

Advances in Computer Vision and Pattern Recognition



Radu Tudor Ionescu
Marius Popescu

Knowledge Transfer between Computer Vision and Text Mining

Similarity-based Learning Approaches

 Springer

The Springer logo, which is a stylized white chess knight (horse) facing left, positioned to the left of the word "Springer" in a white serif font.

Advances in Computer Vision and Pattern Recognition

Founding editor

Sameer Singh, Rail Vision, Castle Donington, UK

Series editor

Sing Bing Kang, Microsoft Research, Redmond, WA, USA

Advisory Board

Horst Bischof, Graz University of Technology, Austria

Richard Bowden, University of Surrey, Guildford, UK

Sven Dickinson, University of Toronto, ON, Canada

Jiaya Jia, The Chinese University of Hong Kong, Hong Kong

Kyoung Mu Lee, Seoul National University, South Korea

Yoichi Sato, The University of Tokyo, Japan

Bernt Schiele, Max Planck Institute for Computer Science, Saarbrücken, Germany

Stan Sclaroff, Boston University, MA, USA

More information about this series at <http://www.springer.com/series/4205>

Radu Tudor Ionescu · Marius Popescu

Knowledge Transfer between Computer Vision and Text Mining

Similarity-based Learning Approaches

 Springer

Radu Tudor Ionescu
Department of Computer Science
University of Bucharest
Bucharest
Romania

Marius Popescu
Department of Computer Science
University of Bucharest
Bucharest
Romania

ISSN 2191-6586 ISSN 2191-6594 (electronic)
Advances in Computer Vision and Pattern Recognition
ISBN 978-3-319-30365-9 ISBN 978-3-319-30367-3 (eBook)
DOI 10.1007/978-3-319-30367-3

Library of Congress Control Number: 2016932522

© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

To our dear families and friends

Foreword

The present book basically studies similarity-based learning approaches for two different fields: computer vision and string processing. However, the discussed text goes far beyond the goal of a general or even of a comprehensive presentation. From the very beginning, the reader is faced with a genuine scientific challenge: accepting the authors' view according to which image and text can and should be treated in a similar fashion.

Computer vision and string processing seem and are traditionally considered two unrelated fields of study. A question which naturally arises is whether this classical view of the two fields can or should be modified.

While learning from data is a central scientific issue nowadays, no one should claim to be a data analyst without having performed string processing. Information retrieval and extraction ultimately depend on string manipulation. From a different angle, computer vision is concerned with the theory behind artificial systems that extract information from images. One is finally concerned with a goal of the same nature: information acquisition. From this perspective, the approach proposed by the authors seems more natural and indeed scientifically justified.

The authors consider treating images as text and improving text processing techniques with knowledge coming from computer vision. Indeed, corresponding concepts like word and visual word do exist, while the existing literature regards, for instance, modeling object recognition as machine translation. The authors present improved methods that exploit such concepts as well as novel approaches, while broadening the meaning of classical concepts like string processing by taking into account tasks ranging from phylogenetic analysis and DNA sequence alignment to native language identification and text categorization by topic. All in all, the authors gradually build a strong case in favor of the theory they are promoting:

knowledge transfer from one of the studied domains to the other being extremely productive. The very topic of this book represents a scientific challenge, one that the authors master with a great precision and that offers interesting perspectives for future scientific research.

Florentina Hristea

Preface

Machine learning is currently a vast area of research with applications in a broad range of fields such as computer vision, bioinformatics, information retrieval, natural language processing, audio processing, data mining, and many others. Among the variety of state-of-the-art machine learning approaches for such applications are the similarity-based learning methods. Learning based on similarity refers to the process of learning based on pairwise similarities between the training samples. The similarity-based learning process can be both supervised and unsupervised, and the pairwise relationship can be either a similarity, a dissimilarity, or a distance function.

This book studies several similarity-based learning approaches, such as nearest neighbor models, local learning, kernel methods, and clustering algorithms. A nearest neighbor model based on a novel dissimilarity for images is presented in this book. It is used for handwritten digit recognition and achieves impressive results. Kernel methods are used in several tasks investigated in this book. First, a novel kernel for visual word histograms is presented. It achieves state-of-the-art performance for object recognition in images. Several kernels based on a pyramid representation are presented next. They are used for facial expression recognition from static images. The same pyramid representation is successfully used for text categorization by topic. Moreover, an approach based on string kernels for native language identification is also presented in this work. The approach achieves state-of-the-art performance levels, while being language independent and theory neutral. An interesting pattern can already be observed, namely that the machine learning tasks approached in this book can be divided into two different areas: computer vision and string processing.

