of objects and one clutter class. For an initial experiment allowing comparison to results from other publications we have taken 50 images from each of the object classes and employed the taxonomy as provided in the report (83). The only changes we made were to add pisa-tower to the taxonomy graph as it seemed to be missing and moved iris to flowers from air animals. Unfortunately, using $50 \cdot 256 \cdot 0.9 = 11520$ samples for training using ten-fold crossvalidation is beyond the scope of the structured prediction baselines on our hardware. Therefore we considered subsets of classes which will be described below. The result for all 256 object classes can be looked up in section 2.4.7.

Caltech256 animals.
We consider all 52 real world animal classes from the Caltech256 dataset (83) which yields 5895 data points (see Figure 2.5). They form a multi-class problem with mutually exclusive classes. We used a taxonomy based on a recherche of biological (phylogenetic) systematics consisting out of 92 nodes constructed a priori. We have chosen this subset for two reasons.

Table 2.4: Synthetic data perfectly aligned to the taxonomy: Differences in taxonomy loss and 0/1 loss to unperturbed SVM outputs and absolute ranks between all four methods. Lower losses are better.

unperturbed	nonscaled M_1	sc M_2	sc M_0	sc M_{-2}
rank δ_T	**1**	3	2	4
rank $\delta_{0/1}$	**1**	3	**1**	4
perturb=+8	nonscaled M_1	sc M_2	sc M_0	sc M_{-2}
diff. δ_T	1.8	0.14	0.04	0.05
rank δ_T	4	3	**1**	2
diff. $\delta_{0/1}$	1.91	0.27	0.15	0.15
rank $\delta_{0/1}$	4	3	**1**	2
perturb=+4	nonscaled M_1	sc M_2	sc M_0	sc M_{-2}
diff. δ_T	0.47	0.14	0.04	0.05
rank δ_T	4	3	**1**	2
diff. $\delta_{0/1}$	0.81	0.26	0.15	0.15
rank $\delta_{0/1}$	4	3	**1**	2
perturb=-4	nonscaled M_1	sc M_2	sc M_0	sc M_{-2}
diff. δ_T	0.26	0.03	0.42	0.75
rank δ_T	2	**1**	3	4
diff. $\delta_{0/1}$	0.34	0.13	0.49	0.73
rank $\delta_{0/1}$	**1**	2	3	4
perturb=-8	nonscaled M_1	sc M_2	sc M_0	sc M_{-2}
diff. δ_T	0.68	0.03	0.7	0.75
rank δ_T	2	**1**	3	4
diff. $\delta_{0/1}$	0.73	0.12	0.74	0.74
rank $\delta_{0/1}$	2	**1**	3	4
80% +4, 20% -4	nonscaled M_1	sc M_2	sc M_0	sc M_{-2}
diff. δ_T	0.41	0.09	0.11	0.12
rank δ_T	4	3	**1**	2
diff. $\delta_{0/1}$	0.53	0.21	0.2	0.23
rank $\delta_{0/1}$	4	3	**1**	2

Firstly, it is a natural multi-class dataset in the multimedia image domain. Secondly, it allows to define a taxonomy in an indisputable way prior to looking at image content, namely via biological systematics. For the remaining 204 classes from Caltech256 we would have to rely on human experience of some sort which might lead to some kind of unintentional appearance-based optimization of when choosing a taxonomy. The technical report on the Caltech256 dataset (83) contains a hierarchy. We have chosen not to use its construction principle because it is somewhat arbitrary as stated by the authors of the technical report themselves and from our own point of view is not biologically plausible. It groups all animals in four flat subgroups: insects, land, air and water based lifeforms. As stated in the introduction the usage of phylogenetic systematics resulted in a taxonomy which is indeed not fully consistent to the subjective visual similarities of the authors which diverge for example for crabs and horseshoe crabs but also as shown in Figure 2.2 potentially for superclasses in the taxonomy. The hierarchy contains in contrast to many preceding works paths with varying lengths. We omitted fantasy animals like Minotaurs and Unicorns from the Caltech256 set, as there is no objective way to incorporate them into biological systematics. The full taxonomy is given in Figure 2.12.

Caltech256 animals thirteen classes subset

For further experiments, we select 13 classes - all Protostomia (praying-mantis, grasshopper, cockroach, house-fly, butterfly, trilobite, centipede, crab, spider, scorpion, horseshoe-crab, octopus, snail) from the *Caltech256 animals* dataset. This corresponds to one subtree in the original taxonomy over all 52 classes. The total number of the images is reduced to 1308. This allows us faster experimentation with the structural approaches which was the main reason for choosing this subset. We deploy as taxonomy the corresponding subtree with 21 nodes of that of *Caltech256 animals* which is still challenging in its topology due to non-balanced tree structure and varying path lengths.

VOC2006 multi-class data

We use the VOC2006 dataset (84) consisting of 10 object classes and 5301 images (see Figure 2.6). We have modified the VOC2006 labels in order to obtain a multi-class problem with mutually exclusive classes. To achieve such exclusive labeling, for each image all positive labels except for a randomly chosen one are suppressed. We remark that this process induces additional label noise.

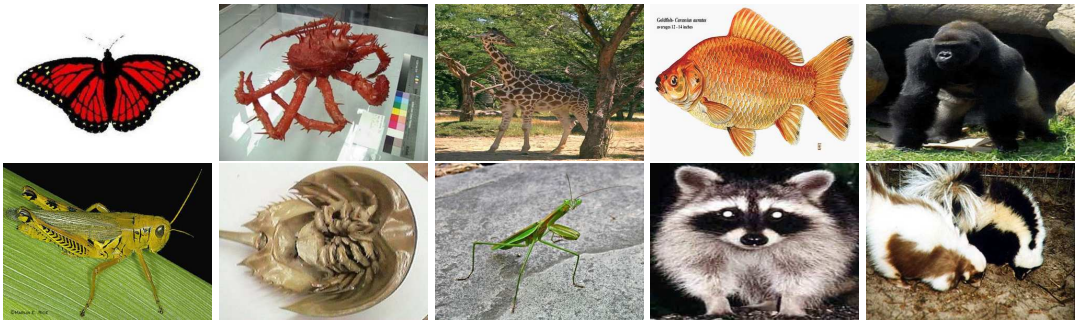


Figure 2.5: Caltech256 animals dataset example images.

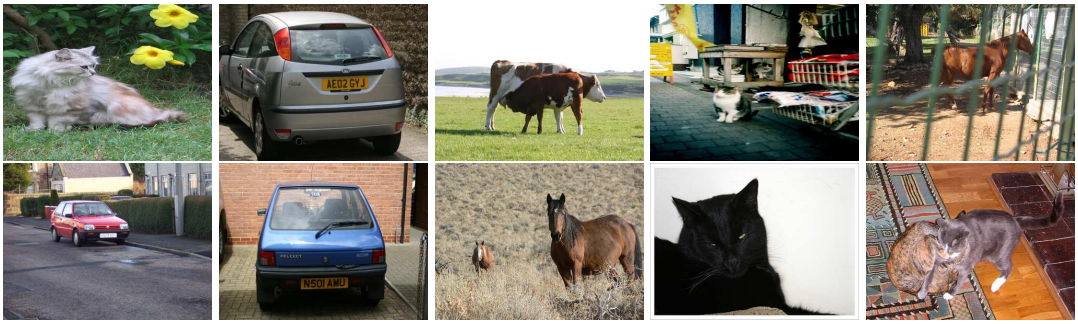


Figure 2.6: VOC2006 dataset example images.

2.4.2 Image Features

For the following experiments, we used bag of words (BoW) representations based on the SIFT descriptors (16) as image features. The BoW features were constructed in a standard way: using the code from (23), the SIFT descriptors were computed on a dense grid of step size six over the color channel triples {red, green, blue} (RGB) and {grey, opponent color 1, 2} (OPP, see equations (1.5),(1.6), (1.7)). Then, for both triples, 8192 visual words (prototypes) were generated by using extremely randomized clustering forest (ERCF) clustering (31) via 16 trees with 512 leaves each based on large sets of SIFT descriptors selected randomly from the training images following (23). For each image, each SIFT feature was assigned to one leaf for each of the 16 trees. We have chosen the supervised ERCF procedure over k-means as it does greatly reduce the time necessary for clustering of visual words and bag of word computation while having comparable performance. The sum of these mappings resulted in one histogram for each image within each cell of the spatial tilings 1×1, 2×2 and 3×1. The idea of a

spatial tiling is to split each image into a set of regularly shaped spatial tiles, to compute one BoW feature for each tile separately and finally to concatenate the BoW features over all tiles into one BoW feature (36, 97). Finally, we obtained 6 BoW features (2 color channels sets $\times 3$ sets of spatial tilings) with dimensionalities 8192, 4×8192 and 3×8192 depending on the spatial tiling. For Caltech256 data we omitted the two kernels based on tilings 2×2 as they did degrade the one-vs-all baseline performance already. We do not aim here at the best possible baseline performance which might be achieved by adding carefully selected sets of additional features. Instead we focus on the effect of a given hierarchy and non-flat loss functions. We note however that high-dimensional bag of words models have been able to achieve superior performance in recent object categorization challenges (23, 98, 99) which motivates our choice of these features.

2.4.3 Image Kernels and Regularization of SVMs

We used the exponential χ^2-Kernel (equation (1.26)) for comparing the image feature histograms (50, 51). The bandwidth σ of the χ^2 kernel in (1.26) is thereby heuristically chosen as the mean χ^2 distance (equation (1.27)) over all pairs of training examples, as done, for example, in (52). All kernels have been normalized to standard deviation in Hilbert space set equal to one which in practice limits the range where to search for an optimal regularization constant. We combined all kernels via addition.

In the local-SVM procedure, we used two regularization constants (one per class) for all binary problems in order to compensate for the unbalanced ratios between positive and negative classes. The regularization constant of the smaller class was obtained by multiplying that of the larger class[1] by the ratio between the two samples. For the structured SVMs we used as regularization parameter $\tilde{C} = 16|V|$ for the taxonomical procedures and $\tilde{C} = 16k$ for the multi-class ones, where $|V|$ and k are the number of nodes and classes, respectively.

This is motivated by comparing the main objective of one local SVM

$$\min_{\tilde{w}_j, \tilde{b}_j, \tilde{\xi}_j} \quad \frac{1}{2}\|\tilde{w}_j\|^2 + \tilde{C}_j \sum_{i=1}^{n} \tilde{\xi}_j^{(i)}$$

to the one from a structured SVM

$$\min_{\tilde{w}, \tilde{\xi}} \quad \sum_{j=1}^{|V|} \frac{1}{2}\|\tilde{w}_j\|^2 + \tilde{C} \sum_{i=1}^{n} \tilde{\xi}^{(i)}.$$

[1] The regularization constant of the larger class was fixed to 16 which corresponds to our experience that high-dimensional Bag-of-words features perform better under hard margin training.

We note that the ratio between the weight norm $\|w\|^2$ and the slacks $\xi^{(i)}$ is roughly up-scaled by a factor equal to the number of nodes. We have checked experimentally that using much lower regularization constants damages the performance of the structural SVMs, while much higher regularization constants did not improve the results anymore. Since the sizes of the object categories are balanced, we do not have to assign one regularization constant for each class separately.

2.4.4 Comparison Methodology

All considered methods can be divided into structured and structure-free as well as taxonomical and taxonomy-free approaches (Table 2.6). Due to limited space, we will use the abbreviations listed in Table 2.6 to in our experimental results.

There are three ways to use the taxonomy. The taxonomy loss as performance measure is used on all methods. The taxonomy loss as part of the training procedure is used in all structured SVMs according to equation (2.8). The taxonomy structure is incorporated in all taxonomical approaches but not in the structured multi class procedures.

We will use as baselines the structure-free one-vs-all classification and taxonomy-free multi class SVMs with margin and slack rescaling trained using zero-one loss $\delta_{0/1}$ or taxonomy loss δ_T. The taxonomy-based algorithms to be tested are, firstly, the structured SVMs with nontrivial taxonomies in margin (2.8) and slack rescaling formulation (2.9) and, secondly, structure-free methods where we obtain scores for each concept class via the arithmetic mean over the component SVM outputs and via generalized means of SVM outputs which are scaled using logistic functions.

We used SVMmulticlass (100) and modified versions thereof for the structured approaches.

Table 2.5: Classification of methods.

	structure-free	structured
taxonomy-free	one vs all (Section 2.2.6)	struct multi-class SVMs (Section 2.2.6)
taxonomical	local taxonomy (Section 2.2.4)	struct taxonomy SVMs (Section 2.2.2)

The non-structured methods have been implemented using shogun toolbox (74) with the SVM-light solver. We note that SVMlight is also deployed in the optimization procedures of the SVMmulticlass implementations.

The error measurement is done for the multi-class problems using the 0/1- and taxonomy loss from equation (2.2). For all multi-class problems we use 20 splits into training and test data with 50 images per class in each split.

Furthermore we use for some experiments the Average Precision (AP) Score for a class (see Equation (2.16)) and the mean Average Precision score (mAP) obtained by averaging the average precisions over all classes.

For computation of the AP score we assume that the pairs of classifer outputs and ground truth labels $(z^{(c)}, y^{(c)})$ for a class in question c are sorted according to the descending order of their output scores $z_k^{(c)}$ over the data sample index k. The average precision (AP) score for $n_+^{(c)} = \sum_{i=1}^n \mathbb{I}\{y_i^{(c)} = 1\}$ positively labeled samples of class c is defined as

$$AP^{(c)}((z_k^{(c)}, y_k^{(c)})_{k=1}^n) := \frac{1}{n_+^{(c)}} \sum_{i=1}^{n} \mathbb{I}\{y_i^{(c)} = 1\} \frac{1}{i} \sum_{k=1}^{i} \mathbb{I}\{y_k^{(c)} = 1\} \tag{2.16}$$

2.4.5 Experimental Results: Performance Comparisons

At first, we would like to remark the difficulty inherent in the datasets. Table 2.7 shows the 0/1 loss and the average precisions (AP score) of the one-vs-all baselines for the three multi-class

Table 2.6: Abbreviations for compared methods.

structured multi-class baseline	
struct mc mr	with margin rescaling
struct mc sr	with slack rescaling
taxonomical structural learning	
struct tax mr	with margin rescaling (2.8)
struct tax sr	with slack rescaling (2.9)
the local procedure with taxonomy	
local tax AM	with arithmetic mean (2.13)
local tax scaled GM	with geometric mean after scaling
M_p	with p-mean after scaling

datasets.

The AP score is a rank-based measure which was deployed as the performance criterion in the recent Pascal VOC challenges. For VOC2006 the results for 20 splits perform worse due to sample size effects as they use only 500 training data in each split as compared to over 5000 points for the 20-fold cross-validation.

Table 2.7: One-vs-all baseline performance on multi-class datasets. Lower losses and higher AP scores are better.

dataset	0/1 Loss	AP score
Cal256 animals	62.56	34.34
Cal256 13 class subset	57.04	43.69
VOC2006, multi-class, 20 splits	50.54	54.75
VOC2006, multi-class, 20-fold crossval	33.56	70.50

The comparisons for Caltech256 animals and its 13 class subset are shown in Tables 2.8 and 2.9. For simplicity, we present only the best result among all options for each of structural multi-class, local taxonomy-based and structural taxonomy-based procedures. The full Tables listing all results can be found in the Appendix (Tables 5.1, 5.2 and 5.3. As expected, the taxonomy-based methods outperform the taxonomy-free baselines in terms of the taxonomy loss by 3-5% relatively. For both datasets, our local SVM procedure improves structure learning with taxonomy by 2-3% relatively. The gains of the taxonomy-based approaches under the taxonomy loss are achieved at the cost of slightly increasing the 0/1 loss. It is notable from Table 2.9 that merely including the taxonomy loss in a structured multi-class algorithm (as an intermediate step of incorporating taxonomical information) does not yield sufficient performance gain under the taxonomy loss. Optimization for taxonomy loss comes at the cost of performance deterioration under the 0/1 loss. This is not surprising, because the baselines, one vs all and structured multi-class models directly optimize for the flat hinge loss which is more closely related to the 0/1 loss than to the taxonomy loss. Since this problem occurs for all hierarchical methods including the structured prediction based methods it does point out the considerable difference between the canonical flat loss and what a user might desire. From an optimization viewpoint minimizing a different loss leads to a different model. Therefore

merely the scale of change might be surprising. The relation of 0/1 loss to AUC scores at internal edges across datasets will be discussed in Section 2.4.8.

Table 2.8: Errors on Caltech256 animals (52 classes), 20 splits. Lower losses are better.

Method	Taxonomy Loss	0/1 Loss
one vs all	30.66 ± 0.46	**62.56** ± 0.67
best local tax: scaled GM	**29.62** ± 0.34	76.19 ± 0.57
best struct tax: mr	30.58 ± 0.31	81.19 ± 0.53

Table 2.9: Errors on Caltech256 animals 13 class subset data, 20 splits. Lower losses are better.

Method	Taxonomy Loss	0/1 Loss
one vs all	42.49 ± 1.46	**57.04** ± 1.98
best struct mc: sr, $\Delta = \delta_{0/1}$	42.48 ± 1.50	57.06 ± 2.00
best local tax: scaled GM	**40.58** ± 1.15	58.33 ± 1.50
best struct tax: mr	41.48 ± 1.22	61.54 ± 1.55

Table 2.10: Errors on VOC2006 as multi-class problem, 20 splits. Lower losses are better.

Method	Taxonomy Loss	0/1 Loss
one vs all	27.09 ± 1.88	50.54 ± 2.51
best struct mc: mr, $\Delta = \delta_T$	26.37 ± 1.77	51.04 ± 2.53
best local tax: scaled GM	25.86 ± 1.56	**50.10** ± 2.29
best struct tax: mr	**25.78** ± 1.67	50.17 ± 2.17

Table 2.10 shows the performance comparison for the VOC2006 multi-class problem. Similar to the Caltech animals datasets, the taxonomy-based methods outperform the one-vs-all

baseline in terms of the taxonomy loss by 5% relatively. On the other hand, there are some differences from the previous cases. At first, our local SVM procedure is rather on par with the structural counterpart. Secondly, the intermediate step, the structure multi-class procedure with the taxonomy loss δ_T improved the one-vs-all baseline significantly under the taxonomy loss. Finally, the taxonomy-based approaches improved slightly the taxonomy-free baselines under the 0/1 loss as it was already the case for the synthetic data example.

As a sanity check for structured implementations we remark that the structure-free methods perform approximately equally well to their structured counterparts for both taxonomy and 0/1 losses. Since for the flat 0/1 loss setting we used SVMstruct in its unmodified formulation, this is clearly a property of the data rather than a potentially faulty implementation of structured approaches.

In summary, we observed that the taxonomical approaches outperform the taxonomy-free baselines under the taxonomy loss, as was the case for the synthetic data. Unlike in the synthetic data the zero-one error was slightly increased by optimization of taxonomy based losses for both Caltech datasets. The choice of the loss function determines the algorithm to be used. It is not expectable in a statistical sense that a taxonomical model improves a flat loss under all circumstances, however there is a tendency for relatedness of zero one loss and differences of AUC scores across levels (see also discussion in Section 2.4.8). The local taxonomy-based methods are slightly worse than structured taxonomy ones on VOC2006 dataset, but considerably better on both Caltech256 animals problems. We would like to emphasize that the way of averaging is important to achieve better performance. Note that the scaled geometrical mean compares favorably with the arithmetic mean. Indeed, when we examined the generalized p-means in a wide rage of the parameter p, parameters close to 0 (i.e. the geometrical mean) achieved the minimum values both under the 0/1 and taxonomy losses.

2.4.6 Remark on Training Time

In all three data sets the local SVMs are much faster to train when compared to structured taxonomy approaches (cf. Table 2.11). The local SVMs can be parallelized by training each edge as a separate optimization problem, an advantageous property when scaling the number of object categories. Another beneficial scaling characteristic when increasing the number of samples is the possibility to reduce the training set for each edge individually since it is sufficient to control the performance of the binary classification problem at each edge separately. Certain steps in the structural approaches like finding the most violated constraints can be parallelized

to e.g. multicore machines which typically accounts for four or at most eight cores. The used code may have potential for further problem-specific optimizations. The speed gains by using local SVMs are large factors of over 10. Thus we do not expect the advantage of the local SVMs to disappear against a multicore-parallelization of structural support vector machines. Furthermore the parallelization of local SVMs into optimization problems restricted to single edges can be achieved generically over more than 8 cores. Another performance reducing factor was excessive main memory usage of structural algorithms of up to 16 Gigabyte per task which in practice leads to additional slowdowns compared to many small tasks as solved by the local SVMs.

Table 2.11: Training times, the multiplier for local models shows separability into independent jobs.

Method	Dataset	Training time
one vs all	Cal256 animals, 52 classes	3.69s $\times$ 52
local tax	Cal256 animals, 52 classes	3.69s $\times$ 92
struct. tax	Cal256 animals, 52 classes	35.13 h
one vs all	Cal256 animals, 13 classes	0.5s $\times$ 13
local tax	Cal256 animals, 13 classes	0.5s $\times$ 21
struct multi-class	Cal256 animals, 13 classes	15.1 min
struct tax	Cal256 animals, 13 classes	44.9 min
one vs all	VOC2006	$<$0.5s $\times$ 10
local tax	VOC2006	$<$0.5s $\times$ 19
struct multi-class	VOC2006	9.4 min
struct tax	VOC2006	28.7 min

2.4.7 Discussion

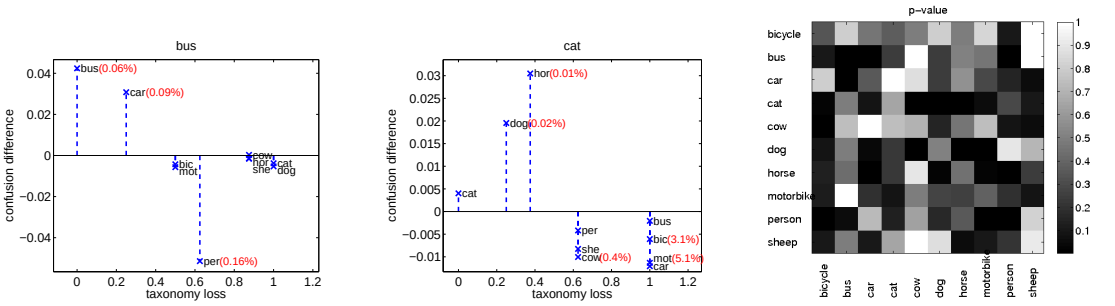
Confusion Between Object Categories Figures 2.8 and 2.9 provide example images where the results from the local taxonomy approach differs compared to the one versus all baseline. Each image comes with a graph on the taxonomy. The ground truth label is green. The choice by one versus all is marked in magenta and the path to the choice by hierarchical classification is given in blue. All relevant paths have attached the SVM outputs to them (see also Figure 2.4).

Figure 2.8 shows typical cases when the hierarchic approach fails. It is caused by false positive outlier votes at internal edges which are too strong in order to be averaged out. Figure 2.9 shows cases when the hierarchical approach improves over a flat one versus all baseline. Typically the votes from internal edges can average out and thus overrule false positive and too negative votes at the leaf edges. The upper part of Figure 2.9 shows a case when a taxonomically more plausible result can be achieved by using a hierarchy even when the classifier for the leaf edge belonging the ground truth gives a too negative vote. In the lower part the hierarchic approach classifies the image correctly.

By comparing the confusion pattern of our taxonomy based procedure with that of the one-vs-all baseline, we observe clear qualitative differences. Figure 2.7 shows confusion differences between the two approaches (y-axis) versus the taxonomy losses (x-axis) for (a) bus and (b) cat of the VOC 2006 data. As expected, we can find the general tendency that the taxonomy based method confused more with the categories with lower taxonomy losses, while it can reduce the error with those with higher taxonomy losses. We also checked significances of all confusion differences by a Wilcoxon signed-rank test from 20 random repetitions. Its p-values are summarized in the panel (c) (row: true classes, column: predicted classes). For instance, for (a) bus class, more images were correctly classified as bus (p-value = 0.06%) and confusion with person reduced significantly (0.16%) at the cost of increasing the error by prediction of cars (0.09%) which is in the taxonomy the closest category to bus. Similar relations hold for (b) cat class: confusions with the closer categories dog and horse increased, which brought improvements in confusions with farther away classes cow (0.4%), bicycle (3.1%) and motorbike (5.1%).

It is worth to point out that the improvement of taxonomy losses by hierarchical classification which was observed in Section 2.3 (see Table 2.3) and Section 2.4.5 implies that erroneous decisions are moved to lower levels in the hierarchy compared to baselines. This yields a more plausible, i.e. more human-like, result based on the taxonomy.

