PERGAMON

Pattern Recognition 36 (2003) 2007–2017

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Hierarchical classification and feature reduction for fast face detection with support vector machines

Bernd Heisele[a,b,*], Thomas Serre[b], Sam Prentice[b], Tomaso Poggio[b]

[a] *Honda Research Institute US, 145 Tremont St., Boston, MA 02111, USA*
[b] *Center for Biological and Computational Learning, MIT, E25-201, 45 Carleton St., Cambridge, MA 02142, USA*

Accepted 15 January 2003

Abstract

We present a two-step method to speed-up object detection systems in computer vision that use support vector machines as classifiers. In the first step we build a hierarchy of classifiers. On the bottom level, a simple and fast linear classifier analyzes the whole image and rejects large parts of the background. On the top level, a slower but more accurate classifier performs the final detection. We propose a new method for automatically building and training a hierarchy of classifiers. In the second step we apply feature reduction to the top level classifier by choosing relevant image features according to a measure derived from statistical learning theory. Experiments with a face detection system show that combining feature reduction with hierarchical classification leads to a speed-up by a factor of 335 with similar classification performance.

Keywords: Face detection; Object detection; Feature reduction; Hierarchical classification; Support vector machines

1. Introduction

A major task in visual scene analysis is to detect objects in images. This is commonly done by shifting a search window over an input image and by categorizing the object in the window with a classifier. The main problem in categorization is the large range of possible variations within a class of objects. The system must generalize not only across different viewing and illumination conditions but also across different exemplars of a class, such as faces of different people for face detection. This requires complex, computationally expensive classifiers. Further contributing to the computational load of the task is the large amount of input data that has to be processed. A real-time vision system has to deal with data streams in the range of several MBytes/s. Speeding-up classification is therefore of major concern when developing systems for real-world applications. In the following we investigate two methods for speed-ups: hierarchical classification and feature reduction.

In Ref. [1] we presented a system for detecting frontal and near-frontal views of faces in still gray images. The system achieved high detection accuracy by classifying 19×19 gray patterns using a non-linear SVM. Searching an image for faces at different scales took several minutes on a PC, far too long for most real-world applications. Experiments with faster classifiers (linear SVMs) gave significantly lower recognition rates. To speed-up the system without losing in classification performance one can exploit the following two characteristics common to most vision-based detection tasks: First, the vast majority of the analyzed patterns in an image belongs to the background class. For example, the ratio of non-face to face patterns in the tests in Ref. [1] was about 50,000 to 1. Second, many of the background patterns can be easily distinguished from the objects. Based on these

* Corresponding author. Honda Research Institute US, 145 Tremont St., Boston, MA 02111, USA. Tel.: +1-617-338-4909; fax: +1-617-338-4085.
E-mail addresses: heisele@ai.mit.edu (B. Heisele), serre@ai.mit.edu (T. Serre), prentice@mit.edu (S. Prentice), tp@ai.mit.edu (T. Poggio).

0031-3203/03/$30.00
doi:10.1016/S0031-3203(03)00062-1

two task-specific characteristics it is sensible to apply a hierarchy of classifiers. Fast classifiers remove large parts of the background on the bottom and middle levels of the hierarchy and a more accurate but slower classifier performs the final detection on the top level. This idea is related to the well known coarse-to-fine template matching [2–4]. In Ref. [5] hierarchical classification is used to speed-up a face detection system. A candidate selection neural network with increased robustness against translation is added as a first layer to an existing face detection neural network.

We present a method for building and training a hierarchy of SVM classifiers given a set of classifiers which operate at different image resolutions. The iterative algorithm starts with the topmost classifier at the highest image resolution and adds a new lower layer. At each iteration, the hierarchies are retrained bottom up and a speed test is performed on a validation set of non-face images to choose the hierarchy with the least number of computations.

