Advances in Computer Vision and Pattern Recognition

Hong Cheng

Sparse Representation, Modeling and Learning in Visual Recognition

Theory, Algorithms and Applications

Springer

Advances in Computer Vision and Pattern Recognition

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

Hong Cheng

Sparse Representation, Modeling and Learning in Visual Recognition

Theory, Algorithms and Applications

Springer

Hong Cheng
University of Electronic Science
and Technology of China
Chengdu, Sichuan
China

ISSN 2191-6586 ISSN 2191-6594 (electronic)
Advances in Computer Vision and Pattern Recognition
ISBN 978-1-4471-6713-6 ISBN 978-1-4471-6714-3 (eBook)
DOI 10.1007/978-1-4471-6714-3

Library of Congress Control Number: 2015938733

Springer London Heidelberg New York Dordrecht

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

Springer-Verlag London Ltd. is part of Springer Science+Business Media (www.springer.com)

Preface

Over the past decade, sparse representation, modeling, and learning has emerged and is widely used in many visual tasks such as feature extraction and learning, object detection, and recognition (i.e., faces, activities). It is rooted in statistics, physics, information theory, neuroscience, optimization theory, algorithms, and data structure. Meanwhile, visual recognition has played a critical role in computer vision as well as in robotics. Recently, sparse representation consists of two basic tasks, data sparsification and encoding features. The first task is to make data more sparse directly. The second is to encode features with sparsity properties in some domain using either strictly or approximately K-Sparsity. Sparse modeling is to model specific tasks by jointly using different disciplines and their sparsity properties. Sparse learning is to learn mapping from input signals to outputs by either representing the sparsity of signals or modeling the sparsity constraints as regularization items in optimization equation. Mathematically, solving sparse representation and learning involves seeking the sparsest linear combination of basic functions from an overcomplete dictionary. The rationale behind this is the sparse connectivity between nodes in the human brain.

The necessity and popularity of sparse representation, modeling, and learning are spread over all major universities and research organizations around the world, with leading scientists from various disciplines. This book presents our recent research work on sparse representation, modeling and learning with emphasis on visual recognition, and is aimed at researchers and graduate students. Our goal in writing this book is threefold. First, it creates an updated reference book of sparse representing, modeling, and learning. Second, this book covers both the theory and application aspects, which benefits readers keen to learn broadly sparse representation, modeling, and learning. Finally, we have provided some applications about visual recognition, as well as some applications about computer vision. We try to link theory, algorithms, and applications to promote compressed sensing research.

This book is divided into four parts. The first part, Introduction and Fundamentals, presents the research motivation and purpose, and briefly introduces the definition of spares representation, modeling, and learning, as well as its applications on visual recognition. The second part, Sparse representation, Modeling and Learning, which includes sparse recovery approaches, robust sparse representation and learning, efficient sparse representation and modeling, introduces large-scale visual recognition, and situations of efficient sparse coding and sparse quantization. The third part, Visual Recognition Applications, which includes feature representation and learning, sparsity-induced similarity, and sparse representation and learning-based classifiers, is the result of combining pattern recognition and compressed sensing. In different SRLCs, sparsity could be used in sample similarity, histogram generation, local feature similarity, and nearest neighbor classifiers. The fourth part, Advanced Topics, discusses the topic beyond the sparse—low-rank representation which is known as two-dimensional sparse representation. Additionally, Mathematics fundamental, and Computer Programming Resources are included in the appendices.

Most of this book refers to our research work at University of Electronic Science and Technology of China, and Carnegie Mellon University. I would like to offer my deep respect to Jie Yang at Carnegie Mellon University who supported my research during my stay at this university, as well as Zicheng Liu at Microsoft Research for his deep discussions with me. I would like to express my sincere thanks to Nan Zhou, Jianmei Su, Yuzhuo Wang, Ratha Pech, Saima who have provided immense help with preparation of figures and with the typesetting of the book. Springer has provided great support throughout the final stages of this book. I would like to thank my commissioning editor, Simon Rees, for his support. Finally, I would like to thank my wife, Xiaohong Feng, who has been supportive of me to write this book.

