Multivariate Time-Series Classification Using the Hidden-Unit Logistic Model

Wenjie Pei, Hamdi Dibeklioğlu, *Member, IEEE*, David M. J. Tax, and Laurens van der Maaten

***Abstract*— We present a new model for multivariate time-series classification, called the hidden-unit logistic model (HULM), that uses binary stochastic hidden units to model latent structure in the data. The hidden units are connected in a chain structure that models temporal dependencies in the data. Compared with the prior models for time-series classification such as the hidden conditional random field, our model can model very complex decision boundaries, because the number of latent states grows exponentially with the number of hidden units. We demonstrate the strong performance of our model in experiments on a variety of (computer vision) tasks, including handwritten character recognition, speech recognition, facial expression, and action recognition. We also present a state-of-the-art system for facial action unit detection based on the HULM.**

***Index Terms*— Hidden unit, latent structure modeling, temporal dependences modeling, time-series classification.**

I. Introduction

TIME series classification is the problem of assigning a single label to a sequence of observations (i.e., to a time series). Time-series classification has a wide range of applications in computer vision. A state-of-the-art model for time-series classification problem is the hidden-state conditional random field (HCRF) [1], which models latent structure in the data using a chain of k-nomial latent variables. The HCRF has been successfully used in, amongst others, gesture recognition [2], object recognition [1], and action recognition [3]. An important limitation of the HCRF is that the number of model parameters grows linearly with the number of latent states in the model. This implies that the training of complex models with a large number of latent states is very prone to overfitting, while models with smaller numbers of parameters may be too simple to represent a good classification function. In this paper, we propose to circumvent this problem of the HCRF by replacing each of the k-nomial latent variables by a collection of H binary stochastic hidden units. To keep inference tractable, the hidden-unit chains are conditionally independent given the time series and the label. Similar ideas have been explored before in discriminative RBMs [4] for standard classification problems and in hidden-unit CRFs [5] for sequence labeling. The binary stochastic hidden units allow the resulting model, which we call the hidden-unit logistic model (HULM), to represent 2^H latent states using only $O(H)$ parameters. This substantially reduces the amount of data needed to successfully train models without overfitting while maintaining the ability to learn complex models with exponentially many latent states. Exact inference in our proposed model is tractable, which makes parameter learning via (stochastic) gradient descent very efficient. We show the merits of our HULM in experiments on computer-vision tasks ranging from online character recognition to activity recognition and facial expression analysis. Moreover, we present a system for facial action unit detection that, with the help of the HULM, achieves the state-of-the-art performance on a commonly used benchmark for facial analysis.

The remainder of this paper is organized as follows. Section II reviews prior work on time-series classification. Section III introduces our HULM and describes how inference and learning can be performed in the model. In Section IV, we present the results of experiments comparing the performance of our model with that of the state-of-the-art time-series classification models on a range of classification tasks. In Section V, we present a new state-of-the-art system for facial action unit detection based on the HULM. Section VI concludes this paper.

II. Related Work

There is a substantial amount of prior work on multivariate time-series classification. Much of this paper is based on the use of (kernels based on) dynamic time warping (e.g., [6]) or on hidden Markov models (HMMs) [7]. The HMM is a generative model that models the time-series data in a chain of latent k-nomial features. Class-conditional HMMs are commonly combined with class priors via Bayes' rule to obtain a time-series classification models. Alternatively, HMMs are also frequently used as the base model for Fisher kernel [8], which constructs a time-series representation that consists of the gradient a particular time series induces in the parameters of the HMM; the resulting representations can be used on standard classifiers such as SVMs. Some recent work has also tried to learn the parameters of the HMM in such a way as to learn Fisher kernel representations that are well suited for nearest-neighbor classification [9]. HMMs have also been used as the base model for probability product kernels [10], which fit a single HMM on each time series and define the similarity

Manuscript received February 26, 2016; revised August 23, 2016 and December 9, 2016; accepted January 2, 2017. This work was supported in part by EU-FP7 INSIDDE and in part by AALSALIG++.

The authors are with the Pattern Recognition Laboratory, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: W.Pei-1@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2651018

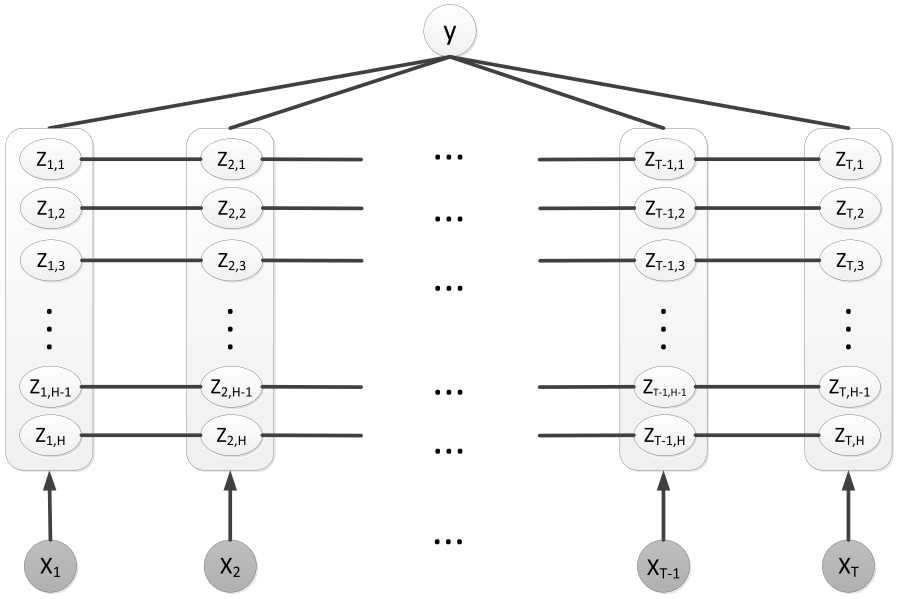


Fig. 1. Graphical model of the HULM.

between two time series as the inner product between the corresponding HMM distributions. A potential drawback of these approaches is that they perform classification based on (rather simple) generative models of the data that may not be well suited for the discriminative task at hand. By contrast, we opt for a discriminative model that does not waste model capacity on features that are irrelevant for classification. In contrast to HMMs, CRFs [11] are discriminative models that are commonly used for sequence labeling of time series using so-called linear-chain CRFs. While standard linear-chain CRFs achieve strong performance on very high-dimensional data (e.g., in natural language processing), the linear nature of most CRF models limits their ability to learn complex decision boundaries. Several sequence labeling models have been proposed to address this limitation, amongst which are latent-dynamic CRFs [12], conditional neural fields [13], neural CRFs [14], and hidden-unit CRFs [5]. These models introduce stochastic or deterministic hidden units that model latent structure in the data, allowing these models to represent nonlinear decision boundaries. As these prior models were designed for sequence labeling (assigning a label to each frame in the time series), they cannot readily be used for time-series classification (assigning a single label to the entire time series). Our HULM may be viewed as an adaptation of sequence labeling models with hidden units to the time-series classification problem. As such, it is closely related to the hidden CRF model [1]. The key difference between our HULM and the hidden CRF is that our model uses a collection of binary stochastic hidden units instead of a single k-nomial hidden unit, which allows our model to represent exponentially more states with the same number of parameters.