Despite the fact that computer vision and string processing seem to be unrelated fields of study, image analysis and string processing are in some ways similar. As will be shown by the end of this book, the concept of treating image and text in a similar fashion has proven to be very fertile for specific applications in computer vision. In fact, one of the state-of-the-art methods for image categorization is inspired by the *bag of words* representation, which is very popular in information

retrieval and natural language processing. Indeed, the *bag of visual words* model, which builds a vocabulary of visual words by clustering local image descriptors extracted from images, has demonstrated impressive levels of performance for image categorization and image retrieval. By adapting string processing techniques for image analysis or the other way around, knowledge from one domain can be transferred to the other. In fact, many breakthrough discoveries have been made by transferring knowledge between different domains. This book follows this line of research and presents novel approaches or improved methods that rely on this concept. First of all, a dissimilarity measure for images is presented. The dissimilarity measure is inspired by the rank distance measure for strings. The main concern is to extend rank distance from one-dimensional input (strings) to two-dimensional input (digital images). While rank distance is a highly accurate measure for strings, the empirical results presented in this book suggest that the proposed extension of rank distance to images is very accurate for handwritten digit recognition and texture analysis. Second of all, a kernel that stems from the same idea is also presented in this book. The kernel is designed to encode the spatial information in an efficient way and it shows performance improvements in object class recognition and text categorization by topic. Third of all, some improvements to the popular bag of visual words model are proposed in the present book. As mentioned before, this model is inspired by the bag of words model from natural language processing and information retrieval. A new distance measure for strings is introduced in this work. It is inspired by the image dissimilarity measure based on patches that is also described in the present book. Designed to conform to more general principles and adapted to DNA strings, it comes to improve several state-of-the-art methods for DNA sequence analysis. Furthermore, another application of this novel distance measure for strings is discussed. More precisely, a kernel based on this distance measure is used for native language identification. To summarize, all the contributions presented in this book come to support the concept of treating image and text in a similar manner.

It is worth mentioning that the studied methods exhibit state-of-the-art performance levels in the approached tasks. A few arguments come to support this claim. First of all, an improved bag of visual words model described in this work obtained the fourth place at the Facial Expression Recognition (FER) Challenge of the ICML 2013 Workshop in Challenges in Representation Learning (WREPL). Second of all, the system based on string kernels presented in this book ranked on third place in the closed Native Language Identification Shared Task of the BEA-8 Workshop of NAACL 2013. Third of all, the PQ kernel for visual word histograms described in this work received the Caianiello Best Young Paper Award at ICIAP 2013. Together, these achievements reflect the significance of the methods described in the present book.

Contents

1	Motivation and Overview	1
1.1	Introduction	1
1.2	Knowledge Transfer between Image and Text	2
1.3	Overview and Organization	7
	References	11
2	Learning based on Similarity	15
2.1	Introduction	15
2.2	Nearest Neighbor Approach	17
2.3	Local Learning	20
2.4	Kernel Methods	22
2.4.1	Mathematical Preliminaries	22
2.4.2	Overview of Kernel Classifiers	24
2.4.3	Kernel Functions	26
2.4.4	Kernel Normalization	28
2.4.5	Generic Kernel Algorithm	29
2.4.6	Multiple Kernel Learning	29
2.5	Cluster Analysis	30
2.5.1	K-Means Clustering	32
2.5.2	Hierarchical Clustering	33
	References	35
Part I Knowledge Transfer from Text Mining to Computer Vision		
3	State-of-the-Art Approaches for Image Classification	41
3.1	Introduction	41
3.2	Image Distance Measures	42
3.2.1	Color Image Distances	42
3.2.2	Grayscale Image Distances	44
3.2.3	Earth Mover’s Distance	44
3.2.4	Tangent Distance	44
3.2.5	Shape Match Distance	45

- 3.3 Patch-based Techniques 45
- 3.4 Image Descriptors 46
- 3.5 Bag of Visual Words 47
 - 3.5.1 Encoding Spatial Information 48
- 3.6 Deep Learning 49
- References 50
- 4 Local Displacement Estimation of Image Patches and Textons 53**
 - 4.1 Introduction 53
 - 4.2 Local Patch Dissimilarity 54
 - 4.2.1 Extending Rank Distance to Images 54
 - 4.2.2 Local Patch Dissimilarity Algorithm 56
 - 4.2.3 LPD Algorithm Optimization 58
 - 4.3 Properties of Local Patch Dissimilarity 60
 - 4.4 Experiments and Results 61
 - 4.4.1 Data Sets Description 61
 - 4.4.2 Learning Methods 62
 - 4.4.3 Parameter Tuning 64
 - 4.4.4 Baseline Experiment 67
 - 4.4.5 Kernel Experiment 72
 - 4.4.6 Difficult Experiment 74
 - 4.4.7 Filter-based Nearest Neighbor Experiment 75
 - 4.4.8 Local Learning Experiment 78
 - 4.4.9 Birds Experiment 79
 - 4.5 Local Texton Dissimilarity 81
 - 4.5.1 Texton-based Methods 81
 - 4.5.2 Texture Features 82
 - 4.5.3 Local Texton Dissimilarity Algorithm 83
 - 4.6 Texture Experiments and Results 85
 - 4.6.1 Data Sets Description 86
 - 4.6.2 Learning Methods 88
 - 4.6.3 Brodatz Experiment 89
 - 4.6.4 UIUCTex Experiment 91
 - 4.6.5 Biomass Experiment 95
 - 4.7 Discussion 96
 - References 96
- 5 Object Recognition with the Bag of Visual Words Model 99**
 - 5.1 Introduction 99
 - 5.2 Bag of Visual Words Model 101
 - 5.3 PQ Kernel for Visual Word Histograms 103
 - 5.4 Spatial Non-Alignment Kernel 107
 - 5.4.1 Translation and Size Invariance 109
 - 5.5 Object Recognition Experiments 110
 - 5.5.1 Data Sets Description 112
 - 5.5.2 Implementation and Evaluation Procedure 113