Chengdu, China
February 2015

Hong Cheng

Contents

Part I Introduction and Fundamentals

Part II Sparse Representation, Modeling and Learning

Part III Visual Recognition Applications

Part IV Advanced Topics

Mathematical Notation

$\mathbb{R}$	Real numbers
$\mathbb{R}^D$	Vector space of real valued D dimensional vectors
$\mathbb{R}^D_K$	The space of K-sparse vector in $\mathbb{R}^D$
$\mathbb{B}^D_K$	The space of binary K-sparse vector built from $\{0, 1\}$
$\mathbb{T}^D_K$	The space of K-sparse vector built from $\{-1, 0, 1\}$
$\boldsymbol{x}$	(Sparse)sampling coefficients
x_i	ith element of vector $\boldsymbol{x}$
$\lvert \cdot \rvert$	If applied to a number, absolute value
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	The inner product of $\boldsymbol{x}, \boldsymbol{y}$, $\sum_i x_i y_i$
$\boldsymbol{y} = A\boldsymbol{x}$	Linear system equation
$\boldsymbol{y} \in \mathbb{Y}$	Sample/observation and sample/observation space
A	Dictionary/codebook
$A_{i.}$	ith row of matrix A
$A_{.j}$	jth column of matrix A
A_{ij}	The ith row jth column element of matrix A
I_n	Identity matrix of size n
$E(\cdot)$	Expected value of a random variable
$N(\boldsymbol{\mu}, \Sigma)$	Gaussian with mean μ and covariance Σ
N	The number of feature vectors
D	The number of feature vector dimensions
$\lVert \mathbf{x} \rVert_{\ell_p}$	For a vector ℓ_p-norm or ℓ_p seminorm defined as $(\sum_i \lvert x_i \rvert^p)^{\frac{1}{p}}$
$\lVert \boldsymbol{x} \rVert_{\ell_0}$	ℓ_0-norm of a vector. Number of nonzero elements in $\boldsymbol{x}$
$(a)_+$	a if $a > 0$ and zero otherwise
$\mathrm{sgn}(\cdot)$	Sign of a number
K	K-sparse
C	The number of classes
c	The index of a class

λ	Lagrange multiplier
$(.)^T$	Matrix transpose
$(.)^\dagger$	The Moore-Penrose pseudoinverse
e, ε	Noise term and its $\ell_2 - norm$
$\mathbf{1}$	A vector with all the elements are 1 $([1, 1, \cdots, 1]^T)$

Part I
Introduction and Fundamentals

Chapter 1
Introduction

1.1 Sparse Representation, Modeling, and Learning

1.1.1 Sparse Representation

Sparse representation consists of two basic tasks, data sparsification and encoding features. The first task is to make data more sparse directly. Moreover, data sparsification is to project original data into a potentially either same-dimensional or higher-dimensional latent space, which guarantees the minimum distance between before-projection and after-projection features [18]. The second task is to encode features with sparsity properties in some domain using either strictly or approximately K-sparsity. First of all, sometimes features are either sparse or compressible in nature. Sparse features mean that only K-coefficients have large magnitude and others are zero. K is much smaller than the dimension of coefficients vector. In other words, the coefficients vector has only K nonzero entries. Compressible features mean that the coefficient vectors in certain codewords are composed of a few large coefficients and other coefficients are of small value. Of course, the compressible features are not sparse ones. However,if we set small coefficients to zero, the remaining large coefficients can represent the original features with certain loss, called sparse quantization. For example,the gradient of a piecewise constant signal is sparse in the time domain. Natural signals or image are either sparse or compressible in the discrete cosine transform (DCT) domain. More generally, most signals/data are dense, such as image intensity, scene depth mapping, and action/gesture trajectories. However, even dense signals/features could be sparse in another domain. The relationship of sparsity between different domains follows the fundamental law of signal resolution for sparse signal representation, uncertainty principal (UP). This is similar to Heigenberg's UP in quantum mechanics, and a fundamental limit to the sparsity properties of data in different domains. Moreover, if signals/features are sparse but not strictly K-sparse, we can use K-sparse to approximate those signals/features, called sparse quantization. Sparse quantization is a basic way to code signals/features for

H. Cheng, *Sparse Representation, Modeling and Learning in Visual Recognition*,
Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-1-4471-6714-3_1

efficient representation and modeling, such as for patch description [8] and visual recognition [9].