An alternative approach to expanding the number of hidden states of the HCRF is the infinite HCRF (iHCRF), which employs a Dirichlet process to determine the number of hidden states. Inference in the iHCRF can be performed via collapsed Gibbs sampling [15] or variational inference [16]. While theoretically facilitating infinitely many states, the modeling power of the iHCRF is, however, limited to the number of "represented" hidden states. Unlike our model, the number of parameters in the iHCRF thus still grows linearly with the number of hidden states.

III. Hidden-Unit Logistic Model

The HULM is a probabilistic graphical model that receives a time series as input, and is trained to produce a single output label for this time series. Like the hidden-state CRF, the model contains a chain of hidden units that aim to model latent temporal features in the data, and that form the basis for the final classification decision. The key difference with the HCRF is that the latent features are model in H binary stochastic hidden units, much like in a (discriminative) Restricted Boltzmann machine (RBM). These hidden units $\mathbf{z}_t$ can model very rich latent structure in the data: one may think about them as carving up the data space into 2^H small clusters, all of which may be associated with particular clusters. The parameters of the temporal chains that connect the hidden units may be used to differentiate between features that are "constant" (i.e., that are likely to be presented for prolonged lengths of time) or that are "volatile" (i.e., that tend to rapidly appear and disappear). Because the hidden-unit chains are conditionally independent given the time series and the label, they can be integrated out analytically when performing inference or learning.

Suppose we are given a time series $\mathbf{x}_{1,\ldots,T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ of length T in which the observation at the tth time step is denoted by $\mathbf{x}_t \in \mathbb{R}^D$. Conditioned on this time series, the HULM outputs a distribution over vectors $\mathbf{y}$ that represent the predicted label using a 1-of-K encoding scheme (i.e., a one-hot encoding): $\forall k : y_k \in \{0, 1\}$ and $\sum_k y_k = 1$.

Denoting the stochastic hidden units at time step t by $\mathbf{z}_t \in \{0, 1\}^H$, the hidden-unit logistic model defines the conditional distribution over label vectors using a Gibbs distribution

in which all hidden units are integrated out

$$p(\mathbf{y}|\mathbf{x}_{1,\ldots,T}) = \frac{\sum_{\mathbf{z}_{1,\ldots,T}} \exp\{E(\mathbf{x}_{1,\ldots,T}, \mathbf{z}_{1,\ldots,T}, \mathbf{y})\}}{Z(\mathbf{x}_{1,\ldots,T})}. \quad (1)$$

Herein, $Z(\mathbf{x}_{1,\ldots,T})$ denotes a partition function that normalizes the distribution, and is given by

$$Z(\mathbf{x}_{1,\ldots,T}) = \sum_{\mathbf{y}'} \sum_{\mathbf{z}'_{1,\ldots,T}} \exp\{E(\mathbf{x}_{1,\ldots,T}, \mathbf{z}'_{1,\ldots,T}, \mathbf{y}')\}. \quad (2)$$

The energy function of the HULM is defined as

$$\begin{aligned} &E(\mathbf{x}_{1,\ldots,T}, \mathbf{z}_{1,\ldots,T}, \mathbf{y}) \\ &\quad = \mathbf{z}_1^\top \boldsymbol{\pi} + \mathbf{z}_T^\top \boldsymbol{\tau} + \mathbf{c}^\top \mathbf{y} \\ &\qquad + \sum_{t=2}^{T} \mathbf{z}_{t-1}^T \mathrm{diag}(\mathbf{A}) \mathbf{z}_t + \sum_{t=1}^{T} \left[\mathbf{z}_t^\top \mathbf{W} \mathbf{x}_t + \mathbf{z}_t^\top \mathbf{V} \mathbf{y} + \mathbf{z}_t^\top \mathbf{b}\right]. \end{aligned} \quad (3)$$

The graphical model of the HULM is shown in Fig. 1.

Next to a number of bias terms, the energy function in (3) consists of three main components: 1) a term with parameters $\mathbf{W}$ that measures to what extent particular latent features are present in the data; 2) a term parametrized by $\mathbf{A}$ that measures the compatibility between corresponding hidden units at time step $t-1$ and t; and 3) a prediction term with parameters $\mathbf{V}$ that measures the compatibility between the latent features $\mathbf{z}_{1,\ldots,T}$ and the label vector $\mathbf{y}$. Please note that hidden units in consecutive time steps are connected using a chain structure rather than fully connected; we opt for this structure, because exact inference is intractable when consecutive hidden units are fully connected. Intuitively, the HULM thus assigns a high probability to a label (for a particular input) when there are hidden unit states that are both "compatible" with the observed data and with a particular label. As the hidden units can take 2^H different states, this leads to a model that can represent highly nonlinear decision boundaries. Sections III-A–C describe the details of inference and learning in the HULM. The whole process is summarized in Algorithm 1.

Algorithm 1 Inference and Learning of *HULM*

Input: A time series $\mathbf{x}_{1,\ldots,T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ and the associated labels $\mathbf{y}$.

Output:

- The conditional distribution over predicted labels $p(\mathbf{y}|\mathbf{x}_{1,\ldots,T})$ (*inference*);
- The conditional log-likelihood of the training data: $\mathcal{L}(\Theta)$ (*inference*);
- The gradient of $\mathcal{L}(\Theta)$ with respect to each parameter $\theta \in \Theta$: $\frac{\partial \mathcal{L}}{\partial \theta}$ (*learning*).

1: Compute the potential functions $\Psi_{t,h}(\mathbf{x}_t, z_{t-1,h}, z_{t,h}, \mathbf{y})$ for each hidden unit h ($1 \le h \le H$) at each time step t ($1 \le t \le T$) as indicated in Equation 5.
2: **for** $t = 1 \rightarrow T$ **do**
3: Calculate the forward message $\alpha_{t,h,k}$ with $k \in \{0, 1\}$ by Equation 9.
4: **end for**
5: **for** $t = T \rightarrow 1$ **do**
6: Compute the backward message $\beta_{t,h,k}$ by Equation 10.
7: **end for**
8: Compute the intermediate term $M(\mathbf{x}_{1,\ldots,T}, \mathbf{y}) = \sum_{\mathbf{z}_{1,\ldots,T}} \exp\{E(\mathbf{x}_{1,\ldots,T}, \mathbf{z}_{1,\ldots,T}, \mathbf{y})$ either with $\alpha_{T,h,k}$ or with $\beta_{1,h,k}$ by Equation 11.
9: Compute the partition function $Z(\mathbf{x}_{1,\ldots,T}) = \sum_{\mathbf{y}'} M(\mathbf{x}_{1,\ldots,T}, \mathbf{y}')$.
10: The conditional distribution over predicted labels is calculated by $p(\mathbf{y}|\mathbf{x}_{1,\ldots,T}) = \frac{M(\mathbf{x}_{1,\ldots,T}, \mathbf{y})}{Z(\mathbf{x}_{1,\ldots,T})}$.
11: The conditional log-likelihood of the training data $\mathcal{L}(\Theta)$ is calculated by Equation 14.
12: Compute the marginal distribution over a chain edge $\xi_{t,h,k,l} = P(z_{t,h} = k, z_{t+1,h} = l|\mathbf{x}_{1,\ldots,T}, \mathbf{y})$ by Equation 13 using forward and backward messages.
13: The gradient of $\mathcal{L}(\Theta)$ with respect to each parameter $\theta \in \Theta$: $\frac{\partial \mathcal{L}}{\partial \theta}$ is calculated by Equation 15 and 16 using marginal distribution $\xi_{t,h,k,l}$.