Comparison with Greedy Walks We also analyzed the performance for local taxonomy approaches with hierarchical classification using greedy path-walks (79). We regard this direction rather as a side topic with respect to our comparison of structured versus local models. In this approach for each node in the taxonomy the set of negative examples is restricted to those with the class labels of the parent node. For example, for the class cat in the taxonomy from Figure 2.3, a binary SVM is trained only with samples of classes Carnivora, i.e. cats and dogs. Such



(a) bus (b) cat (c) p-values of Wilcoxon test

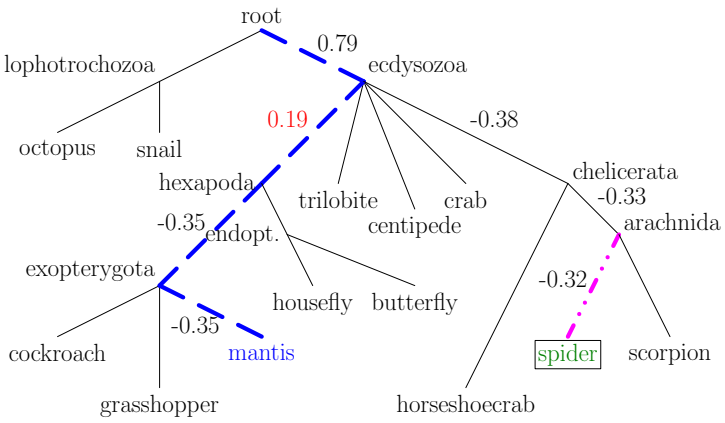
Figure 2.7: Confusion differences between our local SVM with taxonomy and the one-vs-all classification (y-axis) versus the taxonomy losses (x-axis) for (a) bus and (b) cat from VOC 2006 categories (bic = bicycle, hor = horse, mot = motorbike, per = person, she = sheep). Positive values denote more confusions by the proposed method. Significances of the differences are checked by Wilcoxon signed-rank test whose p-values are summarized in (c) (row: true classes, column: predicted classes).

greedy walks lead to performance decrease. This is not surprising. Since the binary SVM at the leaf edge 'cat' takes only images annotated with dog as negative samples, it may give highly positive scores to images containing horses or motorbikes. It is possible that the classifiers at the upper edges, e.g. the nonlife-versus-life or the carnivora-versus-classifier misjudge some of these images and that the cat-versus-dog classifier finally annotates them as cat with very high confidence.

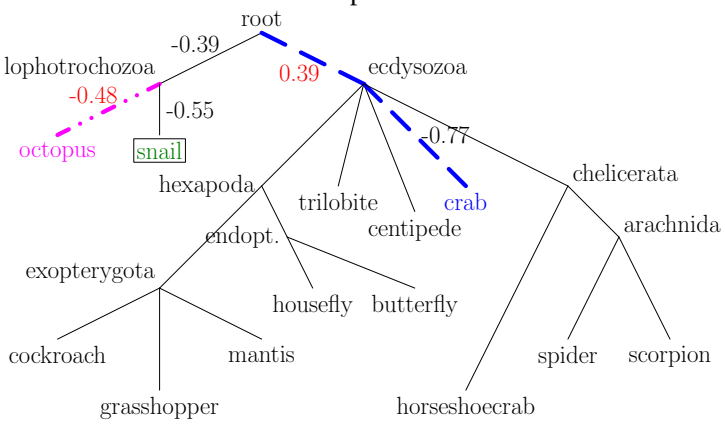
We have found that the greedy walks strategy itself is detrimental. We obtain for both datasets a moderate rise in 0/1 loss and a sharp rise in taxonomy loss. In that sense the local approach adopted here is superior to other possible simpler local solutions. Performances of greedy walks can be found in Appendix (Tables 5.1,5.2,5.3).

The greedy approach has two advantages in running times compared to the local approach presented here. During training it deals at each edge only with classifiers working on subsets of all categories which leads to a reduced amount of training data. During testing we have to follow only one path for each sample. The local approach presented here can be, in principle, modified by subsampling from the set of negative classes during training so that it uses the same amount of training data as the greedy approach. It would still retain the advantage of being able to suppress votes for outlier images as described above, i. e. when a car image is tested in a cat versus dog classifier in a greedy walk scheme. While the greedy approach is the fastest option during test time, the local approach introduced here can be interpreted as a

Figure 2.8: Example images where the hierarchical classifier is inferior to the one versus all baseline on Caltech 256 animals, 13 classes. Boxed green denotes the ground truth label, dashed blue the path to the choice by hierarchical classifier and dash-dotted magenta the decision by one versus all.

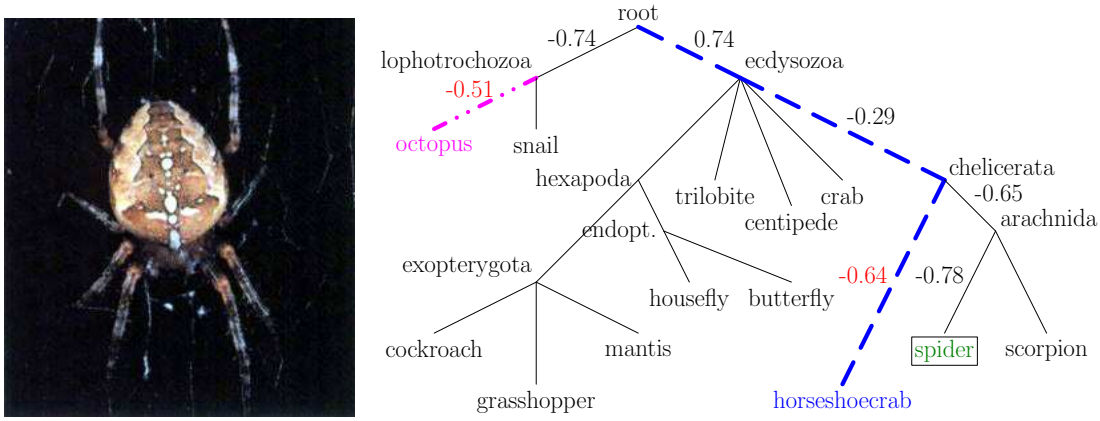


(upper) **hierarchic:** praying-mantis; **one versus all:** spider; **ground truth:** spider; Strong false positive vote for Hexapoda in hierarchical approach, the appearance of the spider does not show 8 legs clearly and is somewhat similar to mantids in pose and color.

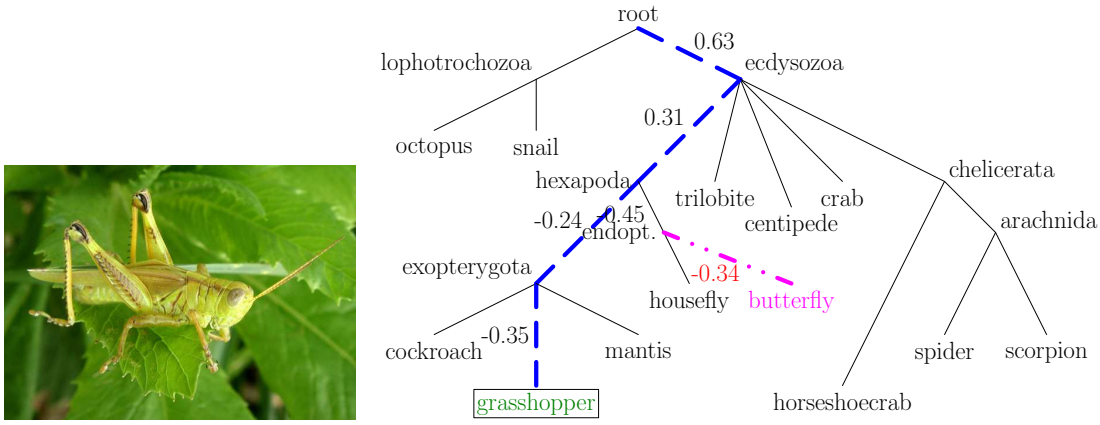


(lower) **hierarchic:** crab; **one versus all:** octopus; **ground truth:** snail; Strong false positive vote for Ecdysozoa causes hierarchy classifier to fail while one vs all predicts a taxonomically closer animal to the ground truth.

Figure 2.9: Example images where the hierarchical classifier outperforms the one versus all baseline on Caltech256 animals, 13 classes. Boxed green denotes the ground truth label, dashed blue the path to the choice by hierarchical classifier and dash-dotted magenta the decision by one versus all.



(upper) **hierarchic:** horseshoe crab; **one versus all:** octopus; **ground truth:** spider; The hierarchical approach predicts a horseshoe crab which belongs to the same subphylum Chelicerata as the spider, the score at the one vs all edge for octopus is too large. The score in the one versus all edge for horseshoe crab is too large, too, which prevents a correct classification as a spider.



(lower) **hierarchic:** grasshopper; **one versus all:** butterfly; **ground truth:** grasshopper; The grasshopper gets classified correctly in the hierarchical approach at the Exopterygota versus all edge which overrules the too low vote at the leaf edges for class grasshopper compared to butterflies.

compromise between the structured SVMs and the greedy walks in terms of training and testing time. It achieves a trade-off between speed and precision.

Outlook - Larger Numbers of Classes: Caltech256 Full Here we consider the results for all 256 object classes from Caltech256. We omitted the clutter class and computed one k-means prototyped Bag of Words kernel based on 1000 words over the RGB color channel. We used 50 images per class and ten-fold crossvalidation which resulted in a training set size of 11520 samples. We were not able to compute the solutions from structured prediction methods however we are still able to compare one versus all against our local SVM approach. We observe in Table 2.12 qualitatively the same results as for the other, smaller, datasets. The taxonomy based approach improves on the taxonomy loss at the cost of setbacks in the zero one loss when compared to one versus all. The one versus all baseline performance ranges between the baseline used in (85) and the best kernel from (101).

Table 2.12: Errors on Caltech256 all classes except for clutter, 10 splits. Lower losses are better.

Method	Taxonomy Loss	0/1 Loss
one vs all	34.31 ± 0.74	**68.93** ± 1.23
local tax AM	33.04 ± 0.7	72.91 ± 1.16
local tax scaled GM	**32.77** ± 0.6	72.55 ± 1.14
local tax greedy path-walk	37.81 ± 0.71	77.96 ± 1.3

2.4.8 Generalization Ability of Learning with Taxonomies

We have formulated in the introduction 2.1 of this chapter a more human-like classification in the sense that errors between taxonomically far categories are reduced as one of our goals. We have observed experimental evidence that taxonomical losses are indeed reduced when using hierarchical classification instead of the one-versus-all baseline.

However, there is a gap between our goal and the experimental results: On the one hand, humans are able to generalize higher level categories very well, seemingly better than more specific low level categories. For example humans can label cars very well even if their optical appearance is quite diverse as with old-timers, converted cars in strange shapes or rare car

models, whereas identifying a car brand or even a specific car model constitutes a much more difficult task for humans. On the other hand, the improvement in taxonomical losses observed here is somewhat limited. For Caltech datasets we observe an increase in flat loss. Please note that for local taxonomic models the difference between the one-versus-all baseline and classification with taxonomies consists of adding classification problems located at intermediate edges of the taxonomy, see also Figure 2.4 for this aspect. If we assume in analogy to our expectation about human capabilities that the problems at intermediate edges are much easier to classify for our system and thus result in much better recognition rates, then the local taxonomic models should result in much better improvement over the one-versus-all baseline.

We would like to identify reasons for this gap in this section, and point to possible improvements for the future. The obvious observation to start with is given in Table 2.13. We can see that for the Caltech datasets AUC scores at intermediate edges are worse than the AUC scores at leaf edges. The classification tasks at the intermediate edges for the Caltech datasets are more difficult and therefore yield more errors compared to classification at leaf edges, which is in clear contrast to our intuition about human capabilities.[1]

Table 2.13: Mean AUCs on leaf edges versus internal edges for the local-SVM methods. Higher values are better.

Dataset	AUC Leaf edges	AUC Internal edges
Caltech256 52 animals	88.49	84.82
Caltech256, 13 class subset	84.00	78.55
VOC2006 multi-class	86.38	91.40
Synthetic data, $\sigma = 1/4$, 16 classes (Sec. 2.3)	92.40	96.64

The task of learning with taxonomies can be divided into two aspects. The first aspect is the optimization of a non flat loss via the taxonomy structure.

The second aspect is that taxonomy based learning is an averaging using classifiers constructed by forming superclasses from sets of single classes. Adding classifiers for these superclasses with higher error rates, as we have done for the Caltech datasets, is likely to raise error

[1] We showed for the synthetic data statistics per level of the taxonomy in Table 2.3. We use here the coarser discrimination between internal edges and leaf edges because for the taxonomies on the real data the notion of level does not imply a constant difference to the nearest leaf. Leafs have varying path lengths and thus, two edges at the same level may have different distances to the nearest leaf. See Figure 2.12 for an example.

rates. This has been observed for the flat 0/1 loss in Tables 2.8 and 2.9. To shed light on the question why classification problems at superclasses can be harder we will consider additional metrics. The first metric are kernel target alignment scores (59). The kernel target alignment is a similarity measure between the kernel from image features and an optimally discriminative kernel computed from the labels of the classification problems located at the edges of the taxonomy. For a short overview of kernel target alignment we refer to section 1.3.4. Higher scores imply that a kernel is potentially more useful for solving a classification task.

Table 2.14: Mean Kernel Target alignment on leaf edges versus internal edges for the local-SVM methods. Higher values are better.

Dataset	KTA Leaf edges	KTA Internal edges
Caltech256 52 animals	0.0147	0.0241
Caltech256, 13 class subset	0.0431	0.0402
VOC2006 multi-class	0.0662	0.1882
Synthetic data, $\sigma = 1/4$, 16 classes (Sec. 2.3)	0.0675	0.2075

We see from Table 2.14 that the Caltech datasets have low gains in kernel target alignment scores at classification problems located at internal edges relative to kernel target alignment scores at leaf edges. This shows that the kernels when applied to classification at intermediate edges do not provide much higher information content than the leaf classifiers for Caltech datasets. Furthermore the Table 2.14 shows that the differences in AUC values seen in Table 2.13 can be explained by properties of the employed kernel. Therefore we will compute another kernel metric for a subsequent complexity analysis.

We claim that some of the classification problems at intermediate edges may have an increased complexity because they have to discriminate two sets of classes in which *both* sets may have a highly varying visual appearance as a consequence of being a union many different classes. In contrast to that the classification problems at the leaf edges need to discriminate one class against a set of all other classes, i.e. one of the sets consists of a single class which may have lower varying visual appearance than a set of many classes. Note that in our experiments we use the same kernel for all classification problems.

For bringing evidence about the complexity of classification problems we will employ Kernel principal component analysis-based (kPCA) label reconstruction agreement. This method

has been discussed in (102) as a measure of complexity for a classification problem with a given kernel. The idea is to compute the principal components of a kernel in the Hilbert space and sort them according to the descending order of their eigenvalues. Note from Lemma 1 in (102) that for a kernel matrix over a fixed finite set of samples the m-th sorted kernel PCA component is equal to the corresponding eigenvector u_m of the kernel matrix.

For a chosen fixed dimensionality d we can project the labels Y onto the first d sorted kPCA components to obtain projected labels $\hat{Y}$:

$$\hat{Y} = \sum_{m=1}^{d} u_m u_m^\top Y \tag{2.17}$$

The projected labels allow to compute an agreement to the true label as one minus the zero one loss:

$$agr_{01}(\hat{Y}, Y) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}\{\text{sign}(\hat{Y}) = Y\} \tag{2.18}$$

If we project on all kPCA components by setting $d = N$, then we recover the ground truth labels $\hat{Y} = Y \Rightarrow agr_{01}(\hat{Y}, Y) = 1$. The idea of relevant dimensionality analysis (102) and kPCA label reconstruction agreement is that for a low-complexity classification problem the majority of information is contained in a small number of the first sorted kPCA components. Thus, for a low-complexity classification problem the projected labels will have a high agreement to the true labels. We compute the agreement between true and projected labels for the first $d = 2^i, i = \{2, \ldots, 8\}$ kPCA components. We show for each number of components the ratio between the agreements in intermediate and leaf edges in Figure 2.10.

The kPCA ratios are all below 1 implying that more kPCA components are needed at intermediate edges to reach the same accuracy in explaining the labels compared to the number of kPCA components at leaf edges. This is consistent to our claim made above that classes representing intermediate edges have on average an increased complexity given the fixed kernel employed here.

Furthermore the ratios between those accuracies are lowest for Caltech animals and higher for VOC2006 and the synthetic dataset. Therefore, classification problems at intermediate edges have a higher relative complexity for the Caltech datasets. This suggests that adding classifiers which were trained on intermediate edges to the one-versus-all classifiers on leaf edges is less likely to improve classification results for the Caltech animal datasets than for VOC2006 and the synthetic data.

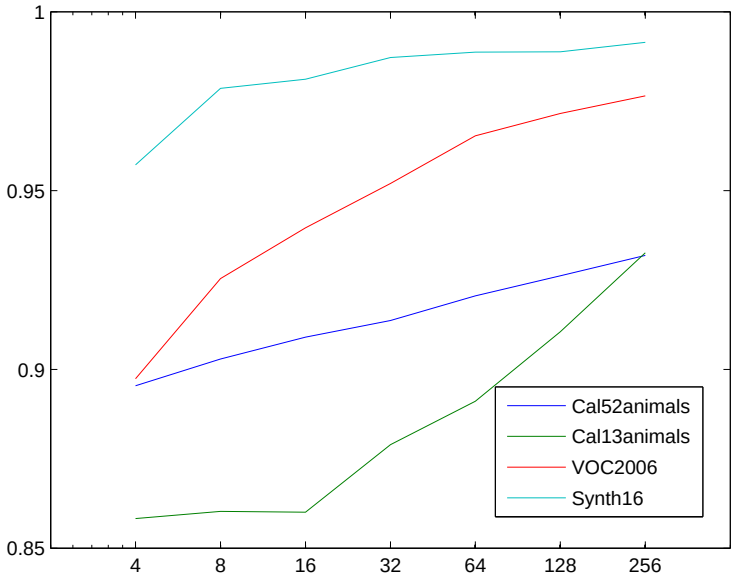


Figure 2.10: Ratios of agreements of kPCA projected labels and ground truth labels. Ratios are computed between classifiers at intermediate edges and leaf edges. The ratios were computed at dimensions 4 to 256. Higher values are better.

This result is what we can expect: both animals taxonomies are built by evolutionary similarities, not visual ones. Visually, a dolphin still looks much more like a fish than a mammal. The visual features are not able to capture genetic similarities - see Figure 2.2 for a convincing example. To give another example, the horse is as part of odd-toed ungulates in a group with cats and dogs while the the look of a horse itself as well as the background appearance of horses, meadows, might be more similar to those of even-toed ungulates as cows and sheep.

The fact that the taxonomies of the Caltech animals are not well aligned to kernels similarities can be validated numerically by computing the cosine angle between the distances induced from the kernel matrices and the taxonomy distance for each of the dataset. The kernel distance between two classes is computed as the mean over the kernel distances for all pairs of samples from both classes using the additional fact that for χ^2-kernels we have k(x,x)=1:

$$d(c_1, c_2) = \frac{1}{|c_1|} \sum_{x_1 \in c_1} \frac{1}{|c_2|} \sum_{x_2 \in c_2} k(x_1, x_1) - 2k(x_1, x_2) + k(x_2, x_2) \tag{2.19}$$

$$= 2 - 2\frac{1}{|c_1|} \sum_{x_1 \in c_1} \frac{1}{|c_2|} \sum_{x_2 \in c_2} k(x_1, x_2) \tag{2.20}$$

From both distance matrices the mean is subtracted so that they have zero mean over their entries. We can see from table 2.15 that both Caltech datasets have a very low alignment between kernel induced distances and taxonomy-induced distances. This may explain the observed increase in flat zero-one-loss when applying taxonomy learning.

Table 2.15: Cosine Angles between taxonomy distances and kernel induced distances. Higher values are better.

Dataset	cosine of angles
Caltech256 52 animals	0.1130
Caltech256, 13 class subset	0.1087
VOC2006 multi-class	0.6314
Synthetic data, $\sigma = 1/4$, 16 classes (Sec. 2.3)	0.9752

The ordering of cosine angles across datasets corresponds well to the order of AUC scores at intermediate edges in Table 2.13. In the Pascal VOC2006 dataset and the synthetic dataset the distances from kernel similarities are more in line with the taxonomic ones. In the synthetic dataset this has been achieved by construction which is also reflected in Table 2.13 and in the KTA ratios from table 2.14.

We have identified the reason for the gap between our expectation for a more human-like classification using taxonomies and the case observed experimentally. The positive message from our experiments is the observation that even in the adversarial case of the low alignment between taxonomy and visual similarities as seen in Caltech animals data, the taxonomic losses can be improved while in the other two more well-behaved cases both losses, taxonomic and flat, can be improved.

A solution for improvement towards more human-like classification is to consider a richer feature representation which allows for a better alignment of the kernel-induced distances to the distances from the taxonomy because a richer feature representation can be used to select for each classifier its own more appropriate subset of features. In this study we used the same kernel for each classifier. Using a better feature set may include features which are not restricted purely visual ones in order to incorporate knowledge from biological systematics which cannot be captured by visual similarities alone. When humans reason about similarities between

known animal species, they use additionally more information than merely visual cues, e.g. they group animals by being insect, mammal or fish.

2.5 Ranking for Multi-label Datasets with hierarchies

Clearly the local SVM approach can also be used in a multi-label setting. In a multi-label setting each concept can be present or absent in each image independent of all other concepts. In particular, each image may contain multiple concepts and, as a consequence, confusions between concepts within an image are not well defined anymore. Therefore the target function evaluated here differs from the multi-class case.

Instead of minimizing confusions between concepts, we aim to enforce for each concept separately an ordering of images such that images of the concept in question and taxonomically close concepts are ranked highest. For this reason we introduce a novel taxonomy-aware ranking score, the ATax score.

2.5.1 The ATax score

Technically we will replace scores based on confusion matrices by threshold-independent ranking scores. A standard flat score function used in the Pascal VOC challenge is the Average Precision (AP) (103) and its mean over all classes. We assume that the pairs of SVM outputs and ground truth labels $(z^{(c)}, y^{(c)})$ for a class in question c are sorted according to the descending order of their output scores $z_k^{(c)}$ over the data sample index k. The average precision (AP) score for $n_+^{(c)} = \sum_{i=1}^{n} \mathbb{I}\{y_i^{(c)} = 1\}$ positively labeled samples of class c is defined as

$$AP^{(c)}((z_k^{(c)}, y_k^{(c)})_{k=1}^{n}) := \frac{1}{n_+^{(c)}} \sum_{i=1}^{n} \mathbb{I}\{y_i^{(c)} = 1\} \frac{1}{i} \sum_{k=1}^{i} \mathbb{I}\{y_k^{(c)} = 1\} \tag{2.21}$$

The AP score is maximized when the images of the class in question c are ranked first. It is invariant against permutation of the ordering of images from all other classes as long as the ranks of images from the class in question c are untouched. However, given relations from a taxonomy, we would prefer a ranking where images from taxonomically close classes are ranked in front of images from taxonomically far classes, even when they do not belong to the class in question c. To incorporate this awareness about the taxonomical structure we will introduce a novel score and call it the Atax score.

For deriving the structure of the Atax score we need two preliminaries.

The first preliminary is the fact that we need to consider for each image the set of all labels. For the ATax score being a taxonomy-aware extension of the AP score we consider instead of one single binary label $y_k^{(c)}$ for the class in question c the set of labels based on *all classes in the multi-label problem* $\{y_k^{(r)} \in \{0,1\}, r \in \{1,\ldots,C\}\}$. $y_k^{(r)}$ is the label for data sample k and class r.

The second preliminary is an representation of the AP score as an average of top-rank-list precisions derived from distance functions over a set of samples.

Let us define for a $[0,1]$-bounded distance function $l(y)$ the top-rank-list precision of the top ranked i samples $Prec[l](i)$ to be

$$Prec[l](i) = \frac{1}{i}\sum_{k=1}^{i} 1 - l(y_k) \tag{2.22}$$

Then average precision can be seen as an average of top-rank-list precisions over a particular set S of samples:

$$AP^{(c)} = \frac{1}{|S|}\sum_{i\in S} Prec[l_{01}^{(c)}](i) \tag{2.23}$$

where the set of samples S is given in according to Equation (2.21) as $S = \{i \mid \mathbb{I}\{y_i^{(c)} = 1\}\}$ and

$$l_{01}^{(c)}(y_k) = \mathbb{I}\{y_k^{(c)} \neq 1\} \tag{2.24}$$

is the zero-one discretized distance of the class label $y^{(c)} \in \{-1,+1\}$ to the label value 1.

