PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 37 (2004) 2069–2079

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components

Simon Günter*, Horst Bunke

Department of Computer Science, University of Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland

Received 11 June 2003; received in revised form 5 April 2004; accepted 5 April 2004

Abstract

In off-line handwriting recognition, classifiers based on hidden Markov models (HMMs) have become very popular. However, while there exist well-established training algorithms which optimize the transition and output probabilities of a given HMM architecture, the architecture itself, and in particular the number of states, must be chosen “by hand”. Also the number of training iterations and the output distributions need to be defined by the system designer. In this paper we examine several optimization strategies for an HMM classifier that works with continuous feature values. The proposed optimization strategies are evaluated in the context of a handwritten word recognition task.

Keywords: Handwritten word recognition; Hidden Markov model (HMM); Training strategy; State number optimization; Gaussian mixture models

1. Introduction

The field of off-line handwriting recognition has been a topic of intensive research for many years. First only the recognition of isolated handwritten characters was investigated [1], but later whole words [2] were addressed. Most of the systems reported in the literature until today consider constrained recognition problems based on vocabularies from specific domains, e.g. the recognition of handwritten check amounts [3] or postal addresses [4]. Free handwriting recognition, without domain specific constraints and large vocabularies, was addressed only recently in a few papers [5,6].

Hidden Markov Model classifiers (HMMs) have become very popular in the domain of handwritten word recognition [4,6,7]. Given an HMM with a predefined architecture, there exist some well-established training algorithms to automatically optimize the parameters of that architecture. An example is the Baum–Welch training procedure [8] which uses the Maximum Likelihood Estimation (MLE) criterion. In the Baum–Welch training the product of the likelihood values of the correct classes of the training patterns is guaranteed to increase after each iteration. The architecture of an HMM as well as the number of Gaussian components per state and the number of training iterations are usually empirically determined. In the speech recognition domain there are some papers that address the optimization of the free parameters of an HMM architecture. In Ref. [9] the number of Gaussian components of the states is optimized by repeatedly splitting Gaussian distributions according to the maximum mutual information estimation (MMIE) criterion, which is also used for training of the HMMs in this paper. Also in Ref. [10] the number of Gaussian components is optimized. Here in each step the Gaussian component with the highest variance is split. Before splitting, Gaussian

* Corresponding author.
E-mail address: sguenter@iam.unibe.ch (S. Günter)

0031-3203/$ - see front matter
doi:10.1016/j.patcog.2004.04.006

components that have too few training samples and are similar to each other are merged to avoid the problem of overfitting. In Ref. [11] a combined optimization strategy for the number of states and the number of Gaussian components per HMM is presented. A disadvantage of this method is that it does not preserve the linearity of the models which is often an important property of HMMs used in handwriting recognition.

In the present paper we propose some optimization strategies for HMM classifiers using Baum–Welch training. These optimization strategies are evaluated in a number of experiments with a handwritten word recognizer. The parameters of the HMM subject to optimization are the number of training iterations and the number of Gaussian components per state. In addition we address the problem of finding the optimum number of states per HMM and investigate two different ways to use the validation set. Finally, the results of a system trained on a large set of words and optimized using the best of the considered strategies are presented.

The rest of the paper is organized as follows. Section 2 describes the handwritten word recognizer and the recognition task. In Section 3 methods for finding the number of states of an HMM are discussed, and in Section 4 a description of the strategies to jointly optimize the number of Gaussian components and training iterations is given. Two approaches to use the validation set for the training of the final system are presented in Section 5. Experiments are then discussed in Section 6 and, finally, conclusions are drawn in Section 7.

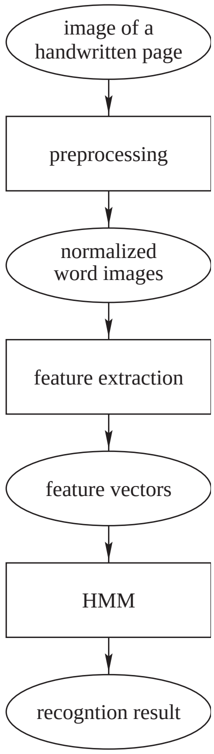

Fig. 1. System overview.

2. Handwritten word recognizer

The basic handwritten text recognizer used in the experiments of this paper is similar to the one described in Ref. [6]. It follows the classical architecture and consists of three main modules (see Fig. 1): the preprocessing, where noise reduction and normalization take place, the feature extraction, where the image of a handwritten text is transformed into a sequence of numerical feature vectors, and the recognizer, which converts this sequence of feature vectors into a word class.

The first step in the processing chain, the preprocessing, is mainly concerned with text image normalization. The goal of the different normalization steps is to produce a uniform image of the writing with less variations of the same character or word across different writers. The aim of feature extraction is to derive a sequence of feature vectors which describe the writing in such a way that different characters and words can be distinguished, but avoiding redundant information as much as possible. In the presented system the features are based on geometrical measurements. At the core of the recognition procedure is an HMM classifier. It receives a sequence of feature vectors as input and outputs a word class. In the following these modules are described in greater detail.

2.1. Preprocessing

The application considered in this paper is the off-line recognition of cursively handwritten words. As basic classifier an HMM-based recognizer is used. This recognizer is similar to the one described in Ref. [6]. Each person has a different writing style with its own characteristics. This fact makes the recognition task complicated. To reduce variations in the handwritten texts as much as possible, a number of preprocessing operations are applied. The input for these preprocessing operations are images of words extracted from the database described in Refs. [12,13]. The images were scanned with a resolution of 300 dpi. Please note that the system is optimized for this resolution. In the presented system the following preprocessing steps are carried out:

- *Skew correction*: The word is horizontally aligned, i.e. rotated, such that the baseline is parallel to the x-axis of the image.
- *Slant correction*: Applying a shear transformation, the writing's slant is transformed into an upright position.
- *Line positioning*: The word's total extent in vertical direction is normalized to a standard value. Moreover, applying a vertical scaling operation the location of the upper and lower baseline are adjusted to a standard position.