A. Inference

The main inferential problem given an observation $\mathbf{x}_{1,\ldots,T}$ is the evaluation of predictive distribution $p(\mathbf{y}|\mathbf{x}_{1,\ldots,T})$. The key difficulty in computing this predictive distribution is the sum over all $2^{H\times T}$ hidden unit states

$$M(\mathbf{x}_{1,\ldots,T}, \mathbf{y}) = \sum_{\mathbf{z}_{1,\ldots,T}} \exp\{E(\mathbf{x}_{1,\ldots,T}, \mathbf{z}_{1,\ldots,T}, \mathbf{y})\}. \quad (4)$$

The chain structure of the HULM allows us to employ a standard forward–backward algorithm that can compute $M(\cdot)$ in computational time linear in T.

Specifically, defining potential functions that contain all terms that involve time t and hidden unit h

$$\begin{aligned} &\Psi_{t,h}(\mathbf{x}_t, z_{t-1,h}, z_{t,h}, \mathbf{y}) \\ &\quad = \exp\{\mathbf{z}_{t-1,h} \mathbf{A}_h z_{t,h} + z_{t,h} \mathbf{W}_h \mathbf{x}_t + z_{t,h} \mathbf{V}_h \mathbf{y} + z_{t,h} b_h\} \end{aligned} \quad (5)$$

ignoring bias terms, and introducing virtual hidden units $\mathbf{z}_0 = \mathbf{0}$ at time $t = 0$, we can rewrite $M(\cdot)$ as

$$\begin{aligned} M(\cdot) &= \sum_{\mathbf{z}_{1,\ldots,T}} \prod_{t=1}^{T} \prod_{h=1}^{H} \Psi_{t,h}(\mathbf{x_t}, z_{t-1,h}, z_{t,h}, \mathbf{y}) \\ &= \prod_{h=1}^{H} \left[\sum_{z_{1,h},\ldots,z_{T,h}} \prod_{t=1}^{T} \Psi_{t,h}(\mathbf{x}_t, z_{t-1,h}, z_{t,h}, \mathbf{y}) \right] \\ &= \prod_{h=1}^{H} \left[\sum_{z_{T-1,h}} \Psi_{T,h}(\mathbf{x}_T, z_{T-1,h}, z_{T,h}, \mathbf{y}) \right. \\ &\qquad \left. \sum_{z_{T-2,h}} \Psi_{T-1,h}(\mathbf{x}_{T-1}, z_{T-2,h}, z_{T-1,h}, \mathbf{y}) \cdots \right]. \end{aligned} \quad (6)$$

In the above derivation, it should be noted that the product over hidden units h can be pulled outside the sum over all states $\mathbf{z}_{1,\ldots,T}$, because the hidden-unit chains are conditionally

independent given the data $\mathbf{x}_{1,\ldots,T}$ and the label $\mathbf{y}$. Subsequently, the product over time t can be pulled outside the sum because of the (first-order) Markovian chain structure of the temporal connections between hidden units.

In particular, the required quantities can be evaluated using the forward–backward algorithm, in which we define the forward messages $\alpha_{t,h,k}$ with $k \in \{0, 1\}$ as

$$\alpha_{t,h,k} = \sum_{z_{1,h},\ldots,z_{t-1,h}} \prod_{t'=1}^{t} \Psi_{t',h}(\mathbf{x}_{t'}, z_{t'-1,h}, z_{t',h} = k, \mathbf{y}) \tag{7}$$

and the backward messages $\beta_{t,h,k}$ as

$$\beta_{t,h,k} = \sum_{z_{t+1,h},\ldots,z_{T,h}} \prod_{t'=t+1}^{T} \Psi_{t',h}(\mathbf{x}_{t'+1}, z_{t',h} = k, z_{t'+1,h}, \mathbf{y}). \tag{8}$$

These messages can be calculated recursively as follows:

$$\alpha_{t,h,k} = \sum_{i\in\{0,1\}} \Psi_{t,h}(\mathbf{x}_t, z_{t-1,h} = i, z_{t,h} = k, \mathbf{y})\alpha_{t-1,h,i} \tag{9}$$

$$\beta_{t,h,k} = \sum_{i\in\{0,1\}} \Psi_{t+1,h}(\mathbf{x}_{t+1}, z_{t,h} = k, z_{t+1,h} = i, \mathbf{y})\beta_{t+1,h,i}. \tag{10}$$

The value of $M(\mathbf{x}_{1,\ldots,T}, \mathbf{y})$ can readily be computed from the resulting forward messages or backward messages

$$\begin{aligned} M(\mathbf{x}_{1,\ldots,T}, \mathbf{y}) &= \prod_{h=1}^{H} \left(\sum_{k\in\{0,1\}} \alpha_{T,h,k} \right) \\ &= \prod_{h=1}^{H} \left(\sum_{k\in\{0,1\}} \beta_{1,h,k} \right). \end{aligned} \tag{11}$$

To complete the evaluation of the predictive distribution, we compute the partition function of the predictive distribution by summing $M(\mathbf{x}_{1,\ldots,T}, \mathbf{y})$ over all K possible labels: $Z(\mathbf{x}_{1,\ldots,T}) = \sum_{\mathbf{y}'} M(\mathbf{x}_{1,\ldots,T}, \mathbf{y}')$. Indeed, inference in the HULM is linear in both the length of the time series T and in the number of hidden units H.