- 5.5.3 PQ Kernel Results on Pascal VOC Experiment 115
- 5.5.4 PQ Kernel Results on Birds Experiment 118
- 5.5.5 SNAK Parameter Tuning 119
- 5.5.6 SNAK Results on Pascal VOC Experiment 120
- 5.5.7 SNAK Results on Birds Experiment 121
- 5.6 Bag of Visual Words for Facial Expression Recognition 122
- 5.7 Local Learning 125
- 5.8 Facial Expression Recognition Experiments 125
 - 5.8.1 Data Set Description 125
 - 5.8.2 Implementation 127
 - 5.8.3 Parameter Tuning and Results 127
- 5.9 Discussion 129
- References 130

Part II Knowledge Transfer from Computer Vision to Text Mining

- 6 State-of-the-Art Approaches for String and Text Analysis 135**
 - 6.1 Introduction 135
 - 6.2 String Distance Measures 136
 - 6.2.1 Hamming Distance 136
 - 6.2.2 Edit Distance 137
 - 6.2.3 Rank Distance 137
 - 6.3 Computational Biology 139
 - 6.3.1 Sequencing and Comparing DNA 139
 - 6.3.2 Phylogenetic Analysis 140
 - 6.4 Text Mining 141
 - 6.4.1 String Kernels 142
 - References 144
- 7 Local Rank Distance 149**
 - 7.1 Introduction 149
 - 7.2 Approach 151
 - 7.3 Local Rank Distance Definition 153
 - 7.4 Local Rank Distance Algorithm 155
 - 7.5 Properties of Local Rank Distance 158
 - 7.6 Local Rank Distance Sequence Aligners 161
 - 7.6.1 Indexing Strategies and Efficiency Improvements 163
 - 7.7 Experiments and Results 165
 - 7.7.1 Data Sets Description 165
 - 7.7.2 Phylogenetic Analysis 167
 - 7.7.3 DNA Comparison 171
 - 7.7.4 Alignment in the Presence of Contaminated Reads 172
 - 7.7.5 Clustering an Unknown Organism 180
 - 7.7.6 Time Evaluation of Sequence Aligners 184
 - 7.7.7 Experiment on Vibrio Species 185

- 7.8 Discussion 187
- References 189
- 8 Native Language Identification with String Kernels 193**
 - 8.1 Introduction 193
 - 8.2 Related Work 195
 - 8.2.1 Native Language Identification 195
 - 8.2.2 Methods that Work at the Character Level 196
 - 8.3 Similarity Measures for Strings 197
 - 8.3.1 String Kernels 197
 - 8.3.2 Kernel based on Local Rank Distance 200
 - 8.4 Learning Methods 200
 - 8.5 Experiments 203
 - 8.5.1 Data Sets Description 203
 - 8.5.2 Parameter Tuning and Implementation Choices 205
 - 8.5.3 Experiment on TOEFL11 Corpus 207
 - 8.5.4 Experiment on ICLE Corpus 210
 - 8.5.5 Experiment on TOEFL11-Big Corpus 211
 - 8.5.6 Cross-Corpus Experiment 213
 - 8.5.7 Experiment on ALC Subset Corpus 214
 - 8.5.8 Experiment on ASK Corpus 217
 - 8.6 Language Transfer Analysis 220
 - 8.7 Discussion 224
 - References 225
- 9 Spatial Information in Text Categorization 229**
 - 9.1 Introduction 229
 - 9.2 Related Work 231
 - 9.3 Methods to Encode Spatial Information 232
 - 9.3.1 Spatial Pyramid for Text 233
 - 9.3.2 Spatial Non-Alignment Kernel for Text 234
 - 9.4 Experiments 236
 - 9.4.1 Data Set Description 236
 - 9.4.2 Implementation Choices 236
 - 9.4.3 Evaluation Procedure 237
 - 9.4.4 Experiment on Reuters-21578 Corpus 238
 - 9.5 Discussion 239
 - References 240
- 10 Conclusions 243**
 - 10.1 Discussion and Conclusions 243
 - References 245
- Index 247**

List of Figures

Figure 1.1	An example in which the context helps to disambiguate an object (kitchen glove), which can easily be mistaken for something else if the rest of the image is not seen. The image belongs to the Pascal VOC 2007 data set. a A picture of a kitchen glove. b A picture of the same glove with context	4
Figure 1.2	An object that can be described by multiple categories such as toy, bear, or both.	5
Figure 2.1	A 3-NN model for handwritten digit recognition. For visual interpretation, digits are represented in a two-dimensional feature space. The figure shows 30 digits sampled from the popular MNIST data set. When the new digit x needs to be recognized, the 3-NN model selects the nearest 3 neighbors and assigns label 4 based on a majority vote	18
Figure 2.2	A 1-NN model for handwritten digit recognition. The figure shows 30 digits sampled from the popular MNIST data set. The decision boundary of the 1-NN model generates a Voronoi partition of the digits.	18
Figure 2.3	Two classification models are used to solve the same binary classification problem. The two test samples depicted in <i>red</i> are misclassified by a global linear classifier (<i>left-hand</i> side). The local learning framework produces a nonlinear decision boundary that fixes this problem (<i>right-hand</i> side). a A global linear classifier misclassifies the test samples depicted in <i>red</i> . b A local learning model based on an underlying linear classifier is able to correctly classify the test samples depicted in <i>red</i>	21