1.1.2 Sparse Modeling

Sparse Modeling is to model specific tasks (e.g., visual recognition tasks) by jointly using different disciplines and their sparsity properties. This is rooted in statistics, physics, information theory, neuroscience, optimization theory, algorithm and data structure, matrix theory, machine learning, and signal processing, shown in Fig. 1.1.

There are many applications of sparse modeling, such as regression [54, 66, 80], classification tasks [14, 78], graphical model selection [3, 50, 56], sparse M-estimators [52, 57], and sparse dimensionality reduction. Sparse modeling is a particularly important issue in many applications of machine learning and statistics where the main objective is to discover predictive patterns in data which would enhance our understanding of underlying physical, biological, and other natural processes, beyond just building accurate 'black-box' predictors. Common examples include biomarker selection in biological applications [80], finding brain areas predictive about 'brain states' based on fMRI data [12], and identifying network bottlenecks best explaining end-to-end performance [15]. Moreover, efficient recovery of high-dimensional sparse signals from a relatively small number of observations is the main focus of compressed sensing [11, 22, 41], and also is a rapidly growing and extremely popular area of signal processing.

More interestingly, sparse modeling is directly related to various computer vision tasks, such as image separation [6], image restoration and denoising [23], face recognition [73, 79], image superresolution [43, 76], recommendation systems [5], EEG analysis [19], text classification [6, 46], subspace methods [25], label propagation [17], and human activity recognition [16].

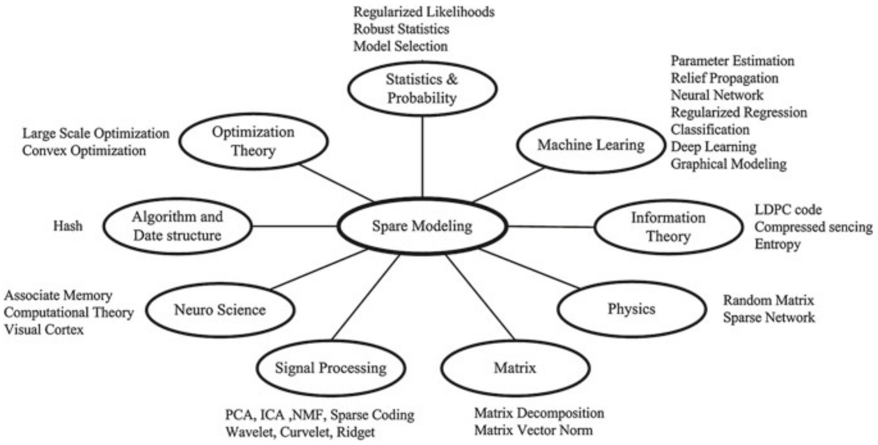

Fig. 1.1 The corresponding knowledge of sparse modeling

1.1.3 Sparse Learning

Sparse learning is to learn mappings from input signals/features to outputs by either representing the sparsity of signals/features or modeling the sparsity constraints as regularization items in optimization equations. Given a specific application, sparse learning is based on both sparse modeling and sparse representation. In general, sparse learning has two branches *supervised sparse learning* and *unsupervised sparse learning*. The most famous work in sparse unsupervised learning is the *sparse feature learning*, such as dictionary learning [40, 49, 68], feature learning and matrix factorization [21, 61, 70, 81]. By adding the sparse constraints in feature learning, it can extract the representative features. Another work in sparse unsupervised learning is *sparse subspace clustering* [24, 25], which uses the sparse representation to cluster the data and get extremely good results. The *supervised sparse learning* is to learn the parameters with the training data with labels which can be used for classification.

