Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Component-based discriminative classification for hidden Markov models

Manuele Bicego[a,b,*], Elżbieta Pękalska[c], David M.J. Tax[d], Robert P.W. Duin[d]

[a]Computer Science Department, University of Verona, Strada le Grazie, 15, 37134 Verona, Italy
[b]DEIR, University of Sassari, via Torre Tonda, 34, 07100 Sassari, Italy
[c]School of Computer Science, University of Manchester, Oxford Road, M13 9PL Manchester, UK
[d]Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

ARTICLE INFO

Article history:
Received 15 September 2008
Received in revised form 10 March 2009
Accepted 18 March 2009

Keywords:
Hidden Markov models
Discriminative classification
Dimensionality reduction
Hybrid models
Generative embeddings

ABSTRACT

Hidden Markov models (HMMs) have been successfully applied to a wide range of sequence modeling problems. In the classification context, one of the simplest approaches is to train a single HMM per class. A test sequence is then assigned to the class whose HMM yields the maximum a posterior (MAP) probability. This generative scenario works well when the models are correctly estimated. However, the results can become poor when improper models are employed, due to the lack of prior knowledge, poor estimates, violated assumptions or insufficient training data.
To improve the results in these cases we propose to combine the descriptive strengths of HMMs with discriminative classifiers. This is achieved by training feature-based classifiers in an HMM-induced vector space defined by specific components of individual hidden Markov models.
We introduce four major ways of building such vector spaces and study which trained combiners are useful in which context. Moreover, we motivate and discuss the merit of our method in comparison to dynamic kernels, in particular, to the Fisher Kernel approach.

1. Introduction

Hidden Markov models (HMMs) represent a powerful statistical learning technique, widely applied to sequence modeling. They have been successful in a variety of fields such as natural language processing (including speech recognition [1,2], speech translation [3] or parsing [4]), pattern recognition and computer vision [5] (including shape recognition [6,7], face and gesture recognition [8,9], image analysis [10]) and also computational biology and genomics [11–13].

HMMs are stochastic finite state machines which generate a sequence of observations, symbols or vectors, by moving among (hidden) states, as governed by transition probabilities. It is assumed that the probability distribution of the current state is conditionally independent of the path of past states (first order Markov property). A symbol is emitted according to the emission probabilities each time a state is visited. Hence, in order to model the underlying structure of a set of sequences one needs to specify the emission probability functions, state transition probabilities and the initial state probabilities. HMMs are computationally efficient to train and evaluate thanks to the assumption of the first order Markov property and the availability of efficient algorithms [1,12].

In the classical HMM-based classification scenario each class ω_c from a set of classes $\Omega = \{\omega_1, \ldots, \omega_C\}$ is modeled by a single HMM. The generative HMM $\boldsymbol{\lambda}_c$ describes the probability density function for an example sequence $\mathbf{O}$ associated with ω_c, i.e. $P(\mathbf{O}|\omega_c; \boldsymbol{\lambda}_c)$. Typically, each model $\boldsymbol{\lambda}_c$ is trained iteratively via the Baum–Welch re-estimation procedure [14,15], i.e. its parameters are updated to maximize the likelihood of the training sequences $\mathcal{O}^c$. Having determined the model parameters $\boldsymbol{\lambda}_c$, sequences can now be randomly generated from the distribution of this model, and, more importantly, the sequence likelihood can be calculated for an example $\mathbf{O}$. Given the prior distribution over the classes, $P(\omega_c)$, $c = 1, \ldots, C$, Bayes rule is used to estimate the posterior probabilities: $P(\omega_c|\mathbf{O}; \boldsymbol{\lambda}_c) = P(\mathbf{O}|\omega_c; \boldsymbol{\lambda}_c)P(\omega_c)/P(\mathbf{O})$. Classification of an unlabeled sequence $\mathbf{O}$ is performed by the MAP (maximum a posteriori) approach: $y = \arg\max_c\{P(\mathbf{O}|\omega_c; \boldsymbol{\lambda}_c)P(\omega_c)\}$, where y is the label assigned to $\mathbf{O}$. The evidence $P(\mathbf{O})$ is omitted as it does not influence the decision. So, we deal with a generative approach to classification, in which probability density models are learned per class, and the class of the largest $P(\mathbf{O}|\omega_c; \boldsymbol{\lambda}_c)P(\omega_c)$is picked for the assignment.

Even if this Bayes rule represents the theoretical optimal decision rule (i.e. leading to the minimum probability of error [16]), in practice, generative HMMs may suffer from poor discriminative abilities.

* Corresponding author at: Computer Science Department, University of Verona, Strada le Grazie, 15, 37134 Verona, Italy. Tel.: +39 045 8027072; fax: +39 045 8027068.
E-mail address: manuele.bicego@univr.it (M. Bicego).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.03.023

This is likely to occur in one of the following scenarios:

- poorly estimated class models, e.g. due to insufficient learning examples,
- improper models, e.g. due to bad model definition or conditional dependence of the states and
- possible class overlap, as may occur in for instance medical problems where patient diagnoses may not be consistent between different medical doctors.

Some of these issues can be addressed by improving and extending the classical HMMs. Many such approaches have been proposed in the past; a non-exhaustive list includes hierarchical HMM [17], input–output HMM [18], factorial hidden Markov models [19], coupled HMM [20] and others. These offer solutions to inhomogeneous class structures, as they are flexible enough to model complex classes. They however become infeasible for small sample size problems, because there is far too little information for the estimation of their parameters. Alternatively, the discriminative skills can be enhanced by training HMMs with discriminative criteria. Two popular examples are based on maximum mutual information (MMI) [21] and minimum Bayes risk (MBR) [22], but other extensions are available, such as [23,24] or the lattice-based framework presented in [25]. See also [26] for an overview. In the context of speech recognition, we read in [25]: "The key issue [with discriminative criteria] is one of generalization and this is affected by the amount of training data available, the number of HMM parameters estimated, and the training scheme used. [..] if steps are taken to improve generalization performance, MMI-trained systems can indeed outperform the best ML-trained systems for even the most complex large vocabulary speech recognition tasks." One must however remember that although discriminative criteria try to reduce the recognition error directly, they require sufficient amount of training data.

Furthermore, there exist generalizations of HMMs towards probabilistic discriminative models. These are conditional random fields (CRFs) [27] and hidden CRFs (HCRFs) [28], in which conditional maximum likelihood is often used to estimate the parameters. In the application to phone classification, Gunawardana [28] shows that HCRFs outperform comparable generative HMMs (trained by ML) and MMI-trained HMMs. Discriminative learning procedures can also be based on a maximum/soft margin criterion [29], a perceptron-like algorithm [30], or a recursive scheme that maximizes the lower bound of the regularized cross-entropy [31]. All these discriminative techniques need complex training procedures, whereas the final classification still relies on the MAP approach: a sequence is assigned to the class whose HMM yields the highest maximum a posterior probability.

In this paper we propose a simpler method to improve the discriminative power of the MAP approach, i.e. to apply discriminative classifiers to features that are extracted from individual HMM class descriptions. This utilizes the lesson from Multiple Classifier Systems that a back-end discriminative classifier is able to compensate for the weak generalization power of individual class-related hidden Markov models; see also [32–34]. We therefore focus on situations in which classical HMMs are "good enough" to capture characteristics of the classes, but the classes are either weakly discriminative (due to the problem definition) or described by poorly estimated models (due to insufficient complexity or small sample size problems).

Our proposal is to map each sequence into an HMM-induced vector space (HHMVS), whose features are derived from specific components of individual HMMs. Every feature measures the relevance of a specific component (like states or transitions) of a given HMM when an input sequence **O** is fed to this HMM, or, better, how a specific component contributes to the explanation of **O**. Discriminative classifiers are then designed in these vector spaces. This paper shows that this method is robust and may drastically improve the standard HMM-based classification scheme. This is particularly evident when the original models are insufficient to solve the problem, usually due to small training sets, incorrect model topology or bad model assumptions. The described approach may be really useful in another crucial scenario, namely when different models—already trained on different data sets—are available, and new data (from a novel class) come. In this case, instead of training the models again (or training new models), we can just build the discriminant on the new data projected to the feature space through the already trained models. This is illustrated in Section 4, where a feature space derived from a model of a *single* class is good enough to enable discrimination between different classes not modeled by HMMs. Such a scenario could be useful for using generic models in different cultural areas (e.g. for speech recognition), and also when a time adaption is required.

The approach proposed here is an extension of some previous work [35–37], in which features were defined as functions measuring similarity between patterns and classes (or clusters). In this paper, we go a step further and define features that describe relevance of particular components of the HMMs for explaining input sequences. It is related to the family of dynamic kernels [38–40], in particular the Fisher Score approach [38,41–43] and its variants, e.g. [44], as well as further generalizations to score spaces [45]. In the classical Fisher Score approach, features are gradients of the log-likelihood with respect to the model's parameters. Even though HMM-induced spaces proposed by us share some structure with variants of the Fisher Score space, the spaces are different because they are derived based on fundamentally different assumptions. We describe both similarities and differences between HMM-induced vector spaces and (Fisher) Score spaces, showing experimentally how and when they differ.

The remainder of the paper is organized as follows. Section 2 describes the proposed generative-discriminative approach, while Section 3 presents an experimental evaluation. Section 4 discusses dimensionality issues. Section 5 focuses on a comparison with the Fisher Score approach. Finally, conclusions are drawn and future perspectives are envisaged in Section 6.

2. Learning in HMM-induced vector spaces

Before introducing our framework, we will first start with some notation.

2.1. Hidden Markov models

A discrete-time first order hidden Markov model [1] is a stochastic finite state machine defined over a set of K states $S=\{S_1,S_2,\ldots,S_K\}$. The states are hidden, i.e. not directly observable. Each state has an associated probability density function encoding the probability of observing a certain symbol being output from that state. Let $\mathbf{Q}=(Q_1,Q_2,\ldots,Q_T)$ be a fixed state sequence of length T with the corresponding observations $\mathbf{O}=(O_1,O_2,\ldots,O_T)$. An HMM is described by a model $\boldsymbol{\lambda}$, determined by a triple $\{\mathbf{A},\mathbf{B},\boldsymbol{\pi}\}$ such that:

- $\mathbf{A}=(a_{ij})$ is a matrix of transition probabilities, in which $a_{ij}=P(Q_t=S_j|Q_{t-1}=S_i)$ denotes the probability of state S_j following state S_i.
- $\mathbf{B}=(b_j(o))$ consists of emission probabilities, in which $b_j(o)=P(O_t=o|Q_t=S_j)$ is the probability of emitting the symbol o when being in state S_j. We deal with a *discrete* HMM if the emitted symbol comes from a finite alphabet, and we deal with a *continuous* HMM, if the emission probability density is modeled by a continuous function as emitted observations are vectors.
- $\boldsymbol{\pi}=(\pi_i)$ is the initial state probability distribution, i.e. $\pi_i=P(Q_1=S_i)$.