Figure 2.4 The function ϕ embeds the data into a feature space where the nonlinear relations now appear linear. Machine learning methods can easily detect such linear relations 24

Figure 4.1 Two images that are compared with LPD. **a** For every position $(x_1; y_1)$ in the first image, LPD tries to find a similar patch in the second image. First, it looks at the same position $(x_1; y_1)$ in the second image. The patches are not similar. **b** LPD gradually looks around position $(x_1; y_1)$ in the second image to find a similar patch. **c** LPD sum up the spatial offset between the similar patches at $(x_1; y_1)$ from the first image and $(x_2; y_2)$ from the second image 57

Figure 4.2 A random sample of 15 handwritten digits from the MNIST data set. 62

Figure 4.3 A random sample of 12 images from the Birds data set. There are two images per class. Images from the same class sit next to each other in this figure. 63

Figure 4.4 Average accuracy rates of the 3-NN based on LPD model with patches of 1×1 pixels at the *top* and 2×2 pixels at the *bottom*. Experiment performed on the MNIST subset of 100 images. **a** Accuracy rates with patches of 1×1 pixels. **b** Accuracy rates with patches of 2×2 pixels. 65

Figure 4.5 Average accuracy rates of the 3-NN based on LPD model with patches of 3×3 pixels at the *top* and 4×4 pixels at the *bottom*. Experiment performed on the MNIST subset of 100 images. **a** Accuracy rates with patches of 3×3 pixels. **b** Accuracy rates with patches of 4×4 pixels. 66

Figure 4.6 Average accuracy rates of the 3-NN based on LPD model with patches of 5×5 pixels at the *top* and 6×6 pixels at the *bottom*. Experiment performed on the MNIST subset of 100 images. **a** Accuracy rates with patches of 5×5 pixels. **b** Accuracy rates with patches of 6×6 pixels. 67

Figure 4.7 Average accuracy rates of the 3-NN based on LPD model with patches of 7×7 pixels at the *top* and 8×8 pixels at the *bottom*. Experiment performed on the MNIST subset of 100 images. **a** Accuracy rates with patches of 7×7 pixels. **b** Accuracy rates with patches of 8×8 pixels. 68

Figure 4.8 Average accuracy rates of the 3-NN based on LPD model with patches of 9×9 pixels at the *top* and 10×10 pixels at the *bottom*. Experiment performed on the MNIST subset of 100 images. **a** Accuracy rates with patches of 9×9 pixels. **b** Accuracy rates with patches of 10×10 pixels 69

Figure 4.9 Average accuracy rates of the 3-NN based on LPD model with patches ranging from 2×2 pixels to 9×9 pixels. Experiment performed on the MNIST subset of 300 images 70

Figure 4.10 Similarity matrix based on LPD with patches of 4×4 pixels and a similarity threshold of 0.12, obtained by computing pairwise dissimilarities between the samples of the MNIST subset of 1000 images 72

Figure 4.11 Euclidean distance matrix based on L_2 -norm, obtained by computing pairwise distances between the samples of the MNIST subset of 1000 images 73

Figure 4.12 Error rate drops as K increases for 3-NN (\circ) and 6-NN (\diamond) classifiers based on LPD with filtering 77

Figure 4.13 Sample images from three classes of the Brodatz data set. 87

Figure 4.14 Sample images from four classes of the UIUCTex data set. Each image is showing a textured surface viewed under different poses. **a** Bark. **b** Pebbles. **c** Brick. **d** Plaid 88

Figure 4.15 Sample images from the biomass texture data set. **a** Wheat. **b** Waste. **c** Corn 89

Figure 4.16 Similarity matrix based on LTD with patches of 32×32 pixels and a similarity threshold of 0.02, obtained by computing pairwise dissimilarities between the texture samples of the Brodatz data set 92

Figure 4.17 Similarity matrix based on LTD with patches of 64×64 pixels and a similarity threshold of 0.02, obtained by computing pairwise dissimilarities between the texture samples of the UIUCTex data set. 94

Figure 5.1 The BOVW learning model for object class recognition. The feature vector consists of SIFT features computed on a regular grid across the image (dense SIFT) and vector quantized into visual words. The frequency of each visual word is then recorded in a histogram. The histograms enter the training stage. Learning is done by a kernel method 102

Figure 5.2 The spatial similarity of two images computed with the SNAK framework. First, the center of mass is computed according to the objectness map. The average position and the standard deviation of the spatial distribution of each visual word are computed next. The images are aligned according to their centers, and the SNAK kernel is computed by summing the distances between the average positions and the standard deviations of each visual word in the two images 111