This representation holds because of

$$\begin{aligned}
\frac{1}{|S|}\sum_{i\in S} Prec[l_{01}^{(c)}](i) &= \frac{1}{|S|}\sum_{i\in S}\frac{1}{i}\sum_{k=1}^{i} 1 - l_{01}^{(c)}(y_k) \\
&= \frac{1}{|S|}\sum_{i\in S}\frac{1}{i}\sum_{k=1}^{i} 1 - \mathbb{I}\{y_k^{(c)} \neq 1\} \\
&= \frac{1}{|S|}\sum_{i\in S}\frac{1}{i}\sum_{k=1}^{i} \mathbb{I}\{y_k^{(c)} = 1\} \\
&= \frac{1}{n_+(c)}\sum_{i\in\{m\mid\mathbb{I}\{y_m^{(c)}=1\}\}}\frac{1}{i}\sum_{k=1}^{i}\mathbb{I}\{y_k^{(c)} = 1\} \\
&= \frac{1}{n_+(c)}\sum_{i=1}^{n}\mathbb{I}\{y_i^{(c)} = 1\}\frac{1}{i}\sum_{k=1}^{i}\mathbb{I}\{y_k^{(c)} = 1\} \\
&= AP^{(c)}((z_k^{(c)}, y_k^{(c)})_{k=1}^{n}) \text{ see Equation (2.21).}
\end{aligned} \tag{2.25}$$

We compute a ranking score for a fixed class in question c of the multi-label problem. Therefore note that we can replace in the original AP score hierarchy-unaware precision score $l_{01}^{(c)}$ by a term *dependent* on the a priori fixed class c. The Atax score will be defined by a replacement term given in equation (2.26) based on the minimal taxonomy distance δ_T between the fixed class c and all positive labels in the ground truth $\{y_k^{(r)}, r \in \{1,\ldots,C\} \mid y_k^{(r)} = 1\}$ of a fixed sample k:

$$l_T^{(c)}(\{y_k^{(r)}, r = 1,\ldots,C\}) = \min_{r\in\{1,\ldots,C\}|y_k^{(r)}=1} \delta(c,r)) \tag{2.26}$$

Again, assume that the data samples $(x_k, \{y_k^{(r)}, r = 1,\ldots,C\})$, and thus their labels $y_k^{(r)}$ for all classes r, are sorted according to the descending order of the SVM outputs $z_k^{(c)}$ for the fixed class c. The set of samples S is given again as $S = \{i \mid \mathbb{I}\{y_i^{(c)} = 1\}\}$.

Then we define the ATax score for class c to be:

$$ATax^{(c)} = \frac{1}{|S|}\sum_{i\in S} Prec[l_T^{(c)}](i) \tag{2.27}$$

$$= \frac{1}{n_+^{(c)}}\sum_{i=1}^{n}\mathbb{I}\{y_i^{(c)} = 1\}\frac{1}{i}\sum_{k=1}^{i} 1 - \min_{r\in\{1,\ldots,C\}|y_k^{(r)}=1} \delta_T(c,r) \tag{2.28}$$

The above derivation shows that the ATax score can be seen as a taxonomy-aware extension of the established AP score. Since the taxonomy distance δ_T from equation (2.2) is scaled to lie in $[0,1]$ and a correct prediction implies scores of $\mathbb{I}\{y_k^{(c)} = 1\} = 1$ respectively $1-l_T^{(c)}(y_k) = 1$, the ATax score is never smaller than the AP score. The precision function used in the AP score can be interpreted as a zero-one discretization of the taxonomy score $1 - l_T^{(c)}(y_k)$. Both scores, AP and ATax, have the advantage of being invariant against the classification threshold and evaluate the ranking of images. We did not use the ranking based scores for the multi-class problem, however. Inspecting the constraints of the structured prediction formulation from (2.8) shows that it aims at classifying each image correctly in the sense of obtaining a correct *ranking of classes* for each image. Its optimization does not aim at obtaining a correct *ranking of images* for each class. Thus, using a ranking score would be a biased measure against the structured approaches.

2.5.2 Datasets

VOC2006 multi-label data

We use the VOC2006 dataset (84) consisting of 10 object classes and 5301 images with its original, unmodified labels. The full taxonomy is given in Figure 2.3.

VOC2009 multi-label classification task data

This dataset consists of 20 classes with 7054 labeled images. It serves as a second multi-label setting for the local algorithms. The full taxonomy is given in Figure 2.13.

2.5.3 Experimental Results

Note that for multi-label data the structured algorithms cannot be applied in their current form as the multi-class constraints are not well-defined anymore. Therefore we will compare one-versus-all classification against local hierarchical approaches. As this frees us of time and memory consumption problems related to the structured algorithms we will use crossvalidation with 20 folds. We will use the same features and kernels as described in sections 2.4.2 and 2.4.3 and measure with AP and ATax scores.

Table 2.16: Ranking scores on VOC06 as multi-label problem, 20-fold crossvalidation. Higher scores are better.

Method	ATax	AP
one versus all	90.10 ± 3.46	80.13 ± 7.21
local tax. scaled geometric mean	$\mathbf{91.29} \pm 3.34$	79.96 ± 7.23
local tax. scaled, harmonic mean	90.85 ± 3.28	$\mathbf{80.61} \pm 7.06$

Table 2.17: Ranking scores on VOC09 as multi-label problem, 20-fold crossvalidation. Higher scores are better.

Method	ATax	AP
one versus all	79.02 ± 8.72	55.92 ± 15.91
local tax. scaled geometric mean	$\mathbf{80.68} \pm 8.20$	54.62 ± 16.08
local tax. scaled, harmonic mean	80.03 ± 8.33	$\mathbf{56.43} \pm 15.77$

Tables 2.16 and 2.17 show that even for a multi-label setting, introducing a taxonomy can improve taxonomy based as well as flat ranking scores, despite we have no notion of avoiding confusions anymore.

This may become relevant when using classifier scores for ranking images for retrieval. A higher ATax score implies that the desired class and similar classes are ranked higher than more distant classes which in effect leads to a subjectively improved ranking result from a human viewpoint. When looking for cats, humans tend to be more impressed by results which return erroneously other pets than cars. Highly ranked images from very distant categories tend to be perceived as strong outliers.

Figure 2.11 shows examples where the hierarchical classifier is able to improve rankings simultaneously for classes which are far apart in the taxonomy given in Figure 2.3. This shows that taxonomy learning for multi-label problems does not lead necessarily to mutual exclusion of taxonomy branches. In both images, the classes under consideration are separated already at the top level. We observe that images can be re-ranked to top positions despite average rankings at all edges. For the upper image this occurs for the cow class, for the lower image this occurs for the motorbike class as can be seen from the rankings given along the paths. This can be explained by the property of the nonpositive p-means to be upper-bounded by the smallest score (see Section 2.2.5). Many images which achieved higher scores and ranks at some edges along the considered path were effectively ranked lower because they received very low scores at one edge at least in the same path. Note that the observed improvement in ranking is independent of the ranking loss.

Table 2.18 compares the performance of scaled versus unscaled combinations of scores for both multi-label problems. We see clearly that scaling of scores onto a compact interval contributes to the good performance of the local models. The good performance of scaled scores is not surprising as one can expect the SVM outputs to have different distribution statistics like variances across the edges. Please note that for one versus all classification the scaling has no influence on the ranking scores as it is monotonous and rank-preserving and the score computation is done for each class separately.

2.6 Conclusions

When classifying complex data such as objects, humans are first of all much better than learning machines and most importantly human and machine errors diverge considerably. Among

Table 2.18: Scaling of outputs is important for multi-label problems, 20 fold crossvalidation. Higher AP and ATax scores are better.

Method: local tax. arith. mean	ATax	AP
VOC06, *unscaled*	84.59 ± 6.73	60.31 ± 15.08
VOC06, *scaled*	89.58 ± 3.89	74.85 ± 8.51
VOC09, *unscaled*	73.35 ± 9.40	35.87 ± 14.73
VOC09, *scaled*	77.30 ± 9.45	46.58 ± 16.61

others, a reason for both findings is the impressing ability of humans to generate abstract representations that implicitly organize hierarchical knowledge and thus to create appropriate task relevant factorizations of the environment, put in one word humans generalize. One aspect of such abstract representation can be captured by taxonomies.

In this chapter we have demonstrated that taxonomy-based learning using structured SVMs and local-SVM-based approaches on real world data yields improved results when measured with taxonomy-based losses. Local algorithms with generalized means voting perform on par to structured models while being considerably faster in training. The geometric mean appears to be a good a priori choice as a sensitivity tradeoff against small and large outliers. Successful minimization of taxonomy losses implies the reduction of confusions between distant categories, i.e. a step towards more human-like decision making. Note, however, that an improved result measured with taxonomy-based losses does not necessarily translate into a better result in a flat loss such as 0/1-loss since more meaningful confusions, i.e. improved quality of decision making does not necessarily come with overall quantitative improvements as other more meaningful confusions may come in addition – as a side effect. In the local SVM framework this can be checked by the AUC scores on the internal edges compared to the leaf edges.

Experiments on synthetic data show, somewhat expectedly, that taxonomy based algorithms work better than the taxonomy-free baseline, when the data is aligned to the taxonomy. They suggest that performance gains are achieved for local procedures by combining classifiers with different trade-offs of false positive versus false negative rates. Interestingly but in fact to be expected, taxonomy based learners tend to make their errors rather close to the leaf-edges of the taxonomy tree thereby confusing 'close' categories, whereas learners based on flat losses incur classification errors uniformly across the tree. The latter behavior is one of the reasons

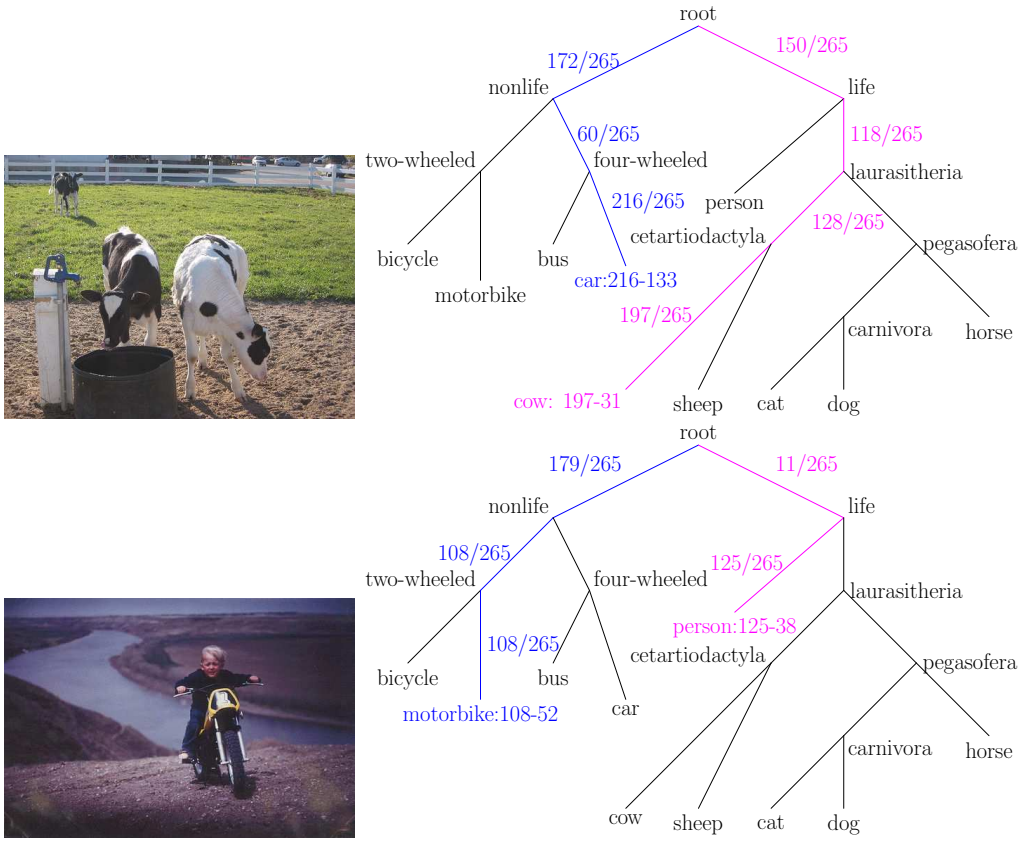


Figure 2.11: Example images where the hierarchical classifier improves rankings for taxonomically distant classes compared the one versus all baseline on VOC2006 multi-label problem. (Upper) car from 216 to 133, cow from 197 to 31. (Lower) motorbike from 108 to 52, person from 125 to 38.

to consider the decisions of taxonomy-based learning machines more human-compatible than their flat loss training based counterparts.

The local as well as structured approaches can be combined with methods which learn taxonomies. The difference to previous approaches would be to measure taxonomy based errors instead of flat losses and to rely in case of local algorithms on vote fusion instead of reduced kernels and greedy path-walks. It is open in such a case how much can be retained of the interpretation of a taxonomy as a weak prior knowledge to define loss functions which penalize dissimilarities as they are perceived by humans.

With respect to learning hierarchies an image might be scored using multiple paths leading from the root to the same visual concept in the local setup. This is related to approaches

learning relaxed hierarchies (85, 90). The idea would be to fix an original hierarchical loss function and its generating hierarchy and check whether learning a different hierarchy (or directed acyclic graph structure) than the original one may improve the original hierarchical loss because the learned hierarchy can encode information about the similarity between image features and thus help to bridge the gap between the similarity between image features which is used for learning classifiers and the similarity encoded in the original hierarchy which is used for evaluation of classifiers. One simple example would be to suppress nodes with associated edges when the classifiers on these edges yield very high error rates.

Another option would be to design local algorithms for the optimization of losses using weighted edges or more general losses. In the structured prediction setup losses using weighted edges can be achieved straightforwardly by weighting $\kappa_i(y) \rightarrow \lambda_i \kappa_i(y)$ in equation (2.5) as shown in Section 2.2.3. Such weights can be even learned via Multiple Kernel Learning on the label kernel from equation 2.3 in which the original label kernel $\mathrm{K}_Y(y, \hat{y}) = \sum_{j=1}^{|V|} \kappa_j(y)\kappa_j(\hat{y})$ from equation 2.3 is replaced by a parametrized variant

$$\mathrm{K}_Y(y, \hat{y})[\boldsymbol{\lambda}] = \sum_{j=1}^{|V|} \lambda_j K_{Y,j}(y, \hat{y}) \tag{2.29}$$

$$K_{Y,j}(y, \hat{y}) = \kappa_j(y)\kappa_j(\hat{y}) \,. \tag{2.30}$$

The difference to the learning of a taxonomy is that the taxonomy and the loss used for evaluation is fixed here. The motivation to do so is the same as for learning a hierarchy, namely to bridge the gap between the similarity between image features which is used for learning classifiers and the similarity encoded in the original hierarchy and its loss function.

In the local setup such learning might be analogously achieved by learning weights in vote fusion as a replacement for the p-means based vote from Section 2.2.5 such as to minimize a regularized weighted loss between prediction and labels. Based on our experience with overfitting of support vector machines on training data at settings where performance on test data is near-optimal (see also Chapter 3) such scores would have to be learned on cross-validated outputs in difference to (78). One meaningful application of weighted edges is to weight each path by the binary power 2^{-d} of its negative depth d in the hierarchy as described in Section 2.2.3. This ensures a strict hierarchy – errors made at higher levels in the hierarchy always count more than errors at lower levels.

Multiple kernel learning (see Chapter 3) or other techniques to fuse information from multiple features can be employed to learn a mixture of feature kernels depending on the position in the edge.

A further direction is to compare the local-SVM procedures versus taxonomy-free multi-task learning approaches on multi-label problems. In these problems we are interested to rank the set of images for each class which demands for threshold-invariant measures like the average precision scores for comparison or the Atax score. Our simulation study on VOC 2006 and 2009 shows encouraging results. In the meantime multi-label structured prediction has been developed in (91). Yet the reported performance results for hierarchical classification were not better than the one versus all baseline which leaves space for improvement.

An open question is the relation between research on attribute classification and hierarchical classification. Clearly the works on attribute-based classification known to the author (104, 105, 106) aim at minimizing flat losses and use additional labels, namely the attribute labels, while the hierarchy approaches work without additional concept labels. Another difference to the visual concepts defined by edges in a hierarchy is that the presence of attributes may vary within a visual concept class (104) which results in a higher flexibility of attributes. Mathematically attribute prediction itself is the same as visual concept prediction. Semantically, however, the attributes are designed to correspond to image content which can be shared among visual object classes (104). Attributes share with internal edges in a hierarchy the fact that they define a new visual concept and use the new visual concepts for aiding to infer the original concepts labels. Learning the weights for attributes as in (105) improves flat losses which makes it interesting.

One direction with respect to practical aspects of hierarchical classification of any kind would be to incorporate early stopping when the decision to descend further along a tree or directed acyclic graph structure becomes statistically uncertain. This could reduce error rates and improve similarity of decisions to human ones. Humans also tend to stop classifying objects at a level of certainty. All humans are able to identify that a cat is a indeed cat easily, however people unfamiliar with those furballs would reject to predict the precise cat breed unless explicitly asked to do so. In that sense humans perform early stopping in the absence of sufficient knowledge. A statistical prediction system can do the same, and avoid to make predictions if the classifier prediction for a sample is unreliable. One easy way would be to determine thresholds for each path in the hierarchy such that classifying images exceeding the

lower or upper threshold yields a fixed accuracy. The threshold can be estimated by cross-validation for example. This could also serve as a way to measure the quality of a classifier. A too poor quality of a classifier in the sense that almost no image can be reliably classified by it because the thresholds are too high could be used as an indicator to remove this path from the hierarchy.

An overall challenge of the field would be to further the generic understanding of the different decision making between human and learning machine, ultimately combining low level machine precision, attribute based features and human abstraction optimally towards a truly cognitive automated decision making machinery.

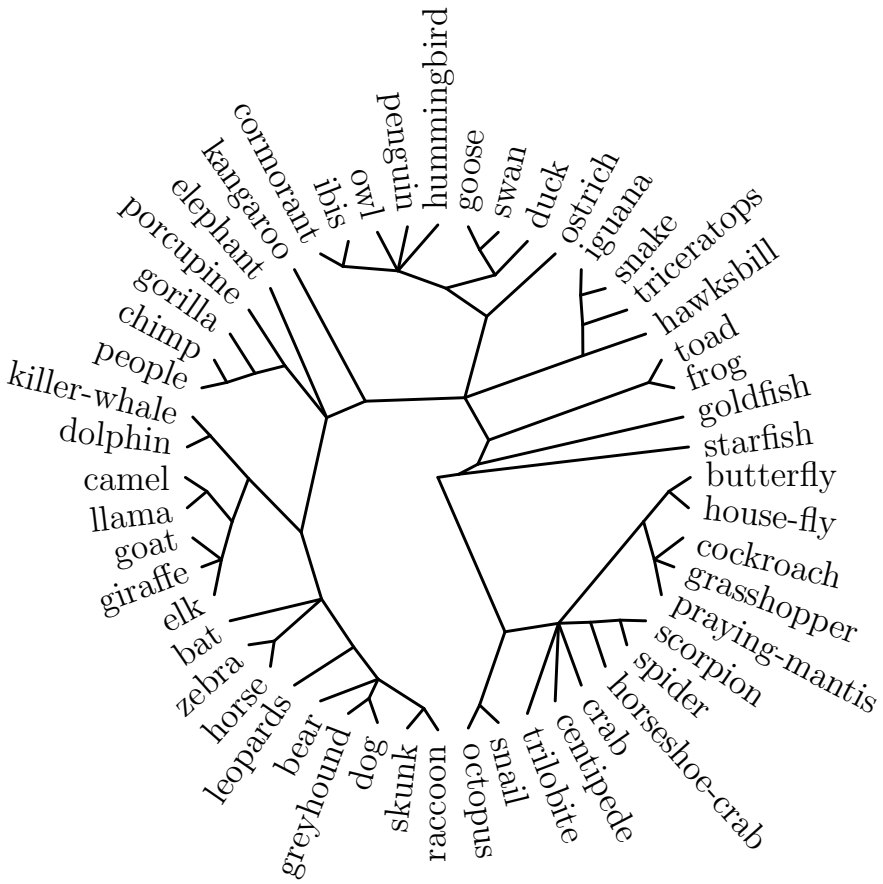


Figure 2.12: Taxonomy on 52 Animals Classes from Caltech256, the 13 class subset taxonomy is contained in the lower left quadrant from octopus to butterfly.

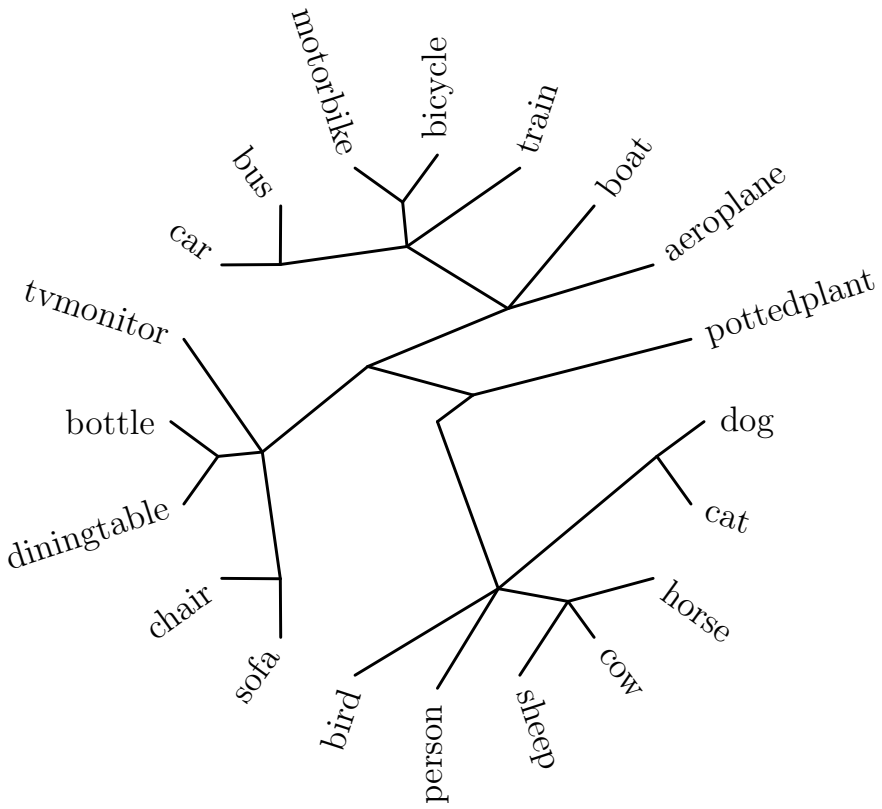


Figure 2.13: Taxonomy on 20 Classes from Pascal VOC2009.

3

Insights from Classifying Visual Concepts with Multiple Kernel Learning

3.1 Motivation for this aspect of Semantic Concept Recognition in Images

Given a set of mercer kernels for image data the problem considered here is to learn a linear combination of these kernels for use with semantic concept ranking with support vector machines.

It is a common strategy in visual object recognition tasks to combine different image representations to capture relevant traits of an image. This results in a set of features for each image as opposed to classifying an image using a single feature. Prominent representations are for instance built from color, texture, and shape information and used to accurately locate and classify the objects of interest. The importance of such image features changes across the tasks. For example, color information may increase the detection rates of stop signs in images substantially but it is almost useless for finding cars. This is because stop sign are usually red in most countries but cars in principle can have any color. As additional but nonessential features not only slow down the computation time but may even harm predictive performance, it is necessary to combine only relevant features for state-of-the-art object recognition systems.

This work is inspired by two factors: firstly, typically many kernels are computed for state of the art submissions to renowned competitions such as ImageCLEF PhotoAnnotation (1) and

Pascal VOC Classification (11). Secondly, many of these submissions do not employ methods to learn kernel combinations. For a person with a background in kernel-based machine learning this leaves the pressing question why methods to learn kernel combinations are not employed in practical settings. Anecdotally it is known that the common sparse ℓ_1-norm multiple kernel learning does not perform well in many settings outside datasets with subjectively low within-class variance like Caltech101 (2) and Oxford Flowers (107). On other datasets it is reported anecdotally to select a very sparse set of kernels with a decrease in the performance which indicates overfitting.

3.1.1 Contributions

The contributions of this chapter are[1]

- We apply a recently developed non-sparse multiple kernel learning (MKL) variant to state-of-the-art concept recognition tasks within computer vision.
- We report empirical results for the PASCAL VOC 2009 Classification and ImageCLEF2010 Photo Annotation challenge data sets.
- We provide insights on benefits and limits of non-sparse MKL and compare it against its direct competitors within the family of algorithms which are based on support vector machines, the sum kernel SVM and the sparse MKL. To this end we identify two limiting factors and one promoting factor for the usage of MKL algorithms over the natural baseline represented by SVMs applied to uniform kernel mixtures in image annotation and ranking tasks. We provide experimental evidence for these factors.
- We introduce a novel measure for the analysis of the diversity of classifiers for the explanation of one of these factors.

This chapter is organized as follows. Section 3.1.2 gives an overview of multiple kernel learning and related algorithms in image annotation tasks. In Section 3.2, we briefly review the machine learning techniques used here; The following section 3.3 we present our experimental results on the VOC2009 and ImageCLEF2010 datasets; in Section 3.4 we discuss promoting and limiting factors of MKL and the sum-kernel SVM in three learning scenarios. We perform experiments in Section 3.4 in order to provide evidence for these factors.

[1]The content of this chapter is based on the author's own peer-reviewed work (63).

3.1.2 Related Work

In the last decades, support vector machines (SVM) (3, 108) have been successfully applied widely to practical problems of image annotation (51). Support vector machines exploit similarities of the data, arising from some (possibly nonlinear) measure. The matrix of pairwise similarities, also known as kernel matrix, allows to abstract the data from the learning algorithm (4).