The most famous work in *supervised sparse learning* is the *sparse regression* [13, 30, 58], which is widely used in medical diagnosis. Another one is the *Sparse Bayesian Learning* [67, 71, 72] which was first proposed by M. E. Tipping et al. [67] used in relevance vector machine (RVM), which adds a sparse constraint prior distribution to the parameters to learn a sparse parameter vector. It can provide more accurate prediction model than support vector machine (SVM).

1.2 Visual Recognition

1.2.1 Feature Representation and Learning

1. Feature Categories

In visual recognition, there are various features extracted from pixel intensities. We will introduce some basic features of images/videos, such as edges, interesting points, saliency, and histogram of oriented gradient.

Edge

Edges are the discontinuity of intensity in some direction. Thus, we can detect edges by localizing the pixels with large derivative values. The large value means a sharp change in an image. More details, the discontinuities in image brightness are likely to correspond to discontinuities in depth and surface orientation, changes in material properties, and variations in scene illumination.

The edge pixels are at local maxima of gradient magnitude. The pixel gradient can be computed by convolving with Gaussian derivatives, and the gradient direction is always perpendicular to the edge direction, shown in Fig. 1.2. In the ideal case, an edge detector may result in a set of connected curves that indicate the boundaries of objects, surface markings, and surface orientation [4]. Thus, the

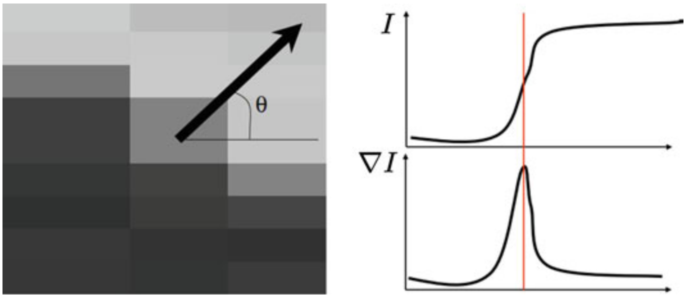

Fig. 1.2 The gradient direction (*left*) and the relation between edge and image

edge detection may significantly reduce the amount of pixel intensities and filter unimportant information, while only preserving the important structural properties of an image. This procedure is a kind of data sparsification.

The edge detection can be divided into two categories, *search-based* and *zero-crossing based*. The *search-based* approaches usually use a first-order derivative to compute the strength of the edge, and then search a local maximum direction of the gradient magnitude. The *zero-crossing based* approaches search for zero crossing in a second-order derivative to find edges, such as the zero-crossings of Laplacian or the zero-crossings of a nonlinear differential expression. Before edge detection, we need to smooth the image first, typically using Gaussian smoothing. J. Mairal et al. use sparse representation for class-specific edge detection problem [48]. In this approach, two dictionaries are trained for each specific class, one of which corresponds to the "Good edges" for this class, and the other corresponds to the "Bad edges" for this class. Figure 1.3 shows us that different directions of derivatives can generate different edges.

Interest Point

Interest points are the junctions of contours in images and can be characterized as follows [59]:

- It is clear and well-defined in concepts, scales, and image spaces;
- Its local image structure is rich;
- It is *stable* under local and global perturbations in the image domain as illumination/brightness variations, such that the interest points can be reliably computed with high degree of reproducibility;

Mathematically, we can localize the interesting points using local displacement sensitivity within a window. In detail, we define a shift change of intensity between a pixel and its neighboring pixels within this window. Furthermore, we have a bilinear approximation using first-order Talyor approximation. Thus, the Harris matrix in the bilinear approximation describes the distribution of intensity within the window, and thus can be used to detect interesting points. Furthermore, 2D interesting point detection can extend to 3D cases, a.k.a spatial temporal interesting points (STIPs) [45].