Figure 5.3 A random sample of 12 images from the Pascal VOC data set. Some of the images contain objects of more than one class. For example, the image at the *top left* shows a dog sitting on a couch, and the image at the *top right* shows a person and a horse. Dog, couch, person, and horse are among the 20 classes of this data set 112

Figure 5.4 A random sample of 12 images from the Birds data set. There are two images per class. Images from the same class sit next to each other in this figure. 113

Figure 5.5 The BOVW learning model for facial expression recognition. The feature vector consists of SIFT features computed on a regular grid across the image (dense SIFT) and vector quantized into visual words. The presence of each visual word is then recorded in a presence vector. Normalized presence vectors enter the training stage. Learning is done by a local kernel method 124

Figure 5.6 An example of SIFT features extracted from two images representing distinct emotions: fear (*left*) and disgust (*right*) 125

Figure 5.7 The six nearest neighbors selected with the presence kernel from the vicinity of the test image are visually more similar than the other six images randomly selected from the training set. Despite of this fact, the nearest neighbors do not adequately indicate the test label (disgust). Thus, a learning method needs to be trained on the selected neighbors to accurately predict the label of the test image. 126

Figure 7.1 Phylogenetic tree obtained for 22 mammalian mtDNA sequences using LRD based on 2-mers. 168

Figure 7.2 Phylogenetic tree obtained for 22 mammalian mtDNA sequences using LRD based on 4-mers. 168

Figure 7.3 Phylogenetic tree obtained for 22 mammalian mtDNA sequences using LRD based on 6-mers. 169

Figure 7.4 Phylogenetic tree obtained for 22 mammalian mtDNA sequences using LRD based on 8-mers. 169

Figure 7.5 Phylogenetic tree obtained for 22 mammalian mtDNA sequences using LRD based on 10-mers. 170

Figure 7.6 Phylogenetic tree obtained for 22 mammalian mtDNA sequences using LRD based on sum of k -mers 170

Figure 7.7 Phylogenetic tree obtained for 27 mammalian mtDNA sequences using LRD based on 18-mers. 171

Figure 7.8 The distance evolution of the best chromosome at each generation for the rat–mouse–cow experiment. The *green line* represents the rat–house mouse (RH) distance, the *blue line* represents the rat–fat dormouse (RF) distance, and the *red line* represents the rat–cow (RC) distance. 173

Figure 7.9 The precision–recall curves of the state-of-the-art aligners versus the precision–recall curves of the two LRD aligners, when 10,000 contaminated reads of length 100 from the orangutan are included. The two variants of the BOWTIE aligner are based on local and global alignment, respectively. The LRD aligner based on hash tables is a fast approximate version of the original LRD aligner 175

Figure 7.10 The precision–recall curves of the state-of-the-art aligners versus the precision–recall curves of the two LRD aligners, when 50,000 contaminated reads of length 100 from 5 mammals are included. The two variants of the BOWTIE aligner are based on local and global alignment, respectively. The LRD aligner based on hash tables is a fast approximate version of the original LRD aligner 178

Figure 7.11 Local Rank Distance computed in the presence of different types of DNA changes such as point mutations, indels, and inversions. In the first three cases **a–c**, a single type of DNA polymorphism is included in the second (*bottom*) string. The last case **d** shows how LRD measures the differences between the two DNA strings when all the types of DNA changes occur in the second string. The nucleotides affected by changes are marked with bold. To compare the results for the different types of DNA changes, the first string is always the same in all the four cases. Note that in all the four examples, LRD is based on 1-mers. In each case, $\Delta_{LRD} = \Delta_{left} + \Delta_{right}$. **a** Measuring LRD with point mutations. The *T* at index 7 is substituted with *C*. **b** Measuring LRD with indels. The substring *GT* is deleted. **c** Measuring LRD with inversions. The substring *AGTT* is inverted. **d** Measuring LRD with point mutations, indels, and inversions 188

Figure 8.1 An example with three classes that illustrates the masking problem. Class A is masked by classes B and C. 203

List of Tables

Table 4.1	Results of the experiment performed on the MNIST subset of 300 images, using the 3-NN based on LPD model with patches ranging from 2×2 pixels to 9×9 pixels.	70
Table 4.2	Results of the experiment performed on the MNIST subset of 300 images, using various maximum offsets, patches of 4×4 pixels, and a similarity threshold of 0.12.	71
Table 4.3	Baseline 3-NN versus 3-NN based on LPD	71
Table 4.4	Accuracy rates of several classifiers based on LPD versus the accuracy rates of the standard SVM and KRR.	73
Table 4.5	Comparison of several classifiers (some based on LPD).	74
Table 4.6	Error and time of the 3-NN classifier based on LPD with filtering, for various K values.	76
Table 4.7	Confusion matrix of the 3-NN based on LPD with filtering using $K = 50$	78
Table 4.8	Error rates on the entire MNIST data set for baseline 3-NN, k -NN based on Tangent distance, and k -NN based on LPD with filtering	78
Table 4.9	Error rates of different k -NN models on Birds data set	80
Table 4.10	Error on Birds data set for texton learning methods of Lazebnik et al. (2005a) and kernel methods based on LPD	80
Table 4.11	Accuracy rates on the Brodatz data set using 3 random samples per class for training	90
Table 4.12	Accuracy rates of several MKL approaches that include LTD compared with state-of-the-art methods on the Brodatz data set	90
Table 4.13	Accuracy rates on the UIUCTex data set using 20 random samples per class for training	93
Table 4.14	Accuracy rates of several MKL approaches that include LTD compared with state-of-the-art methods on the UIUCTex data set.	93