There are two additional assumptions that an HMM should obey, first order Markov property and output independence. The former property states that information in the current state is conditionally independent from the previous states, i.e. $P(Q_t|Q_{t-1},\dots,Q_1)=P(Q_t|Q_{t-1})$. The second assumption, $P(O_t|Q_t,Q_{t-1},\dots,Q_1,O_{t-1},\dots,O_1)=P(O_t|Q_t)$, demands that the output at time t depends only on the current state.

A crucial procedure is the so-called *forward–backward* procedure [1], used to recursively compute the probability $P(\mathbf{O}|\boldsymbol{\lambda}_c)$ for a test sequence $\mathbf{O}$, i.e. the probability of generating $\mathbf{O}$ by model $\boldsymbol{\lambda}_c$ via the most likely path. This algorithm is used multiple times to derive the quantities needed in our proposal; see Appendix A.2 for details.

2.2. Our methodology

We will now describe a framework for combing the strengths of one-class HMMs with discriminative feature-based learners via the use of component-based HMM-induced vector spaces. Let $\mathscr{O}=\{\mathbf{O}_1,\dots,\mathbf{O}_n\}$ be a set of observation sequences (of variable lengths) with the corresponding labels $\mathscr{Y}=\{y_1,\dots,y_n\}$, indicating class memberships to one of C classes, $\omega_1,\dots,\omega_C$. Let $\mathscr{O}^c=\{\mathbf{O}^c_1,\dots,\mathbf{O}^c_{n_c}\}$ denote sequences of the class ω_c. The HMM-based discriminative classification relies on the following steps:

1. Training individual HMMs.
2. Definition of an HMM-induced vector space.
3. Classifier training: data projection to HMMVS and training a feature-based classifier.
4. Classifier testing: data projection to HMMVS and testing.

2.3. Training individual HMMs

Each class ω_c is described by a generative K-state hidden Markov model $\boldsymbol{\lambda}_c$, with the states $S^c=\{S^c_1,S^c_2,\dots,S^c_K\}$, trained on the sequences from $\mathscr{O}^c$. Each $\boldsymbol{\lambda}_c$ is estimated via the Baum–Welch re-estimation algorithm such that $P(\mathbf{O}^c_i|\boldsymbol{\lambda}_c)$ is maximized for all $\mathbf{O}^c_i$ [1]. This results in a battery of C HMMs $\{\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\dots,\boldsymbol{\lambda}_C\}$.

2.4. HMM-induced vector space

The HMMs are fitted to model a single class well, but this may lead to poor discrimination as the models are not optimized to differentiate among the classes. We propose to derive a fixed-dimensional feature space from the trained generative HMMs, in which discriminative classifiers are trained. We call this an HMMVS, equipped with the traditional norm and Euclidean metric. Every feature is extracted from a specific HMM and conveys information about the corresponding class. In essence, this approach maps variable-length observation sequences into a vector space, and by doing this it integrates the modeling potential of one-class models with discriminative classifiers.

HMMVS are based on "Component Information" features, $\mathscr{CI}$s, which describe some relevant information extracted from particular components of the models, in relation to the input sequence $\mathbf{O}$. A $\mathscr{CI}$ feature either characterizes some properties of the generation path of the sequence $\mathbf{O}$ through the model $\boldsymbol{\lambda}_c$ or the strength with which a specific component of $\boldsymbol{\lambda}_c$ "responds" to $\mathbf{O}$. More formally, $\mathscr{F}_{\mathscr{CI}}(\cdot,\boldsymbol{\lambda}_c):\mathscr{O}^c\to\mathbb{R}^{m_c}$ is a model-dependent mapping defined by m_c components derived from $\boldsymbol{\lambda}_c$. The final HMM-induced vector space is a Cartesian product of all $\mathscr{CI}$-spaces (one for each class), $\mathscr{F}_{\mathscr{CI}}(\cdot,\boldsymbol{\lambda}_1)\times\cdots\times\mathscr{F}_{\mathscr{CI}}(\cdot,\boldsymbol{\lambda}_C)$ such that

$$\mathscr{F}_{HMM}(\mathbf{O})=[\mathscr{F}_{\mathscr{CI}}(\mathbf{O},\boldsymbol{\lambda}_1),\mathscr{F}_{\mathscr{CI}}(\mathbf{O},\boldsymbol{\lambda}_2),\dots,\mathscr{F}_{\mathscr{CI}}(\mathbf{O},\boldsymbol{\lambda}_C)]^T. \tag{1}$$

We introduce now four different $\mathscr{CI}$-mappings, leading to four different HMM-induced spaces (see Appendix A.2 on how to compute the relevant quantities):

1. LL-*space* (log-likelihood space): This is our reference space, similar to the ones proposed in [35,36,45]. The $\mathscr{CI}$-feature becomes the logarithm of the probability that the model $\boldsymbol{\lambda}_c$ has generated the sequence $\mathbf{O}$, i.e. $\mathscr{CI}^{\mathrm{LL}}(\mathbf{O},\boldsymbol{\lambda}_c)=\log P(\mathbf{O}|\boldsymbol{\lambda}_c)$. So, the extracted Component Information tells us how well a sequence is modeled by the given HMM. The resulting C-dimensional vector in HMMVS is defined as $\mathscr{F}^{\mathrm{LL}}_{HMM}(\mathbf{O})=[\log P(\mathbf{O}|\boldsymbol{\lambda}_1),\dots,\log P(\mathbf{O}|\boldsymbol{\lambda}_C)]^T$.
2. state-*space*: Here we measure the contribution of individual states to the most likely generation of the observation sequence. The $\mathscr{CI}$-feature describes how often (and with which probability) the model $\boldsymbol{\lambda}_c$ passes through a particular state S^c_i when observing the sequence $\mathbf{O}$. We start by considering the variable γ^t_i, obtained from the forward–backward procedure. It represents the probability of being in state S^c_i at time t, given the sequence $\mathbf{O}$ of the length T and the model $\boldsymbol{\lambda}_c$. We have

$$\gamma^t_i(\mathbf{O},\boldsymbol{\lambda}_c)=P(Q_t=S^c_i|\mathbf{O},\boldsymbol{\lambda}_c). \tag{2}$$

The relevant information for the state S^c_i of a given model $\boldsymbol{\lambda}_c$ is therefore the sum over time of the probability that $\boldsymbol{\lambda}_c$ passes through that state while emitting $\mathbf{O}$, i.e.

$$\bar{\gamma}_i(\mathbf{O},\boldsymbol{\lambda}_c)=\sum_{t=1}^{T}P(Q_t=S^c_i|\mathbf{O},\boldsymbol{\lambda}_c)=\sum_{t=1}^{T}\gamma^t_i(\mathbf{O},\boldsymbol{\lambda}_c). \tag{3}$$

Notice that $\bar{\gamma}_i(\mathbf{O},\boldsymbol{\lambda}_c)$, the sum of γ^t_i over time t, can be interpreted as the expected number of transitions from S^c_i [1]. It is therefore a natural measure of the importance of state S^c_i in the process of deriving $P(\mathbf{O}|\boldsymbol{\lambda}_c)$. Hence, our $\mathscr{CI}$-feature is defined as $\bar{\gamma}_i(\mathbf{O},\boldsymbol{\lambda}_c)$. If $\bar{\gamma}_i(\mathbf{O},\boldsymbol{\lambda}_c)$ is scaled by T, then the $\mathscr{CI}$-feature becomes independent of sequence length. We, however, do not follow this idea here. It appears that sequence length can be very informative in the experiments, because it captures complexity of the structure encoded in a sequences. We are not disturbed by variable-length sequences as the resulting $\mathscr{CI}$-features can appropriately be scaled for the use of discriminative classifiers.
The total Component Information for the class ω_c is captured by a K-element vector:

$$\mathscr{F}^{\mathrm{S}}_{\mathscr{CI}}(\mathbf{O},\boldsymbol{\lambda}_c)=[\bar{\gamma}_1(\mathbf{O},\boldsymbol{\lambda}_c),\dots,\bar{\gamma}_K(\mathbf{O},\boldsymbol{\lambda}_c)]^T. \tag{4}$$

Finally, each sequence is mapped into a CK-dimensional feature vector which summarizes the importance of each state S^c_i for every model $\boldsymbol{\lambda}_c$ with respect to the input sequence $\mathbf{O}$, i.e. $\mathscr{F}^{\mathrm{S}}_{HMM}(\mathbf{O})=[\mathscr{F}^{\mathrm{S}}_{\mathscr{CI}}(\mathbf{O},\boldsymbol{\lambda}_1),\dots,\mathscr{F}^{\mathrm{S}}_{\mathscr{CI}}(\mathbf{O},\boldsymbol{\lambda}_C)]$.
3. trans-*space*: We focus now on the importance of individual transitions in an HMM. Hence, we characterize the property of $\boldsymbol{\lambda}_c$ by the frequency with which each connection is used in the process of generating $\mathbf{O}$. The basic variable, easily computed from the forward–backward algorithm, is $\xi^t_{(i,j)}$. It represents the probability of passing from state S^c_i at time t to state S^c_j at time $(t+1)$, given the observations and the model, i.e.

$$\xi^t_{(i,j)}(\mathbf{O},\boldsymbol{\lambda}_c)=P(Q_t=S^c_i,Q_{t+1}=S^c_j|\mathbf{O},\boldsymbol{\lambda}_c). \tag{5}$$

Similarly to the previous case we compute the sum over t, deriving:

$$\bar{\xi}_{(i,j)}(\mathbf{O},\boldsymbol{\lambda})=\sum_{t=1}^{T-1}P(Q_t=S^c_i,Q_{t+1}=S^c_j|\mathbf{O},\boldsymbol{\lambda}_c)=\sum_{t=1}^{T-1}\xi^t_{(i,j)}(\mathbf{O},\boldsymbol{\lambda}_c). \tag{6}$$

$\bar{\xi}_{(i,j)}(\mathbf{O},\boldsymbol{\lambda}_c)$ can be interpreted as the expected number of transitions from state S^c_i to state S^c_j [1]. Therefore, our $\mathscr{CI}$-feature is

$\bar{\xi}_{(i,j)}(\mathbf{O},\boldsymbol{\lambda}_c)$. The total Component Information for the class ω_c becomes now a K^2-dimensional vector, defined as

$$\mathscr{F}^{\mathrm{T}}_{\mathscr{C}\mathscr{I}}(\mathbf{O},\boldsymbol{\lambda}_c)=[\bar{\xi}_{(1,1)}(\mathbf{O},\boldsymbol{\lambda}_c),\ldots,\bar{\xi}_{(1,K)}(\mathbf{O},\boldsymbol{\lambda}_c),\bar{\xi}_{(2,1)}(\mathbf{O},\boldsymbol{\lambda}_c),\ldots,\bar{\xi}_{(K,K)}(\mathbf{O},\boldsymbol{\lambda}_c)]^T. \tag{7}$$

The HMMVS is defined by all models, hence each sequence is mapped into a CK^2-dimensional feature vector, summarizing the importance of all transitions $\{S^c_i \to S^c_j\}$ for each model $\boldsymbol{\lambda}_c$ and a sequence $\mathbf{O}$, i.e. $\mathscr{F}^{\mathrm{T}}_{HMM}(\mathbf{O})=[\mathscr{F}^{\mathrm{T}}_{\mathscr{C}\mathscr{I}}(\mathbf{O},\boldsymbol{\lambda}_1),\ldots,\mathscr{F}^{\mathrm{T}}_{\mathscr{C}\mathscr{I}}(\mathbf{O},\boldsymbol{\lambda}_C)]$.