Another inferential problem that needs to be solved during parameter learning is the evaluation of the marginal distribution over a chain edge

$$\xi_{t,h,k,l} = P(z_{t,h} = k, z_{t+1,h} = l|\mathbf{x}_{1,\ldots,T}, \mathbf{y}). \tag{12}$$

Using a similar derivation, it can be shown that this quantity can also be computed from the forward and backward messages

$$\xi_{t,h,k,l} = \frac{\alpha_{t,h,k} \cdot \Psi_{t+1,h}(\mathbf{x}_{t+1}, z_{t,h} = k, z_{t+1,h} = l, y) \cdot \beta_{t+1,h,l}}{\sum_{k\in\{0,1\}} \alpha_{T,h,k}}. \tag{13}$$

B. Parameter Learning

Given a training set $\mathcal{D} = \{(\mathbf{x}^{(}n)_{1,\ldots,T}, \mathbf{y}^{(n)})\}_{n=1,\ldots,N}$ containing N pairs of time series and their associated label. We learn the parameters $\mathbf{\Theta} = \{\pi, \tau, \mathbf{A}, \mathbf{W}, \mathbf{V}, \mathbf{b}, \mathbf{c}\}$ of the HULM

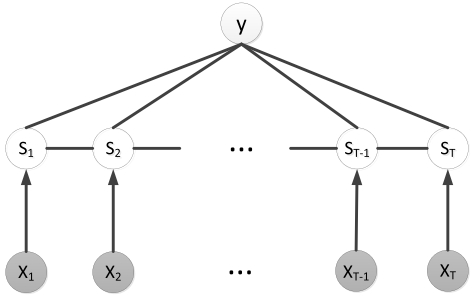


Fig. 2. Graphical model of the HCRF.

by maximizing the conditional log-likelihood of the training data with respect to the parameters

$$\begin{aligned} \mathcal{L}(\Theta) &= \sum_{n=1}^{N} \log p(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}_{1,\ldots,T}) \\ &= \sum_{n=1}^{N} \left[\log M\left(\mathbf{x}^{(n)}_{1,\ldots,T}, \mathbf{y}^{(n)}\right) - \log \sum_{\mathbf{y}'} M\left(\mathbf{x}^{(n)}_{1,\ldots,T}, \mathbf{y}'\right) \right]. \end{aligned} \tag{14}$$

We augment the conditional log-likelihood with L2-regularization terms on the parameters $\mathbf{A}$, $\mathbf{W}$, and $\mathbf{V}$. As the objective function is not amenable to closed-form optimization (in fact, it is not even a convex function), we perform optimization using stochastic gradient descent on the negative conditional log-likelihood. The gradient of the conditional log-likelihood with respect to a parameter $\theta \in \Theta$ is given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta} &= \mathbb{E}\left[\frac{\partial E(\mathbf{x}_{1,\ldots,T}, \mathbf{z}_{1,\ldots,T}, \mathbf{y})}{\partial \theta} \right]_{P(\mathbf{z}_{1,\ldots,T}|\mathbf{x}_{1,\ldots,T}, \mathbf{y})} \\ &\quad - \mathbb{E}\left[\frac{\partial E(\mathbf{x}_{1,\ldots,T}, \mathbf{z}_{1,\ldots,T}, \mathbf{y})}{\partial \theta} \right]_{P(\mathbf{z}_{1,\ldots,T}, \mathbf{y}|\mathbf{x}_{1,\ldots,T})}. \end{aligned} \tag{15}$$

where we omitted the sum over training examples for brevity. The required expectations can readily be computed using the inference algorithm described in Section III-A.

For example, defining $r(\Theta) = z_{t-1,h}\mathbf{A}_h z_{t,h} + z_{t,h}\mathbf{W}_h\mathbf{x}_t + z_{t,h}\mathbf{V}_h y + z_{t,h}b_h$ for notational simplicity, the first expectation can be computed as follows:

$$\begin{aligned} &\mathbb{E}\left[\frac{\partial E(\mathbf{x}_{1,\ldots,T}, \mathbf{z}_{1,\ldots,T}, \mathbf{y})}{\partial \theta} \right]_{P(\mathbf{z}_{1,\ldots,T}|\mathbf{x}_{1,\ldots,T}, \mathbf{y})} \\ &= \sum_{\mathbf{z}_{1,\ldots,T}} P(\mathbf{z}_{1,\ldots,T}|\mathbf{x}_{1,\ldots,T}, \mathbf{y}) \left(\sum_{t=1}^{T} \sum_{h=1}^{H} \frac{\partial r(\Theta)}{\partial \theta} \right) \\ &= \sum_{t=1}^{T} \sum_{k\in\{0,1\}} \sum_{l\in\{0,1\}} \left(\xi_{t-1,h,k,l} \cdot \frac{\partial r(\Theta)}{\partial \theta} \right). \end{aligned} \tag{16}$$

The second expectation is simply an average of these expectations over all K possible labels $\mathbf{y}$.

C. Comparison With HCRF

The hidden-state CRF's graphical model, shown in Fig. 2, is similar to that of the HULM. They are both discriminative

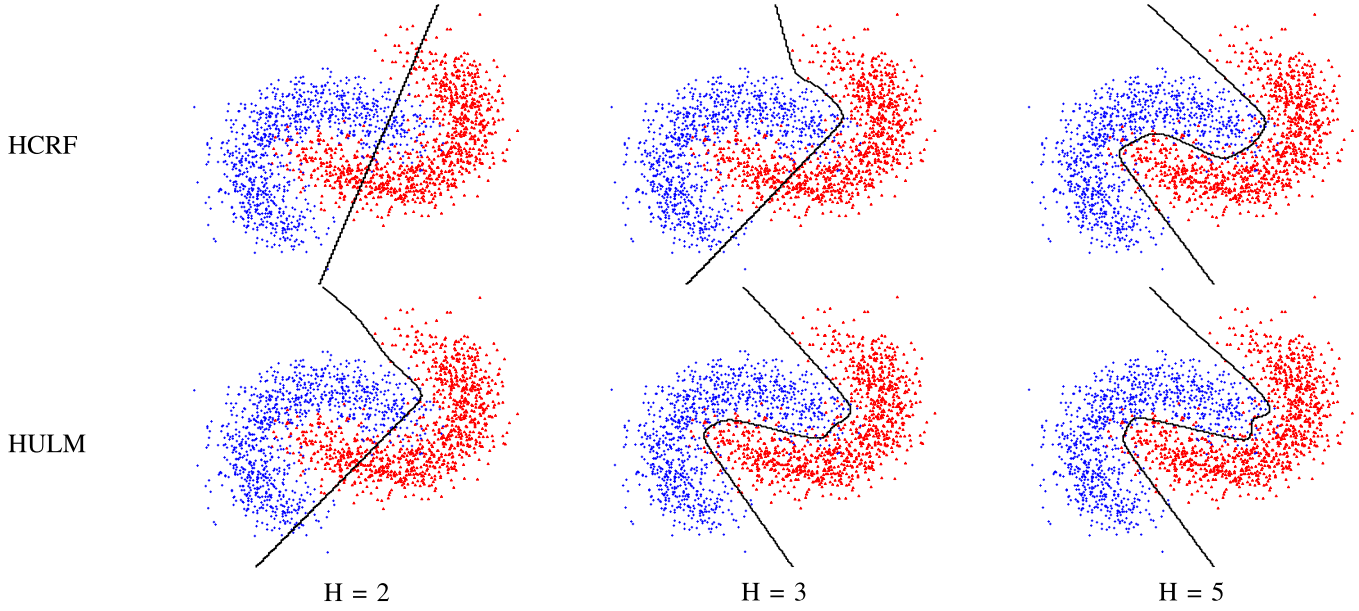


Fig. 3. Comparison of HCRF and HULM for binary classification on the banana data set (ignoring the time series aspect of the models) with the same number of hidden units H. Black lines: decision boundaries learned by both models.