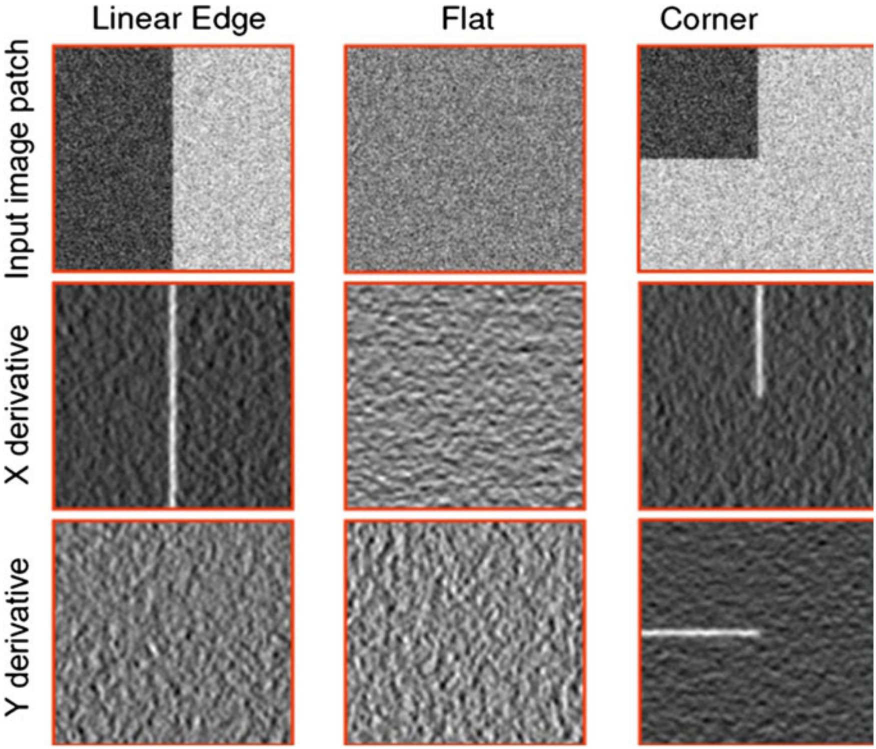

Fig. 1.3 The edges corresponding to different direction of derivatives

Visual Saliency

One of the most severe problems of perception for either humans or machines is information overload [32]. It is challenging especially for machines to process all the information from peripheral sensors all the time. The neural system of humans is good at making decisions on which part of the available information is to be selected for further, and which parts are to be discarded. This procedure, selection, and ordering process is called selective attention. Detecting and recognizing objects in unknown environments is an important yet challenging task, especially for machines. Visual saliency approaches guide the attention of people/machines to some positions in images/videos in a computational procedure. Typical features used in visual saliency are color, gradient, and contrast for distinguishing salient regions from the rest. Upon those features, we can generate various saliency mappings using visual saliency approaches. Moreover, saliency mappings root in feature integration theory (FIT) [69], and its framework are shown in Fig. 1.4.

The saliency map consists of two basic models, top-down and bottom-up. The visual saliency involves feature learning and saliency computation. The top-down visual saliency is driven by expectation and tasks while bottom-up visual saliency

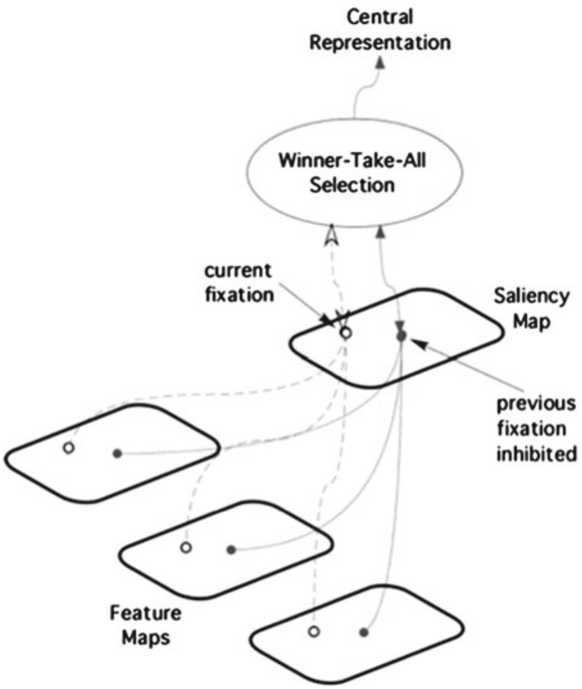

Fig. 1.4 The basic saliency map model as proposed by Koch and Ullman [44]

is determined by feature properties in the context. Moreover, according to the mathematical models of visual saliency, we can divide saliency algorithms into five kinds, i.e., local contrast models [39], information maximization models [10], graph-based models [33], low-rank matrix recovery models [62, 74], and spectral residual models [37]. As we may know, visual saliency is widely used in mobile robots [63], visual recognition [77], and text detection from images/videos [38, 42, 60].