4. emit-*space*: In some applications, emission probabilities may represent the most meaningful part of an HMM. Hence, we characterize the property of $\boldsymbol{\lambda}_c$ by the sum of emission probabilities at a given state. The $\mathscr{C}\mathscr{I}$-feature is defined as $\tilde{\rho}_i(\mathbf{O},\boldsymbol{\lambda}_c)=\sum_{t=1}^T b(O_t|S^c_i)$. The total Component Information for the class ω_c is a K-dimensional vector,

$$\mathscr{F}^{\mathrm{E}}_{\mathscr{C}\mathscr{I}}(\mathbf{O},\boldsymbol{\lambda}_c)=[\tilde{\rho}_1(\mathbf{O},\boldsymbol{\lambda}_c),\ldots,\tilde{\rho}_K(\mathbf{O},\boldsymbol{\lambda}_c)]^T. \tag{8}$$

The final HMMVS is described by all models, hence each sequence is mapped into a CK-dimensional feature vector, $\mathscr{F}^{\mathrm{E}}_{HMM}(\mathbf{O})=[\mathscr{F}^{\mathrm{E}}_{\mathscr{C}\mathscr{I}}(\mathbf{O},\boldsymbol{\lambda}_1),\ldots,\mathscr{F}^{\mathrm{E}}_{\mathscr{C}\mathscr{I}}(\mathbf{O},\boldsymbol{\lambda}_C)]$.

The spaces presented above are the ones most intuitively derived from HMMs. Other spaces can be defined by following the same principle, e.g. such that Component Information is summarized over a couple of states or over a set of transitions entering a particular state. Another possible extension is a fusion of (parts of) the spaces. The flexibility in the definition of Component Information can be utilized to tailor this generic approach into a specific application.

The main problem in our approach is the high dimensionality of some HMM-induced vector spaces, in particular the trans-space. Any traditional feature reduction technique, such as principal component analysis (PCA) can be applied to diminish the impact of curse of dimensionality [16] on classifier performance. Section 4 discusses a few methods that effectively deal with this issue.

2.5. Classifier training in HMMVS

The classification model is now an ensemble $\{\{\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_C\},\mathscr{F}_{HMM},\mathscr{C}\}$, where $\{\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_C\}$ are the individual HMMs describing the classes, $\mathscr{F}_{HMM}$ is a specific HMM-induced vector space, and $\mathscr{C}$ is the final classifier. Since the original sequences are mapped as vectors to an HMMVS via Component Information derived from the trained models, these resulting vectors do not follow independent distributions. Discriminative classifiers are therefore preferred here, because they directly focus on estimating class posterior probabilities instead of modeling class distributions. Such classifiers should also be less affected by the curse of dimensionality.

Ideally, the training set is split into two independent subsets, one used for learning HMMs and the other used for training a classifier in the HMMVS. In practice, however, there is often insufficient data to allow for this. In our study we use a single set, both to determine the HMMs and to train the final classifier. Such data re-use may lead to overtraining. The practice of Multiple Classifier Systems shows, however, that we can still benefit from this generative-discriminative combining scheme, provided that the basic generative HMMs are capable of characterizing informative features (however poorly they may be estimated) and the combiner is simple enough not to overfit. The main task is, therefore, to get sufficiently rich $\mathscr{C}\mathscr{I}$-features; see also [34].

Feature normalization in HMMVS can be important depending on the chosen classifier. Two usual techniques are standardization (zero mean and unit variance) and linear scaling to the $[0,1]$ domain. Normalization is necessary for the LL-space, as it relies on unbounded logarithms of probabilities, which may become very large in modulus for small probabilities. state-, emit- and trans-spaces are bounded, because their features are summations over sequences of probabilities of a fixed maximal length. In this case, it is not clear whether normalization is required. In our experiments we compare normalized and non-normalized spaces, showing that the usefulness of this operation strictly depends on the chosen application and the classifier. Nevertheless, we remove non-informative features in HMMVS with zero (or close to zero in the machine precision) variances as judged on the training set. This may especially be the case, e.g. for trans-space when particular transitions never occur.

a

$$A=\begin{bmatrix}0.10&0.40&0.40&0.10\\0.20&0.40&0.40&0.00\\0.10&0.40&0.30&0.20\\0.50&0.30&0.10&0.10\end{bmatrix}\quad \pi=\begin{bmatrix}0.25\\0.25\\0.25\\0.25\end{bmatrix}\quad B=\begin{bmatrix}\mu_1=1&\sigma_1^2=1\\\mu_2=3&\sigma_2^2=1\\\mu_3=5&\sigma_3^2=1\\\mu_4=7&\sigma_4^2=1\end{bmatrix}$$

Parameters of the HMM generating sequences of class1.

b

$$A=\begin{bmatrix}0.25&0.25&0.25&0.25\\0.25&0.25&0.25&0.25\\0.25&0.25&0.25&0.25\\0.25&0.25&0.25&0.25\end{bmatrix}\quad \pi=\begin{bmatrix}0.25\\0.25\\0.25\\0.25\end{bmatrix}\quad B=\begin{bmatrix}\mu_1=1.1&\sigma_1^2=1\\\mu_2=3.1&\sigma_2^2=1\\\mu_3=5.1&\sigma_3^2=1\\\mu_4=7.1&\sigma_4^2=1\end{bmatrix}$$

Parameters of the HMM generating sequences of class2.

Fig. 1. Two-class synthetic data; sequences are generated by two HMMs. A is a probability transition matrix, $\boldsymbol{\pi}$ denotes the initial state probabilities and $(\boldsymbol{\mu},\boldsymbol{\sigma})$ are the parameters of the Gaussian emission density function.

2.6. Illustrative synthetic example

Now we illustrate both the advantages and drawbacks of the proposed approach with a synthetic example. Our goal is to highlight the situations for which classifiers in HMMVS are superior to the traditional MAP approach. If the training set is sufficiently large and class models are well estimated (i.e. with the correct number of states and topology, and well estimated probability functions), then the MAP approach is more beneficial.

We consider a two-class problem, in which sequences of length 10 are generated from the two 4-state Gaussian HMMs specified in Fig. 1. Since these two models are very similar (especially in the emission probability density models), the discrimination becomes difficult. Our experimental evaluation is specified as follows:

- Disjoint training and test sets are randomly generated. The training set varies from 2 to 500 sequences per class. The test set contains 1000 sequences, 500 examples per class.
- Standard either 4-state or 8-state HMMs are trained per class, stopping at the likelihood convergence. The former case corresponds to a perfect model topology, while the latter simulates an incorrect model.
- Non-normalized state- and trans-spaces are constructed on the training set. The state-space is either 8D or 16D, while the trans-space is either 32D or 128D, depending on the number of states in the HMMs.
- Test sequences are classified either by the MAP approach applied to the two trained HMMs, or by classifiers trained in the HMM-induced vector spaces. The classifiers are chosen such that they do not suffer much from high-dimensional feature spaces

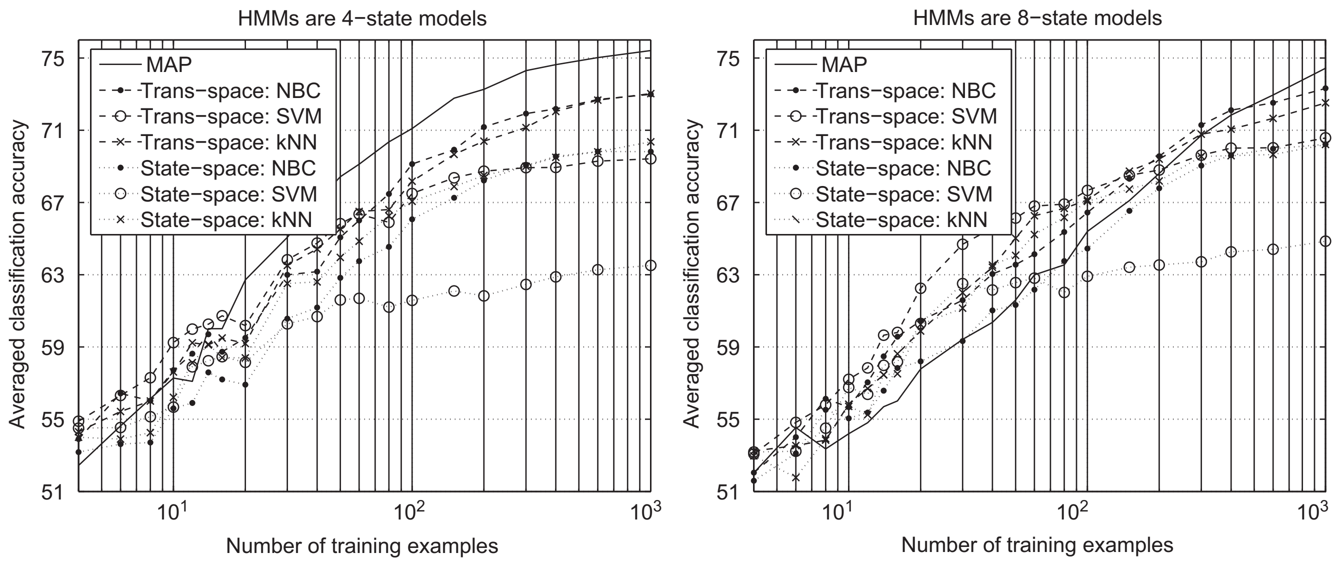

Fig. 2. Averaged classification accuracy (in %) for the synthetic data as a function of the training size n_{tr}. Note that x-axis has a logarithmic scale. Concerning the correct 4-state class models (left), the standard errors of the average accuracy vary from 1.0% to 1.2% for $n_{tr} = 6$, via 0.9% (trans-space) or 1.2% (state-space) for $n_{tr} = 20\%$ to 0.55% for $n_{tr} = 100$ and, finally, 0.3–0.45% for large n_{tr}. Concerning the incorrect 8-state class models (right), the standard errors of the average [49] results vary from 0.9% to 1.1% for $n_{tr} = 6$, via 0.7–0.8% for $n_{tr} = 20$ to $\approx 0.5\%$ for $n_{tr} = 100$ and, finally, 0.3–0.4% for large n_{tr}.