Table 4.15 Accuracy rates on the Biomass Texture data set using 20, 30 and 40 random samples per class for training and 70, 60 and 50 for testing, respectively. 95

Table 5.1 Mean AP on Pascal VOC 2007 data set for SVM based on different kernels 115

Table 5.2 Mean AP on the 20 classes of the Pascal VOC 2007 data set for the SVM classifier based on 3000 visual words using the spatial pyramid representation and different kernels 117

Table 5.3 Running time required by each kernel to compute the two kernel matrices for training and testing, respectively 117

Table 5.4 Classification accuracy on the Birds data set for SVM based on different kernels 118

Table 5.5 Mean AP on Pascal VOC 2007 data set for different representations that encode spatial information into the BOVW model 120

Table 5.6 Classification accuracy on the Birds data set for different representations that encode spatial information into the BOVW model 122

Table 5.7 Accuracy levels for several BOVW models obtained on the FER validation, test, and private test sets. 128

Table 7.1 The 27 mammals from the EMBL database used in the phylogenetic experiments. 166

Table 7.2 The genomic sequence information of three vibrio pathogens consisting of two circular chromosomes 167

Table 7.3 The number of misclustered mammals for different clustering techniques on the 22 mammals data set. 171

Table 7.4 Closest string results for the genetic algorithm based on LRD with 3-mers 172

Table 7.5 Several statistics of the state-of-the-art aligners versus the LRD aligner, when 10,000 contaminated reads of length 100 sampled from the orangutan genome are included. 176

Table 7.6 Metrics of the human reads mapped to the human mitochondrial genome (true positives) by the hash LRD aligner versus the human reads that are not mapped to the genome (false negatives) 176

Table 7.7 Several statistics of the state-of-the-art aligners versus the LRD aligner, when 50,000 contaminated reads of length 100 sampled from the genomes of five mammals are included 178

Table 7.8 The recall at best precision of the state-of-the-art aligners versus the LRD aligner, when 10,000 contaminated reads of length 100 sampled from the orangutan genome are included 179

Table 7.9	The recall at best precision of the state-of-the-art aligners versus the LRD aligner, when 40,000 contaminated reads of length 100 sampled from the blue whale, the harbor seal, the donkey, and the house mouse genomes are included, respectively	179
Table 7.10	The results for the real-world setting experiment on mammals	182
Table 7.11	The results for the hard setting experiment on mammals	183
Table 7.12	The running times of the BWA aligner, the BLAST aligner, the BOWTIE aligner, and the LRD aligner	184
Table 7.13	The results of the rank-based aligner on vibrio species	186
Table 8.1	Summary of corpora used in the experiments	203
Table 8.2	Distribution of the documents per native language in the ALC subset	204
Table 8.3	Average word length and optimal p -gram range for the TOEFL11 corpus (English L2), the ALC subset (Arabic L2), and the ASK corpus (Norwegian L2)	206
Table 8.4	Accuracy rates on TOEFL11 corpus (English L2) of various classification systems based on string kernels compared with other state-of-the-art approaches	208
Table 8.5	Accuracy rates on the raw text documents of the TOEFL11 corpus (English L2) of various classification systems based on string kernels	209
Table 8.6	Accuracy rates on ICLEv2 corpus (English L2) of various classification systems based on string kernels compared with a state-of-the-art approach.	211
Table 8.7	Accuracy rates on TOEFL11-Big (English L2) corpus of various classification systems based on string kernels compared with a state-of-the-art approach	212
Table 8.8	Accuracy rates on TOEFL11-Big corpus (English L2) of various classification systems based on string kernels compared with a state-of-the-art approach	213
Table 8.9	Accuracy rates on ALC subset (Arabic L2) of various classification systems based on string kernels compared with a state-of-the-art approach.	215
Table 8.10	Accuracy rates on three subsets of five languages of the ASK corpus (Norwegian L2) of various classification systems based on string kernels compared with a state-of-the-art approach.	218
Table 8.11	Accuracy rates on the ASK corpus (Norwegian L2) of various classification systems based on string kernels	220

Table 8.12	Examples of discriminant overused character sequences with their ranks (left) according to the KRR model based on blended spectrum presence bits kernel extracted from the TOEFL11 corpus (English L2)	222
Table 8.13	Examples of discriminant underused character sequences (ranks omitted for readability) according to the KRR model based on blended spectrum presence bits kernel extracted from the TOEFL11 corpus (English L2)	223
Table 9.1	Confusion matrix (also known as contingency table) of a binary classifier with labels +1 or -1	237
Table 9.2	Empirical results on the Reuters-21578 corpus obtained by the standard bag of words versus two methods that encode spatial information, namely spatial pyramids and SNAK	238