models, which employ hidden variables to model the latent structures. The key difference between the two models is in the way that the hidden units are defined: whereas the HULM uses a large number of (conditionally independent) binary stochastic hidden units to represent the latent state, the HCRF uses a single multinomial unit (much like an HMM). As a result, there are substantial differences in the distributions that the HCRF and HULM can model. In particular, the HULM is a *product of experts* model,[1] whereas the HCRF is a *mixture of experts* model [17], [18]. A potential advantage of product distributions over mixture distributions is in the "sharpness" of the distributions [17]. Consider, for instance, two univariate Gaussian distributions with equal variance but different means: whereas a mixture those distributions will have higher variance than each of the individual Gaussians, a product of the distribution will have lower variance and, therefore, model a much sharper distribution. This can be a substantial advantage when modeling high-dimensional distributions in which much of the probability mass tends to be lost in the tails. There also appear to be differences in the total number of modes that can be modeled by product and mixture distributions in high-dimensional spaces (although it is hitherto unknown how many modes a mixture of unimodal distributions maximally contains [19]). Indeed, theoretical results suggest that product distributions have more modeling power with the same number of parameters than mixture distributions; for certain distributions, mixture distributions even require exponentially more parameters than their product counterparts [20].

To empirically explore these differences, we performed a simple experiment in which we ignore the temporal component of the HULM and HCRF models (to facilitate visualizations), and train the models on a binary 2-D classification problem.

[1]The expression of $M(\cdot)$ presented earlier clearly shows that HULM models a distribution that is a product over H experts.

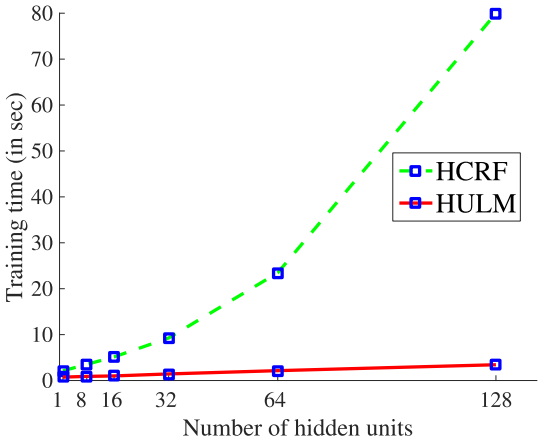


Fig. 4. Running time of a single training epoch of the HULM and HCRF model on the facial expression data (CK+) described in Section IV-A as a function of the number of hidden units. We used stochastic gradient descent with the same configuration to train both the HULM and the HCRF.

Fig. 3 shows the decision boundaries learned by HULM and HCRF models with the same number of hidden parameters on our test data set. Indeed, the results suggest that the HULM can model more complex decision boundaries than HCRFs with the same number of parameters.

In our experiments, we also observed that HULM models can be trained faster than HCRF models. We illustrate this in Fig. 4, which shows the training time of both models (with the same experimental configuration) on a facial expression data set. While these differences in training speed may be partly due to implementation differences, they are also the result of the constraint we introduce that the transition matrix between hidden units in consecutive time steps is diagonal. As a result, the computation of the forward message α in 7 and backward message β in 8 is linear in the number of

hidden units H. Consequently, the quantities $M(\mathbf{x}_{1,\ldots,T}, \mathbf{y})$ in 11 and marginal distribution $\xi_{t,h,k,l}$ in 12 can be calculated in $O(THD)$. Taking into account the number of label classes Y, the overall computational complexity of HULM is $O(TH(D+Y))$. By contrast, the complexity of HCRF is $O(TH^2(D+Y))$ [1]. This difference facilitates the use or larger numbers of hidden units H in the HULM model than in the HCRF model (admittedly, it is straightforward to develop a diagonal version of the HCRF model, also).

IV. Experiments

To evaluate the performance of the HULM, we conducted classification experiments on eight different problems involving seven time series data sets. Since univariate times series can be considered as a special case of multivariate time series, we first performed experiments on two univariate time-series data sets introduced by UCR Archive [21]: 1) Synthetic Control and 2) Swedish Leaf, and subsequently, we evaluated our models on five multivariate time-series data sets: 1) an online handwritten character (OHC) data set [22]; 2) a data set of Arabic spoken digits (ASDs) [23]; 3) the Cohn–Kanade extended facial expression data set (CK+) [24]; 4) the MSR Action 3-D data set (Action) [25]; and 5) the MSR Daily Activity 3-D data set (Activity) [26]. The seven data sets are introduced in IV-A, the experimental setup is presented in IV-B, and the results of the experiments are in IV-C.

A. Data Sets

1) Univariate Time-Series Data Sets: We performed experiments on two univariate UCR data sets: *Synthetic Control* and *Swedish Leaf*. *Synthetic Control* is a relatively easy data set containing 300 training samples and 300 test samples grouped into 6 classes. All samples in it have the identical length of time series equaling to 60. We enrich the univariate feature by windowing 10 frames into 1 frame resulting in the ten dimensions for each frame. *Swedish Leaf* is a challenging data set which consists of 500 training samples and 625 test samples with the length of 128 frames spreading in 15 classes. Similarly, we preprocess the data by windowing the features of 30 frames into 1 frame with 30-D feature.

2) Multivariate Time-Series Data Sets: The OHC data set [22] is a pen-trajectory time-series data set that consists of three dimensions at each time step, *viz.*, the pen movement in the x-direction and y-direction, and the pen pressure. The data set contains 2858 time series with an average length of 120 frames. Each time series corresponds to a single handwritten character that has one of 20 labels. We preprocess the data by windowing the features of 10 frames into a single feature vector with 30 dimensions.

The ASD data set contains 8800 utterances [23], which were collected by asking 88 Arabic native speakers to utter all 10 digits ten times. Each time series consists of 13-D Mel-Frequency Cepstral Coefficients (MFCCs), which were sampled at 11 025Hz, 16 b using a Hamming window. We enrich the features by windowing 3 frames into 1 frames resulting in the 13×3 dimensions for each frame of the features while keeping the same length of time series. We use two different versions of the spoken digit data set: 1) a *digit* version in which the uttered digit is the class label and 2) a *voice* version in which the speaker of a digit is the class label.

The Cohn–Kanade extended facial expression data set [24] contains 593 image sequences (videos) from 123 subjects. Each video shows a single facial expression. The videos have an average length of 18 frames. A subset of 327 of the videos, which have validated label corresponding to one of seven emotions (anger, contempt, disgust, fear, happiness, sadness, and surprise), are used in our experiments. We adopt the publicly available shape features used in [27] as the feature representation for our experiments. These features represent each frame by the variation of 68 feature point locations (x, y) with respect to the first frame [24], which leads to 136-dimensional feature representation for each frame in the video.