Gradients

In mathematics, gradient is the derivative of a function in one dimension to a function in several dimensions, which means the change of function. It can model many physical phenomena in images/videos, such edges, interest points, and saliency since gradient encodes edges and local contrast very well. The rationale behind this is the human visual system is very sensitive to gradients. Thus, the image/video gradient is important in various computer vision tasks, such as human detection [20], local descriptors [47], and high-resolution [64]. Image/video gradients include two kinds of information, magnitude and orientation. The former denotes how quickly the images/vidoes are changing while the latter tells us the direction in which the images/videos is changing most rapidly.

The first important application of image gradients is human detection [20]. Histogram of Oriented Gradients (HOG) is the dense local feature for describing one image, and is widely used in object detection. The core idea of HOG is that the

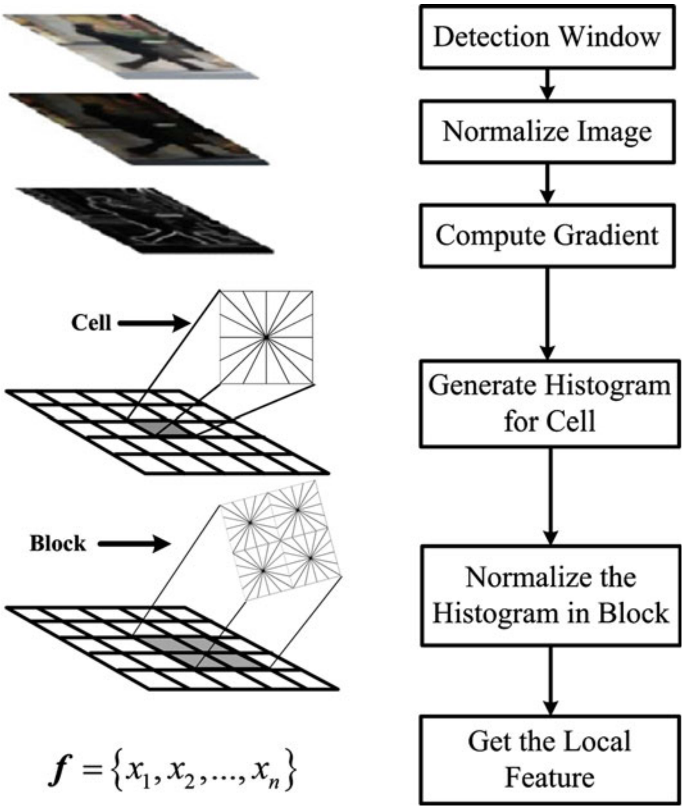
Detection Window
Normalize Image
Compute Gradient
Generate Histogram for Cell
Normalize the Histogram in Block
Get the Local Feature
Cell
Block
$f=\{x_1,x_2,\ldots,x_n\}$

Fig. 1.5 The flowchart of how to get HOG feature

local appearance and shape of one patch can be represented using the distribution of gradient. The merit of HOG is that it has good invariable property for changes in illumination and geometry. The flowchart of how to get the HOG feature is illustrated in Fig. 1.5.

One more application of HOG is scale invariant feature transform (SIFT) in image matching and classification. It is similar to HOG feature. Compared to HOG, the SIFT is not the dense local feature in one image. They are obtained by searching the extreme point in scale spaces as shown in Fig. 1.6. Moreover, Fig. 1.7 shows the procedure of generating the signature of SIFT.