Chapter 1

Motivation and Overview

1.1 Introduction

Machine learning is a branch of artificial intelligence that studies computer systems that can learn from data. In this context, learning is about recognizing complex patterns and making intelligent decisions based on data. In the early years of artificial intelligence, the idea that human thinking could be rendered logically in a numerical computing machine emerged, but it was unclear if such a machine could model the complex human brain, until Alan Turing proposed a test to measure its performance in 1950. The Turing test states that a machine exhibits human-level intelligence if a human judge engages in a natural language conversation with the machine and cannot distinguish it from another human. Despite the fact that intelligent machines that can pass the Turing test have not been developed yet, many interesting and useful systems that can learn from data have been proposed since then.

Several learning paradigms have been proposed in the context of machine learning. The two most popular ones are supervised and unsupervised learning. *Supervised learning* refers to the task of building a classifier using labeled training data. The most studied approaches in machine learning are supervised and they include Support Vector Machines (Cortes and Vapnik 1995), Naïve Bayes classifiers (Manning et al. 2008), neural networks (Bishop 1995; Krizhevsky et al. 2012; LeCun et al. 2015), Random Forests (Breiman 2001), and many others (Caruana and Niculescu-Mizil 2006). *Unsupervised learning* refers to the task of finding hidden structure in unlabeled data. The best known form of unsupervised learning is *cluster analysis*, which aims at clustering objects into groups based on their similarity. Among the other learning paradigms are *semi-supervised learning*, which combines both labeled and unlabeled data, and *reinforcement learning*, which learns to take actions in an environment in order to maximize a long-term reward. Depending on the desired outcome of the machine learning algorithm or on the type of training input available for an application, a particular learning paradigm may be more suitable than the others.

Machine learning is currently a vast area of research with applications in a broad range of fields, such as computer vision (Fei-Fei and Perona 2005; Forsyth and Ponce 2002; Sebastiani 2002; Zhang et al. 2007), bioinformatics (Dinu and Ionescu 2013; Inza et al. 2010; Leslie et al. 2002) information retrieval (Chifu and Ionescu 2012; Ionescu et al. 2015b; Manning et al. 2008), natural language processing (Lodhi et al. 2002; Popescu and Grozea 2012; Sebastiani 2002), and many others (Ionescu et al. 2015a). Among the variety of state-of-the-art machine learning approaches for such applications are the similarity-based learning methods (Chen et al. 2009).

This book studies similarity-based learning approaches such as nearest neighbor models, kernel methods (Shawe-Taylor and Cristianini 2004), and clustering algorithms. The studied approaches have interesting applications and exhibit state-of-the-art performance levels in two different areas: computer vision and string processing. It is important to note that, in this book, *string processing* refers to any task that needs to process string data such as text documents, DNA sequences, and so on. This work investigates string processing tasks ranging from phylogenetic analysis (Ionescu 2013) and sequence alignment (Dinu et al. 2014) to native language identification (Ionescu et al. 2014b; Popescu and Ionescu 2013) and text categorization by topic, from a machine learning perspective. These tasks belong to one of two separate fields, namely text mining or computational biology, but they are gathered under one umbrella called string processing. On the other hand, a broad variety of computer vision tasks are also investigated in this book, including object recognition (Ionescu and Popescu 2013b, 2015a, b), facial expression recognition (Ionescu et al. 2013), optical character recognition (Dinu et al. 2012; Ionescu and Popescu 2013a) and texture classification (Ionescu et al. 2014a). While all the topics enumerated so far seem to be unrelated, each and every one of them includes at least a concept that is borrowed from the other fields of study covered by this book. Further details about this transfer of knowledge between domains are given in the following section. Before going into the next section, it is worth mentioning that the core part of this book is mostly based on recently published works by the authors, yet, it also includes (previously) unpublished work and results.

1.2 Knowledge Transfer between Image and Text

In recent years, computer science specialists are faced with the challenge of processing massive amounts of data. The largest part of this data is actually unstructured and semi-structured data, available in the form of text documents, images, audio files, video files, and so on. Researchers have developed methods and tools that extract relevant information and support efficient access to unstructured and semi-structured content. Such methods that aim at providing access to information are mainly studied by researchers in machine learning and related fields. In fact, a tremendous amount of effort has been dedicated to this line of research (Agarwal and Roth 2002; Lazebnik et al. 2005, 2006; Leung and Malik 2001; Manning et al. 2008). In the context of machine learning, the aim is to obtain a good representation of the data that can later

be used to build an efficient classifier. In computer vision, image representations are obtained by *feature detection* and *feature extraction*. Most of the feature extraction methods are handcrafted by researchers that have a good understanding of the application and a vast experience. This is the case of the bag of visual words model (Leung and Malik 2001; Sivic et al. 2005) in computer vision. A different approach is *representation learning*, which aims at discovering a better representation of the data provided during training. This is the case of deep learning algorithms (Bengio 2009; LeCun et al. 2015; Montavon et al. 2012) that aim at discovering multiple levels of representation, or a hierarchy of features. Deep algorithms learn to transform one representation into another, by better disentangling the factors of variation that explain the observed data.