In image annotation and ranking, translating information from various features into a set of several kernels has now become a standard technique (23). Consequently, the choice of finding the right kernel changes to finding an appropriate way of fusing the kernel information; however, finding the right combination for a particular application is so far often a matter of a judicious choice (or trial and error).

In the absence of principled approaches, practitioners frequently resort to heuristics such as uniform mixtures of normalized kernels (36, 50, 98) that have proven to work well. Nevertheless, this may lead to sub-optimal kernel mixtures.

An alternative approach is multiple kernel learning (MKL), which has been applied to object classification tasks involving various image features (101, 109). Multiple kernel learning (110, 111, 112, 113) generalizes the support-vector-machine framework and aims at *simultaneously* learning the optimal kernel mixture *and* the model parameters of the SVM. To obtain a well-defined optimization problem, many MKL approaches promote sparse mixtures by incorporating a 1-norm constraint on the mixing coefficients. Compared to heuristic approaches, MKL has the appealing property of automatically selecting kernels in a mathematical sound way and converges quickly as it can be wrapped around a regular support vector machine (112). However, some evidence shows that sparse kernel mixtures are often outperformed by an unweighted-sum kernel (114). As a remedy, (115, 116) propose ℓ_2-norm regularized MKL variants, which promote non-sparse kernel mixtures and subsequently have been extended to ℓ_p-norms (56, 117).

Multiple Kernel approaches have been applied to various computer vision problems outside our scope of multi-label ranking such multi-class problems (118), which require in distinction to the general multi-label case mutually exclusive labels[1] and object detection (119, 120) in the sense of finding object regions in an image. The latter reaches its limits when image concepts

[1]We make a distinction between the general case of multi-label classification and the more special case of multi-class classification with mutually exclusive classes.

cannot anymore be represented by an object region such as the *Outdoor*,*Overall Quality* or *Boring* concepts in the ImageCLEF2010 dataset that we will use.

The family of MKL algorithms is not restricted to SVM-based ones. Another competitor, for example, is Multiple Kernel Learning based on Kernel Discriminant Analysis (KDA) (121, 122). The difference between MKL-SVM and MKL-KDA lies in the underlying single kernel optimization criterion while the regularization over kernel weights is the same.

Fusing information from multiple features include algorithms relying on a significantly larger number of parameters, for example, (123), who use logistic regression as base criterion; their approach results in a number of optimization parameters equal to the number of samples times the number of input features. Since the approach in (123) a priori uses much more optimization variables, it poses a more challenging and potentially more time consuming optimization problem, which limits the number of applicable features.

Further alternatives use more general combinations of kernels such as products with kernel widths as weighting parameters (101, 124). As (124) point out, the corresponding optimization problems are no longer convex. Consequently, they may find suboptimal solutions and it is more difficult to assess using how much gain can be achieved by learning the kernel weights.

3.2 Methods

This section briefly introduces multiple kernel learning (MKL). For an extensive treatment see the surveys in (125, 126).

Multiple Kernel Learning

Given a finite number m of different kernels each of which implies the existence of a feature mapping $\psi_j : \mathcal{X} \to \mathcal{H}_j$ onto a Hilbert space

$$k_j(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \langle \psi_j(\boldsymbol{x}), \psi_j(\bar{\boldsymbol{x}}) \rangle_{\mathcal{H}_j}$$

the goal of multiple kernel learning is to learn SVM parameters $(\boldsymbol{\alpha}, b)$ and kernel weights $\{\beta_l, l = 1, \ldots, m\}$ for a linear combination of these m kernels $K = \sum_l \beta_l k_l$ simultaneously.

This can be cast as the following optimization problem which reduces to support vector

machines (3, 5) in the special case of on kernel $m = 1$

$$\begin{aligned} \min_{\boldsymbol{\beta},\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\sum_{j=1}^{m}\beta_j \boldsymbol{w}_j'\boldsymbol{w}_j + C\|\boldsymbol{\xi}\|_1 \\ \text{s.t.} \quad & \forall i:\ y_i\left(\sum_{j=1}^{m}\beta_j \boldsymbol{w}_j'\psi_j(\boldsymbol{x}_i) + b\right) \geq 1-\xi_i \\ & \boldsymbol{\xi}\geq \mathbf{0}\,; \quad \boldsymbol{\beta}\geq \mathbf{0}\,; \quad \|\boldsymbol{\beta}\|_p \leq 1. \end{aligned} \tag{3.1}$$

The explicit usage of kernel mixtures $\sum_l \beta_l k_l$ is permitted through its partially dualized form:

$$\begin{aligned} \min_{\boldsymbol{\beta}}\max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,l=1}^{n}\alpha_i\alpha_l y_i y_l \sum_{j=1}^{m}\beta_j k_j(\boldsymbol{x}_i,\boldsymbol{x}_l) \\ \text{s.t.} \quad & \forall_{i=1}^{n}:\ 0\leq\alpha_i\leq C; \quad \sum_{i=1}^{n} y_i\alpha_i = 0; \\ & \forall_{j=1}^{m}:\ \beta_j\geq 0; \quad \|\boldsymbol{\beta}\|_p \leq 1. \end{aligned} \tag{3.2}$$

For details on the solution of this optimization problem and its kernelization we refer to (56). This optimization problem has two parameters: the regularization constant C and a parameter p on the constraint for the kernel weights $\boldsymbol{\beta}$. The regularization constant is known from support vector machines; it balances the margin term $C\|\boldsymbol{\xi}\|_1$ from equation (3.1) over the regularization term $\sum_{j=1}^{m}\beta_j\boldsymbol{w}_j'\boldsymbol{w}_j$. A high value of the regularization constant C puts more emphasis on achieving high classification margins $y_i\left(\sum_{j=1}^{m}\boldsymbol{\beta}_j w_j'\psi_j(\boldsymbol{x}_i)+b\right)$ on the training data while a low value emphasizes the regularization term as a measure against overfitting on training data.

While prior work on MKL imposes a 1-norm constraint on the mixing coefficients to enforce sparse solutions lying on a standard simplex (54, 111, 112, 127), we employ a generalized ℓ_p-norm constraint $\|\boldsymbol{\beta}\|_p \leq 1$ for $p \geq 1$ as used in (56, 117). The implications of this modification in the context of image concept classification will be discussed throughout this chapter.

3.3 Empirical Evaluation

In this section, we evaluate ℓ_p-norm MKL in real-world image categorization tasks, experimenting on the VOC2009 and ImageCLEF2010 data sets. We also provide insights on *when* and *why* ℓ_p-norm MKL can help performance in image classification applications. The evaluation measure for both datasets is the average precision (AP) over all recall values based on the precision-recall (PR) curves.

3.3.1 Data Sets

We experiment on the following data sets:

1. PASCAL2 VOC Challenge 2009 (Multi-label data) The first dataset is the official data set of the *PASCAL2 Visual Object Classes Challenge 2009* (VOC2009) (64), which consists of 13979 images. We use the official split into 3473 training, 3581 validation, and 6925 test examples provided by the challenge organizers. The organizers also provided annotation for 20 object categories; It is a multi-label dataset, i.e. an image may be labeled with multiple classes. The task is to solve 20 binary classification problems, i.e. predicting whether at least one object from a class k is visible in the test image. Although the test labels are undisclosed, the more recent VOC datasets permit to evaluate AP scores on the test set via the challenge website (the number of allowed submissions per week being limited).

2. ImageCLEF 2010 PhotoAnnotation (Multi-label data) The ImageCLEF2010 PhotoAnnotation data set (128) consists of 8000 labeled training images taken from flickr and a test set with recently disclosed labels. The images are annotated by 93 concept classes having highly variable concepts—they contain both well defined objects such as *lake, river, plants, trees, flowers*, as well as many rather ambiguously defined concepts such as *winter, boring, architecture, macro, artificial, motion blur,*—however, those concepts might not always be connected to objects present in an image or captured by a bounding box. This makes it highly challenging for any recognition system. As for VOC2009 we decompose the problem into 93 binary classification problems. Again, many concept classes are challenging to rank or classify by an object detection approach due to their inherent non-object nature. As for the previous dataset each image can be labeled with multiple concepts.

3.3.2 Image Features and Base Kernels

In all of our experiments we deploy 32 kernels capturing various aspects of the images. Our choice of features is inspired by the VOC 2007 winner (66) and our own experiences from our submissions to the VOC2009 and ImageCLEF2009 challenges. It is known from the top-ranked submissions in recent Pascal VOC Classification and ImageCLEF PhotoAnnotation Challenges that Bag-of-Words features are necessary for state-of-the-art performance results when the focus lies on visual concept classification and ranking. At the same time adding simpler features together with multiple kernel learning may improve the ranking performance for some visual concepts as well as the average performance measured over all visual concepts (shown in (73)).

For the ImageCLEF2010 dataset the test data annotations have been disclosed and we checked that adding the simpler features listed below improves, indeed, the average-kernel performance compared to relying on BoW-S features (see next section) alone. Our choice of features was furthermore guided by the intention to have several different feature types that empirically have been proven to be useful and to use gradient and color information. Furthermore the features should have reasonable computation times without the need for excessive tuning of many parameters and they should be able to capture objects and visual concept cues of varying sizes and positions. For this reason, we used bag of word features and global histograms based on color and gradient information.

All these features were computed over sets of color channels as inspired by (23). The features obtained for each color channel of one set were concatenated to yield one feature for each color channel set. The color channel sets used here are

- red, green, and blue (RGB)
- grey (equation (1.5))
- grey (equation (1.5)), opponent color 1 (equation (1.6)) and opponent color 2 (OPP) (equation (1.7))
- normalized RGB (nRGB)(equation (1.8))
- normalized opponent colors (nOPP) (equation (1.9))

The features used in the following are derived from histograms that a priori contain *no spatial information*. We therefore enrich the respective representations by using regular spatial tilings 1×1, 3×1, 2×2, 4×4, 8×8, which correspond to single levels of the pyramidal approach in (36, 97). Furthermore, we apply a exponential χ^2 kernel (equation (1.26)) on top of the enriched histogram features, which has proven effective for histogram features (50, 51). The bandwidth σ of the χ^2 kernel in (1.26) is thereby heuristically chosen as the mean χ^2 distance (equation (1.27)) over all pairs of training examples, as done, for example, in (52).

Histogram over a bag of visual words over SIFT features (BoW-S)

Histograms over a bag of visual words over SIFT features are known to yield excellent performance for visual concept recognition both when used as single features alone as well as in combination with other features. This can be observed by checking the top-ranked submissions

in the recent ImageCLEF PhotoAnnotation and Pascal VOC Classification challenges and noting their general usage in publications on visual concept ranking. It has also recently been successfully deployed to object detection (46) on a large data set of images within the ImageNet Large Scale Visual Recognition Challenge. For an introduction on bag of word features the reader is referred to Section 1.3.1.

The BoW features (10) were constructed with parameters that were established in past image annotation challenges so as to yield good results. At first, the SIFT features (16) were calculated on a regular grid with six pixel pitch for each image. We computed the SIFT features over the following color channel sets: RGB, nRGB, OPP, and nOPP; in addition, we also use a simple gray channel. For visual words we used a code book of size 4000 obtained by k-means clustering (with a random initialization of centers and using 600000 local features taken randomly from the training set). Finally, all SIFT features were assigned to the visual words (so-called *prototypes*) by hard mapping as in equation (1.10) and then summarized into histograms within entire images or sub-regions. The BoW feature was normalized to an ℓ_1-norm of 1. Note that five color channel sets times three spatial tilings 1×1, 2×2 and 3×1 yield 15 features in total.

Histogram over a bag of visual words over color intensity histograms (BoW-C)

This feature has been computed in a similar manner as the BoW-S feature. However, for the local feature, we employed low-dimensional color histograms instead of SIFT features, which combines the established BoW computation principle of aggregating local features into a global feature with color intensity information – this was our motivation for employing them. The color histograms were calculated on a regular grid with nine pixel pitch for each image over a descriptor support of radius 12 and histogram dimension 15 per color channel (SIFT: 128). We computed the color histograms over the following color combinations: RGB, OPP, gray only and, finally, the hue weighted by the grey value in the pixels. For the latter the weighting implies that the hue receives a higher weight in bright pixels as a countermeasure against the known difficulties to estimate the hue in dark regions of an image.

For visual words we used a code book of size 900 obtained by k-means clustering. The lower dimensionality in local features and visual words yielded a much faster computation compared to the BoW-S feature. Otherwise we used the same settings as for BoW-S. Four color channel sets times two spatial tilings 1×1 and 3×1 resulted in 8 BoW-C features in total.

Histogram of oriented gradients (HoG)

The histogram of oriented gradients has proven to be useful (97) on the seminal Caltech101 Dataset (2). It serves as an alternative and much faster way to incorporate gradient information compared to the BoW-S features. The HoG feature is based on discretizing the orientation of the gradient vector at each pixel into bins and then summarizing the discretized orientations into histograms within image regions (97, 129). Canny detectors (130) are used to discard contributions from pixels, around which the image is almost uniform. We computed HoG features over the following color channel sets: RGB, OPP and gray only, every time using 24 histogram bins for gradient orientations for each color channel and spatial tilings 4×4 and 8×8.

In the experiments we deploy four kernels: a product kernel created from the two kernels with different spatial tilings using the RGB color channel set, a product kernel created from the two kernels having the color channel set OPP, and the two kernels using the gray channel alone (differing in their spatial tiling). Note that building a product kernel out of χ^2 kernels boils down to concatenating feature blocks (but using a separate kernel width for each feature block).

This choice allows to employ gradient information for a specific color channel set – independent of spatial resolution – via the first two kernels and for a specific spatial resolution (independent of color channels) via the last two kernels. This is a principled way to yield diverse features: one subset varies over color channel sets and the other over spatial tilings. In total we have four HoG features.

Histogram of pixel color intensities (HoC)

The histogram of color intensities is known to be able to improve ranking performance of BoW-S features as shown in (73), which motivated us to use it here. The HoC features were constructed by discretizing pixel-wise color values and computing their bin histograms within image regions. We computed HoC features over the following color channel combinations: RGB, OPP and gray only, every time using 15 histogram bins for color intensities for each color channel and spatial tilings 3×1, 2×2 and 4×4.

In the experiments we deploy five kernels: a product kernel created from the three kernels with different spatial tilings with color channel set RGB, a product kernel created from the three kernels with color combination OPP, and the three kernels using the gray channel alone

(differing in their spatial tiling). Again, please note the relation between feature concatenation and taking the product of χ^2-kernels. The last three kernels are HoC features from the gray channel and the two spatial tilings. This choice allows to employ color information for a specific color channel set independent of spatial resolution via the first two kernels and for a specific spatial resolution independent of color channels via the last two kernels. In total we have five HoC features.

For the HoG and HoC feature we used higher spatial tilings because these two features are much faster to compute compared to BoW features, thus allowing to increase their dimensionality by the spatial tilings, and due to our empirical experience that choices of finer spatial tilings beyond 2×2 tend to yield a higher improvement for such simpler features as compared to BoW-based features.

Summary of used features

We can summarize the employed kernels by the following types of basic features:

- Histogram over a bag of visual words over SIFT features (BoW-S), 15 kernels
- Histogram over a bag of visual words over color intensity histograms (BoW-C), 8 kernels
- Histogram of oriented gradients (HoG), 4 kernels
- Histogram of pixel color intensities (HoC), 5 kernels.

We used a higher fraction of bag-of-word-based features as we knew from our challenge submissions that they have a better performance than global histogram features. The intention was, however, to use a variety of different feature types that have been proven to be effective on the above datasets in the past—but at the same time obeying memory limitations of maximally ca. 25GB per job as required by computer facilities used in our experiments (we used a cluster of 23 nodes having in total 256 AMD64 CPUs and with memory limitations ranging in 32–96 GB RAM per node).

In practice, the normalization of kernels is as important for MKL as the normalization of features is for training regularized linear or single-kernel models. Optimal feature / kernel weights are requested to be small by the ℓ_p-norm constraint in the optimization problem given by equation (3.1), implying a bias to towards excessively up-scaled kernels. In general,

there are several ways of normalizing kernel functions. We apply the following normalization method, proposed in (54, 55) and entitled *multiplicative normalization* in (56);

$$K \longmapsto \frac{K}{\frac{1}{n}\mathrm{tr}(K) - \frac{1}{n^2}\mathbf{1}^\top K \mathbf{1}}. \tag{3.3}$$

The denominator is an estimator of the variance in the embedding Hilbert space computed over the given dataset D by replacing the expectation operator $\mathbb{E}[\cdot]$ by the discrete average over the data points $x_i \in D$.

$$\begin{aligned}\mathrm{Var}(\phi)_{\mathcal{H}} &= \mathbb{E}\left[\|\phi(X) - \mathbb{E}[\phi]\|^2_{\mathcal{H}}\right] \\ &= \mathbb{E}\langle \phi(X) - \mathbb{E}[\phi], \phi(X) - \mathbb{E}[\phi]\rangle_{\mathcal{H}} \approx_D \frac{1}{n}\mathrm{tr}(K) - \frac{1}{n^2}\mathbf{1}^\top K \mathbf{1}\end{aligned} \tag{3.4}$$

Thus dividing the kernel matrix $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle_{\mathcal{H}}$ by this term is equivalent to dividing each embedded feature $\phi(x)$ by its standard deviation over the data. This normalization corresponds to rescaling the data samples to unit variance in the Hilbert space used for SVM and MKL classification.

3.3.3 Experimental Setup

We treat the multi-label data set as binary classification problems, that is, for each object category we trained a one-vs.-rest classifier. Multiple labels per image render multi-class methods inapplicable as these require mutually exclusive labels for the images. The classifiers used here were trained using the open sourced Shogun toolbox *www.shogun-toolbox.org* (74). In order to shed light on the nature of the presented techniques from a statistical viewpoint, we first pooled all labeled data and then created 20 random cross-validation splits for VOC2009 and 12 splits for the larger dataset ImageCLEF2010.

For each of the 12 or 20 splits, the training images were used for learning the classifiers, while the SVM/MKL regularization parameter C and the norm parameter p were chosen based on the maximal AP score on the validation images. Thereby, the regularization constant C is optimized by class-wise grid search over $C \in \{10^i \,|\, i = -1, -0.5, 0, 0.5, 1\}$. Preliminary runs indicated that this way the optimal solutions are attained inside the grid. Note that for $p = \infty$ the ℓ_p-norm MKL boils down to a simple SVM using a uniform kernel combination (subsequently called sum-kernel SVM). In our experiments, we used the average kernel SVM instead of the sum-kernel one. This is no limitation in this as both lead to identical result for an appropriate choice of the SVM regularization parameter.

For a rigorous evaluation, we would have to construct a separate codebook for each cross validation split. However, creating codebooks and assigning features to visual words is a time-consuming process. Therefore, in our experiments we resort to the common practice of using a single codebook created from all training images contained in the official split. Although this could result in a slight overestimation of the AP scores, this affects all methods equally and does not favor any classification method more than another—our focus lies on a *relative* comparison of the different classification methods; therefore there is no loss in exploiting this computational shortcut.

3.3.4 Results

In this section we report on the empirical results achieved by ℓ_p-norm MKL in our visual object recognition experiments.

VOC 2009 Table 3.1 shows the AP scores attained on the official test split of the VOC2009 data set (scores obtained by evaluation via the challenge website). The class-wise optimal regularization constant has been selected by cross-validation-based model selection on the training data set. We can observe that non-sparse MKL outperforms the baselines ℓ_1-MKL and the sum-kernel SVM in this sound evaluation setup. We also report on the cross-validation performance achieved on the training data set (Table 3.2). Comparing the two results, one can observe a small overestimation for the cross-validation approach (for the reasons argued in Section 3.3.3)—however, the amount by which this happens is equal for all methods; in particular, the ranking of the compared methods (SVM versus ℓ_p-norm MKL for various values of p) is preserved for the average over all classes and most of the classes (exceptions are the bottle and bird class); this shows the reliability of the cross-validation-based evaluation method in practice. Note that the observed variance in the AP measure across concepts can be explained in part by the variations in the label distributions across concepts and cross-validation splits. Unlike for the AUC measure (96) which is also commonly used for the evaluation of rankings of classifier predictions, the average score of the AP measure under randomly ranked images depends on the ratio of positive and negative labeled samples.

A reason why the bottle class shows such a strong deviation towards sparse methods could be the varying but often small fraction of image area covered by bottles leading to overfitting when using spatial tilings.

We can also remark that $\ell_{1.333}$-norm achieves the best result of all compared methods on the VOC dataset, slightly followed by $\ell_{1.125}$-norm MKL. To evaluate the statistical significance of

Table 3.1: AP scores on VOC2009 test data with fixed ℓ_p-norm. Higher scores are better.

	average	aeroplane	bicycle	bird	boat	bottle	bus
ℓ_1	54.58	**81.13**	54.52	56.14	62.44	**28.10**	**68.92**
$\ell_{1.125}$	56.43	81.01	56.36	58.49	62.84	25.75	68.22
$\ell_{1.333}$	**56.70**	80.77	**56.79**	**58.88**	63.11	25.26	67.80
ℓ_2	56.34	80.41	56.34	58.72	**63.13**	24.55	67.70
ℓ_∞	55.85	79.80	55.68	58.32	62.76	24.23	67.79
	car	cat	chair	cow	diningtable	dog	horse
ℓ_1	52.33	55.50	52.22	36.17	45.84	41.90	61.90
$\ell_{1.125}$	55.71	57.79	53.66	40.77	**48.40**	46.36	**63.10**
$\ell_{1.333}$	**55.98**	**58.00**	**53.87**	**43.14**	48.17	46.54	63.08
ℓ_2	55.54	57.98	53.47	40.95	48.07	**46.59**	63.02
ℓ_∞	55.38	57.30	53.07	39.74	47.27	45.87	62.49
	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
ℓ_1	57.58	81.73	31.57	36.68	45.72	**80.52**	61.41
$\ell_{1.125}$	60.89	82.65	**34.61**	41.91	**46.59**	80.13	63.51
$\ell_{1.333}$	**61.28**	**82.72**	34.60	44.14	46.42	79.93	**63.60**
ℓ_2	60.91	82.52	33.40	**44.81**	45.98	79.53	63.26
ℓ_∞	60.55	82.20	32.76	44.15	45.69	79.03	63.00

AP scores were obtained on request from the challenge organizers due to undisclosed annotations. Regularization constants were selected via AP scores computed via cross-validation on the training set. Best methods are marked boldface.

our findings, we perform a Wilcoxon signed-rank test for the cross-validation-based results (see Table 3.2; significant results are marked in boldface). We find that in 15 out of the 20 classes the optimal result is achieved by truly non-sparse ℓ_p-norm MKL (which means $p \in]1, \infty[$), thus outperforming the baseline significantly.

ImageCLEF Table 3.3 shows the AP scores averaged over all classes achieved on the ImageCLEF2010 data set. We observe that the best result is achieved by the non-sparse ℓ_p-norm MKL algorithms with norm parameters $p = 1.125$ and $p = 1.333$. The detailed results for all 93 classes are shown in the appendix in Tables 5.4,5.5 and 5.6. We can see from the detailed results that in 37 out of the 93 classes the optimal result attained by non-sparse ℓ_p-norm MKL was significantly better than the sum kernel according to a Wilcoxon signed-rank test.

We also show the results for optimizing the norm parameter p *class-wise* on the training set and measuring the performance on the test set (see Table 3.4 for the VOC dataset and Table 3.5

Table 3.2: AP scores obtained on the VOC2009 data set with fixed ℓ_p-norm. Higher scores are better.