The MSR Action 3-D data set [25] consists of RGB-D videos of people performing certain actions. The data set contains 567 videos with an average length of 41 frames. Each video should be classified into one of 20 actions such as "high arm wave," "horizontal arm wave," and "hammer." We use the real-time skeleton tracking algorithm of [28] to extract the 3-D joint positions from the depth sequences. We use the 3-D joint position features (pairwise relative positions) proposed in [26] as the feature representation for the frames in the videos. Since we track a total of 20 joints, the dimensionality of the resulting feature representation is $3 \times \binom{20}{2} = 570$, where $\binom{20}{2}$ is the number of pairwise distances between joints and 3 is dimensionality of the (x, y, z) coordinate vectors. It should be noted that we only extract the joints features to evaluate performances of different time-series classification models mentioned in this paper rather than pursue state-of-the-art action-recognition performance, and hence, it is not fair to compare the reported results in Table I directly to the performance of the *ad hoc* action-recognition methods, which employ 2-D/3-D appearance features.

The MSR Daily Activity 3-D data set [26] contains RGB-D videos of people performing daily activities. The data set also contains 3-D skeletal joint positions, which are extracted using the Kinect SDK. The videos need to be classified into one of 16 activity types, which include "drinking," "eating," "reading book," and so on. Each activity is performed by ten subjects in two different poses (namely, while sitting on a sofa and while standing), which leads to a total of 320 videos. The videos have an average length of 193 frames. To represent each frame, we extract 570-D 3-D joint position features.

B. Experimental Setup

In our experiments, the model parameters $\mathbf{A}, \mathbf{W}$, and $\mathbf{V}$ of the HULM were initialized by sampling them from a Gaussian distribution with a variance of 10^{-3}. The initial-state parameter $\boldsymbol{\pi}$, final-state parameter $\boldsymbol{\tau}$, and the bias parameters $\mathbf{b}, \mathbf{c}$ were initialized to 0. Training of our model is performed using a standard stochastic gradient descent procedure; the learning rate is decayed during training. We set the number of hidden units H to 100. The L2-regularization parameter λ was tuned by minimizing the error on a small validation set.

TABLE I

GENERALIZATION ERRORS (%) ON ALL EIGHT PROBLEMS BY FOUR TIME-SERIES CLASSIFICATION MODELS: THE NL MODEL, FKL, THE HCRF, AND THE HULM. THE BEST PERFORMANCE ON EACH DATA SET IS BOLDFACED. SEE TEXT FOR DETAILS

Dataset	Dim.	Classes	Model			
			NL	FKL	HCRF	HULM
Synthetic Control	1×10	6	20.00	2.33	1.67	**1.33**
Swedish Leaf	1×30	15	52.64	10.24	12.80	**10.08**
OHC	3×10	20	23.67	**0.97**	1.58	1.30
ASD-digit	13×3	10	25.50	6.91	**3.68**	4.68
ASD-voice	13×3	88	36.91	6.36	20.40	**5.45**
CK+	136	7	9.20	10.81	11.04	**6.44**
Action	570	20	40.40	40.74	**34.68**	35.69
Activity	570	16	59.38	**43.13**	62.50	45.63
Avg. rank	–	–	3.50	2.38	2.63	**1.50**

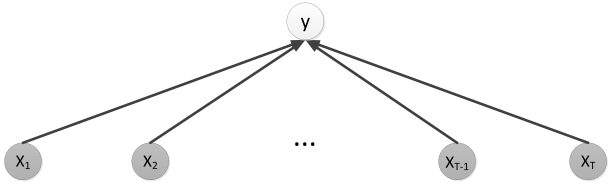


Fig. 5. Graphical model of the NL model.

Code reproducing the results of our experiments is available on *https://github.com/wenjiepei/HULM*.

We compare the performance of our HULM with that of three other time-series classification models: 1) the naive logistic (NL) model shown in Fig. 5; 2) the popular HCRF model [1]; and 3) Fisher kernel learning (FKL) model [9]. The details of these models are given in the following.

1) Naive Logistic Model: The NL model is a linear logistic model that shares parameters between all time steps, and makes a prediction by summing (or equivalently, averaging) the inner products between the model weights and feature vectors over time before applying the softmax function. Specifically, the NL model defined the following conditional distribution over the label y given the time-series data $\mathbf{x}_{1,\ldots,T}$:

$$p(\mathbf{y}|\mathbf{x}_{1,\ldots,T}) = \frac{\exp\{E(\mathbf{x}_{1,\ldots,T}, \mathbf{y})\}}{Z(\mathbf{x}_{1,\ldots,T})}$$

where the energy function is defined as

$$E(\mathbf{x}_{1,\ldots,T}, \mathbf{y}) = \sum_{t=1}^{T} (\mathbf{y}^T \mathbf{W} \mathbf{x}_t) + \mathbf{c}^T \mathbf{y}.$$

The corresponding graphical model is shown in Fig. 5. We include the NL model in our experiments to investigate the effect of adding hidden units to models that average energy contributions over time.

2) Hidden CRF: The Hidden-state CRF model is similar to HULM and thereby an important baseline. We performed experiments using the hidden CRF implementation of [29]. Following [1], we trained HCRFs with ten latent states on all data sets (we found it was computationally infeasible to train HCRFs with more than ten latent states). We tune the L2-regularization parameter of the HCRF on a small validation set.

3) Fisher Kernel Learning: In addition to compared with HCRFs, we compare the performance of our model with that of the recently proposed FKL model [9]. We selected the FKL model for our experiments, because [9] reports strong performance on a range of time-series classification problems. We trained FKL models based on HMMs with ten hidden states (the number of hidden states was set identical to that of the hidden CRF). Subsequently, we computed the Fisher kernel representation and trained a linear SVM on the resulting features to obtain the final classifier. The slack parameter C of the SVM is tuned on a small validation set.

C. Results

We perform two sets of experiments with the HULM: 1) a set of experiments in which we evaluate the performance of the model (and of the hidden CRF) as a function of the number of hidden units and 2) a set of experiments in which we compare the performance of all models on all data sets. The two sets of experiments are described separately in the following.

1) Effect of Varying the Number of Hidden Units: We first conduct experiments on the ASD data set to investigate the performance of the HULM as a function of the number of hidden units. The results of these experiments are shown in Fig. 6. The results presented in the figure show that the error initially decreases when the number of hidden unit increases, because adding hidden units adds complexity to the model that allows it to better fit the data. However, as the hidden unit number increases further, the model starts to overfit on the training data despite the use of L2-regularization.

We performed a similar experiment on the CK+ facial expression data set, in which we also performed comparisons with the hidden CRF for a range of values for the number of hidden states. Fig. 7 shows the results of these experiments. On the CK+ data set, there are no large fluctuations in the

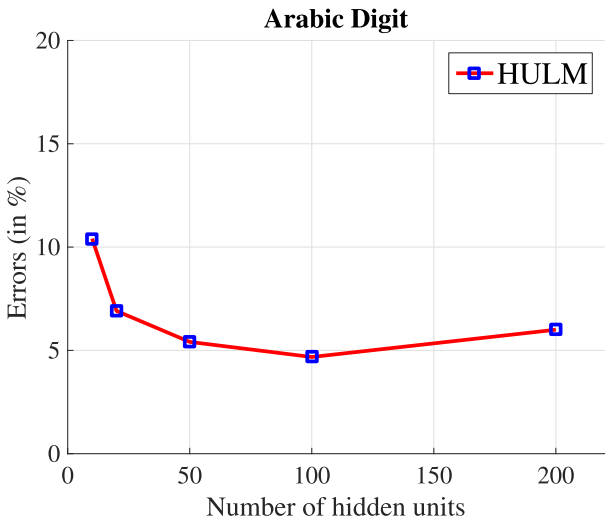


Fig. 6. Generalization error (in %) of the HULM on the Arabic speech data set as a function of the number of hidden units.