Table 1
Characteristics of the data.

Data	C	Size	No. per class (min–max)	Taining/test	Sequences	Sq. length (min–max)	K
Chicken	5	446	61–117	20× 50–50% holdout	Discr. chain-codes	167–48	3
Chicken	5	446	61–117	20× 50–50% holdout	1D curvature	10–81	3
Auslan	10	270	27	20× fivefold CV	22D sensors data	30–102	3
Japan-Vowel	9	640	61–118	Fixed: 270/370	12D LPC coef.	7–29	5
Alcoholic	2	600	300	Fixed: 600–600	2D EEG signals	256	9

C denotes the number of classes. *K* is the number of states in the HMMs.

(zero-valued features that occur during training are neglected). The classifiers are:

- ○ the naive Bayes classifier (NBC—[48]), assuming that features are conditionally independent; individual class-conditional marginal densities are estimated by 1D histograms;
- ○ the support vector machine (SVM—[50]) with a Gaussian kernel. ν is estimated by the leave-one-out nearest neighbor error on the training set, the scale σ of the Gaussian kernel is determined in a 20-step optimization based on the fivefold cross-validation error estimation; and
- ○ the k-nearest neighbor rule (k-NN—[47]). k is estimated on the training set by minimizing the leave-one-out classification error.

- The experiments are repeated 30 times and the average accuracy is reported.

Fig. 2 shows curves of average classification accuracy. We can observe that classification performance increases with the increasing training size, and that the state-space is inferior to trans-space, especially when SVM is trained. Most importantly, given the correct model topology ($K = 4$, left subplot), we can observe that the MAP results become increasingly better with a growing number of training sequences, finally outperforming the classifier in the HMMVS. For small sample size situations, however, the classifiers in the trans-space outperform the MAP approach. The right subplot of Fig. 2 shows the results for the situation where an incorrect model topology, 8-state HMMs, is used. In this case, the accuracy deteriorates for all classification strategies, but the MAP is suffering the most. The trained classifiers are capable of recovering from the improper estimates by using supervised training information in their discriminative learning. The k-NN and SVM trained in the trans-space provide a substantial improvement over the baseline MAP result for small and moderate training sizes and a similar result for large training sizes. This experiment supports the idea that our approach can be recommended for small training sets or when the chosen HMMs do not fit to the observed phenomena.

3. Experimental evaluation

The usefulness of HMMVS is evaluated on four applications: shape recognition (*Chicken* data), gesture classification (*Auslan* data), speaker verification (*Japan-Vowel* data) and EEG signal classification (*Alcoholic* data). These problems represent various scenarios, such as discrete symbol sequences versus continuous signals, difficult tasks versus easy tasks, small versus large number of classes, and finally, small or moderate training sets. The data sets are summarized in Table 1 and their further descriptions (including derivation of sequences) are given in Appendix A.1.

3.1. Experimental details and parameter setting

We assume that we deal with fully ergodic HMMs. Initialization is random both for the transition probabilities and initial state probabilities. In case of continuous signals, the emission probability models are initialized by a Gaussian Mixture clustering. In case of discrete symbol sequences, 20 independent training runs are

Table 2
Average classification accuracy (in %) in HMM-induced vector spaces for the *Chicken* data.

Classifier	LL (N) 5D	state (N) 15D	state 15D	trans (N) 45D	trans 45D	emit (N) 15D	emit 15D
Discrete sequences							
`lda`	62.2 (1.0)	72.2 (0.8)	72.2 (0.8)	82.6 (0.6)	82.8 (0.6)	72.3 (0.7)	72.3 (0.7)
`loglc`	64.2 (0.9)	72.3 (0.8)	72.3 (0.8)	46.6 (4.7)	49.5 (5.0)	73.0 (0.5)	73.0 (0.5)
`nbc`	53.5 (0.7)	62.5 (0.7)	62.5 (0.7)	80.7 (0.6)	80.7 (0.6)	64.7 (0.5)	64.7 (0.5)
`svm`	61.9 (1.0)	68.7 (1.1)	64.2 (0.7)	80.4 (0.6)	69.4 (0.8)	69.0 (0.6)	62.6 (0.7)
`rbsvm`	62.0 (0.7)	72.8 (0.7)	71.3 (0.7)	81.1 (0.5)	74.9 (0.5)	75.8 (0.6)	74.0 (0.7)
`knn`	55.9 (0.7)	67.8 (0.7)	67.8 (0.9)	72.7 (0.9)	68.2 (0.7)	69.4 (0.7)	69.2 (0.5)
Continuous sequences							
`lda`	70.7 (0.5)	72.5 (0.5)	72.5 (0.5)	75.5 (0.4)	76.1 (0.4)	73.0 (0.8)	73.0 (0.8)
`loglc`	69.5 (0.6)	71.9 (0.7)	71.9 (0.7)	36.5 (2.5)	36.0 (2.5)	71.9 (0.7)	71.9 (0.7)
`nbc`	58.0 (0.9)	70.0 (0.7)	70.0 (0.7)	71.5 (0.9)	71.5 (0.9)	68.9 (0.9)	68.9 (0.9)
`svm`	63.8 (0.8)	69.9 (0.8)	55.7 (0.9)	75.2 (0.6)	65.8 (0.8)	73.7 (0.6)	60.0 (0.9)
`rbsvm`	75.5 (0.5)	79.9 (0.6)	79.1 (0.5)	80.0 (0.6)	79.8 (0.4)	80.1 (0.5)	78.4 (0.5)
`knn`	71.0 (0.7)	77.4 (0.5)	77.3 (0.5)	74.4 (0.5)	77.0 (0.7)	77.2 (0.4)	75.2 (0.5)

The numbers in parenthesis describe the standard errors of the mean results. (N) denotes a normalized vector space.

Table 3
Average classification accuracy (in %) in HMM-induced vector spaces for the *Auslan* data.

Classifier	LL (N) 10D	state (N) 30D	state 30D	trans (N) 90D	trans 90D	emit (N) 30D	emit 30D
`lda`	87.2 (0.4)	82.7 (0.6)	82.7 (0.6)	85.1 (0.4)	85.2 (0.5)	51.1 (0.6)	10.0 (0.0)
`loglc`	82.7 (0.5)	59.9 (2.0)	59.9 (2.0)	70.3 (1.1)	70.3 (1.1)	11.6 (0.4)	11.6 (0.4)
`nbc`	63.8 (0.5)	77.1 (0.6)	77.1 (0.6)	84.4 (0.5)	84.4 (0.5)	30.2 (0.7)	10.0 (0.0)
`svm`	84.5 (0.5)	78.5 (0.5)	52.2 (0.9)	82.2 (0.5)	51.3 (0.9)	60.6 (0.5)	17.1 (0.5)
`rbsvm`	86.1 (0.4)	82.1 (0.5)	77.1 (0.6)	84.5 (0.6)	77.3 (0.5)	62.0 (0.6)	11.4 (0.4)
`knn`	79.2 (0.5)	70.4 (0.9)	62.8 (1.0)	72.0 (0.7)	63.0 (1.0)	47.1 (0.5)	11.5 (0.4)

The numbers in parenthesis describe the standard errors of the mean results. (N) indicates a normalized vector space.

performed, starting from a random initialization, picking the best likelihood model as the representative. Our implementation relies on the Murphy's Matlow hidden Markov model toolbox.[1]

For simplicity, the number of states is fixed for all classes in each problem. It is chosen beforehand by using some preliminary analysis or based on a priori knowledge (e.g. published papers). The same HMMs are used in the MAP scheme as in the proposed generative-discriminative framework.

Different strategies are used to estimate classification accuracy for each problem in order to make valid comparisons with the already published results. Various classifiers, as implemented in `Prtools` [46], are tested in our HMM-induced vector spaces:

`lda` *linear discriminant analysis* [47]: It assumes identical class conditional densities, modeled by Gaussians. The linear decision function is determined by the Bayes rule.

`loglc` *logistic linear classifier* [48]: It models the log-odds (logarithm of the ratio of class posterior probabilities) as linear functions. The weights are optimized by ML.

`nbc` *naive Bayes classifier* [48]: It assumes that features are conditionally independent. Here the individual class-conditional marginal densities are estimated by 1D histograms.

`knn` *k-nearest neighbor rule* [47]: k is estimated on the training set by minimizing the leave-one-out classification error.

`svm` *linear support vector machine* [50]: This is the ν-SVM rule applied to a linear kernel. ν is estimated by the leave-one-out nearest neighbor error on the training set.

`rbsvm` *radial basis support vector machine* [50]: This is the ν-SVM rule applied to a Gaussian kernel. ν is estimated by the leave-one-out nearest neighbor error on the training set. The scale σ of the Gaussian kernel is determined in a 20-step optimization based on the fivefold cross-validation error estimation.

`1-svm` 1-*norm Support Vector Machine* [51]: This is a linear programming machine optimizing the ℓ_1-norm of the weights. It may serve as a feature selector.

`lda`, `loglc` and `nbc` are not affected by a linear scaling of features, in contrast to classifiers that depend on a distance or inner product. Note also that both `lda` and `nbc` cannot be any longer considered as purely generative classifiers in our framework. The reason is that every feature in the HMMVS conveys information related to a specific class. If all models are used, features provide class-related responses of the trained HMMs. So, even if we model class-conditional densities, they incorporate information from all the classes thanks to the class-related features. The final classifier is trained in the HMMVS, after removing features with zero variances, which also reduces the space dimension.