Norm	**Average**	Aeroplane	Bicycle	Bird	Boat	Bottle
ℓ_1	54.94 ± 12.3	**84.84** ± 5.86	55.35 ± 10.5	59.38 ± 10.1	66.83 ± 12.4	25.91 ± 10.2
$\ell_{1.125}$	57.07 ± 12.7	**84.82** ± 5.91	**57.25** ± 10.6	62.4 ± 9.13	**67.89** ± 12.8	**27.88** ± 9.91
$\ell_{1.333}$	57.2 ± 12.8	**84.51** ± 6.27	**57.41** ± 10.8	**62.75** ± 9.07	**67.99** ± 13	**27.44** ± 9.77
ℓ_2	56.53 ± 12.8	84.12 ± 5.92	56.89 ± 10.9	**62.53** ± 8.9	67.69 ± 13	26.68 ± 9.94
ℓ_∞	56.08 ± 12.7	83.67 ± 5.99	56.09 ± 10.9	61.91 ± 8.81	67.52 ± 12.9	26.5 ± 9.5

Norm	Bus	Car	Cat	Chair	Cow	Diningtable
ℓ_1	**71.15** ± 23.2	54.54 ± 7.33	59.5 ± 8.22	53.3 ± 11.7	23.13 ± 13.2	**48.51** ± 19.9
$\ell_{1.125}$	**71.7** ± 22.8	**56.59** ± 8.93	**61.59** ± 8.26	**54.3** ± 12.1	**29.59** ± 16.2	**49.32** ± 19.5
$\ell_{1.333}$	**71.33** ± 23.1	**56.75** ± 9.28	**61.74** ± 8.41	**54.25** ± 12.3	**29.89** ± 15.8	48.4 ± 19.3
ℓ_2	70.33 ± 22.3	55.92 ± 9.49	61.39 ± 8.37	**53.85** ± 12.4	28.39 ± 16.2	47 ± 18.7
ℓ_∞	70.13 ± 22.2	55.58 ± 9.47	**61.25** ± 8.28	53.13 ± 12.4	27.56 ± 16.2	46.29 ± 18.8

Norm	Dog	Horse	Motorbike	Person	Pottedplant	Sheep
ℓ_1	41.72 ± 9.44	57.67 ± 12.2	55 ± 13.2	81.32 ± 9.49	35.14 ± 13.4	38.13 ± 19.2
$\ell_{1.125}$	**45.57** ± 10.6	**59.4** ± 12.2	**57.66** ± 13.1	**82.18** ± 9.3	39.05 ± 14.9	**43.65** ± 20.5
$\ell_{1.333}$	**45.85** ± 10.9	**59.4** ± 11.9	**57.57** ± 13	**82.27** ± 9.29	**39.7** ± 14.6	**46.28** ± 23.9
ℓ_2	45.14 ± 10.8	58.61 ± 11.9	**56.9** ± 13.2	82.19 ± 9.3	38.97 ± 14.8	**45.88** ± 24
ℓ_∞	44.63 ± 10.6	58.32 ± 11.7	**56.45** ± 13.1	82 ± 9.37	38.46 ± 14.1	**45.93** ± 24

Norm	Sofa	Train	Tvmonitor
ℓ_1	**48.15** ± 11.8	**75.33** ± 14.1	63.97 ± 10.2
$\ell_{1.125}$	**48.72** ± 13	**75.79** ± 14.4	**65.99** ± 9.83
$\ell_{1.333}$	**48.76** ± 11.9	**75.75** ± 14.3	**66.07** ± 9.59
ℓ_2	47.29 ± 11.7	**75.29** ± 14.5	**65.55** ± 10.1
ℓ_∞	46.08 ± 11.8	74.89 ± 14.5	**65.19** ± 10.2

AP scores were computed by cross-validation on the training set. Bold faces show the best method and all other ones that are not statistical-significantly worse by a Wilcoxon's signed rank test with a p-value of 0.05.

for the ImageCLEF dataset). We can see from Table 3.5 that optimizing the ℓ_p-norm class-wise is beneficial: selecting the best $p \in]1, \infty[$ class-wise, the result is increased to an AP of 37.02—this is almost 0.6 AP better than the result for the vanilla sum-kernel SVM. Including the ℓ_1-norm MKL in the candidate set results in no gains. Similarly, including the sum-kernel SVM to the set of models, the AP score does not increase compared to using ℓ_p-Norms in $]1, \infty[$ alone. A qualitatively similar result can be seen from Table 3.4 for the VOC 2009 dataset where we observe a gain of 0.9 AP compared to the sum-kernel SVM.

Table 3.3: Average AP scores obtained on the ImageCLEF2010 test data set with ℓ_p-norm fixed for all classes. Higher scores are better.

ℓ_p-Norm	1	1.125	1.333	2	∞
	34.61	37.01	36.97	36.62	36.45

AP scores computed on the test set. Regularization constants were selected via AP scores computed via 12-fold cross-validation on the training set.

Table 3.4: Average AP scores on the VOC2009 test data with ℓ_p-norm class-wise optimized on training data. Higher scores are better.

∞	$\{1, \infty\}$	$\{1.125, 1.333, 2\}$	$\{1.125, 1.333, 2, \infty\}$	$\{1, 1.125, 1.333, 2\}$	all norms from the left
55.85	55.94	56.75	56.76	56.75	56.76

AP scores on test data were obtained on request from the challenge organizers due to undisclosed annotations. The class-wise selection of ℓ_p-norm and regularization constant relied on AP scores obtained via cross-validation on the training set.

Table 3.5: Average AP scores on the ImageCLEF2010 test data with ℓ_p-norm class-wise optimized. Higher scores are better.

∞	$\{1.125, 1.333, 2\}$	$\{1.125, 1.333, 2, \infty\}$	$\{1, 1.125, 1.333, 2\}$	all norms from the left
36.45	37.02	37.00	36.94	36.95

AP scores computed on the test set. The class-wise selection of ℓ_p-norm and regularization constant relied on AP scores obtained via cross-validation on the training set.

We conclude that optimizing the norm parameter p class-wise improves performance compared to the sum kernel SVM and, more importantly, model selection for the class-wise optimal ℓ_p-norm on the training set is stable in the sense that the choices make sense by their AP scores on the test set; additionally, one can rely on ℓ_p-norm MKL alone without the need to additionally include the sum-kernel-SVM to the set of models. Tables 3.2 and 3.1 show that the gain in performance for MKL varies considerably on the actual concept class. The same also holds for the ImageCLEF2010 dataset.

3.3.5 Analysis and Interpretation

Analysis of the Chosen Kernel Set with Kernel Alignment

We now analyze the kernel set in an explorative manner; to this end, our methodological tools are the following

1. Pairwise kernel alignment scores (KKA)
2. Kernel-target alignment scores (KTA).

Both are based on measuring angles between kernel matrices embedded in a vector space and are explained briefly in section 1.3.4. The KKA score measures a similarity between two kernels computed from image features. The KTA score measures a similarity between one of our computed feature kernels and an optimally discriminative kernel derived from the visual concept labels. Alternatively RDE (102) can be used which on these datasets did not yield conclusive results. For an introduction to kernel alignment we refer to section 1.3.4 and the work in (59).

To start with, we computed the pairwise kernel alignment scores of the 32 base kernels: they are shown in Fig. 3.1. We recall that the kernels can be classified into the following groups: Kernels 1–15 and 16–23 employ BoW-S and BoW-C features, respectively; Kernels 24 to 27 are product kernels associated with the HoG and HoC features; Kernels 28–30 deploy HoC, and, finally, Kernels 31–32 are based on HoG features over the gray channel. We see from the block-diagonal structure that features that are of the same type (but are generated for different parameter values, color channels, or spatial tilings) are strongly correlated. Furthermore the BoW-S kernels (Kernels 1–15) are weakly correlated with the BoW-C kernels (Kernels 16–23). Both, the BoW-S and HoG kernels (Kernels 24–25,31–32) use gradients and therefore are moderately correlated; the same holds for the BoW-C and HoC kernel groups (Kernels 26–30). This corresponds to our original intention to have a broad range of feature types which are, however, useful for the task at hand. The principle usefulness of our feature set can be seen a posteriori from the fact that ℓ_1-MKL achieves the worst performance of all methods included in the comparison while the sum-kernel SVM performs moderately well. Clearly, a higher fraction of noise kernels would further harm the sum-kernel SVM and favor the sparse MKL instead.

Based on the observation that the BoW-S kernel subset shows high KTA scores, we also evaluated the performance restricted to the 15 BoW-S kernels only. Unsurprisingly, this setup

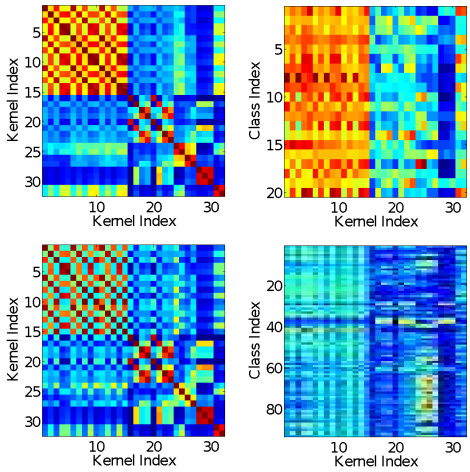


Figure 3.1: Similarity of the kernels for the VOC2009 (TOP) and ImageCLEF2010 (BOTTOM) data sets in terms of pairwise kernel alignments (LEFT) and kernel target alignments (RIGHT), respectively. In both data sets, five groups can be identified: 'BoW-S' (Kernels 1–15), 'BoW-C' (Kernels 16–23), 'products of HoG and HoC kernels' (Kernels 24–27), 'HoC single' (Kernels 28–30), and 'HoG single' (Kernels 31–32). On the left side rows and columns correspond to single kernels. On the right side columns correspond to kernels while rows correspond to visual concepts.

favors the sum-kernel SVM, which achieves higher results on VOC2009 for most classes; compared to ℓ_p-norm MKL using all 32 classes, the sum-kernel SVM restricted to 15 classes achieves slightly better AP scores for 11 classes, but also slightly worse for 9 classes. Furthermore, the sum kernel SVM, ℓ_2-MKL, and $\ell_{1.333}$-MKL were on par with differences fairly below 0.01 AP. This is again not surprising as the kernels from the BoW-S kernel set are strongly correlated with each other for the VOC data which can be seen in the top left image in Fig. 3.1. For the ImageCLEF data we observed a quite different picture: the sum-kernel SVM restricted to the 15 BoW-S kernels performed significantly worse, when, again, being compared to non-sparse ℓ_p-norm MKL using all 32 kernels. To achieve top state-of-the-art performance, one could optimize the scores for both datasets by considering the class-wise maxima over learning methods *and* kernel sets. However, since the intention here is not to win a challenge but a relative comparison of models, giving insights in the nature of the methods—we therefore discard the time-consuming optimization over the kernel subsets.

From the above analysis, the question arises why restricting the kernel set to the 15 BoW-S kernels affects the performance of the compared methods differently, for the VOC2009 and

ImageCLEF2010 data sets. This can be explained by comparing the KKA/KTA scores of the kernels attained on VOC and on ImageCLEF (see Fig. 3.1 (RIGHT)): for the ImageCLEF data set the KTA scores are substantially more spread along all kernels; there is neither a dominance of the BoW-S subset in the KTA scores nor a particularly strong correlation within the BoW-S subset in the KKA scores. We attribute this to the less object-based and more ambiguous nature of many of the concepts contained in the ImageCLEF data set. Furthermore, the KKA scores for the ImageCLEF data (see Fig. 3.1 (LEFT)) show that this dataset exhibits a higher variance among kernels—this is because the correlations between all kinds of kernels are weaker for the ImageCLEF data.

Therefore, because of this non-uniformity in the spread of the information content among the kernels, we can conclude that indeed our experimental setting falls into the situation where non-sparse MKL can outperform the baseline procedures. For example, the BoW features are more informative than HoG and HoC, and thus the uniform-sum-kernel-SVM is suboptimal. On the other hand, because of the fact that typical image features are only moderately informative, HoG and HoC still convey a certain amount of complementary information—this is what allows the performance gains reported in Tables 3.2 and 3.3.

Note that we class-wise normalized the KTA scores to sum to one. This is because we are rather interested in a comparison of the relative contributions of the particular kernels than in their absolute information content, which anyway can be more precisely derived from the AP scores already reported in Tables 3.2 and 3.3. Furthermore, note that we consider *centered* KKA and KTA scores, since it was argued in (60) that only those correctly reflect the test errors attained by established learners such as SVMs.

The Role of the Choice of ℓ_p-norm

Next, we turn to the interpretation of the norm parameter p in our algorithm. We observe a big gap in performance between $\ell_{1.125}$-norm MKL and the sparse ℓ_1-norm MKL. The reason is that for $p > 1$ MKL is reluctant to set kernel weights to zero, as can be seen from Figure 3.2. In contrast, ℓ_1-norm MKL eliminates 62.5% of the kernels from the working set. The difference between the ℓ_p-norms for $p > 1$ lies solely in the ratio by which the less informative kernels are down-weighted—they are never assigned with true zeros.

However, as proved in (56), in the computational optimum, the kernel weights are accessed by the MKL algorithm via the information content of the particular kernels given by a ℓ_p-norm-dependent formula (see Eq. (3.7); this will be discussed in detail in Section 3.4.1). We mention

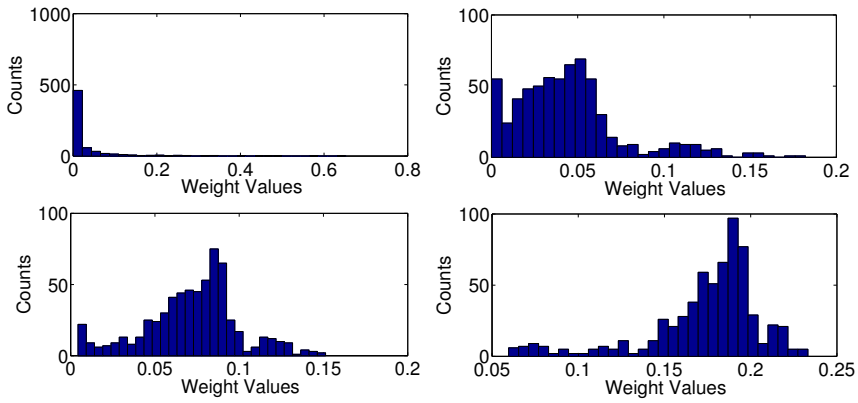


Figure 3.2: Histograms of kernel weights as output by ℓ_p-norm MKL for the various classes on the VOC2009 data set (32 kernels $\times$ 20 classes, resulting in 640 values). ℓ_1-norm (TOP LEFT)), $\ell_{1.125}$-norm (TOP RIGHT), $\ell_{1.333}$-norm (BOTTOM LEFT), and ℓ_2-norm (BOTTOM RIGHT).

at this point that the kernel weights all converge to the same, uniform value for $p \to \infty$. We can confirm these theoretical findings empirically: the histograms of the kernel weights shown in Fig. 3.2 clearly indicate an increasing uniformity in the distribution of kernel weights when letting $p \to \infty$. Higher values of p thus cause the weight distribution to shift away from zero and become slanted to the right while smaller ones tend to increase its skewness to the left.

Selection of the ℓ_p-norm permits to tune the strength of the regularization of the learning of kernel weights. In this sense the sum-kernel SVM clearly is an extreme, namely fixing the kernel weights, obtained when letting $p \to \infty$. The sparse MKL marks another extreme case: ℓ_p-norms with p below 1 loose the convexity property so that $p = 1$ is the maximally sparse choice preserving convexity at the same time. Sparsity can be interpreted here that only a few kernels are selected which are considered most informative according to the optimization objective. Thus, the ℓ_p-norm acts as a prior parameter for how much we trust in the informativeness of a kernel. In conclusion, this interpretation justifies the usage of ℓ_p-norm outside the existing choices ℓ_1 and ℓ_2. The fact that the sum-kernel SVM is a reasonable choice in the context of image annotation will be discussed further in Section 3.4.1.

Our empirical findings on ImageCLEF and VOC seem to contradict previous ones about the usefulness of MKL reported in the literature, where ℓ_1 is frequently to be outperformed by a simple sum-kernel SVM (for example, see (101, 131))—however, in these studies the sum-kernel SVM is compared to ℓ_1-norm or ℓ_2-norm MKL only. In fact, our results *confirm* these findings: ℓ_1-norm MKL is outperformed by the sum-kernel SVM in all of our experiments.

Nevertheless, in this chapter, we show that by using the more general ℓ_p-norm regularization, the prediction accuracy of MKL can be considerably leveraged, even clearly outperforming the sum-kernel SVM, which has been shown to be a tough competitor in the past (101). But of course also the simpler sum-kernel SVM also has its advantage, although on the computational side only: in our experiments it was about a factor of ten faster than its MKL competitors. Further information about running times of MKL algorithms compared to sum kernel SVMs can be taken from (56).

Remarks for Particular Concepts Finally, we show images from classes where MKL helps performance and discuss relationships to kernel weights. We have seen above that the sparsity-inducing ℓ_1-norm MKL clearly outperforms all other methods on the *bottle* class (see Table 3.1). Fig. 3.3 shows two typical highly ranked images and the corresponding kernel weights as output by ℓ_1-norm (LEFT) and $\ell_{1.333}$-norm MKL (RIGHT), respectively, on the bottle class. We observe that ℓ_1-norm MKL tends to rank highly party and people group scenes. We conjecture that this has two reasons: first, many people group and party scenes come along with co-occurring bottles. Second, people group scenes have similar gradient distributions to images of large upright standing bottles sharing many dominant vertical lines and a thinner head section—see the left- and right-hand images in Fig. 3.3. Sparse ℓ_1-norm MKL strongly focuses on the dominant HoG product kernel, which is able to capture the aforementioned special gradient distributions, giving small weights to two HoC product kernels and almost completely discarding all other kernels.

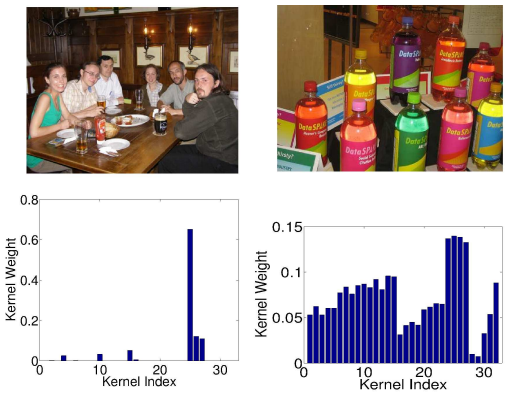


Figure 3.3: Images of typical highly ranked bottle images and kernel weights from ℓ_1-MKL (left) and $\ell_{1.333}$-MKL (right).

Next, we turn to the *cow* class, for which we have seen above that $\ell_{1.333}$-norm MKL outperforms all other methods clearly. Fig. 3.4 shows a typical high-ranked image of that class and also the corresponding kernel weights as output by ℓ_1-norm (LEFT) and $\ell_{1.333}$-norm (RIGHT) MKL, respectively. We observe that ℓ_1-MKL focuses on the two HoC product kernels; this is justified by typical cow images having green grass in the background. This allows the HoC kernels to easily to distinguish the cow images from the indoor and vehicle classes such as *car* or *sofa*. However, horse and sheep images have such a green background, too. They differ in sheep usually being black-white, and horses having a brown-black color bias (in VOC data); cows have rather variable colors. Here, we observe that the rather complex yet somewhat color-based BoW-C and BoW-S features help performance—it is also those kernels that are selected by the non-sparse $\ell_{1.333}$-MKL, which is the best performing model on those classes. In contrast, the sum-kernel SVM suffers from including the five gray-channel-based features, which are hardly useful for the horse and sheep classes and mostly introduce additional noise. MKL (all variants) succeed in identifying those kernels and assign those kernels with low weights.

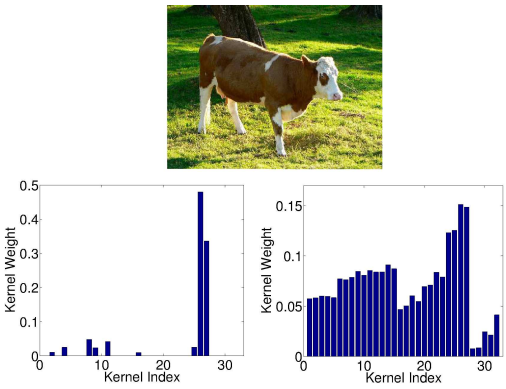


Figure 3.4: Images of a typical highly ranked cow image and kernel weights from ℓ_1-MKL (left) and $\ell_{1.333}$-MKL (right).

3.4 Promoting and Limiting Factors for Multiple Kernel Learning

In the previous section we presented empirical evidence that ℓ_p-norm MKL considerably can help performance in visual image categorization tasks. We also observed that the gain is class-specific and limited for some classes when compared to the sum-kernel SVM, see again Tables 3.2 and 3.1. The same also holds for the ImageCLEF2010 dataset. In this section, we aim

to shed light on the reasons of this behavior, in particular discussing strengths of the average kernel in Section 3.4.1, trade-off effects in Section 3.4.2 and strengths of MKL in Section 3.4.3. Since these scenarios are based on statistical properties of kernels which can be observed in concept recognition tasks within computer vision we expect the results to be transferable to other algorithms which learn linear models over kernels such as (122, 123).

3.4.1 One Argument For the Sum Kernel: Randomness in Feature Extraction

We would like to draw attention to one aspect present in BoW features, namely the amount of randomness induced by the visual word generation stage acting as noise with respect to kernel selection procedures.

Experimental setup We consider the following experiment, similar to the one undertaken in (131): we compute a BoW kernel ten times each time using the same local features, identical spatial pyramid tilings, and identical kernel functions; the only difference between subsequent repetitions of the experiment lies in the randomness involved in the generation of the codebook of visual words. Note that we use SIFT features over the gray channel that are densely sampled over a grid of step size six, 512 visual words (for computational feasibility of the clustering), and a χ^2 kernel. This procedure results in ten kernels that only differ in the randomness stemming from the codebook generation. We then compare the performance of the sum-kernel SVM built from the ten kernels to the one of the best single-kernel SVM determined by cross-validation-based model selection.

In contrast to (131) we try *two* codebook generation procedures, which differ by their intrinsic amount of randomness: first, we deploy k-means clustering, with random initialization of the centers and a bootstrap-like selection of the best initialization (similar to the option 'cluster' in MATLAB's k-means routine). Second, we deploy *extremely randomized clustering forests* (ERCF) (31, 132), that are, ensembles of randomized trees—the latter procedure involves a considerably higher amount of randomization compared to k-means.

Results The results are shown in Table 3.6. For both clustering procedures, we observe that the sum-kernel SVM outperforms the best single-kernel SVM. In particular, this confirms earlier findings of (131) carried out for k-means-based clustering. We also observe that the difference between the sum-kernel SVM and the best single-kernel SVM is much more pronounced for ERCF-based kernels—we conclude that this stems from a higher amount of randomness is involved in the ERCF clustering method when compared to conventional k-means. The standard

deviations of the kernels in Table 3.6 confirm this conclusion. For each class we computed the conditional standard deviation

$$\mathrm{std}(K \mid y_i = y_j) + \mathrm{std}(K \mid y_i \neq y_j) \tag{3.5}$$

averaged over all classes. The usage of a conditional variance estimator is justified because the ideal similarity in kernel target alignment (cf. equation (1.32)) does have a variance over the kernel as a whole however the conditional deviations in equation (3.5) would be zero for the ideal kernel. Similarly, the fundamental MKL optimization formula (3.7) relies on a statistic based on the two conditional kernels used in formula (3.5). Finally, ERCF clustering uses label information. Therefore averaging the class-wise conditional standard deviations over all classes is not expected to be identical to the standard deviation of the whole kernel.

Table 3.6: AP Scores and standard deviations showing amount of randomness in feature extraction. Higher AP scores are better.

Method	Best Single Kernel	Sum Kernel
VOC-KM	AP: 44.42 $\pm$ 12.82	**45.84** $\pm$ 12.94
VOC-KM	Std: **30.81**	30.74
VOC-ERCF	AP: 42.60 $\pm$ 12.50	**47.49** $\pm$ 12.89
VOC-ERCF	Std: **38.12**	37.89
CLEF-KM	AP: 31.09 $\pm$ 5.56	**31.73** $\pm$ 5.57
CLEF-KM	Std: **30.51**	30.50
CLEF-ERCF	AP: 29.91 $\pm$ 5.39	**32.77** $\pm$ 5.93
CLEF-ERCF	Std: **38.58**	38.10

AP Scores and standard deviations showing amount of randomness in feature extraction: Results from repeated computations of BoW Kernels with randomly initialized codebooks. VOC-KM denotes VOC2009 dataset and k-means for visual word generation, VOC-ERCF denotes VOC2009 dataset and ERCF for visual word generation. Similarly CLEF denotes ImageCLEF2010 dataset.

We observe in Table 3.6 that the standard deviations are lower for the sum kernels. Comparing ERCF and k-means shows that the former not only exhibits larger absolute standard deviations but also greater differences between single-best and sum-kernel as well as larger differences in AP scores.

We can thus postulate that the reason for the superior performance of the sum-kernel SVM stems from averaging out the randomness contained in the BoW kernels (stemming from the visual-word generation). This can be explained by the fact that averaging is a way of reducing

the variance in the predictors/models (133). We can also remark that such variance reduction effects can also be observed when averaging BoW kernels with varying color combinations or other parameters; this stems from the randomness induced by the visual word generation.

Note that in the above experimental setup each kernel uses the *same* information provided via the local features. Consequently, the best we can do is *averaging*—learning kernel weights in such a scenario is likely to suffer from overfitting to the noise contained in the kernels and can only decrease performance.

To further analyze this, we recall that, in the computational optimum, the information content of a kernel is measured by ℓ_p-norm MKL via the following quantity, as proved in (56):

$$\beta \propto \|w\|_2^{\frac{2}{p+1}} = \left(\sum_{i,j} \alpha_i y_i K_{ij} \alpha_j y_j \right)^{\frac{2}{p+1}} . \tag{3.6}$$

In this chapter we deliver a novel interpretation of the above quantity; to this end, we decompose the right-hand term into two terms as follows:

$$\sum_{i,j} \alpha_i y_i K_{ij} \alpha_j y_j = \sum_{i,j|y_i=y_j} \alpha_i K_{ij} \alpha_j - \sum_{i,j|y_i \neq y_j} \alpha_i K_{ij} \alpha_j \, .$$

The above term can be interpreted as a difference of the support-vector-weighted sub-kernel restricted to consistent labels *and* the support-vector-weighted sub-kernel over the opposing labels. Equation (3.6) thus can be rewritten as

$$\beta \propto \left(\sum_{i,j|y_i=y_j} \alpha_i K_{ij} \alpha_j - \sum_{i,j|y_i \neq y_j} \alpha_i K_{ij} \alpha_j \right)^{\frac{2}{p+1}} . \tag{3.7}$$

Thus, we observe that random influences in the features combined with overfitting support vectors can suggest a falsely high information content in this measure for *some* kernels. SVMs do overfit on BoW features. Using the scores attained on the training data subset we can observe that many classes are deceptive-perfectly predicted with AP scores fairly above 0.9. At this point, non-sparse $\ell_{p>1}$-norm MKL offers a parameter p for regularizing the kernel weights—thus hardening the algorithm to become robust against random noise, yet permitting to use some degree of information given by Equation (3.7).