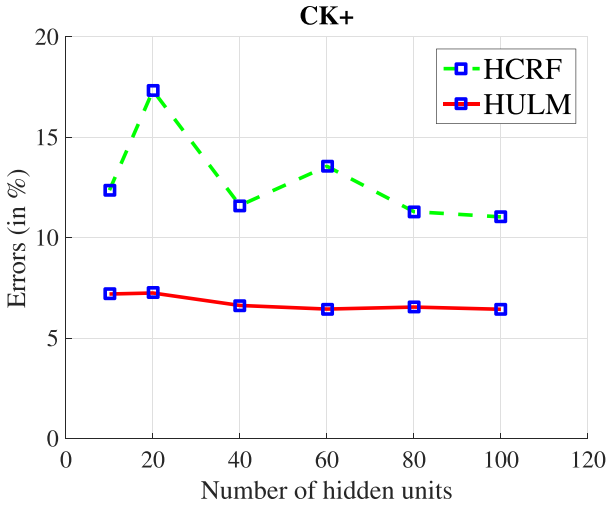


Fig. 7. Generalization error (in %) of the HULM and the hidden CRF on the CK+ data set as a function of the number of hidden units.

errors of the HULM as the hidden parameter number increases. The figure also shows that the HULM outperforms the hidden CRF irrespective of the number of hidden units. For instance, an HULM with 10 hidden units outperforms even a hidden CRF with 100 hidden parameters. This result illustrates the potential merits of using models in which the number of latent states grows exponentially with the number of parameters.

2) Comparison With Modern Time-Series Classifiers: In a second set of experiments, we compare the performance of the HULM with that of the NL model, FKL, and the hidden CRF on all eight problems. In our experiments, the number of hidden units in the HULM was set to 100; following [1], the hidden CRF used ten latent states. The results of our experiments are presented in Table I, and are discussed for each data set separately in the following.

a) Synthetic control: Synthetic Control is a simple univariate time-series classification problem from the UCR time-series classification archive [21]. Table I shows the generalization errors by four time-series classification models mentioned earlier. HULM model achieves the best performance with 1.33%, which is close to the state-of-the-art performance on this data set (0.7%) reported in [21]. This is an encouraging result, in particular, because the HULM method is not at all tuned toward solving univariate time-series classification problems.

b) Swedish leaf: Swedish Leaf is a much more challenging univariate time-series classification problem, whereas the NL model performs very poorly on this data set, all other three models achieves good performance, with the HULM slightly outperforming the other methods. It is worth mentioning that all three methods outperform the dynamic time warping approach that achieves 15.4% on this data set reported in [21]. We surmise that the strong performance of our models is due to the nonlinear features transformations these models perform. The state-of-the-art performance (6.24%) on this data set is obtained by the recursive edit distance kernels (REDKs) [30], which aims to embed (univariate) time series in time-warped Hilbert spaces while preserving the properties of elastic measure.

c) Online handwritten character data set: Following the experimental setup in [9], we measure the generalization error of all four models on the OHC data set using tenfold cross validation. The average generalization error of each model is shown in Table I. While the NL model performs very poorly on this data set, all three other methods achieve very low error rates. The best performance is obtained by FKL, but the differences between the models are very small on this data set, presumably, due to a ceiling effect.

d) ASDs data set (ASD-digit): Following [23], the error rates for the ASDs data set with *digit* as the class label in Table I were measured using a fixed training/test division: 75% of samples are used for training and left 25% of samples compose test set. The best performance on this data set is obtained by the hidden CRF model (3.68%), while our model has a slightly higher error of 4.68%, which in turn is better than the performance of FKL. It should be noted that the performance of the hidden CRF and the HULM is better than the error rate of 6.88% reported in [23] (on the same training/test division).

e) Arabic spoken digits data set (ASD-voice): In the experiment setup in which the speaker of a digit is the class label for the ASD data set, the classification problem becomes much harder than the *digit* version due to much more classes involved (88 subjects). Table I shows that HULM achieves the best performance and FKL also performs very well. While the NL model unsurprisingly performs very poorly, it should be noted that HULM significantly outperforms HCRF, which reveals the advantage of HULM in the case of challenging classification problem.

f) Facial expression data set (CK+): Table I presents generalization errors measured using tenfold cross validation. Folds are constructed in such a way that all videos by the same subject are in the same fold (the subjects appearing in test videos were not present in the training set). On the CK+ data set, the HULM substantially outperforms the hidden CRF model, obtaining an error of 6.44%. Somewhat surprisingly, the NL model also outperforms the hidden CRF model with

an error of 9.20%. A possible explanation for this result is that the classifying these data successfully does not require exploitation of temporal structure: many of the expressions can also be recognized well from a single frame. As a result, the NL model may perform well even though it simply averages over time. This result also suggests that the hidden CRF model may perform poorly on high-dimensional data (the CK+ data is 136-D) despite performing well on low-dimensional data such as the handwritten character data set (3-D) and the Arabic spoken data set (13-D).

g) MSR Action 3-D data set (Action): To measure the generalization error of the time-series classification models on the MSR Action 3-D data set, we followed the experimental setup of [26]: we used all videos of the five subjects for training, and used the videos of the remaining five subjects for testing. Table I presents the average generalization error on the videos of the five test subjects. The four models perform quite similarly, although the hidden CRF and the HULM do appear to outperform the other two models somewhat. The state-of-the-art performance on this data set is achieved by [31], which performs temporal downsampling associated with elastic kernel machine learning. Nevertheless, it performs cross validation on the all possible (252) combinations of training/test subject divisions. Hence, the direct comparison with our model is not straightforward.

h) MSR Daily Activity 3-D data set (Activity): On the MSR Daily Activity data set, we use the same experimental setup as on the action data set: five subjects are used for training and five for testing. The results in Table I show that the HULM substantially outperforms the hidden CRF on this challenging data set (but FKL performs slightly better).

In terms of the average rank over all data sets, the HULM performs very strongly. Indeed, it substantially outperforms the hidden CRF model, which illustrates that using a collection of (conditionally independent) hidden units may be a more effective way to represent latent states than a single multinomial unit. FKL also performs quite well in our experiments, although its performance is slightly worse than that of the HULM. However, it should be noted here that FKL scales poorly to large data sets: its computational complexity is quadratic in the number of time series, which limits its applicability to relatively small data sets (with fewer than, say, 10 000 time series). By contrast, the training of HULMs scales linearly in the number of time series and, moreover, can be performed using stochastic gradient descent.