3.2. Results and discussion

Tables 2–5 report the results of our experimental evaluation. When appropriate (in case of cross-validation or hold-out experiments), averaged classification accuracies and their standard errors are displayed. The tables present classification performance in both normalized (by unit variance) and non-normalized state-, emit- and trans-spaces for the *Chicken*, *Auslan*, *Japan-Vowel* and *Alcoholic* data, correspondingly. The performance of the traditional MAP approach and the best performances in the proposed framework are shown in Table 6. While analyzing all results, the following observations can be made:

- *Discriminative classifiers in the* HMMVS *versus MAP*: In general, the proposed approach is better than the standard MAP classification scheme. The improvement is negligible when HMMs are performing very well, as observed for the *Japan-Vowel* data. Improvements in accuracy are impressive when the estimated HMMs are weak, either undertrained or badly estimated, as it can be observed for the *Chicken* data.

[1] Downloadable from http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

Table 4
Average classification accuracy (in %) in HMM-induced vector spaces for the *Japan-Vowel* data.

Classifier	LL (N) 9D	state (N) 45D	state 45D	trans (N) 225D	trans 225D	emit (N) 45D	emit 45D
lda	95.7	89.7	89.7	83.2	84.6	66.5	61.9
loglc	94.6	86.5	86.5	80.5	80.5	24.1	24.1
nbc	90.0	86.0	86.0	84.9	84.9	66.0	66.0
svm	96.8	90.0	68.1	90.0	64.3	83.2	65.7
rbsvm	96.2	91.1	92.2	89.5	91.6	87.6	88.1
knn	94.1	84.9	85.7	85.7	85.7	84.1	90.0

The training and test sets are fixed. (N) indicates a normalized vector space.

Table 5
Average classification accuracy (in %) in HMM-induced vector spaces for the *Alcoholic* data.

Classifier	LL (N) 2D	state (N) 18D	state 18D	trans (N) 162D	trans 162D	emit (N) 18D	emit 18D
lda	58.8	59.0	59.0	59.8	60.0	58.2	58.2
loglc	57.0	59.5	59.5	46.2	46.2	60.0	60.0
nbc	54.7	61.8	61.8	61.5	61.5	60.8	60.8
svm	57.0	60.5	60.5	59.2	60.2	59.8	60.3
rbsvm	58.8	61.8	62.5	58.3	64.2	65.0	62.2
knn	58.3	62.3	63.2	61.5	59.2	62.2	58.5

The training and test sets are fixed. (N) indicates a normalized vector space.

Table 6
Comparison between the standard MAP results and the best results of the proposed scenario.

Problem	MAP	Best	Type	2nd best	Type
Chicken (chain-code)	51.0 (0.8)	82.8 (0.6)	lda + trans-space	81.1 (0.5)	rbsvm + trans-space
Chicken (curvature)	57.4 (0.8)	80.1 (0.5)	rbsvm + emit-space	80.0 (0.6)	rbsvm + trans-space
Auslan	82.2 (0.4)	87.2 (0.4)	lda + LL-space	86.1 (0.4)	rbsvm + LL-space
Japan-Vowel	97.6	96.8	svm + LL-space	96.2	rbsvm + LL-space
Alcoholic	56.7	65.0	rbsvm + emit-space	64.2	rbsvm + trans-space

- *Comparison between HMM-induced vector spaces*: Given C classes, each modeled by a K-state HMM, the dimensions of the LL-space, state-space, emit-space and trans-space are C, CK, CK and CK^2, respectively. If the number of classes is large and/or generative HMMs are well estimated, then the LL-space captures sufficient discriminative information in order to enable a very good classification performance. For instance, the best results for the 9-class *Japan-Vowel* data are reached in the LL-space. In this case, the high increase of the number of features in the state- and trans-spaces does not improve discrimination. On the other hand, when the HMMs are poor, then building an HMMVS via the model components works very well, as can clearly be observed for the *Chicken* and *Alcoholic* data. The best results are achieved by rbsvm in the emit-space and trans-space. Obviously, trans-space is the most flexible, but may also be inadvisable when the dimensionality grows exponentially (due to a high number of states and many classes). emit-space or state-space should be considered in such a case.
- *Normalization*: Normalization in the LL-space is recommended to prevent unbounded feature values. In other cases, normalization can be useful, depending on the task and the classifier. For instance, svm, the linear ν-SVM seems to especially benefit from feature scaling, as observed for the *Chicken*, *Auslan* and *Japan-Vowel* data.
- *Choice of classifier*: There is no single best classifier, although ν-SVM based either on linear (svm) or Gaussian kernel (rbsvm) often performs best or close to the best. In addition, also lda gives very good results.

4. Dealing with space dimensionality

A potential drawback of using HMM-induced vector spaces is the curse of dimensionality [52]. The log-likelihood space is C-dimensional, state-space and emit-space are CK-dimensional, while trans-space is CK^2-dimensional. The problem is even more enhanced when two or more feature spaces are fused via Cartesian product. This is relevant for density-based classifiers, such as the linear or quadratic discriminant, which suffer from a large number of features. In the trans-space, the training size has to increase exponentially in order to permit a reliable estimation of the class-conditional densities. Supervised and unsupervised feature reduction techniques are standard solutions to diminish the negative effect of high dimensionality [47]. Here, we will perform both unsupervised and supervised reduction, by using principal component analysis and forward feature selection (FFS) based on the 1-NN classification criterion.

Other classifiers are less affected by the curse of dimensionality. The most notable example is SVM [53], whose complexity depends on the number of support vectors instead of the space dimension. It appears to perform well in high-dimensional spaces. A variant of SVM is a 1-norm SVM [51,54], optimizing the ℓ_1-norm of its weight vector, which leads to a sparse solution. Hence, this method implements a feature selection mechanism inside the classifier training.

An alternative possibility of feature reduction is to focus on *a single HMM* to represent the information on sequences from *all the classes*. If an HMM is trained on a particular class such that it can capture the class characteristics well, then sequences from other classes may respond in a way that reflects the differences between the classes, hence provide discriminative information. Some substructures in a new sequence may be similar to the HMM-description of the chosen class, while other substructures may be very different. As a result, a few states or transitions may have very high Component Information, while others will have a low one. For a sufficiently rich HMM, several classes may respond fundamentally differently to this HMM, and may therefore be separable in the vector space induced by a single one-class HMM. This is in line with recent outcomes obtained for the Fisher Score approach presented in [55], where a subset of models have been successfully employed to build the space.

Table 7
Classification performance in the normalized LL + trans-space (in %) for the *Chicken* data (curvature sequences).

HMMVS	Dim	`lda`	`knn`	`1-svm`	`svm`	`rbsvm`
HMMs are 2-state models						
Original	25	**76.8 (0.5)**	**75.0 (0.8)**	**75.3 (0.6)**	**75.9 (0.6)**	**79.7 (0.5)**
FeatSel	11	73.4 (0.7)	**73.2 (0.5)**	69.0 (1.1)	71.6 (1.2)	77.9 (0.5)
PCA99	11	73.9 (0.8)	**74.8 (0.8)**	71.2 (1.0)	**76.5 (0.6)**	**79.4 (0.6)**
HMM-1	5	59.1 (0.8)	68.8 (0.6)	41.4 (1.4)	58.9 (0.9)	69.4 (0.5)
HMM-2	5	61.5 (0.7)	65.0 (0.6)	45.8 (1.2)	61.3 (0.8)	67.6 (0.6)
HMM-3	5	59.7 (0.9)	67.4 (0.7)	41.4 (1.5)	59.8 (1.3)	69.3 (0.6)
HMM-4	5	60.1 (0.8)	68.2 (0.8)	47.8 (1.0)	60.4 (0.9)	69.1 (0.5)
HMM-5	5	65.0 (0.7)	68.6 (0.7)	47.8 (1.4)	67.5 (0.8)	70.9 (0.6)
HMMs are 5-state models						
Original	130	66.1 (0.6)	**77.6 (0.6)**	**75.0 (2.6)**	79.0 (0.6)	**79.6 (0.6)**
FeatSel	74	68.8 (0.8)	**76.0 (0.9)**	**72.8 (2.5)**	71.7 (1.7)	**77.9 (0.9)**
PCA99	33	**76.1 (0.6)**	**77.8 (0.4)**	**76.5 (0.6)**	**81.9 (0.6)**	**79.3 (0.5)**
HMM-1	26	73.8 (0.4)	74.7 (0.6)	72.4 (0.7)	75.4 (0.9)	73.3 (0.8)
HMM-2	26	72.9 (0.8)	73.5 (0.8)	71.4 (0.8)	72.7 (0.7)	73.1 (0.9)
HMM-3	26	74.2 (0.7)	74.2 (0.6)	**71.8 (2.5)**	75.4 (0.7)	76.3 (0.7)
HMM-4	26	73.2 (0.7)	74.7 (0.8)	73.8 (0.6)	75.5 (0.6)	75.8 (0.8)
HMM-5	26	**75.1 (0.6)**	73.8 (0.7)	74.0 (0.7)	74.3 (0.7)	74.1 (0.7)

The best performance per classifier (and the ones that are not significantly worse) are indicated in bold. Results are averaged over 20 cross-validation runs.

We use *Chicken* data to investigate the usefulness of the above-discussed feature reduction approaches for the construction of HMMVS. Table 7 shows the classification performances of several classifiers in the normalized LL + trans-space (obtained by a Cartesian product of the LL-space and trans-space with features scaled to unit variances) for different feature reduction techniques. HMMs are either 2-state or 5-state models. First, the results in the upper part show the performance in the original HMM-induced space based on 2-state HMMs and in the spaces reduced by forward feature selection (FeatSel) and PCA retaining 99% of the variance (PCA99). On average, these techniques reduce the dimension to 11 (depending on the fold in the cross-validation runs). Next, we present results when just a single HMM, trained on a particular class, is used to build HMMVS.

The performances of five classifiers, `lda`, `knn`, `1-svm`, `svm` and `rbsvm`, indicate that feature reduction is counterproductive for these HMM-induced spaces. Due to the low number of states in the HMMs ($K = 2$), the original HMMVS space has a relatively low dimension (25D), and complex and flexible models such as `knn` and `rbsvm` perform well in this space. The best classification performance is obtained by `rbsvm` and `svm`. The `1-svm` classifier is somewhat too aggressive in dimensionality reduction as it often finds solutions based on 2–3 features only.

Surprisingly, the performance in vector spaces induced by a single HMM is very high. Although such spaces are only 5D, `knn` and `rbsvm` reach more than 68% accuracy. This is very competitive with the standard MAP performance of $\approx 57\%$. The HMM trained for the fifth class, HMM-5, seems to provide the best discrimination; the other class models work similarly well.