(131) reported in accordance to our idea about overfitting of SVMs that ℓ_2-MKL and ℓ_1-MKL show no gain in such a scenario while ℓ_1-MKL even reduces performance for some datasets. This result is not surprising as the overly sparse ℓ_1-MKL has a stronger tendency to

overfit to the randomness contained in the kernels / feature generation. The observed amount of randomness in the state-of-the-art BoW features could be an explanation why the sum-kernel SVM has shown to be a quite hard-to-beat competitor for semantic concept classification and ranking problems.

3.4.2 MKL and Prior Knowledge

For solving a learning problem, there is nothing more valuable than *prior knowledge*. Our empirical findings on the VOC2009 and ImageCLEF09 data sets suggested that our experimental setup was actually biased towards the sum-kernel SVM via usage of prior knowledge when choosing the set of kernels / image features. We deployed kernels based on four features types: BoW-S, BoW-C, HoC and HoG. However, the *number* of kernels taken from each feature type is not equal. Based on our experience with the VOC and ImageCLEF challenges we used a higher fraction of BoW kernels and less kernels of other types such as histograms of colors or gradients because we already knew that BoW kernels have superior performance.

To investigate to what extend our choice of kernels introduces a bias towards the sum-kernel SVM, we also performed another experiment, where we deployed a higher fraction of weaker kernels for VOC2009. The difference to our previous experiments lies in that we summarized the 15 BOW-S kernels in 5 product kernels reducing the number of kernels from 32 to 22. The results are given in Table 3.7; when compared to the results of the original 32-kernel experiment (shown in Table 3.2), we observe that the AP scores are in average about 4 points smaller. This can be attributed to the fraction of weak kernels being higher as in the original experiment; consequently, the gain from using ($\ell_{1.333}$-norm) MKL compared to the sum-kernel SVM is now more pronounced: over 2 AP points—again, this can be explained by the higher fraction of weak (i.e., noisy) kernels in the working set.

In summary, this experiment should remind us that semantic classification setups use a substantial amount of prior knowledge. Prior knowledge implies a *pre-selection* of highly effective kernels—a carefully chosen set of strong kernels constitutes a bias towards the sum kernel. Clearly, pre-selection of strong kernels reduces the need for learning kernel weights; however, in settings where prior knowledge is sparse, statistical (or even adaptive, adversarial) noise is inherently contained in the feature extraction—thus, beneficial effects of MKL are expected to be more pronounced in such a scenario.

Table 3.7: MKL versus Prior Knowledge: AP Scores for a set of kernels with a smaller fraction of well scoring kernels. Higher scores are better.

Class / ℓ_p-norm	1.333	∞
Aeroplane	**77.82** $\pm$ 7.701	76.28 $\pm$ 8.168
Bicycle	**50.75** $\pm$ 11.06	46.39 $\pm$ 12.37
Bird	**57.7** $\pm$ 8.451	55.09 $\pm$ 8.224
Boat	**62.8** $\pm$ 13.29	60.9 $\pm$ 14.01
Bottle	**26.14** $\pm$ 9.274	25.05 $\pm$ 9.213
Bus	**68.15** $\pm$ 22.55	67.24 $\pm$ 22.8
Car	**51.72** $\pm$ 8.822	49.51 $\pm$ 9.447
Cat	**56.69** $\pm$ 9.103	55.55 $\pm$ 9.317
Chair	**51.67** $\pm$ 12.24	49.85 $\pm$ 12
Cow	**25.33** $\pm$ 13.8	22.22 $\pm$ 12.41
Diningtable	**45.91** $\pm$ 19.63	42.96 $\pm$ 20.17
Dog	**41.22** $\pm$ 10.14	39.04 $\pm$ 9.565
Horse	**52.45** $\pm$ 13.41	50.01 $\pm$ 13.88
Motorbike	**54.37** $\pm$ 12.91	52.63 $\pm$ 12.66
Person	**80.12** $\pm$ 10.13	79.17 $\pm$ 10.51
Pottedplant	**35.69** $\pm$ 13.37	34.6 $\pm$ 14.09
Sheep	**37.05** $\pm$ 18.04	34.65 $\pm$ 18.68
Sofa	**41.15** $\pm$ 11.21	37.88 $\pm$ 11.11
Train	**70.03** $\pm$ 15.67	67.87 $\pm$ 16.37
Tvmonitor	**59.88** $\pm$ 10.66	57.77 $\pm$ 10.91
Average	**52.33** $\pm$ 12.57	50.23 $\pm$ 12.79

In this set only five instead of 15 Bow-S kernels are used leading to a lower fraction of BoW-based kernels compared to kernels over global histogram features.

3.4.3 One Argument for Learning the Multiple Kernel Weights: Varying Informative Subsets of Data

In the previous sections, we have presented evidence for why the sum-kernel SVM is considered to be an efficient learner in visual image categorization. Nevertheless, in our experiments we have observed gains in accuracy by using non-sparse MKL for many concepts. In this section, we investigate causes for this performance gain.

We formulate a hypothesis for the performance gains achieved by MKL: each kernel is informative for a subset of the data in the sense that the kernel, when used in a SVM, classifies that subset well. These subsets can be partially disjoint between kernels and have varying

sizes. The MKL information criterion given in Eq. (3.7) is able to exploit such differences in informative subsets and is able to weight kernels properly despite being a *global* information measure that is computed over the support vectors (which in turn are chosen over the *whole dataset*).

In this section, we will present experimental evidence for this hypothesis in two steps. In the first step we show that our kernels computed from the real ImageCLEF2010 dataset indeed have fairly disjoint informative subsets. This suggests that our observed performance gains achieved by MKL could be explained by MKL being able to exploit such a scenario. In the second step we will create a toy dataset such that the informative subsets of kernels are disjoint by design. We will show that, in this controlled toy scenario, MKL outperforms average-kernel SVMs in a statistically significant manner. These two steps together will serve as evidence for our hypothesis given above.

The main question for the first step is how to determine which set of samples is informative for a given kernel matrix and how to measure the diversity of two sets defined by two kernels. Despite using ranking measures for most of the paper, we will stick here to a simple definition. Consider one binary classification problem. The set of all true positively and true negatively classified test examples using a SVM will be the informative subset for a kernel. If we restrict the kernel to the union of these two subsets of the test data set, then the resulting classifier would discriminate the two classes perfectly. Since we do not have test data labels for the Pascal VOC dataset, we will restrict ourselves to the ImageCLEF data.

The diversity measure will be defined in two steps: at first for two sets, then for a pair of kernels. The diversity measure $d(S_1, S_2)$ for two sets S_1, S_2 should have two properties: it should be 1 if these sets are maximally disjoint and be equal to zero if one set is contained in the other. The second property follows the idea that if the informative set of one kernel is contained in the informative set of another, then the first kernel is inferior to the second and we would like to reflect this in our diversity measure by setting it to zero as we would expect little gain from adding the first kernel to the second one in SVMs or MKL algorithms – we would say the inferior kernel does not add any diversity.

Using these two conditions we note that two sets S_1, S_2 are maximally disjoint if $|S_1 \cup S_2| = \min(|S_1| + |S_2|, N_{test})$,where N_{test} is the total number of test samples. Analogously, if one set is contained in the other, then $|S_1 \cup S_2| = \min(|S_1|, |S_2|)$. Linear interpolation between

these two extremes yield the diversity measure for a pair of sets S_1, S_2:

$$\bar{d}(S_1, S_2) = \frac{|S_1 \cup S_2| - \min(|S_1|, |S_2|)}{\min(|S_1| + |S_2|, N_{test}) - \min(|S_1|, |S_2|)} \tag{3.8}$$

Note that we do not use the symmetric difference here because this would be non-empty if one set was contained in the other.

The diversity measure $d(k_1, k_2)$ for two kernels k_1, k_2, still given a fixed binary classification problem, will be defined as the sum of the diversities between the two true positive sets from both kernels and the two true negative sets from both kernels. Let $TP(k)$ be the set of true positive samples of kernel k, and $TN(k)$ the corresponding set of true negative samples. Then we define

$$d(k_1, k_2) = \frac{\bar{d}(TP(k_1), TP(k_2)) + \bar{d}(TN(k_1), TN(k_2))}{2} \tag{3.9}$$

Treating true positives and true negatives separately makes sense because for most of the classes the positive labeled samples constitute only a small fraction of all samples which has its impact on the maximal number of true positives.

The diversity measure is actually a function of a classifier even though the difference in our case is made by varying the underlying kernels. In contrast to kernel target alignment (59) (see Section 1.3.4), or relevant dimensionality estimation (RDE) (102) it incorporates information about the classifiers itself by using true positives and true negatives. The former two methods rely on kernels and ground truth labels alone. Support vector machines do not use the whole kernel matrix in practice. The support vectors select and re-weight a subset of the kernel matrix corresponding to training data samples close to the decision hyperplane in kernel space. Thus, the above alternative measures, which consider the whole kernel matrix, may not be always optimal for explaining results of support vector machines. The motivation for introducing this novel measure is that incorporating extra information from support vector machines may help to validate a hypothesis related to classification results of support vector machines.

Since the ImageCLEF2010 dataset has 93 classes, we consider the average diversity of a pair of kernels over all classes and the maximal diversity of a pair of kernels over all classes. Figure 3.5 shows both diversities. We can see an interesting phenomenon: the diversities are low between the first 15 BoW-S kernels. This may serve as an explanation for anecdotal experiences that using MKL on BoW-S features alone yields no gains. The diversity is low but the randomness in feature extraction as discussed in a subsection above results in overfitting. However for the whole kernel set of all 32 kernels the diversities are large. The mean average

diversity (when the mean is computed over all pairs of kernels and the average of all 93 binary classification problems) is 37.77, the mean maximal diversity over all kernel pairs is 71.68 when the maximum is computed over all 93 binary classification problems. This concludes the first step: our kernel set does have partially disjoint sets of true positive and true negative samples between pairs of kernels. The informative subsets of kernels are fairly disjoint.

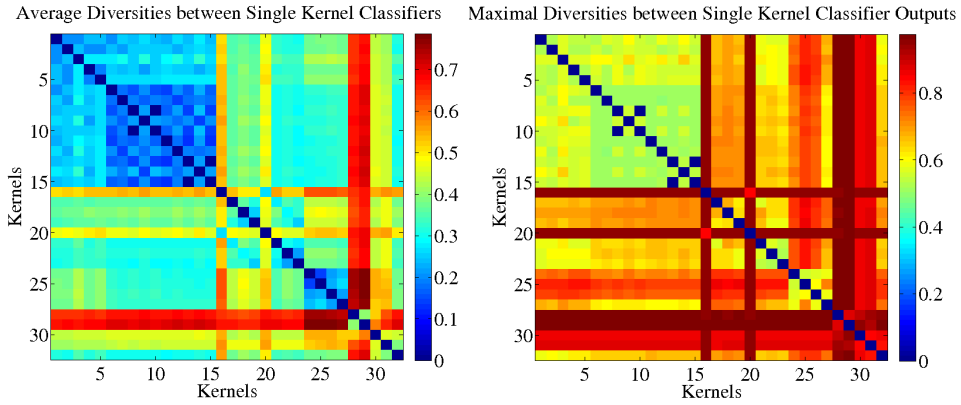


Figure 3.5: Diversity measure from Equation (3.9) between correctly classified samples for all pairs of 32 kernels. Left: Average over all concept classes. Right: Maximum over all concept classes. Rows and columns correspond to entries for a particular kernel index. Red colors correspond to highest diversity, blue to lowest.

In the second step we will construct two toy data sets in which by design we have kernels with disjoint informative subsets of varying sizes. The goal is to show that MKL outperforms the average kernel SVM under such conditions. This implies that the MKL information criterion given in Eq. (3.7) is able to capture such differences in informative subsets despite being a *global* information measure. In other words, the kernel weights are global weights that uniformly hold in all regions of the input space. While on the first look it appears to be a disadvantage, explicitly finding informative subsets of the input space on real data may not only imply a too high computational burden (note that the number of partitions of an n-element training set is exponential in n) but also is very likely to lead to overfitting.

We performed the following toy experiment. The coarse idea is that we create n features of dimension $6k$, where n is the number of data samples. We will compute k kernels such that the i-th kernel is computed only from the i-th consecutive block of 6 feature dimensions from all available $6k$ dimensions. We want the i-th kernel to have an informative subset of samples

and an uninformative complement. After drawing labels for all n samples, we partition all data samples into k blocks of varying size. The precise sizes of the blocks n_l will be given below.

The i-th block of data samples will be the informative subset for the i-th kernel. This will be achieved in the following way: for the i-th block of samples the i-th block of dimensions will be drawn from two Gaussians having different means such that the chosen Gaussian depends on the label of the data sample. This implies that each of the two Gaussians is responsible for creating the samples of one label. For all other samples (except for the i-th block of samples) the i-th block of dimensions will be drawn from an unconditional mixture of two Gaussians, i.e. which Gaussian is used will be independent of the sample labels. Therefore the i-th kernel which is computed from the i-th block of dimensions contains discriminative information only for the samples coming from the i-th block of samples. For all other samples, the i-th kernel uses features from a mixture of Gaussians independent of the sample labels which allows no discrimination of labels. By this construction the i-th kernel will have the i-th set of samples as discriminative subset. Furthermore, all kernels will have mutually disjoint informative subsets, because the i-th kernel is discriminative only on the i-th subset.

We generated a fraction of $p_+ = 0.25$ of positively labeled and $p_- = 0.75$ of negatively labeled training examples (motivated by the unbalancedness of training sets usually encountered in computer vision). The precise data creation protocol is given in the experimental section parts for experiments one and two.

We consider two experimental setups for sampling the data, which differ in the number of employed kernels k and the sizes of the informative sets. In both cases, the informative features are drawn from two sufficiently distant normal distributions (one for each class) while the uninformative features are just Gaussian noise (mixture of Gaussians). The experimental setup of the first experiment can be summarized as follows:

Experimental Settings for Experiment 1 (k=3 kernels):

Let n_l be the size of the l-th informative subset and $n = \sum_{l=1}^{k} n_l$ the total sample size. $\{f_i \in \mathbb{R}^{6k} \mid i = \{1 : n\}\}$ are the features to be drawn where $f_i^{(r)}$ is the r-th dimension of the i-th feature.

$$n_{l=1,2,3} := (300, 300, 500)$$

$$p_+ := P(y = +1) = 0.25$$

$$S_1 = \{1 : n_1\},\ S_{l>1} = \{n_{l-1} + 1 : n_l\}$$

$$f_i^{(r)} \in \begin{cases} \text{informative subset} & \text{if } i \in S_l \text{ and } r \in \{1+6(l-1):l\} \\ \text{uninformative subset} & \text{else} \end{cases} \tag{3.10}$$

The features for the informative subset are drawn according to

$$f_i^{(r)} \sim \begin{cases} N(0.0, \sigma_l) & \text{if } y_i = -1 \\ N(0.4, \sigma_l) & \text{if } y_i = +1 \end{cases} \tag{3.11}$$

$$\sigma_l = \begin{cases} 0.3 & \text{if } l = 1, 2 \\ 0.4 & \text{if } l = 3 \end{cases} \tag{3.12}$$

The features for the uninformative subset are drawn according to

$$f_i^{(r)} \sim (1 - p_+)N(0.0, 0.5) + p_+ N(0.4, 0.5). \tag{3.13}$$

Finally the l-th kernel is defined as

$$k_l(f_1, f_2) = \exp(-\sigma \|\pi_{\{1+6(l-1):l\}}(f_1 - f_2)\|_2^2),\ l = 1, \ldots, k \tag{3.14}$$

where $\pi_{\{1+6(l-1):l\}}(\cdot)$ is the projection on the feature dimensions ranging in the set $\{1+6(l-1):l\}$.

For Experiment 1 the three kernels had disjoint informative subsets of sizes $n_{k=1,2,3} = (300, 300, 500)$. We used 1100 data points for training and the same amount for testing. We repeated this experiment 500 times with different random draws of the data.

Note that the features used for the uninformative subsets are drawn as a mixture of the Gaussians with a higher variance, though. The increased variance encodes the assumption that the feature extraction produces unreliable results on the uninformative data subset. None of these kernels are pure noise or irrelevant. Each kernel is the only informative one for its own informative subset of data points.

We now turn to the experimental setup of the second experiment which is an extension to five kernels:

Experimental Settings for Experiment 2 (k=5 kernels):

Let n_l be the size of the l-th informative subset and $n = \sum_{l=1}^{k} n_l$ the total sample size. $\{f_i \in \mathbb{R}^{6k} \mid i = \{1:n\}\}$ are the features to be drawn where $f_i^{(r)}$ is the r-th dimension of the i-th feature.

$$n_{l=1,2,3,4,5} = (300, 300, 500, 200, 500),$$
$$p_+ := P(y = +1) = 0.25$$

$$S_1 = \{1 : n_1\},\ S_{l>1} = \{n_{l-1} + 1 : n_l\}$$

$$f_i^{(r)} \in \begin{cases} \text{informative subset} & \text{if } i \in S_l \text{ and } r \in \{1 + 6(l-1) : l\} \\ \text{uninformative subset} & \text{else} \end{cases} \tag{3.15}$$

The features for the informative subset are drawn according to

$$f_i^{(r)} \sim \begin{cases} N(0.0, \sigma_l) & \text{if } y_i = -1 \\ N(m_l, \sigma_l) & \text{if } y_i = +1 \end{cases} \tag{3.16}$$

$$m_l = \begin{cases} 0.4 & \text{if } l = 1, 2, 3 \\ 0.2 & \text{if } l = 4, 5 \end{cases} \tag{3.17}$$

$$\sigma_l = \begin{cases} 0.3 & \text{if } l = 1, 2 \\ 0.4 & \text{if } l = 3, 4, 5 \end{cases} \tag{3.18}$$

The features for the uninformative subset are drawn according to

$$f^{(r)} \sim (1 - p_+)N(0.0, 0.5) + p_+ N(m_l, 0.5) \tag{3.19}$$

Finally the l-th kernel is defined as

$$k_l(f_1, f_2) = \exp(-\sigma \|\pi_{\{1+6(l-1):l\}}(f_1 - f_2)\|_2^2),\ l = 1, \ldots, k \tag{3.20}$$

where $\pi_{\{1+6(l-1):l\}}(\cdot)$ is the projection on the feature dimensions ranging in the set $\{1 + 6(l - 1) : l\}$.

As for the real experiments, we normalized the kernels to having standard deviation 1 in Hilbert space and optimized the regularization constant by grid search in $C \in \{10^i \,|\, i = -2, -1.5, \ldots, 2\}$.

Table 3.8 shows the results. The null hypothesis of equal means is rejected by a t-test with a p-value of 0.000266 and 0.0000047, respectively, for Experiment 1 and 2, which is highly significant.

Experiment 2 shows that the design of the Experiment 1 is no singular lucky find: we can extend the setting of experiment 1 and observe similar results again when using more kernels; the performance gaps then even increased. Experiment 2 uses five kernels instead of just three. Again, the informative subsets are disjoint, but this time of sizes 300, 300, 500, 200, and 500; the the Gaussians are centered at 0.4, 0.4, 0.4, 0.2, and 0.2, respectively, for the positive class; and the variance is taken as $\sigma_k = (0.3, 0.3, 0.4, 0.4, 0.4)$. Compared to Experiment 1,

Table 3.8: AP Scores in Toy experiment using Kernels with disjoint informative subsets of Data. Higher scores are better. Lower p-values imply higher statistical significance of differences in scores.

Setup	ℓ_∞-SVM	$\ell_{1.0625}$-MKL	t-test p-value
1	68.72 ± 3.27	69.49 ± 3.17	0.000266
2	55.07 ± 2.86	56.39 ± 2.84	$4.7 \cdot 10^{-6}$

this results in even bigger performance gaps between the sum-kernel SVM and the non-sparse $\ell_{1.0625}$-MKL. One can imagine to create learning scenarios with more and more kernels in the above way, thus increasing the performance gaps—since we aim at a relative comparison, this, however, would not further contribute to validating or rejecting our hypothesis.

Furthermore, we also investigated the single-kernel performance of each kernel: we observed the best single-kernel SVM (which attained AP scores of 43.60, 43.40, and 58.90 for Experiment 1) being inferior to both MKL (regardless of the employed norm parameter p) and the sum-kernel SVM over the whole set of kernels. The differences were significant with fairly small p-values (for example, for $\ell_{1.25}$-MKL the p-value was still about 0.02).

We emphasize that we did not design the example in order to achieve a maximal performance gap between the non sparse MKL and its competitors. For such an example, see the toy experiment of (56). Our focus here was to confirm our hypothesis that kernels in semantic concept classification are based on varying informative subsets of the data—although MKL computes global weights, it emphasizes on kernels that are relevant on the largest informative set and thus approximates the infeasible combinatorial problem of computing an optimal partition/grid of the space into regions which underlie identical optimal weights. Though, in practice, we expect the situation to be more complicated as informative subsets may overlap between kernels instead of being disjoint as modeled here.

Nevertheless, our hypothesis also opens the way to new directions for learning of kernel weights, namely restricted to subsets of data chosen according to a meaningful principle. Finding such principles is one the future goals of MKL—we sketched one possibility: locality in feature space. A first starting point may be the work of (134, 135) on localized MKL.

We conclude the second step. MKL did outperform the average kernel SVM in this controlled toy data scenario with disjoint informative subsets for each kernel. It may serve as empirical evidence for our hypothesis why we observe gains using MKL on real data: MKL

with its global information criterion can exploit scenarios in which each kernel is informative for a subset of the data and these subsets are partially disjoint between kernels.

3.5 Conclusions

When measuring data with different measuring devices, it is always a challenge to combine the respective devices' uncertainties in order to fuse all available sensor information optimally. For images using many different features is a common strategy in visual object recognition. This raises the question of *how* to combine these features.

In this chapter, we revisited this important topic and discussed machine learning approaches to adaptively combine different image features in a systematic and theoretically well founded manner. While MKL approaches in principle solve this problem it has been observed that the standard ℓ_1-norm based MKL often cannot outperform SVMs that use an average of a large number of kernels. One hypothesis why this seemingly un-intuitive result may occur is that the sparsity prior may not be appropriate in many real world problems—especially, when prior knowledge is already at hand. We tested whether this hypothesis holds true for computer vision and applied the recently developed non-sparse ℓ_p MKL algorithms to object classification tasks. The ℓ_p-norm constitutes a less severe method of sparsification. By choosing p as a hyperparameter, which controls the degree of non-sparsity and regularization, from a set of candidate values with the help of a validation data, we showed that ℓ_p-MKL significantly improves SVMs with averaged kernels and the standard sparse ℓ_1 MKL.

From a theoretical viewpoint the works in (136, 137) show that under certain conditions, like differing decay rates of eigenspectra of kernel operators between kernels, non-sparse MKL yields faster convergence rates for increasing sample sizes compared to sparse ℓ_1-norm MKL. However, the analysis undertaken in this chapter identified overfitting of support vector machines as one source of issues with information fusion in practice. The work in (67) and stacking in (68) used cross-validation to generate SVM outputs which were subsequently used for computing kernels employed in information fusion. From a practical viewpoint designing multiple kernel learning criteria based on outputs computed by cross-validation is one potential direction for reducing the overfitting issues with the current MKL approaches. When compared to more heuristic schemes of iteratively removing the weakest kernel from or adding the next best kernel into a uniform mixture and evaluating the kernel mixture using crossvalidation the

approach to use MKL on crossvalidated outputs might offer an advantage when a non-uniform mixture of a subset of all kernels yields the optimal performance.

This approach based on outputs computed by cross-validation can be applied to settings where higher overfitting is expected due to a more flexible or higher-dimensional parametrization such as localized MKL (123, 126).

Future work may study the application of MKL in structured prediction setups as suggested for label kernels used in classification with taxonomies in Section 2.6 of Chapter 2. Another interesting direction is MKL-KDA (121, 122). The difference to the method studied in the present paper lies in the base optimization criterion: KDA (138) leads to non-sparse solutions in the support vectors $\boldsymbol{\alpha}$ of the SVM while ours leads to sparse ones (i.e., a low number of support vectors). While on the computational side the latter is expected to be advantageous, the first one might lead to more accurate solutions. We expect that the identical regularization over kernel weights (i.e., the choice of the norm parameter p) yields similar effects for MKL-KDA like for MKL-SVM. Another reason to believe in observing similar effects in KDA is that the first two observed effects in this study discussed in Section 3.4 originate from feature and kernel design such that any kernel-based algorithm will have to deal with them.