2. Feature Learning

Feature learning is a set of techniques in machine learning that learn a transformation of "raw" inputs to a representation that can be effectively exploited in either supervised or unsupervised learning tasks. The main goal of feature learning is to learn good representations from data by eliminating irrelevant features while preserving the useful features for object detection, recognition, classification, or

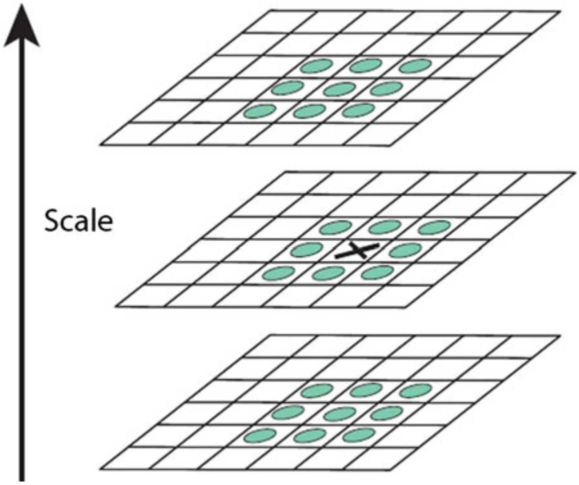

Fig. 1.6 The SIFT is obtained by searching the extreme point in scale space [47]

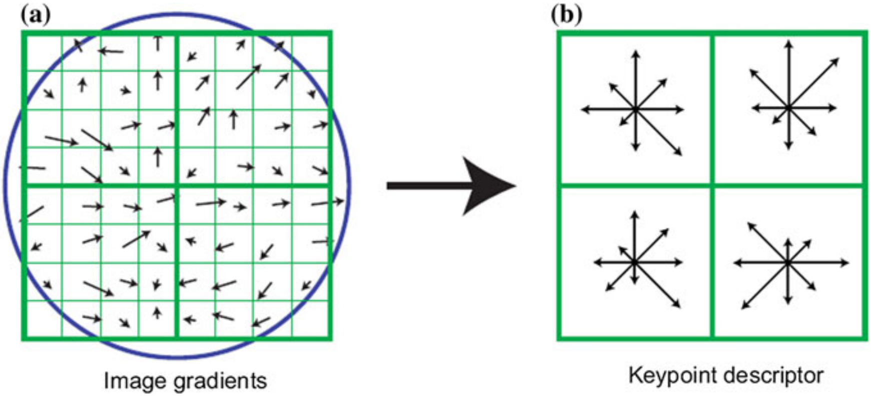

Fig. 1.7 The procedure of calculating the signature of SIFT gives an interesting point: **a** The magnitude and orientation of image gradients; **b** The SIFT descriptor [47]

prediction. Roughly, feature learning can be divided into two categories, unsupervised and supervised. Typical unsupervised learning consists of PCA/SPCA/RSPCA, K-means, mean shift, KSVD, and expectation maximization (EM), and supervised learning consists of neural networks, support vector machines (SVMs), restricted Boltzmann machines (RBM), and linear discriminant analysis. Among these algorithms, sparse feature learning is more and more popular in the computer vision and machine learning committee, such as sparse SVM, recursive feature elimination. Moreover, deep learning (a.k.a deep representation learning) is widely used in visual recognition, such as face recognition and image classification.

1.2.2 Distance Metric Learning

Distance Metric Learning is to learn a distance metric for input feature spaces in either supervised or unsupervised way. Learning a good distance metric in feature space is critical for various computer vision tasks, such as image classification and content-based image retrieval. For example, the retrieval quality of content-based image retrieval (CBIR) systems is known to be highly dependent on the criterion that is used to define similarity between images. Also, it has motivated significant advancement in learning good distance metrics from training data. Moreover, in the K-nearest-neighbor (KNN) classifier, one of the key issues is to select a good distance measure for the specific task. The previous work [34–36, 51] has shown that good distance metric learning can improve the performance of KNN classifier, which is better than the standard Euclidean distance.