Whether the representation of the data is obtained through a handcrafted method or learned by a fully automatic process, common concepts of treating different kinds of unstructured and semi-structured data, such as image and text, naturally arise. Despite the fact that computer vision and string processing seem to be unrelated fields of study, the concept of treating image and text in a similar fashion has proven to be very fertile for several applications. Furthermore, by adapting string processing techniques to image analysis or the other way around, knowledge from one domain can be transferred to the other.

An example of similarity between text and image is discussed next. It refers to word sense disambiguation and object recognition in images. *Word sense disambiguation* (WSD) is a core research problem in computational linguistics and natural language processing, which was recognized since the beginning of the scientific interest in machine translation, and in artificial intelligence, in general. WSD is about determining the meaning of a word in a specific context. Actually all the WSD methods use the context to determine the meaning of an ambiguous word, because the entire information about the word sense is contained in the context (Agirre and Edmonds 2006). The basic concept is to extract features from the context that could help the WSD process. In a similar fashion, an object in an image can be recognized using the entire image as a context. For example, a method that could detect the presence of a kitchen glove in the image would have to look for distinctive features such as the texture of the material, the shape, and perhaps even the color. However, there could be other objects that have similar shape or color, and in more difficult situations, such as illustrated in Fig. 1.1a, it may be almost impossible to distinguish the glove. Thus, a better approach could be to look for other distinctive features in the image provided by the context. For instance, a human can easily figure out that a glove is hanging by a kitchen cabinet knob in the scene illustrated in Fig. 1.1b. It is easier to understand the entire scene as a whole than taking the glove out of context. In conclusion, the idea of using the context can help to avoid any confusion. Not surprisingly, this intuitive idea has already been studied in the computer vision literature (Galleguillos and Belongie 2010; Rabinovich et al. 2007). In (Rabinovich et al. 2007), the semantic context is incorporated into object categorization to reduce ambiguity in objects' visual appearance and to improve accuracy. The paper of (Galleguillos and Belongie 2010) goes even further and makes a distinction between three types of context, namely semantic context, spatial context, and scale context.

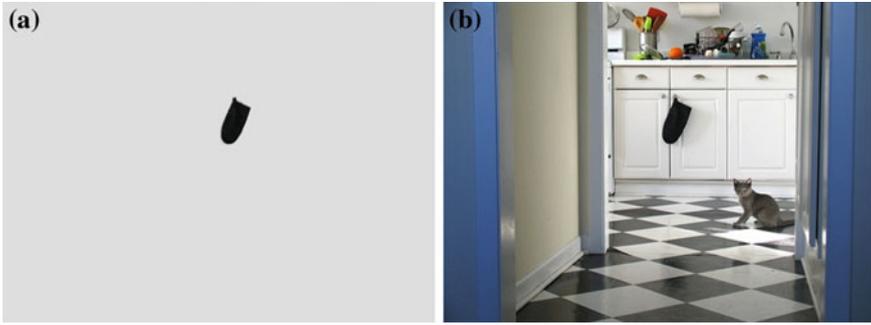


Fig. 1.1 An example in which the context helps to disambiguate an object (kitchen glove), which can easily be mistaken for something else if the rest of the image is not seen. The image belongs to the Pascal VOC 2007 data set. **a** A picture of a kitchen glove. **b** A picture of the same glove with context

Another example of treating image and text in a similar manner is a state-of-the-art method for image categorization and image retrieval inspired by the *bag of words* representation, which is very popular in information retrieval and natural language processing. The bag of words model represents a text as an unordered collection of words, completely disregarding grammar, word order, and syntactic groups. The bag of words model has many applications from information retrieval (Manning et al. 2008) to natural language processing (Manning and Schütze 1999) and word sense disambiguation (Agirre and Edmonds 2006; Chifu and Ionescu 2012). In the context of image analysis, the concept of *word* needs to be somehow defined. Computer vision researchers have introduced the concept of *visual word*. Local image descriptors, such as SIFT (Lowe 1999), are vector quantized to obtain a vocabulary of visual words. The vector quantization process can be done, for example, by k-means clustering (Leung and Malik 2001) or by probabilistic Latent Semantic Analysis (Sivic et al. 2005). The frequency of each visual word is then recorded in a histogram which represents the final feature vector for the image. This histogram is the equivalent of the bag of words representation for text. The idea of representing images as *bag of visual words* has demonstrated very good performance for image categorization (Zhang et al. 2007) and image retrieval (Philbin et al. 2007).

One of the most important problems in computer vision is object recognition. Machine learning methods represent the state-of-the-art approach for the object recognition problem. A common approach is to make some assumptions in order to treat object recognition as a classification problem. First, object categories are considered to be fixed and known. Second, each instance belongs to a single category. However, some researchers argue that these assumptions do not adequately describe the reality. The following example shows that these assumptions are indeed wrong. The object presented in Fig. 1.2 can be described either as a toy, a bear, or both. It is clear that the object does not belong to a single category. Furthermore, the category of the object might be irrelevant for particular applications. Another