Information fusion based on multiple kernel learning does matter in practice. non-sparse MKL was employed for the best purely visual submission by mAP measure in the ImageCLEF2011 Photo Annotation challenge (1, 18).

3. INSIGHTS FROM CLASSIFYING VISUAL CONCEPTS WITH MULTIPLE KERNEL LEARNING

4

Outlook

This thesis is naturally not a complete treatment of the field of image annotation and ranking. I left some closely-related questions aside. One can try a similar analysis of what I did with MKL-SVM in chapter 3 using MKL-KDA. I did not expect a qualitatively new insights from it, however MKL-KDA seems to be much slower than MKL-KDA, resulting in much time for experiments without any new message. Similarly, I did not combine the hierarchical classification analyzed in chapter 2 with MKL for the classifiers at the edges. I have no doubts that one can see improvement from this combination compared to flat classification with or without MKL for some datasets. Again, I did not expect any qualitatively new insights from that straightforward combination.

There are many interesting questions which go beyond the setting of pure image annotation and ranking. One example is incorporation of more prior knowledge in problems like human action recognition. Segmentation did not prove very useful for image annotation with highly varying concepts in the sense that it is not used in the top submissions to recent benchmark competitions on concept classification. It may however be useful in human action recognition where the images are expected to come from a narrower domain. They are more restricted by showing humans being centered and of a certain minimum scale in the image. Another successful example from a narrower domain is (139) where segmentation is used to segment Cats and Dogs for discriminating between breeds. The images show animals centered and covering a larger part of the image which constitutes a difference to generic concept recognition where the scale and position of parts contributing to a concept can be small. I think incorporating prior knowledge about a problem without ending up in messy engineering is a true art.

Another direction would be to consider more complex settings compared to per category ranking up the far goal of understanding an image by guessing the interaction of components in it. What happens if one wants not only to annotate concepts but understand what parts of an image contribute to them? If one is interested in extracting interactions between concepts or regions which contribute to the classification of belonging to a concept? More complex problems could break the dominance of Bag of Word features or even kernel-based methods, in particular when the complexity of a problem makes it hard to design one unified loss function or a score to be optimized. This hypothesis can be supported by the fact that discriminatively trained part models are dominating in image detection (140). Part models have been revived in that setting. One extension of Bag of word features for representation of relations between parts (beyond weighted but orderless sets of features as done in (37)) would be a view of images as sets of local graphs with weighted edges and local features at the nodes. The idea is to represent an image by some way of aggregating many small graphs to circumvent problems from noise-corrupted edge weights in single graphs. In contrast to earlier approaches an image would not correspond to one single large graph but to a set of smaller ones and a local feature can be part of multiple disjoint graphs. The graphs allow to aggregate smaller regions into larger ones and encode relations between parts, yet avoiding the rigidity of early part models which tried to represent one object by one graph rigidly. The challenge would be to generate the graphs and to aggregate the graphs into one representation as it is done with mappings of local features into a BoW feature. However the first step would be to define a meaningful way to understand an image via an interpretation of relations in it.

A general question related to more complex image understanding settings is at what point generative methods may have advantages over discriminative ones. Clearly discriminative methods are strong when an objective function can be formulated and optimized. As with BoW features, discriminative methods may be limiting on very complex image understanding problems where the design of one loss function to be used for optimization becomes difficult. When a large number of different concepts and relations is to be predicted, generative methods could become more attractive again.

5

Appendix

5.1 Tables for Chapter 2: Semantic Concept Recognition with a Tree Structure over Concepts

The full comparison for Caltech256 animals 13 class subset and VOC2006 is shown in Tables 5.2 and 5.3.

Table 5.1: Errors on Caltech256 52 animals classes, 20 splits. Lower losses are better.

Method	Taxonomy Loss	0/1 Loss
one vs all	30.66 ± 0.46	**62.56** ± 0.67
struct mc mr $\Delta = \delta_T$	32.29 ± 0.35	66.91 ± 0.64
struct mc sr $\Delta = \delta_T$	33.48 ± 0.39	68.86 ± 0.60
struct mc sr $\Delta = \delta_{0/1}$	34.09 ± 0.38	68.05 ± 0.64
local tax AM	30.01 ± 0.31	79.82 ± 0.55
local tax scaled GM	**29.62** ± 0.34	76.19 ± 0.57
local tax greedy path-walk	40.31 ± 0.34	77.65 ± 0.46
struct tax mr $\Delta = \delta_T$	30.58 ± 0.31	81.19 ± 0.53
struct tax sr $\Delta = \delta_T$	– [a] $\pm$ –	– $\pm$ –
struct tax sr $\Delta = \delta_{0/1}$	39.16 ± 0.45	76.85 ± 0.59

[a] Did not terminate after over seven days. Jobs consume over 20GB.

Table 5.2: Errors on Caltech256 animals 13 class subset data, 20 splits. Lower losses are better.

Method	Taxonomy Loss	0/1 Loss
one vs all	42.49 ± 1.46	**57.04** ± 1.98
struct mc mr $\Delta = \delta_T$	42.76 ± 0.96	64.35 ± 1.40
struct mc sr $\Delta = \delta_T$	42.49 ± 1.49	57.06 ± 2.01
struct mc sr $\Delta = \delta_{0/1}$	42.40 ± 1.29	57.05 ± 1.77
local tax AM	41.78 ± 1.16	62.57 ± 1.42
local tax scaled GM	**40.58** ± 1.15	58.33 ± 1.50
local tax greedy path-walk	47.65 ± 1.13	63.33 ± 1.57
struct tax mr $\Delta = \delta_T$	41.48 ± 1.22	61.54 ± 1.55
struct tax sr $\Delta = \delta_T$	41.55 ± 1.65	58.21 ± 2.20
struct tax sr $\Delta = \delta_{0/1}$	44.32 ± 1.07	59.22 ± 1.51

Table 5.3: Errors on VOC2006 as multi-class problem, 20 splits. Lower losses are better.

Method	Taxonomy Loss	0/1 Loss
one vs all	27.09 ± 1.88	50.54 ± 2.51
struct mc mr $\Delta = \delta_T$	26.37 ± 1.77	51.04 ± 2.53
struct mc sr $\Delta = \delta_T$	27.20 ± 1.89	50.73 ± 2.54
struct mc sr $\Delta = \delta_{0/1}$	27.18 ± 1.87	50.70 ± 2.41
local tax AM	26.02 ± 1.66	50.48 ± 2.34
local tax scaled GM	25.86 ± 1.56	**50.10** ± 2.29
local tax greedy path-walk	27.15 ± 1.65	51.85 ± 2.28
struct tax mr $\Delta = \delta_T$	**25.78** ± 1.67	50.17 ± 2.17
struct tax sr $\Delta = \delta_T$	27.24 ± 1.61	52.55 ± 2.23
struct tax sr $\Delta = \delta_{0/1}$	27.63 ± 1.71	51.73 ± 2.50

5.2 Tables for Chapter 3: Insights from Classifying Visual Concepts with Multiple Kernel Learning

This supplement delivers the average precision **(AP)** scores for the ImageCLEF2010 test dataset listed for all 93 visual concepts and all ℓ_p-norms used including the average kernel as the special case ℓ^∞.

Table 5.4: AP scores on ImageCLEF2010 test data with fixed ℓ_p-norm. Higher scores are better. Part 1.

	Partylife	FamilyFriends	Beach	BuildSights	Snow	Citylife
ℓ^1	28.41	50.82	39.36	54.94	12.75	50.14
$\ell^{1.125}$	30.52	**52.55**	**42.75**	**57.23**	19.97	52.79
$\ell^{1.333}$	**30.84**	52.26	42.71	56.87	20.38	**52.8**
ℓ^2	30.46	51.54	41.77	55.72	19.94	52.34
ℓ^∞	30.55	50.76	40.78	55.26	**20.49**	51.69
	Landscape	Sports	Desert	Spring	Summer	Autumn
ℓ^1	81.42	7.464	10.85	5.962	28.39	26.12
$\ell^{1.125}$	**81.97**	**10.37**	15.3	13.52	29.12	32.79
$\ell^{1.333}$	81.8	10.33	15.12	15.59	**29.42**	33.49
ℓ^2	81.48	10.19	**16.55**	16	29.34	33.26
ℓ^∞	81.16	10.07	15.82	**16.54**	29.3	**33.58**
	Winter	NoSeason	Indoor	Outdoor	NoPlace	Plants
ℓ^1	15.66	96.51	61.8	90.79	60.1	78.04
$\ell^{1.125}$	19.49	96.61	**62.53**	91.39	60.65	79.28
$\ell^{1.333}$	**20.11**	**96.61**	62.44	**91.49**	**60.92**	**79.44**
ℓ^2	20.09	96.53	62.12	91.43	60.33	79.23
ℓ^∞	19.81	96.47	61.69	91.26	60.06	78.85
	Flowers	Trees	Sky	Clouds	Water	Lake
ℓ^1	43.25	63.03	91.39	87.65	62.69	26.24
$\ell^{1.125}$	46.42	65.35	**91.8**	**88**	65.43	26.95
$\ell^{1.333}$	47.47	**65.39**	91.73	87.93	**66.03**	**27.13**
ℓ^2	47.89	64.87	91.64	87.77	66.01	26.92
ℓ^∞	**47.91**	64.13	91.39	87.54	65.79	25.79

Table 5.5: AP scores on ImageCLEF2010 test data with fixed ℓ_p-norm. Higher scores are better. Part 2.

	River	Sea	Mountains	Day	Night	NoTime
ℓ^1	15.68	47.55	**53.21**	88.03	55.89	80.1
$\ell^{1.125}$	**19.75**	48.74	52.86	88.68	57.85	80.83
$\ell^{1.333}$	18.92	**48.79**	51.95	**88.69**	**58.19**	**80.83**
ℓ^2	18.57	48.19	51.03	88.54	58.13	80.62
ℓ^∞	17.8	47.77	50.36	88.4	57.85	80.38
	Sunny	Sunset	StillLife	Macro	Portrait	Overexpos
ℓ^1	46.51	81.16	37.64	48.5	65.58	17.43
$\ell^{1.125}$	49.82	**81.58**	**40.72**	**50.2**	67.58	**19.9**
$\ell^{1.333}$	**50.13**	81.37	40.65	49.66	**67.62**	18.9
ℓ^2	49.97	81.09	39.76	49.07	67.24	18.51
ℓ^∞	50.08	80.77	39.54	50.02	66.72	17.61
	Underexpos	NeutralIllum	MotionBlur	Outoffocus	PartBlur	NoBlur
ℓ^1	27.74	98.38	13.35	10.28	72.37	90.92
$\ell^{1.125}$	28	98.4	**19.82**	**15.08**	**74.26**	**91.39**
$\ell^{1.333}$	27.43	98.31	19.72	14.88	74.2	91.14
ℓ^2	26.99	98.26	19.22	14.21	73.8	91.21
ℓ^∞	**29.22**	**98.49**	18.47	13.47	73.31	91.06
	SinglePers	SmallGroup	BigGroup	NoPersons	Animals	Food
ℓ^1	54.52	30.74	34.31	91.5	44.24	49.57
$\ell^{1.125}$	**55.85**	**32.88**	41.11	91.99	49.78	52.73
$\ell^{1.333}$	55.78	32.78	**41.81**	**92.03**	**50.08**	**53.31**
ℓ^2	55.34	32.28	41.29	92	49.78	53.26
ℓ^∞	54.81	31.83	40.5	91.81	49.17	52.81
	Vehicle	Aesthetic	OverallQuality	Fancy	Architecture	Street
ℓ^1	45.17	**28.63**	**22.6**	17.14	27.04	29.46
$\ell^{1.125}$	**47.62**	28.25	22.41	17.95	28.8	33.7
$\ell^{1.333}$	47.35	27.14	21.57	17.15	**29.25**	**33.91**
ℓ^2	47	26.01	20.77	16.92	28.91	33.42
ℓ^∞	46.25	28.34	22.46	**18.82**	27.84	32.79
	Church	Bridge	ParkGarden	Rain	Toy	MusicInstr
ℓ^1	5.29	5.087	42.02	0.6378	15.27	5.066
$\ell^{1.125}$	**8.15**	7.437	44.44	0.8926	22.05	5.231
$\ell^{1.333}$	7.441	**7.546**	**44.75**	0.9725	**22.35**	5.445
ℓ^2	6.577	7.243	44.53	0.9875	21.97	**5.609**
ℓ^∞	6.241	7.117	43.91	**1.017**	20.58	5.33

Table 5.6: AP scores on ImageCLEF2010 test data with fixed ℓ_p-norm. Higher scores are better. Part 3.

	Shadow	Bodypart	Travel	Work	Birthday	VisualArt
ℓ^1	**11.23**	22.46	11.68	4.264	**1.143**	32.98
$\ell^{1.125}$	10.93	23.84	**12.89**	**4.596**	0.9434	32.99
$\ell^{1.333}$	10.15	**24.15**	12.49	4.468	0.9152	32.62
ℓ^2	9.702	23.63	12.33	4.314	0.8556	31.97
ℓ^∞	10.89	23.07	12.69	4.257	0.8731	**33.05**
	Graffiti	Painting	Artificial	Natural	Technical	Abstract
ℓ^1	3.411	12.66	12.64	71.16	5.979	**2.553**
$\ell^{1.125}$	**4.467**	18.57	13.96	**71.66**	**6.107**	2.33
$\ell^{1.333}$	4.273	18.83	13.67	71.64	5.853	2.137
ℓ^2	4.094	18.9	13.18	70.62	5.82	2.099
ℓ^∞	3.882	**19.58**	**13.97**	71.32	6.01	2.025
	Boring	Cute	Dog	Cat	Bird	Horse
ℓ^1	7.281	**59.58**	22.04	2.132	13.02	1.48
$\ell^{1.125}$	**7.68**	59.13	31.54	8.586	23.87	**4.414**
$\ell^{1.333}$	7.388	59.46	**31.99**	**8.97**	**23.98**	3.931
ℓ^2	7.23	58.08	31.85	8.208	23.33	3.408
ℓ^∞	7.167	58.88	31.11	7.626	22.7	3.279
	Fish	Insect	Car	Bicycle	Ship	Train
ℓ^1	0.915	11.51	31.27	18.9	8.157	12.97
$\ell^{1.125}$	**1.844**	**16.2**	**34**	**26.17**	**9.749**	**15.42**
$\ell^{1.333}$	1.684	15.6	33.89	26.13	9.164	14.4
ℓ^2	1.594	14.94	33.51	25.53	8.688	13.45
ℓ^∞	1.605	15.06	32.54	24.5	8.581	12.48
	Airplane	Skateboard	Female	Male	Baby	Child
ℓ^1	5.913	0.2205	44.4	20.65	8.028	6.304
$\ell^{1.125}$	11.08	**0.4211**	**45.78**	21.02	17.85	10.36
$\ell^{1.333}$	**11.14**	0.41	45.51	21.01	**18.14**	**11.01**
ℓ^2	10.22	0.3963	44.78	**21.03**	17.12	10.8
ℓ^∞	10.18	0.4172	43.58	20.86	15.22	10.67
	Teenager	Adult	Oldperson			
ℓ^1	21.32	53.03	5.068			
$\ell^{1.125}$	23.69	**54.33**	5.624			
$\ell^{1.333}$	23.35	53.96	**5.66**			
ℓ^2	23.03	53.4	5.58			
ℓ^∞	**23.78**	53	5.46			

5. APPENDIX

References

[1] Stefanie Nowak, Karolin Nagel, and Judith Liebetrau. **The CLEF 2011 Photo Annotation and Concept-based Retrieval Tasks**. In *CLEF (Notebook Papers/Labs/Workshop) 2011* (153). 2, 9, 13, 22, 77, 111

[2] Fei-Fei Li, Robert Fergus, and Pietro Perona. **One-Shot Learning of Object Categories**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(4):594–611, 2006. 3, 78, 85

[3] Corinna Cortes and Vladimir Vapnik. **Support-Vector Networks**. *Machine Learning*, **20**(3):273–297, 1995. 7, 9, 37, 79, 81

[4] Klaus-Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf. **An introduction to kernel-based learning algorithms**. *IEEE Transactions on Neural Networks*, **12**(2):181–201, 2001. 7, 9, 16, 37, 79

[5] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002. 7, 81

[6] Pascal Massart and Elodie Nedelec. **Risk bounds for statistical learning**, February 2007. 7

[7] Enno Mammen and Alexandre B. Tsybakov. **Smooth Discrimination Analysis**. *Ann. Statist*, **27**:1808–1829, 1999. 8

[8] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. **Content-Based Image Retrieval at the End of the Early Years**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(12):1349–1380, 2000. 8, 27

[9] Deng Cai, Xiaofei He, and Jiawei Han. **Efficient Kernel Discriminant Analysis via Spectral Regression**. In *ICDM*, pages 427–432. IEEE Computer Society, 2007. 9

[10] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. **Visual categorization with bags of keypoints**. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004. 9, 84

REFERENCES

[11] MARK EVERINGHAM, LUC VAN GOOL, CHRIS K. I. WILLIAMS, JOHN WINN, AND ANDREW ZISSERMAN. **The PASCAL Visual Object Classes Challenge (VOC)**. http://www.pascal-network.org/challenges/VOC/. 9, 78

[12] BART THOMEE, ADRIAN POPESCU, ROBERTO PAREDES, AND MAURICIO VILLEGAS. **The ImageCLEF 2012 Photo Challenge**. http://www.imageclef.org/2012/photo/. 9

[13] ALEX KRIZHEVSKY. **Learning Multiple Layers of Features from Tiny Images**. Technical report, University of Toronto, 2009. 9

[14] ALEX KRIZHEVSKY, ILYA SUTSKEVER, AND GEOFFREY HINTON. **ImageNet Classification with Deep Convolutional Neural Networks**. In *NIPS*, 2012. 9

[15] JIA DENG, ALEX BERG, SANJEEV SATHEESH, HAO SU, ADITYA KHOSLA, AND FEI-FEI LI. **The ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)**. http://www.image-net.org/challenges/LSVRC/2012/. 9, 14

[16] DAVID G. LOWE. **Distinctive Image Features from Scale-Invariant Keypoints**. *International Journal of Computer Vision*, **60**(2):91–110, 2004. 10, 15, 48, 84

[17] ALEXANDER BINDER, WOJCIECH SAMEK, KLAUS-ROBERT MÜLLER, AND MOTOAKI KAWANABE. **Enhanced Representation and Multi-Task Learning for Image Annotation**. *Computer Vision and Image Understanding*, 2012. accepted. 10, 13, 21, 25

[18] ALEXANDER BINDER, WOJCIECH SAMEK, MARIUS KLOFT, CHRISTINA MÜLLER, KLAUS-ROBERT MÜLLER, AND MOTOAKI KAWANABE. **The Joint Submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF2011 Photo Annotation Task**. In *CLEF (Notebook Papers/Labs/Workshop)* (153). 10, 13, 21, 26, 111

[19] LEI YANG, NANNING ZHENG, JIE YANG, MEI CHEN, AND HONG CHEN. **A biased sampling strategy for object categorization**. In *ICCV 2009* (143), pages 1141–1148. 10

[20] FRÉDÉRIC JURIE AND BILL TRIGGS. **Creating Efficient Codebooks for Visual Recognition**. In *ICCV 2005*, pages 604–610. IEEE Computer Society, 2005. 10, 11

[21] KOEN E. A. VAN DE SANDE AND THEO GEVERS. **The University of Amsterdam's Concept Detection System at ImageCLEF 2010**. In Braschler et al. (144). 10

[22] MARCIN MARSZALEK, CORDELIA SCHMID, HEDI HARZALLAH, AND JOOST VAN DE WEIJER. **Learning Representations for Visual Object Class Recognition**. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/workshop/marszalek.pdf. 10

[23] KOEN E. A. VAN DE SANDE, THEO GEVERS, AND CEES G. M. SNOEK. **Evaluating Color Descriptors for Object and Scene Recognition**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**(9):1582–1596, 2010. 10, 15, 48, 49, 79, 83

[24] THOMAS HOFMANN. **Probabilistic Latent Semantic Analysis**. In *UAI 1999*, pages 289–296. Morgan Kaufmann, 1999. 11

[25] GABRIELA CSURKA AND FLORENT PERRONNIN. **Fisher Vectors: Beyond Bag-of-Visual–Words Image Representations**. In PAUL RICHARD AND JOSÉ BRAZ, editors, *VISIGRAPP (Selected Papers)*, **229** of *Communications in Computer and Information Science*, pages 28–42. Springer, 2010. 11, 14

[26] JIANCHAO YANG, KAI YU, YIHONG GONG, AND THOMAS S. HUANG. **Linear spatial pyramid matching using sparse coding for image classification**. In *CVPR 2009* (148), pages 1794–1801. 11

[27] JAN VAN GEMERT, JAN-MARK GEUSEBROEK, COR J. VEENMAN, AND ARNOLD W. M. SMEULDERS. **Kernel Codebooks for Scene Categorization**. In *ECCV 2008(3)*, **5304** of *Lecture Notes in Computer Science*, pages 696–709. Springer, 2008. 12

[28] JINJUN WANG, JIANCHAO YANG, KAI YU, FENGJUN LV, THOMAS S. HUANG, AND YIHONG GONG. **Locality-constrained Linear Coding for image classification**. In *CVPR 2010* (152), pages 3360–3367. 12, 14

[29] DAVID NISTÉR AND HENRIK STEWÉNIUS. **Scalable Recognition with a Vocabulary Tree**. In *CVPR 2006(2)* (150), pages 2161–2168. 12

[30] WOJCIECH WOJCIKIEWICZ, ALEXANDER BINDER, AND MOTOAKI KAWANABE. **Enhancing Image Classification with Class-wise Clustered Vocabularies**. In *ICPR 2010*, pages 1060–1063. IEEE, 2010. 12, 23, 25

[31] FRANK MOOSMANN, ERIC NOWAK, AND FRÉDÉRIC JURIE. **Randomized Clustering Forests for Image Classification**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(9):1632–1646, 2008. 12, 23, 48, 98

[32] ALEXANDER BINDER, WOJCIECH WOJCIKIEWICZ, CHRISTINA MÜLLER, AND MOTOAKI KAWANABE. **A Hybrid Supervised-Unsupervised Vocabulary Generation Algorithm for Visual Concept Recognition**. In *ACCV 2010(3)*, **6494** of *Lecture Notes in Computer Science*, pages 95–108. Springer, 2010. 12, 21, 23, 25

[33] KAI YU, TONG ZHANG, AND YIHONG GONG. **Nonlinear Learning using Local Coordinate Coding**. In Bengio et al. (142), pages 2223–2231. 12

[34] LINGQIAO LIU, LEI WANG, AND XINWANG LIU. **In defense of soft-assignment coding**. In *ICCV* (154), pages 2486–2493. 13, 14

[35] K. CHATFIELD, V. LEMPITSKY, A. VEDALDI, AND A. ZISSERMAN. **The devil is in the details: an evaluation of recent feature encoding methods**. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2011. 14

REFERENCES

[36] SVETLANA LAZEBNIK, CORDELIA SCHMID, AND JEAN PONCE. **Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories**. In *CVPR 2006(2)* (150), pages 2169–2178. 14, 49, 79, 83

[37] TEÓFILO EMÍDIO DE CAMPOS, GABRIELA CSURKA, AND FLORENT PERRONNIN. **Images as sets of locally weighted features**. *Computer Vision and Image Understanding*, **116**(1):68–85, 2012. 15, 114

[38] HERBERT BAY, ANDREAS ESS, TINNE TUYTELAARS, AND LUC VAN GOOL. **Speeded-Up Robust Features (SURF)**. *Comput. Vis. Image Underst.*, **110**:346–359, June 2008. 15

[39] JASPER R. R. UIJLINGS, ARNOLD W. M. SMEULDERS, AND REMKO J. H. SCHA. **Real-Time Visual Concept Classification**. *IEEE Transactions on Multimedia*, **12**(7):665–681, 2010. 15

[40] KOEN E. A. VAN DE SANDE, THEO GEVERS, AND CEES G. M. SNOEK. **Empowering Visual Categorization With the GPU**. *IEEE Transactions on Multimedia*, **13**(1):60–70, 2011. 15

[41] DEVI PARIKH. **Recognizing jumbled images: The role of local and global information in image classification**. In *ICCV 2011* (154), pages 519–526. 15

[42] LIU YANG, RONG JIN, RAHUL SUKTHANKAR, AND FRÉDÉRIC JURIE. **Unifying discriminative visual codebook generation with classifier training for object category recognition**. In *CVPR 2008* (151). 16

[43] HICHEM SAHBI, JEAN-YVES AUDIBERT, AND RENAUD KERIVEN. **Context-Dependent Kernels for Object Classification**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **33**(4):699–708, 2011. 16

[44] OREN BOIMAN, ELI SHECHTMAN, AND MICHAL IRANI. **In defense of Nearest-Neighbor based image classification**. In *CVPR 2008* (151). 16