The performance can be improved further by increasing the complexity of individual HMM models. The bottom part in Table 7 shows classification accuracies for HMM-induced spaces with 5-state HMMs. The original 130D LL+trans-space is relatively "empty", which enhances the separability between the classes for classifiers that can handle such high-dimensional spaces. The linear SVM, `svm`, is overall the best. `lda` fails in this space. When the dimension is reduced by PCA, all classifiers achieve their best performance.

Good results are also obtained in an HMM-induced vector space defined by a single model; the performance reaches more than 76%. A single 5-state HMM can encode the most important characteristics of all the classes in the resulting HMMVS, such that each class has a different fit to the submodels (components) in the HMM. By using all models, the performance is improved from 76% to $\approx 81\%$, indicating that different HMMs are only somewhat complementary. Some HMMs seem to have a wider range of submodels (components) than other HMMs, and are therefore more suitable for encoding all the classes. For instance, the HMM-4 seems to lead to somewhat better performance than other models do. A subset selection of HMMs may therefore be a worthwhile effort.

In summary, the experiments demonstrate that a hidden Markov model of a single class can be used to differentiate between classes in the HMM-induced space, provided that it is a sufficiently complex model. This holds even for the classes for which the model was never trained. This is possible because of different class-related responses to the model, as the extracted component features are informative enough to discriminate between the classes. This alleviates the demand of estimating an HMM for each class separately, which becomes hard to satisfy when many classes are present, or when new classes may appear. In these cases one or just a few generic HMMs can be used to represent all the objects from all the classes.

This opens the possibility of a very general class description, without using specific characteristics of the class at hand. In the case of 2D shape recognition, one complex HMM may be trained on a very large data set that contains most of the possible variations in curves that may be expected. This HMM is then hoped to characterize typical substructures in these data, that later can be used as features for a discriminative classifier trained on the HMM outputs.

5. Comparison with the Fisher Score approach

Our approach exploits a set of generative models (namely HMMs) to derive a feature space, in which a discriminative classifier is trained. It is therefore related to the family of dynamic kernels [38–40,56,57], in which functions describing the similarity between variable-length sequences are proposed. Such kernels, employed with kernel-based methods, are shown to be useful and appropriate in several different scenarios. The most famous and investigated one is the so-called Fisher Kernel [38], first advocated in the context of protein sequence analysis and proposed as a general way of mixing generative and discriminative models for classification. The basic idea is to employ a generative model to project objects to a suitable feature space, where a meaningful similarity/distance measure could be defined, leading to a kernel. In particular, the Fisher Kernel approach measures the relation between the objects by comparing them in the tangent space induced by the trained generative model. This space has a number of desirable characteristics, such as the possibility of measuring geodesic distances between points along the manifold (leading to the concept of natural gradients [58]). In practice, each object is represented by a feature vector, whose components are called Fisher Scores, defined by derivatives of the log-likelihood of the generative model with respect to all parameters; the dimensionality equals the number of parameters. The kernel can be defined in different ways in the resulting space; the inner product was used in [38].

Some interesting extensions of the basic Fisher Kernel approach have been proposed in order to enhance the scheme: e.g. in [41] the derivatives were computed on the ratio of the likelihoods of generative models (this increasing the discriminative information); in [45], different derivative-based feature spaces (called Score Spaces) were obtained by exploiting high order Taylor expansions; in [44], finally, higher order derivatives were also considered when building the spaces.

It is evident that the proposed HMM-induced spaces share some structure with these spaces, because all rely on the generative models for extracting suitable features. Nevertheless, the spaces are different in a more general setting because they are derived based on fundamentally different assumptions. In the Fisher Score approach

Table 8
Percentage improvements of the HMM-induced vector space over the Fisher Score space.

Classifier	*Chicken* (chain-code)	*Chicken* (curvature)	*Auslan*	*Japan-Vowel*	*Alcoholic*	Average over experiments
`lda`	+1.2	−0.4	+77.2	+87.3	+2.2	+33.5
`loglc`	+22.6	+5.5	−14.0	+11.6	+5.3	+6.2
`nbc`	+32.8	−1.0	+44.1	+81.6	−3.0	+30.9
`svm`	+1.1	−3.9	−11.3	+7.6	−0.8	−1.5
`rbsvm`	+0.1	−1.5	−10.7	+13.5	+4.2	+1.1
`knn`	+2.5	+0.7	−14.5	+6.8	+1.7	−0.5
Avr. over classifiers	+10.0	−0.1	+11.8	+34.7	+1.6	+11.6

(and in almost all its extensions) the basic tool is *differentiation*, namely features are obtained through the gradient of the model's log-likelihood, whereas in our case the basic tool is *marginalization*, namely we compute marginalized probabilities through the generative models (which, depending on the case, may resemble gradients).

We will now compare the proposed spaces with the Fisher Score spaces, as the latter represent the baseline approach, and their usefulness has been largely demonstrated in the HMM-based classification [38,42,43].

5.1. Fisher Score space versus HMM-induced spaces

In order to have a thorough comparison, we repeat the experiments described in Section 3 by using a *normalized* version of the Fisher Score space. This is achieved by a normalization step applied to the space defined in [38], as proposed in [41]. In the original work, the typical application scenario is a two-class case: positive class versus negative class. A single generative model is trained on the examples of the positive class (or on both classes) and used to define a Fisher Score space, in which a linear SVM is employed. Although proposals exist that effectively exploit the multi-class scenario, e.g. [26], here we use a simple extension, which permits us to have a direct and clear comparison between the information extracted by Fisher Scores and Component Information features in our approach. In particular, we train one HMM per class, mapping each object to a Fisher Score vector depending on all models. Different classifiers are trained in the resulting space.

For each experiment, the Fisher Score space is computed by using exactly the same HMMs as in our HMM-induced spaces. To have a direct comparison, Table 8 shows the improvements obtained by our best spaces with respect to the Fisher Score approach, in terms of percentage. A positive value means that the proposed approach is able to improve the accuracy with respect to the Fisher Score. The accuracies are shown for all experiments and all classifiers. Moreover, the averages among experiments and among classifiers are also noted.

We can see that our HMM-induced space is generally better than the Fisher Score space, depending on the classifier. The latter is very competitive when using support vector machines (similarly as in [38]) as observed for the 10-class *Auslan* data. In other cases, the Fisher Score approach possibly suffers from the high dimensionality of the space, leading to very bad classifiers. This is especially evident for `lda`, which estimates singular covariance matrices. Actually, it should be noted that in the Fisher Score case, the features are explicitly linked to the model parameters (because they are derivatives with respect to model parameters), whereas in our approach they can also be derived from general components, such as states or emission functions, leading to more manageable space dimensions.

In addition, we also want to emphasize that normalization of the original Fisher Score spaces [38] is essential. Without normalization, the classification performance deteriorates significantly for `rbsvm` and `svm` as well as for other classifiers. When we compared classification performance in both original Fisher Score spaces and HMM-induced spaces (not reported due to lack of space), the latter led to far better results. As such, normalization is not always necessary in HMMVS.

6. Final discussion and future perspectives

Hidden Markov models are traditionally applied to sequence modeling problems that occur in a variety of fields. In the standard classification context, a model is trained per class and a sequence is assigned to the class based on the maximum a posteriori probability. This works well provided that the HMMs characterize the classes well. The accuracy of the HMM-based classification usually deteriorates when imperfect models are estimated, typically due to the lack of knowledge or insufficient training data. As a simple solution, we propose to use an integrated generative-discriminative framework leading to a component-based discriminative classification. This provides a simple alternative to sophisticated and already well addressed complex extensions of hidden Markov models [17–20], discriminatively trained HMMs [21–24] and further generalizations towards probabilistic discriminative models [27,28].

The essence of our proposal is to train feature-based classifiers in a vector space induced by the HMMs, trained individually per class. The features are chosen to encode meaningful information about specific components of the models. Given K-state one-class HMMs, $\boldsymbol{\lambda}_c$, $c = 1, 2, \ldots, C$, four spaces are studied here:

1. C-dimensional log-likelihood space, in which every feature conveys the importance, i.e. the log-likelihood, of a particular HMM.
2. CK-dimensional state-space, in which each feature derived from $\boldsymbol{\lambda}_c$ encodes the importance of state S_i^c while computing $P(\mathbf{O}|\boldsymbol{\lambda}_c)$ for a test sequence $\mathbf{O}$. The feature describes the expected number of transitions from S_i^c.
3. CK-dimensional emit-space, in which each feature derived from $\boldsymbol{\lambda}_c$ encodes the importance of the emission probability of a given state S_i^c.
4. CK^2-dimensional trans-space, in which each feature derived from $\boldsymbol{\lambda}_c$ encodes the importance of the transition $S_i^c \rightarrow S_j^c$ while computing $P(\mathbf{O}|\boldsymbol{\lambda}_c)$ for a test sequence $\mathbf{O}$. The feature describes the expected number of transitions from S_i^c to S_j^c.

In our experimental evaluation on four different applications we observe that the proposed approach outperforms the standard MAP scheme. The improvements are particularly significant when classical HMMs are not suitable to solve the problem, due to bad assumptions, incorrect model definition or small training sets. When HMMs work properly, the proposed scheme leads to a comparable classification accuracy. The main drawback of our approach lies in a possibly high dimension of the resulting HMM-induced vector spaces, especially trans-space. Some ways of dealing with this issue include linear feature extraction, such as PCA, or forward feature selection.

In addition, we also compare discriminative power of HMM-induced vector spaces with that of the Fisher Score spaces. In essence, both proposals train a discriminative classifier in a fixed-dimensional vector space derived from generative HMMs. Nevertheless, in the case of Fisher Score space, the features are explicitly linked to the model parameters, whereas in our approach they are derived from general components (such as states). These components can be associated with either a single parameter (trans-space) or a group of parameters (state-space), which leads to spaces with better manageable sizes. Moreover, SVM is often the only classifier used in the Fisher Score space, while other classifiers may be more effective. However, the most significant difference is the procedure used to derive the features. In the Fisher Score approach they are obtained through *differentiation*, while in our case, they are obtained through *marginalization*. Experimentally, we found that the Fisher Score spaces are competitive to HMM-induced vector spaces for SVM, but generally inferior with respect to other classifiers, such as `lda`, `loglc`, `nbc` and `knn`. Note that *normalized* Fisher Score spaces are employed in our experimental comparison as otherwise simply poor results were obtained.