Another possibility of speeding-up classification is to reduce the number of features by selecting a subset of relevant features. Feature reduction is a more generic tool than the above described hierarchical classification and can be applied to any classification problem. There are basically two types of feature selection methods in the literature: filter and wrapper methods [6]. Filter methods are preprocessing steps performed independently of the classification algorithm or its error criteria. Wrapper methods attempt to search through the space of feature subsets using the criterion of the classification algorithm to select the optimal feature subset [7]. We present a new wrapper method to reduce the dimensions of both input and feature space of a non-linear SVM. For our final detection system we combine feature selection with hierarchical classification by putting a non-linear SVM with feature selection on top of the hierarchy of linear SVMs. A similar idea of combining feature selection with hierarchical classification has been recently proposed in Ref. [8] for frontal face detection. They use AdaBoost to train a cascade of linear classifiers and to select features from an over complete set of Haar wavelet features. In contrast to our approach, however, the complexity of the classifiers in the final hierarchy is only controlled by the number of features and not by the class of decision functions (i.e. class of kernel functions).

The outline of the paper is as follows: In Section 2 we give a brief overview of SVM classification. In Section 3 we describe how to build and train the hierarchical system. Sections 4 and 5 describe the feature selection methods for the input and the feature space of the SVM, respectively. In Section 6 we present experimental results of the hierarchical system with feature reduction. The paper is concluded in Section 7.

2. Background on support vector machines

2.1. Theory

An SVM [9] performs pattern recognition for a two-class problem by determining the separating hyperplane that has maximum distance to the closest points of the training set. These closest points are called support vectors. To perform a non-linear separation in the input space a non-linear transformation $\Phi(\cdot)$ maps the data points $\mathbf{x}$ of the input space $\mathbb{R}^n$ into a high-dimensional space, called feature space $\mathbb{R}^p$ ($p > n$). The mapping $\Phi(\cdot)$ is represented in the SVM classifier by a kernel function $K(\cdot,\cdot)$ which defines an inner product in $\mathbb{R}^p$. Given ℓ examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$, the decision function of the SVM is linear in the feature space and can be written as

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b = \sum_{i=1}^{\ell} \alpha_i^0 y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{1}$$

The optimal hyperplane is the one with the maximal distance in space $\mathbb{R}^p$ to the closest points $\Phi(\mathbf{x}_i)$ of the training data. Determining that hyperplane leads to maximizing the following functional with respect to α:

$$W^2(\alpha) = 2\sum_{i=1}^{\ell} \alpha_i - \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

under constraints $\sum_{i=1}^{\ell} \alpha_i y_i = 0$ and $C \geqslant \alpha_i \geqslant 0$, $i = 1, \ldots, \ell$. An upper bound on the expected error probability EP_{err} of an SVM classifier is given by [9]

$$EP_{err} \leqslant \frac{1}{\ell} E\left(\frac{R^2}{M^2}\right) = \frac{1}{\ell} E(R^2 W^2(\alpha^0)), \tag{3}$$

where $M = 1/W(\alpha^0)$ is the distance between the support vectors and the separating hyperplane and R is the radius of the smallest sphere including all points $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_\ell)$ of the training data in the feature space. In the following, we will use this bound on the expected error probability to rank and select features.

2.2. Computational issues

The only non-linear kernel investigated in this paper is a second-degree polynomial kernel $K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^2$ which has been successfully applied to various object detection tasks [1,10]. Eq. (1) shows two ways of computing the decision function. When using the kernel representation on the right side of Eq. (1) the number of multiplications required to calculate the decision function for a second-degree polynomial kernel is

$$M_{k,poly2} = (n + 2) \cdot s, \tag{4}$$

where n is the dimension of the input space and s is the number of support vectors. This number is independent of the dimensionality of the feature space. It depends on the number of support vectors which is linear with the size ℓ of the training data [9]. On the other hand, the computation of the decision function in the feature space is independent of the size of training samples, it only depends on the dimensionality p of the feature space. For the second-degree polynomial kernel the feature space $\mathbb{R}^p$ has dimension $p = (n + 3)n/2$ and is given by $\mathbf{x}^* = (\sqrt{2}x_1, \ldots, \sqrt{2}x_n, x_1^2, \ldots, x_n^2, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_{n-1}x_n)$.