[45] NAKAMASA INOUE, YUSUKE KAMISHIMA, TOSHIYA WADA, KOICHI SHINODA, AND SHUNSUKE SATO. **TokyoTech+Canon at TRECVID 2011**. In *2011 TREC Video Retrieval Evaluation*, 2011. 16

[46] KOEN E. A. VAN DE SANDE, JASPER R. R. UIJLINGS, THEO GEVERS, AND ARNOLD W. M. SMEULDERS. **Segmentation as selective search for object recognition**. In *ICCV 2011* (154), pages 1879–1886. 16, 84

[47] BARBARA ANDRÉ, TOM VERCAUTEREN, ANNA M. BUCHNER, MICHAEL B. WALLACE, AND NICHOLAS AYACHE. **Retrieval Evaluation and Distance Learning from Perceived Similarity between Endomicroscopy Videos**. In Fichtinger et al. (155), pages 297–304. 16

[48] RUI XU, YASUSHI HIRANO, RIE TACHIBANA, AND SHOJI KIDO. **Classification of Diffuse Lung Disease Patterns on High-Resolution Computed Tomography by a Bag of Words Approach**. In Fichtinger et al. (155), pages 183–190. 16

[49] ANGEL CRUZ-ROA, JUAN C. CAICEDO, AND FABIO A. GONZÁLEZ. **Visual pattern mining in histology image collections using bag of features**. *Artificial Intelligence in Medicine*, **52**(2):91–106, 2011. 16

[50] JIANGUO ZHANG, MARCIN MARSZALEK, SVETLANA LAZEBNIK, AND CORDELIA SCHMID. **Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study**. *International Journal of Computer Vision*, **73**(2):213–238, 2007. 18, 49, 79, 83

[51] OLIVIER CHAPELLE, PATRICK HAFFNER, AND VLADIMIR VAPNIK. **Support vector machines for histogram-based image classification**. *IEEE Transactions on Neural Networks*, **10**(5):1055–1064, 1999. 18, 49, 79, 83

[52] CHRISTOPH H. LAMPERT AND MATTHEW B. BLASCHKO. **A Multiple Kernel Learning Approach to Joint Multi-class Object Detection**. In *DAGM 2008*, **5096** of *Lecture Notes in Computer Science*, pages 31–40. Springer, 2008. 18, 32, 49, 83

[53] ANDREA VEDALDI AND ANDREW ZISSERMAN. **Efficient Additive Kernels via Explicit Feature Maps**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **34**(3):480–492, 2012. 18, 19

[54] ALEXANDER ZIEN AND CHENG SOON ONG. **Multiclass multiple kernel learning**. In *ICML 2007* (145), pages 1191–1198. 19, 81, 87

[55] O. CHAPELLE AND A. RAKOTOMAMONJY. **Second Order Optimization of Kernel Parameters**. In *Proc. of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008. 19, 87

[56] MARIUS KLOFT, ULF BREFELD, SÖREN SONNENBURG, AND ALEXANDER ZIEN. **l_p-Norm Multiple Kernel Learning**. *Journal of Machine Learning Research*, **12**:953–997, 2011. 19, 22, 79, 81, 87, 94, 96, 100, 109

[57] ALI RAHIMI AND BENJAMIN RECHT. **Random Features for Large-Scale Kernel Machines**. In JOHN C. PLATT, DAPHNE KOLLER, YORAM SINGER, AND SAM T. ROWEIS, editors, *NIPS*. Curran Associates, Inc., 2007. 19

[58] FUXIN LI, GUY LEBANON, AND CRISTIAN SMINCHISESCU. **Chebyshev approximations to the histogram χ^2 kernel**. In *CVPR* (157), pages 2424–2431. 19

[59] NELLO CRISTIANINI, JOHN SHAWE-TAYLOR, ANDRÉ ELISSEEFF, AND JAZ S. KANDOLA. **On Kernel-Target Alignment**. In *NIPS 2001*, pages 367–373. MIT Press, 2001. 19, 62, 92, 104

[60] CORINNA CORTES, MEHRYAR MOHRI, AND AFSHIN ROSTAMIZADEH. **Two-Stage Learning Kernel Algorithms**. In *ICML 2010*, pages 239–246. Omnipress, 2010. 19, 94

[61] SEBASTIAN MIKA, GUNNAR RÄTSCH, JASON WESTON, BERNHARD SCHÖLKOPF, ALEX J. SMOLA, AND KLAUS-ROBERT MÜLLER. **Constructing Descriptive and Discriminative Nonlinear Features: Rayleigh Coefficients in Kernel Feature Spaces**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(5):623–633, 2003. 19

[62] ALEXANDER BINDER, KLAUS-ROBERT MÜLLER, AND MOTOAKI KAWANABE. **On Taxonomies for Multi-class Image Categorization**. *International Journal of Computer Vision*, **99**(3):281–301, 2012. Accepted January 2011. 22, 24, 29

[63] ALEXANDER BINDER, SHINICHI NAKAJIMA, MARIUS KLOFT, CHRISTINA MÜLLER, WOJCIECH SAMEK, ULF BREFELD, KLAUS-ROBERT MÜLLER, AND MOTOAKI KAWANABE. **Insights from Classifying Visual Concepts with Multiple Kernel Learning**. *PLoS ONE*, **7**(8):e38897, 08 2012. 22, 24, 78

[64] MARK EVERINGHAM, LUC VAN GOOL, CHRIS K. I. WILLIAMS, JOHN WINN, AND ANDREW ZISSERMAN. **The PASCAL Visual Object Classes Challenge 2009 (VOC2009)**. http://www.pascal-network.org/challenges/VOC/voc2009/ workshop/index.html, 2009. 22, 82

[65] STEFANIE NOWAK AND PETER DUNKER. **Overview of the CLEF 2009 Large-Scale Visual Concept Detection and Annotation Task**. In *CLEF 2009 (2)* (146), pages 94–109. 22

[66] M. MARSZALEK AND C. SCHMID. **Learning Representations for Visual Object Class Recognition**. 22, 82

[67] WOJCIECH SAMEK, ALEXANDER BINDER, AND MOTOAKI KAWANABE. **Multi-task Learning via Non-sparse Multiple Kernel Learning**. In *CAIP 2011(1)*, **6854** of *Lecture Notes in Computer Science*, pages 335–342. Springer, 2011. 23, 25, 110

[68] MUHAMMAD AWAIS, FEI YAN, KRYSTIAN MIKOLAJCZYK, AND JOSEF KITTLER. **Novel Fusion Methods for Pattern Recognition**. In DIMITRIOS GUNOPULOS, THOMAS HOFMANN, DONATO MALERBA, AND MICHALIS VAZIRGIANNIS, editors, *ECML/PKDD (1)*, **6911** of *Lecture Notes in Computer Science*, pages 140–155. Springer, 2011. 23, 110

[69] SHINICHI NAKAJIMA, ALEXANDER BINDER, CHRISTINA MÜLLER, WOJCIECH WOJCIKIEWICZ, MARIUS KLOFT, ULF BREFELD, KLAUS-ROBERT MÜLLER, AND MOTOAKI KAWANABE. **Multiple kernel learning for object classification**. In *Proceedings of the 12th Workshop on Information-based Induction Sciences*, 2009. 24

[70] MOTOAKI KAWANABE, SHINICHI NAKAJIMA, AND ALEXANDER BINDER. **A procedure of adaptive kernel combination with kernel-target alignment for object classification**. In STÉPHANE MARCHAND-MAILLET AND YIANNIS KOMPATSIARIS, editors, *CIVR*. ACM, 2009. 24

[71] MOTOAKI KAWANABE, ALEXANDER BINDER, CHRISTINA MÜLLER, AND WOJCIECH WOJCIKIEWICZ. **Multi-modal visual concept classification of images via Markov random walk over tags**. In *WACV*, pages 396–401. IEEE Computer Society, 2011. 25

[72] WOJCIECH WOJCIKIEWICZ, ALEXANDER BINDER, AND MOTOAKI KAWANABE. **Shrinking large visual vocabularies using multi-label agglomerative information bottleneck**. In *ICIP*, pages 3849–3852. IEEE, 2010. 25

[73] ALEXANDER BINDER AND MOTOAKI KAWANABE. **Enhancing Recognition of Visual Concepts with Primitive Color Histograms via Non-sparse Multiple Kernel Learning**. In *CLEF (2)* (146), pages 269–276. 26, 82, 85

[74] SÖREN SONNENBURG, GUNNAR RÄTSCH, SEBASTIAN HENSCHEL, CHRISTIAN WIDMER, JONAS BEHR, ALEXANDER ZIEN, FABIO DE BONA, ALEXANDER BINDER, CHRISTIAN GEHL, AND VOJTECH FRANC. **The SHOGUN Machine Learning Toolbox**. *Journal of Machine Learning Research*, **11**:1799–1802, 2010. 26, 51, 87

[75] Flickr™. http://www.flickr.com. 27

[76] KOBUS BARNARD, PINAR DUYGULU, DAVID A. FORSYTH, NANDO DE FREITAS, DAVID M. BLEI, AND MICHAEL I. JORDAN. **Matching Words and Pictures**. *Journal of Machine Learning Research*, **3**:1107–1135, 2003. 27

[77] GUO-JUN QI, XIAN-SHENG HUA, AND HONG-JIANG ZHANG. **Learning semantic distance from community-tagged media collection**. In *ACM Multimedia 2009*, pages 243–252. ACM, 2009. 27

[78] ALON ZWEIG AND DAPHNA WEINSHALL. **Exploiting Object Hierarchy: Combining Models from Different Category Levels**. In *ICCV 2007* (149), pages 1–8. 30, 44, 73

[79] GREGORY GRIFFIN AND PIETRO PERONA. **Learning and using taxonomies for fast visual categorization**. In *CVPR 2008* (151). 30, 32, 56

[80] BENJAMIN TASKAR, CARLOS GUESTRIN, AND DAPHNE KOLLER. **Max-Margin Markov Networks**. In SEBASTIAN THRUN, LAWRENCE K. SAUL, AND BERNHARD SCHÖLKOPF, editors, *NIPS*. MIT Press, 2003. 31, 34

[81] IOANNIS TSOCHANTARIDIS, 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. 31, 34, 36

[82] LIJUAN CAI AND THOMAS HOFMANN. **Hierarchical document categorization with support vector machines**. In *CIKM 2004*, pages 78–87. ACM, 2004. 31, 35, 38

[83] G. GRIFFIN, A. HOLUB, AND P. PERONA. **Caltech-256 Object Category Dataset**. Technical Report 7694, California Institute of Technology, 2007. 31, 45, 47

[84] MARK EVERINGHAM, ANDREW ZISSERMAN, CHRIS K. I. WILLIAMS, AND LUC VAN GOOL. **The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results**. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf. 31, 45, 47, 69

[85] MARCIN MARSZALEK AND CORDELIA SCHMID. **Constructing Category Hierarchies for Visual Recognition**. In *ECCV 2008(4)*, **5305** of *Lecture Notes in Computer Science*, pages 479–491. Springer, 2008. 32, 44, 60, 73

[86] MATTHEW B. BLASCHKO AND ARTHUR GRETTON. **Learning Taxonomies by Dependence Maximization**. In *NIPS 2008*, pages 153–160. Curran Associates, Inc., 2008. 32

[87] ROBERT TIBSHIRANI AND TREVOR HASTIE. **Margin Trees for High-dimensional Classification**. *Journal of Machine Learning Research*, **8**:637–652, 2007. 32

[88] XIAODONG FAN. **Efficient Multiclass Object Detection by a Hierarchy of Classifiers**. In *CVPR 2005(1)* (147), pages 716–723. 32

[89] MARCIN MARSZALEK AND CORDELIA SCHMID. **Semantic Hierarchies for Visual Object Recognition**. In *CVPR 2007*. IEEE Computer Society, 2007. 32

[90] TIANSHI GAO AND DAPHNE KOLLER. **Discriminative learning of relaxed hierarchy for large-scale visual recognition**. In *ICCV* (154), pages 2072–2079. 33, 73

[91] CHRISTOPH H. LAMPERT. **Maximum Margin Multi-Label Structured Prediction**. In Shawe-Taylor et al. (156), pages 289–297. 33, 74

[92] JOHN D. LAFFERTY, XIAOJIN ZHU, AND YAN LIU. **Kernel conditional random fields: representation and clique selection**. In Brodley (141). 35

[93] JASON WESTON AND CHRIS WATKINS. **Support vector machines for multi-class pattern recognition**. In *ESANN 1999*, pages 219–224, 1999. 35

[94] SARIEL HAR-PELED, DAN ROTH, AND DAV ZIMAK. **Constraint Classification: A New Approach to Multiclass Classification**. In *ALT 2002*, **2533** of *Lecture Notes in Computer Science*, pages 365–379. Springer, 2002. 35

[95] JOHN C. PLATT. **Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods**. In BERNHARD SCHÖLKOPF, CHRISTOPHER J. C. BURGES, AND ALEX SMOLA, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 61–74. Cambridge, MA: MIT Press, 2000. 41

[96] ANDREW P. BRADLEY. **The use of the area under the ROC curve in the evaluation of machine learning algorithms**. *Pattern Recognition*, **30**(7):1145–1159, 1997. 43, 88

[97] ANNA BOSCH, ANDREW ZISSERMAN, AND XAVIER MUÑOZ. **Representing shape with a spatial pyramid kernel**. In *CIVR 2007*, pages 401–408. ACM, 2007. 49, 83, 85

[98] MUHAMMAD A. TAHIR, KOEN VAN DE SANDE, JASPER UIJLINGS, FEI YAN, XIRONG LI, KRYSTIAN MIKOLAJCZYK, JOSEF KITTLER, THEO GEVERS, AND ARNOLD SMEULDERS. **SurreyUVA SRKDA method**. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/workshop/tahir.pdf. 49, 79

[99] KOEN E. A. VAN DE SANDE, THEO GEVERS, AND ARNOLD W. M. SMEULDERS. **The University of Amsterdam's Concept Detection System at ImageCLEF 2009**. In *CLEF 2009(2)* (146), pages 261–268. 49

[100] THORSTEN JOACHIMS. **Making large-Scale SVM Learning Practical**. In B. SCHÖLKOPF, C. BURGES, AND A. SMOLA, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11. MIT Press, Cambridge, MA, 2000. 50

[101] PETER V. GEHLER AND SEBASTIAN NOWOZIN. **On feature combination for multiclass object classification**. In *ICCV 2009* (143), pages 221–228. 60, 79, 80, 95, 96

[102] MIKIO L. BRAUN, JOACHIM M. BUHMANN, AND KLAUS-ROBERT MÜLLER. **On Relevant Dimensions in Kernel Feature Spaces**. *Journal of Machine Learning Research*, **9**:1875–1908, 2008. 63, 92, 104

[103] K. KISHIDA. **Property of average precision and its generalization: an examination of evaluation indicator for information retrieval experiments**. Technical report, National Institute of Informatics, Japan, 2005. 66

[104] ALI FARHADI, IAN ENDRES, DEREK HOIEM, AND DAVID A. FORSYTH. **Describing objects by their attributes**. In *CVPR 2009* (148), pages 1778–1785. 74

[105] YANG WANG AND GREG MORI. **A Discriminative Latent Model of Object Classes and Attributes**. In KOSTAS DANIILIDIS, PETROS MARAGOS, AND NIKOS PARAGIOS, editors, *ECCV (5)*, **6315** of *Lecture Notes in Computer Science*, pages 155–168. Springer, 2010. 74

[106] SUNG JU HWANG, FEI SHA, AND KRISTEN GRAUMAN. **Sharing features between objects and their attributes**. In *CVPR*, pages 1761–1768. IEEE, 2011. 74

[107] MARIA-ELENA NILSBACK AND ANDREW ZISSERMAN. **Delving deeper into the whorl of flower segmentation**. *Image Vision Comput.*, **28**(6):1049–1062, 2010. 78

[108] VLADIMIR VAPNIK. *Statistical learning theory*. Wiley, 1998. 79

[109] ANKITA KUMAR AND CRISTIAN SMINCHISESCU. **Support Kernel Machines for Object Recognition**. In *ICCV* (149), pages 1–8. 79

[110] GERT R. G. LANCKRIET, NELLO CRISTIANINI, PETER L. BARTLETT, LAURENT EL GHAOUI, AND MICHAEL I. JORDAN. **Learning the Kernel Matrix with Semidefinite Programming**. *Journal of Machine Learning Research*, **5**:27–72, 2004. 79

[111] FRANCIS R. BACH, GERT R. G. LANCKRIET, AND MICHAEL I. JORDAN. **Multiple kernel learning, conic duality, and the SMO algorithm**. In Brodley (141). 79, 81

[112] SÖREN SONNENBURG, GUNNAR RÄTSCH, CHRISTIN SCHÄFER, AND BERNHARD SCHÖLKOPF. **Large Scale Multiple Kernel Learning**. *Journal of Machine Learning Research*, **7**:1531–1565, 2006. 79, 81

[113] A. RAKOTOMAMONJY, F. BACH, S. CANU, AND Y. GRANDVALET. **SimpleMKL**. *Journal of Machine Learning Research*, **9**:2491–2521, 2008. 79

[114] C. CORTES, A. GRETTON, G. LANCKRIET, M. MOHRI, AND A. ROSTAMIZADEH. **Proceedings of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels**, 2008. 79

[115] MARIUS KLOFT, ULF BREFELD, PAVEL LASKOV, AND SÖREN SONNENBURG. **Non-Sparse Multiple Kernel Learning**. In *Proc. of the NIPS Workshop on Kernel Learning: Automatic Selection of Kernels*, dec 2008. 79

[116] CORINNA CORTES, MEHRYAR MOHRI, AND AFSHIN ROSTAMIZADEH. **L2 Regularization for Learning Kernels**. In *UAI 2009*, pages 109–116. AUAI Press, 2009. 79

[117] MARIUS KLOFT, ULF BREFELD, SÖREN SONNENBURG, PAVEL LASKOV, KLAUS-ROBERT MÜLLER, AND ALEXANDER ZIEN. **Efficient and Accurate Lp-Norm Multiple Kernel Learning**. In Bengio et al. (142), pages 997–1005. 79, 81

[118] FRANCESCO ORABONA, JIE LUO, AND BARBARA CAPUTO. **Online-batch strongly convex Multi Kernel Learning**. In *CVPR 2010* (152), pages 787–794. 79

[119] ANDREA VEDALDI, VARUN GULSHAN, MANIK VARMA, AND ANDREW ZISSERMAN. **Multiple kernels for object detection**. In *ICCV 2009* (143), pages 606–613. 79

[120] CAROLINA GALLEGUILLOS, BRIAN MCFEEAND SERGE J. BELONGIE, AND GERT R. G. LANCKRIET. **Multi-class object localization by combining local contextual interactions**. In *CVPR 2010* (152), pages 113–120. 79

[121] FEI YAN, JOSEF KITTLER, KRYSTIAN MIKOLAJCZYK, AND MUHAMMAD ATIF TAHIR. **Non-sparse Multiple Kernel Learning for Fisher Discriminant Analysis**. In *ICDM 2009*, pages 1064–1069. IEEE Computer Society, 2009. 80, 111

[122] FEI YAN, KRYSTIAN MIKOLAJCZYK, MARK BARNARD, HONGPING CAI, AND JOSEF KITTLER. ***l*p norm multiple kernel Fisher discriminant analysis for object and image categorisation**. In *CVPR 2010* (152), pages 3626–3632. 80, 98, 111

[123] LIANGLIANG CAO, JIEBO LUO, FENG LIANG, AND THOMAS S. HUANG. **Heterogeneous feature machines for visual recognition**. In *ICCV 2009* (143), pages 1095–1102. 80, 98, 111

[124] MANIK VARMA AND BODLA RAKESH BABU. **More generality in efficient multiple kernel learning**. In *ICML 2009*, **382** of *ACM International Conference Proceeding Series*, page 134. ACM, 2009. 80

[125] MARIUS KLOFT. *ℓ_p-Norm Multiple Kernel Learning*. PhD thesis, Berlin Institute of Technology, Oct 2011. 80

[126] MEHMET GÖNEN AND ETHEM ALPAYDIN. **Multiple Kernel Learning Algorithms**. *Journal of Machine Learning Research*, **12**:2211–2268, 2011. 80, 111

[127] ALAIN RAKOTOMAMONJY, FRANCIS BACH, STÉPHANE CANU, AND YVES GRANDVALET. **More efficiency in multiple kernel learning**. In *ICML 2007* (145), pages 775–782. 81

[128] STEFANIE NOWAK AND MARK J. HUISKES. **New Strategies for Image Annotation: Overview of the Photo Annotation Task at ImageCLEF 2010**. In Braschler et al. (144). 82

[129] NAVNEET DALAL AND BILL TRIGGS. **Histograms of Oriented Gradients for Human Detection**. In *CVPR 2005(1)* (147), pages 886–893. 85

[130] JOHN CANNY. **A Computational Approach to Edge Detection**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**(6):679–698, 1986. 85

[131] PETER V. GEHLER AND SEBASTIAN NOWOZIN. **Let the kernel figure it out; Principled learning of pre-processing for kernel classifiers**. In *CVPR* (148), pages 2836–2843. 95, 98, 100

[132] FRANK MOOSMANN, BILL TRIGGS, AND FRÉDÉRIC JURIE. **Fast Discriminative Visual Codebooks using Randomized Clustering Forests**. In *NIPS 2006*, pages 985–992. MIT Press, 2006. 98

[133] LEO BREIMAN. **Bagging Predictors**. *Machine Learning*, **24**(2):123–140, 1996. 100

[134] MEHMET GÖNEN AND ETHEM ALPAYDIN. **Localized multiple kernel learning**. In *ICML 2008*, **307** of *ACM International Conference Proceeding Series*, pages 352–359. ACM, 2008. 109

[135] JINGJING YANG, YUANNING LI, YONGHONG TIAN, LINGYU DUAN, AND WEN GAO. **Group-sensitive multiple kernel learning for object categorization**. In *ICCV 2009* (143), pages 436–443. 109

[136] MARIUS KLOFT AND GILLES BLANCHARD. **The Local Rademacher Complexity of Lp-Norm Multiple Kernel Learning**. In Shawe-Taylor et al. (156), pages 2438–2446. 110

[137] TAIJI SUZUKI. **Unifying Framework for Fast Learning Rate of Non-Sparse Multiple Kernel Learning**. In Shawe-Taylor et al. (156), pages 1575–1583. 110

[138] S. MIKA, G. RÄTSCH, J. WESTON, B. SCHÖLKOPF, AND K.-R. MÜLLER. **Fisher Discriminant Analysis with Kernels**. In Y.-H. HU, J. LARSEN, E. WILSON, AND S. DOUGLAS, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999. 111

[139] OMKAR M. PARKHI, ANDREA VEDALDI, ANDREW ZISSERMAN, AND C. V. JAWAHAR. **Cats and dogs**. In *CVPR* (157), pages 3498–3505. 113

[140] PEDRO F. FELZENSZWALB, ROSS B. GIRSHICK, DAVID A. MCALLESTER, AND DEVA RAMANAN. **Object Detection with Discriminatively Trained Part-Based Models**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**(9):1627–1645, 2010. 114

[141] CARLA E. BRODLEY, editor. *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, **69** of *ACM International Conference Proceeding Series*. ACM, 2004. 128, 130

[142] YOSHUA BENGIO, DALE SCHUURMANS, JOHN D. LAFFERTY, CHRISTOPHER K. I. WILLIAMS, AND ARON CULOTTA, editors. *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*. Curran Associates, Inc., 2009. 123, 130

[143] *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*. IEEE, 2009. 122, 129, 130, 131

[144] MARTIN BRASCHLER, DONNA HARMAN, AND EMANUELE PIANTA, editors. *CLEF 2010 LABs and Workshops, Notebook Papers, 22-23 September 2010, Padua, Italy*, 2010. 122, 131

[145] *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, **227** of *ACM International Conference Proceeding Series*. ACM, 2007. 125, 131

[146] *Multilingual Information Access Evaluation II. Multimedia Experiments - 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2, 2009, Revised Selected Papers*, **6242** of *Lecture Notes in Computer Science*. Springer, 2010. 126, 127, 129

[147] *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. IEEE Computer Society, 2005. 128, 131

[148] *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE, 2009. 123, 129, 131

[149] *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*. IEEE, 2007. 127, 129

[150] *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*. IEEE Computer Society, 2006. 123, 124

[151] *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008. 124, 127

[152] *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. IEEE, 2010. 123, 130

[153] *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, 2011. 121, 122

[154] *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*. IEEE, 2011. 123, 124, 128

[155] GABOR FICHTINGER, ANNE L. MARTEL, AND TERRY M. PETERS, editors. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2011 - 14th International Conference, Toronto, Canada, September 18-22, 2011, Proceedings, Part III*, **6893** of *Lecture Notes in Computer Science*. Springer, 2011. 124

[156] JOHN SHAWE-TAYLOR, RICHARD S. ZEMEL, PETER L. BARTLETT, FERNANDO C. N. PEREIRA, AND KILIAN Q. WEINBERGER, editors. *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, 2011. 128, 131

[157] *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*. IEEE, 2012. 125, 132