In summary, the proposed component-based discriminative classification for hidden Markov models can be recommended as a good alternative to the classical HMM-based classification, especially for imperfect models or small sample size problems. In particular, this scheme may be successfully exploited in the applications, where pre-trained models are already available (for example when new data arrive) or are partially known (due to partial prior knowledge on the problem). Another possible field is the set of applications where some of the models are appropriate and others are not, due to either partial or poor knowledge about the problem or due to partial/inequal observability of the phenomenon. In such a scenario it seems possible to enrich the discrimination by building a good discriminative space from only the accurate models.

New interesting questions, of both immediate and further scope, can now be posed. These are:

- Is it sufficient to build one or a few complex models in the generative step, or is it preferred to train many simple models, instead? When a few models are chosen, this may also effectively address the issue of the dimensionality reduction in the HMM-induced vector spaces.
- Is it a sound strategy to employ models trained on many different data and classes, such as a "general shape HMM", for example?
- How can we benefit by incorporating label information in the generative step?
- How big should the representation set (i.e. the set used to learn the HMMs and to build the corresponding vector space) be with respect to the set used for training the classifiers? Moreover, should these be disjoint?
- Which are the best components to be employed for a given problem?

Some of these issues are currently under investigation.

Finally, our proposed framework of deriving fixed-dimensional vector spaces from HMMs can naturally be employed for other generative models: this is especially recommended with respect to the Fisher Score spaces (and its derivation) when component-based marginalization is possible and easier than computation of derivatives of the log-likelihood with respect to the parameters.

Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (Contract 213250). This research is also supported by the Dutch Technology Foundation STW, applied science division of Netherlands Organization for Scientific Research (NWO), the technology program of the Dutch Ministry of Economic Affairs and the Engineering and Physical Sciences Research Council fellowship no. EP/D066883/1 in the UK. Elżbieta Pękalska wishes to thank Magnus Rattray for clarifying some issues on emission probabilities in HMMs. Manuele Bicego wants to thank Marco Cristani for helpful discussions.

Appendix A.

A.1. Description of data sets

In this section some details about the classification problems addressed in the experimental evaluation are given.

A.1.1. The 2D shape recognition

Recognition of 2D shapes is an unconventional application of HMMs, even though promising results have been reported [6,7,59]. Here, we choose to study the Chicken Pieces Database, http://algoval.essex.ac.uk:8080/data/sequence/chicken/, denoted also as *Chicken* data [60]. This set consists of 446 binary images of chicken pieces. Each piece belongs to one of the five classes, representing specific parts of a chicken: wing (117 samples), back (76), drumstick (96), thigh and back (61), and breast (96). The shapes are usually first described by contours, which are further encoded by suitable sequences. This poses a difficult classification task. The results published in [61] report a baseline leave-one-out accuracy of $\approx 66\%$ by using the 1-NN on the Levenshtein (non-cyclic) edit distance.

In our experiments, two different sequence representations are used to model contours, chain codes and curvature angles. In the first case, a standard 8-direction chain encoding procedure is applied to each image. The resulting chain codes are then transformed into rotation-invariant representations. Finally, discrete HMMs are used to model these classes of symbol sequences. In the second case, we derive curvature sequences as in [7]. First, contours are extracted by using the *Canny* edge detector; the boundary is then approximated by segments of approximately fixed length. Then, at any given point x the curvature value is derived as an angle between two consecutive segments intersecting at x. The initial point is the rightmost point lying on the horizontal line passing through the object centroid, following the boundary in a counterclockwise manner. Classes of curvature sequences are finally modeled by continuous Gaussian HMMs.

After a preliminary evaluation, HMMs are trained with $K=3$ states for all classes and both representations. The original set is split into the training and test sets, in the ratio of 50–50%. The classification runs are averaged over 20 hold-out experiments.

A.1.2. Gesture recognition

We study here high-quality recordings of Australian sign language signs. This data set consists of sample of Australian signs [62]; see http://kdd.ics.uci.edu/databases/auslan2/auslan.data.html. We will denote it *Auslan* data. Samples from a single native signer were collected over a period of nine weeks, using high-quality position trackers and instrumented gloves (resulting in 22-D observations). Twenty-seven samples per sign were collected, the average recording length of each sign is approximately 57 frames. In the reference paper [63], two different scenario are considered: (1) 95 sign-classes, with 2565 signs in total and (2) 10 sign-classes. We follow the second scenario here, i.e. $C = 10$.

Continuous Gaussian 3-state HMMs are employed, directly modeling the signals acquired from the sensors. In order to get comparable results to the ones presented in [63], the performance of our classification schemes is computed by using 20 repetitions of a five-fold cross-validation.

A.1.3. Speaker verification

Speech analysis is the historical application of HMMs, with a plethora of available publications. We address here a speaker recognition problem by using the Japanese Vowel data, http://kdd.ics.uci.edu/databases/JapaneseVowels/JapaneseVowels.html, denoted also as *Japan-Vowel* data. Nine male speakers uttered two Japanese vowels /ae/ successively. For each utterance a discrete-time series with 12 LPC cepstrum coefficients was obtained; see [64]. Time series length lies in the range of 7–29. The number of the time series is 640 in total. Subdivision into the training and test sets is already performed: the former is composed of 270 series (30 utterances for nine speakers), whereas the latter contains 370 series (24–88 utterances by the same nine speakers in different conditions). All HMMs, modeling the nine classes, are chosen to have $K=5$ states, as in [64].

A.1.4. EEG signal recognition

This application aims at the examination of EEG signals in order to distinguish between alcoholic and control subjects, http://kdd.ics.uci.edu/databases/eeg. Each subject was exposed to either a single stimulus (S1) or two stimuli (S1 and S2) which were pictures of objects chosen from the 1980s Snodgrass and Vanderwart picture set. When two stimuli were shown, they were presented in either a matched condition where S1 was identical to S2 or in a non-matched condition where S1 differed from S2. There are three different versions of the data. In our case, we use the Large Data Set, denoted here denote it as *Alcoholic* data, in which the training and test sets are already pre-defined. The training set contains data for 10 alcoholic and 10 control subjects, with 10 runs per subject per paradigm. This results in 600 training sequences. The test data use the same alcoholic and control subjects, but with 10 out-of-sample runs per subject per paradigm. This results in 600 test sequences.

Each data set contains measurements from 64 electrodes placed on the scalp sampled at 256 Hz (3.9-ms epoch) for 1 s. We select the first two channels only, as they permitted an almost perfect discrimination in the case of “small dataset”. All HMMs are trained with the same number of states, $K=9$.

A.2. Computation of the relevant quantities for building HMMVS

The computation of the quantities needed for the construction of the HMMVS is based on two variables, the *forward* variable $\alpha_t(i)$ and the *backward* variable $\beta_t(i)$. The former ($\alpha_t(i)$) is defined as

$$\alpha_t(i)=P(O_1\dots O_t,Q_t=S_i^c|\lambda_c) \tag{9}$$

and represents the probability to have observed the sequence $O_1\dots O_t$ up to time t, and being in state S_i^c. It is recursively computed by the following formulas:

$$\alpha_1(i)=\pi_i b(O_1|S_i^c),\quad 1\leqslant i\leqslant K,$$

$$\alpha_{t+1}(i)=\left[\sum_{j=1}^{K}\alpha_t(j)a_{ji}\right]b(O_{t+1}|S_i^c),\quad 1\leqslant t\leqslant T-1,\ \ 1\leqslant i\leqslant K.$$

The *backward* variable is defined as

$$\beta_t(i)=P(O_{t+1}\dots O_T|Q_t=S_i^c,\lambda_c) \tag{10}$$

and represents the probability to observe the symbols $O_{t+1}\dots O_T$, being in the state S_i^c at time t. This variable is recursively computed:

$$\beta_T(i)=1,\quad 1\leqslant i\leqslant K,$$

$$\beta_t(i)=\sum_{j=1}^{K}a_{ij}b(O_{t+1}|S_j^c)\beta_{t+1}(j),\quad t=T-1,\dots,1,\ \ 1\leqslant i\leqslant K.$$

Given the variables above, the required quantities are computed as follows:

1. *Likelihood* $P(\mathbf{O}|\lambda_c)$:

$$P(\mathbf{O}|\lambda_c)=\sum_{i=1}^{K}\alpha_t(i)\beta_t(i)\quad\forall t. \tag{11}$$

By fixing $t=T$ we obtain

$$P(\mathbf{O}|\lambda_c)=\sum_{i=1}^{K}\alpha_T(i). \tag{12}$$

2. *Variable* $\xi_{(i,j)}^t$:

$$\begin{aligned}\xi_{(i,j)}^t&=P(Q_t=S_i^c,Q_{t+1}=S_j^c|\mathbf{O},\lambda_c)\\&=\frac{P(Q_t=S_i^c,Q_{t+1}=S_j^c,\mathbf{O}|\lambda_c)}{P(\mathbf{O}|\lambda_c)}\\&=\frac{\alpha_t(i)a_{ij}b(O_{t+1}|S_j^c)\beta_{t+1}(j)}{P(\mathbf{O}|\lambda_c)}\\&=\frac{\alpha_t(i)a_{ij}b(O_{t+1}|S_j^c)\beta_{t+1}(j)}{\sum_i\sum_j\alpha_t(i)a_{ij}b(O_{t+1}|S_j^c)\beta_{t+1}(j)}.\end{aligned} \tag{13}$$

Note that the sum of $\xi_{(i,j)}^t$ over time t can be interpreted as the expected number of transitions from state S_i^c to state S_j^c.