Much effort has been made on distance metric learning over the past few years. We can divide distance metric learning into two categories depending on the availability of the training examples: *supervised distance metric learning* and *unsupervised distance metric learning*. In supervised learning, training samples are cast into pairwise constraints: the equivalence constraints where sample pairs belong to the same classes, and inequivalence constraints where pairs of samples belong to different classes. The supervised distance metric learning uses two different strategies, the *global learning*, and the *local learning*. The first strategy learns the metric in a global sense, i.e., to satisfy the constraints of all the training samples pairwise simultaneously. The second strategy is to learn the distance metric in a local setting, i.e., only to satisfy the local pairwise constraints. The main idea for unsupervised distance metric learning is to learn an underlying low-dimensional manifold where geometric relationships between most of the observed data are preserved. The *sparse induced similarity* is an unsupervised distance metric learning method [17], which uses sparse representation as a distance metric.

1.2.3 Classification

1. The Nearest Neighbor Classifier

The *Nearest Neighbor Classifier* (NNC) is a nonparametric approach and also the oldest yet best one, especially for large-scale visual recognition. Nearest neighbor rule is a suboptimal procedure. Its use will usually lead to an error rate greater than the minimum possible, the Bayes error rate [53], shown in Fig. 1.8. We can see that $P^* \leq P \leq P^*(2 - \frac{c}{c-1}P^*)$, where P^* is the Bayesian probability of error rate, P is the nearest error rate. The critical issues are to choose the proper distance measure and efficient NN searching.There are many alternative distance measures to use in NNCs, such as Euclidean distance, Gaussian kernel similarity, sparsity-induced similarity [17], and distance metric learning. Moreover, for large scale NN

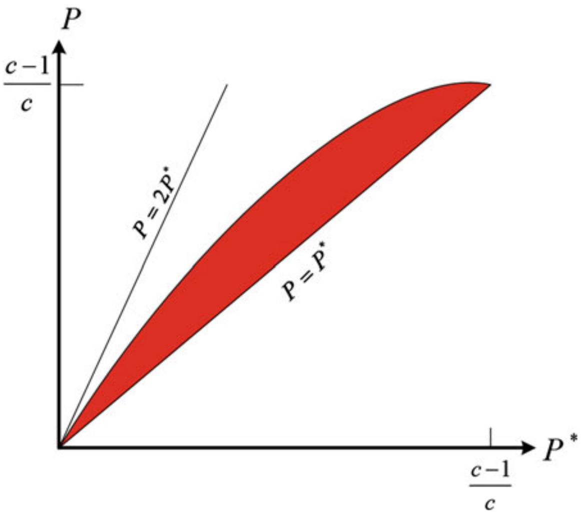

Fig. 1.8 The nearest neighbor error rate. P^* is the Bayesian probability of error rate, P is the nearest neighbor error rate

searching, we can use approximate nearest neighbors [1] to speed up the searching procedures.

2. The Bag of Feature Approach

In computer vision, the bag-of-features (BoF) approaches are widely used in image classification, by treating local image features as visual words [27]. In principal, a BoF is a vector of occurrence counts of a vocabulary of local image features.

There are three steps in the BoF model: (1) Feature detection; (2) Feature description; (3) Codebook generation. A definition of the BoF model can be the histogram representation based on independent features [27]. This image representation method is first used in content-based image indexing and retrieval (CBIR) [55]. Feature representation methods deal with how to represent the patches as numerical vectors. These vectors are called as the local descriptors. A good descriptor should be insensitive with intensity, rotation, scale, and affine variations. One of the most famous descriptor is scale-invariant feature transform (SIFT) in image classification [55]. The SIFT converts each patch to be a 128-dimensional vector. After this step, the image is represented as a collection of vectors of the same dimension. If we want to represent the image in BoF model, we need to generate the “codebook" which is used to quantize each local descriptor to be one word of the codebook. One simple method is using K-means clustering over all the vectors to generate the codebook. The number of the clusters is the codebook size. Thus, each patch in an image is mapped to a certain codeword by computing the similarity with each word in codebook and the image can be represented by the histogram of the codewords. The BoF model is illustrated visually in Fig. 1.9.