3. *Variable* γ_i^t :

$$\begin{aligned}\gamma_i^t&=P(Q_t=S_i^c|\mathbf{O},\lambda_c)\\&=\frac{P(Q_t=S_i^c,\mathbf{O}|\lambda_c)}{P(\mathbf{O}|\lambda_c)}\\&=\frac{\alpha_t(i)\beta_t(i)}{\sum_i\alpha_t(i)\beta_t(i)}.\end{aligned} \tag{14}$$

The variable γ_i^t can also be expressed in terms of the variable $\xi_{(i,j)}^t$, giving

$$\gamma_i^t=\sum_{j=1}^{K}\xi_{(i,j)}^t. \tag{15}$$

References

[1] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.
[2] F. Jelinek, Statistical Methods for Speech Recognition, MIT Press, Cambridge, MA, 1998.
[3] E.V.F. Casacuberta, J.M. Vilar, Architectures for speech-to-speech translation using finite-state models, in: Workshop on Speech-to-Speech Translation: Algorithms and Systems, ACL, Philadelphia, 2002, pp. 39–44.
[4] A. Molina, F. Pla, Shallow parsing using specialized HMMs, Journal on Machine Learning Research 2 (2002) 595–613.
[5] H. Bunke, T. Caelli, Hidden Markov Models Applications in Computer Vision, Series in Machine Perception and Artificial Intelligence, vol. 45, World Scientific, Singapore, 2001.
[6] Y. He, A. Kundu, 2-D shape classification using hidden Markov model, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (11) (1991) 1172–1184.
[7] M. Bicego, V. Murino, Investigating hidden Markov models' capabilities in 2D shape classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 281–286.
[8] M. Bicego, U. Castellani, V. Murino, Using hidden Markov models and wavelets for face recognition, in: International Conference on Image Analysis and Processing, 2003, pp. 52–56.
[9] S. Eickeler, A. Kosmala, G. Rigoll, Hidden Markov model based continuous online gesture recognition, in: International Conference on Pattern Recognition, vol. 2, 1998, pp. 1206–1208.
[10] R. Nowak, Multiscale hidden Markov models for Bayesian image analysis, in: B. Vidakovic, P. Muller (Eds.), Bayesian Inference in Wavelet Based Models, Lecture Notes in Statistics, vol. 141, Springer, Berlin, 1999.
[11] A. Krogh, M. Brown, I. Mian, K. Sjolander, D. Haussler, Hidden Markov models in computational biology: applications to protein modeling, Journal of Molecular Biology 235 (1994) 1501–1531.

[12] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, Cambridge, 1998.
[13] J. Gough, K. Karplus, R. Hughey, C. Chothia, Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure, Journal of Molecular Biology 313 (2001) 903–919.
[14] L. Baum, J. Egon, An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology, Bulletin of the American Meteorology Society 73 (1967) 360–363.
[15] L. Baum, An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, Inequality 3 (1970) 1–8.
[16] R. Duda, P. Hart, D. Stork, Pattern Classification, second ed., Wiley, New York, 2001.
[17] S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov model: analysis and applications, Machine Learning 32 (1998) 41–62.
[18] Y. Bengio, V.-P. Lauzon, R. Ducharme, Experiments on the application of IOHMMs to model financial returns series, IEEE Transactions on Neural Networks 12 (1) (2001) 113–123.
[19] Z. Ghahramani, M. Jordan, Factorial hidden Markov models, Machine Learning 29 (1997) 245–273.
[20] M. Brand, N. Oliver, S. Pentland, Coupled hidden Markov models for complex action recognition, in: IEEE International Conference on Computer Vision and Pattern Recognition, 1997.
[21] L. Bahl, P. Brown, P. de Souza, R. Mercer, Maximum mutual information estimation of hidden Markov model parameters for speech recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 86), vol. 11, 1986, pp. 49–52.
[22] Z. Kaiser, B. Horvat, Z. Kacic, A novel loss function for the overall risk criterion based discriminative training of HMM models, in: International Conference on Spoken Language Processing, vol. 2, Beijing, China, 2000, pp. 887–890.
[23] K. Na, B. Jeon, D. Chang, S. Chae, S. Ann, Discriminative training of hidden Markov models using overall risk criterion and reduced gradient method, in: European Conference on Speech Communication and Technology, Madrid, Spain, 1995, pp. 97–100.
[24] Z. Kaiser, B. Horvat, Z. Kacic, Overall risk criterion estimation of hidden Markov model parameters, Speech Communication 38 (3–4) (2002) 383–398.
[25] P.C. Woodland, D. Povey, Large scale discriminative training of hidden Markov models for speech recognition, Computer Speech and Language 16 (2002) 25–47.
[26] M. Gales, Discriminative models for speech recognition, in: Information Theory and Applications Workshop, 2007.
[27] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labelling sequence data, in: International Conference on Machine Learning, 2001, pp. 591–598.
[28] A. Gunawardana, M. Mahajan, A. Acero, J. Platt, Hidden conditional random fields for phone classification, in: Interspeech, Lisbon, Portugal, 2005, pp. 1117–1120.
[29] Y. Altun, I. Tsochantaridis, T. Hofmann, Hidden Markov support vector machines, in: International Conference on Machine Learning, 2003, pp. 3–10.
[30] M. Collins, Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithm, in: Conference on Empirical Methods in Natural Language Processing, 2002.
[31] S. Chakrabartty, G. Cauwenberghs, Forward decoding kernel machines: a hybrid HMM/SVM approach to sequence recognition, in: International Conference on Pattern Recognition: SVM Workshop, 2002.
[32] D. Wolpert, Stacked generalization, Neural Networks 5 (2) (1992) 241–260.
[33] K. Ting, I.H. Witten, Issues in stacked generalization, Journal of Artificial Intelligence Research 10 (1999) 271–289.
[34] R. Duin, The combining classifier: to train or not to train? in: International Conference on Pattern Recognition, vol. II, Canada, 2002, pp. 765–770.
[35] M. Bicego, V. Murino, M. Figueiredo, Similarity-based classification of sequences using hidden Markov models, Pattern Recognition 37 (12) (2004) 2281–2291.
[36] M. Bicego, E. Pękalska, R. Duin, Group-induced vector spaces, in: M. Haindl, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 4472, Springer, Berlin, 2007, pp. 190–199.
[37] C. Lai, D. Tax, R. Duin, E. Pękalska, P. Paclík, A study on combining image representations for image classification and retrieval, International Journal of Pattern Recognition and Artificial Intelligence 18 (5) (2004) 867–890.
[38] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: Advances in Neural Information Processing Systems, 1999, pp. 487–493.
[39] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins, Text classification using string kernels, Journal of Machine Learning Research 2 (2002) 419–444.
[40] C. Cortes, P. Haffner, M. Mohri, Rational kernels: theory and algorithms, Journal of Machine Learning Research 5 (2004) 1035–1062.
[41] N. Smith, M. Gales, Speech recognition using svms, in: Advances in Neural Information Processing Systems, 2002, pp. 1197–1204.
[42] L. Chen, H. Man, Combination of Fisher Scores and appearance based features for face recognition, in: ACM SIGMM Multimedia Biometrics Methods and Applications Workshop, 2003, pp. 74–81.
[43] L. Chen, H. Man, Discriminant analysis of stochastic models and its application to face recognition, in: International Workshop on Analysis and Modeling of Faces and Gestures, 2003, pp. 5–10.
[44] M. Layton, M. Gales, Augmented statistical models: exploiting generative models in discriminative classifiers, in: Advances in Neural Information Processing Systems, 2005.
[45] N. Smith, Using augmented statistical models and score spaces for classification, Ph.D. Thesis, Engineering Department, Cambridge University, 2003.
[46] R. Duin, P. Juszczak, P. Paclík, E. Pękalska, D. De Ridder, D. Tax, Prtools4, a matlab toolbox for pattern recognition, Delft University of Technology, 2004 ⟨http://www.prtools.org⟩.
[47] K. Fukanaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, San Diego, 1990.
[48] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, Berlin, 2001.
[49] B.S. Everitt, The Cambridge Dictionary of Statistics, 2003.
[50] B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.
[51] K. Bennett, Combining support vector and mathematical programming methods for induction, in: B. Schölkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods—SV Learning, MIT Press, Cambridge, MA, 1999, pp. 307–326.
[52] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, New York, 1961.
[53] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[54] C. Bhattacharyya, L. Grate, A. Rizki, D. Radisky, F. Molina, M. Jordan, M. Bissel, I. Mian, Simultaneous classification and relevant feature identification in high-dimensional spaces: application to molecular profiling data, Signal Processing 83 (2003) 729–743.
[55] L. Chen, H. Man, A.V. Nefian, Face recognition based on multi-class mapping of Fisher Scores, Pattern Recognition 38 (2005) 799–811.
[56] K. Tsuda, T. Kin, K. Asai, Marginalised kernels for biological sequences, Bioinformatics 18 (2002) 268–275.
[57] M. Layton, M. Gales, Acoustic modelling using continuous rational kernels, Journal of VLSI Signal Processing 48 (2007) 67–82.
[58] S. Amari, Natural gradient works efficiently in learning, Neural Computation 10 (1998) 251–276.
[59] N. Arica, F. Yarman-Vural, A shape descriptor based on circular hidden Markov model, in: International Conference on Pattern Recognition, vol. 1, 2000, pp. 924–927.
[60] G. Andreu, A. Crespo, J. Valiente, Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition, in: International Conference on Neural Networks, vol. 2, 1997, pp. 1341–1346.
[61] R. Mollineda, E. Vidal, F. Casacuberta, Cyclic sequence alignments: approximate versus optimal techniques, International Journal of Pattern Recognition and Artificial Intelligence 16 (3) (2002) 291–299.
[62] M. Kadous, Temporal classification: extending the classification paradigm to multivariate time series, Ph.D. Thesis, School of Computer Science and Engineering, University of New South Wales, 2002.
[63] M. Kadous, Learning comprehensible descriptions of multivariate time series, in: International Conference on Machine Learning, 1999, pp. 454–463.
[64] M. Kudo, J. Toyama, M. Shimbo, Multidimensional curve classification using passing-through regions, Pattern Recognition Letters 20 (11–13) (1999) 1103–1111.

About the Author—MANUELE BICEGO received the Ph.D. degree in 2003. During 2000–2003, he was with the Verona University, Italy; from 2004 to 2008 he was with the Sassari University. Currently he is researcher at the University of Verona (Italy) and member of the VIPS (Vision Image Processing and Sound Lab). His main research interests are in probabilistic models for classification and clustering, biometrics and video analysis.

About the Author—ELŻBIETA PĘKALSKA received the Ph.D. degree in 2005. During 1998–2006, she was with Delft University of Technology, The Netherlands, where she worked on both fundamental and applied projects in pattern recognition. She is currently an EPSRC Research Fellow at the University of Manchester, UK. She is the coauthor of 43 publications.

About the Author—DAVID M.J. TAX promoted in the Pattern Recognition group at the Delft University of Technology in 2001 on the subject 'One-class classification' under the supervision of R.P.W. Duin. His main research interest is in the learning and development of classifiers that optimize alternative performance criteria, like ordering criteria using the Area under the ROC curve or a Precision-Recall graph in unbalanced and noisy data.

About the Author—ROBERT P.W. DUIN received the Ph.D. degree in applied physics in 1978 for a thesis on statistical pattern recognition. After studying hardware and software architectures for image analysis he returned to pattern recognition. At present he is an associate professor of the EEMCS Faculty of Delft University of Technology.