V. Application to Facial AU Detection

In this section, we present a system for facial action unit (AU) detection that is based on the HULM. We evaluate our system on the Cohn–Kanade extended facial expression database (CK+) [24], evaluating its ability to detect ten prominent facial action units: namely, AU1, AU2, AU4, AU5, AU6, AU7, AU12, AU15, AU17, and AU25. We compare the performance of our facial action unit detection system with that of state-of-the-art systems for this problem. Before describing the results of these experiments, we first describe the feature extraction of our AU detection system and the setup of our experiments.

A. Facial Features

We extract two types of features from the video frames in the CK+ data set: 1) shape features and 2) appearance features. Our features are identical to the features used by the system described in [27]; the features are publicly available online. For completeness, we briefly describe both types of features in the following.

The *shape features* represent each frame by the vertical/horizontal displacements of facial landmarks with respect to the first frame. To this end, automatically detected/tracked 68 landmarks are used to form 136-D time series. All landmark displacements are normalized by removing rigid transformations (translation, rotation, and scale).

The *appearance features* are based on grayscale intensity values. To capture the change in facial appearance, face images are warped onto a base shape, where feature points are in the same location for each face. After this shape normalization procedure, the grayscale intensity values of the warped faces can be readily compared. The final appearance features are extracted by subtracting the warped textures from the warped texture in the first frame. The dimensionality of the appearance feature vectors is reduced using principal components analysis as to retain 90% of the variance in the data. This leads to 439-D appearance feature vectors, which are combined with the shape features to form the final feature representation for the video frames. For further details on the feature extraction, we refer to [27].

B. Experimental Setup

To gauge the effectiveness of the HULM in facial AU detection, we performed experiments on the CK+ database [24]. The database consists of 593 image sequences (videos) from 123 subjects with an average length of 18.1 frames. The videos show expressions from neutral face to peak formation, and include annotations for 30 AUs. In our experiments, we only consider the ten most frequent AUs.

Our AU detection system employs ten separate binary classifiers for detecting AUs in the videos. In other words, we train a separate HULM for each facial AU. An individual model thus distinguishes between the presence and nonpresence of the corresponding AU. We use a tenfold cross-validation scheme to measure the performance of the resulting AU detection system: we randomly select one test fold containing 10% of the videos, and use remaining nine folds are used to train the system. The folds are constructed such that there is no subject overlap between folds, i.e., subjects appearing in the test data were not present in the training data.

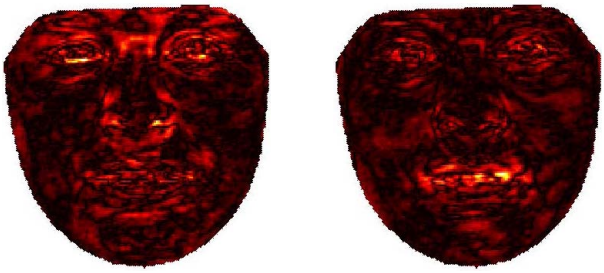
C. Results

We ran experiments using the HULM on three feature sets: 1) shape features; 2) appearance features; and 3) a concatenation of both feature vectors. We measure the performance of our system using the area under ROC curve (AUC). Table II shows the results for HULM, and for the baseline in [27]. The results show that the HULM outperforms the CRF baseline of [27], with our best model achieving an AUC that is approximately 0.03 higher than the best result of [27].

TABLE II

AUC OF THE HULM AND THE CRF BASELINE IN [27] FOR THREE FEATURE SETS. *IN [27], THE COMBINED FEATURE SET ALSO INCLUDES SIFT FEATURES

Method	Feature Set		
	Shape	Appear-ance	Combina-tion
HULM	0.9101	0.9197	0.9253
[27]	0.8902	0.8971	0.8647*



(a) (b)

Fig. 8. Visualization of $|\mathbf{W}|$ for (a) AU4 and (b) AU25. Brighter colors correspond to image regions with higher weights.

TABLE III

AVERAGE F1-SCORES OF OUR SYSTEM AND SEVEN STATE-OF-THE-ART SYSTEMS ON THE CK+ DATA SET. THE F1 SCORES FOR ALL METHODS WERE OBTAINED FROM THE LITERATURE. NOTE THAT THE AVERAGES ARE NOT OVER THE SAME AUS, AND CANNOT READILY BE COMPARED. THE BEST PERFORMANCE FOR EACH CONDITION IS BOLDFACED

AU	HULM	[32]	[33]	[34]	[35]	[36]	[37]
1	**0.91**	0.87	0.83	0.66	0.78	0.76	0.88
2	0.85	0.90	0.83	0.57	0.80	0.76	**0.92**
4	0.76	0.73	0.63	0.71	0.77	0.79	**0.89**
5	0.63	**0.80**	0.60	–	0.64	–	–
6	0.69	0.80	0.80	**0.94**	0.77	0.70	0.93
7	0.57	0.47	0.29	**0.87**	0.62	0.63	–
12	0.88	0.84	0.84	0.88	**0.90**	0.87	**0.90**
15	0.72	0.70	0.36	**0.84**	0.70	0.71	0.73
17	**0.89**	0.76	–	0.79	0.81	0.86	0.76
25	**0.96**	**0.96**	0.75	–	0.88	–	0.73
Avg.	0.79	0.78	0.66	0.78	0.77	0.76	0.84

To obtain insight in what features are modeled by the HULM hidden units, we visualized a single column of $|\mathbf{W}|$ in Fig. 8 for the AU4 and AU25 models that were trained on appearance features. Specifically, we selected the hidden unit with the highest corresponding $\mathbf{V}$-value for visualization, as this hidden unit apparently models the most discriminative features. The figure shows that the appearance of the eyebrows is most important in the AU4 model (brow lowerer), whereas the mouth region is most important in the AU25 model (lips part).

In Table III, we compare the performance of our AU detection system with that of seven other state-of-the-art systems in terms of the more commonly used F1-score (please note that the averages are not over the same AUs, and cannot readily be compared). The results in the table show that our system achieves the best F1 scores for AU1, AU17, and AU25. It performs very strongly on most of the other AUs, illustrating the potential of the HULM. Note that the state-of-the-art methods used in this comparison have specifically designed and optimized for AU detection task, while our approach is a direct application of the proposed HULM.

TABLE IV

PERFORMANCE OF HULM FOR DIFFERENT AUS USING COMBINED FEATURES. *P* SHOWS THE NUMBER OF POSITIVE SAMPLES. ACC, RC, AND DENOTE DETECTION ACCURACY, RECALL, AND PRECISION, RESPECTIVELY

AU	P	ACC	RC	PR	F1	AUC
1	175	0.95	0.88	0.93	0.91	0.96
2	117	0.94	0.84	0.86	0.85	0.96
4	194	0.86	0.71	0.83	0.76	0.90
5	102	0.88	0.62	0.64	0.63	0.88
6	123	0.88	0.63	0.77	0.69	0.92
7	121	0.82	0.58	0.56	0.57	0.81
12	131	0.95	0.88	0.89	0.88	0.95
15	95	0.91	0.75	0.70	0.72	0.92
17	203	0.92	0.91	0.87	0.89	0.97
25	324	0.95	0.95	0.97	0.96	0.97
Avg.	-	0.91	0.77	0.80	0.79	0.93

The detailed performance analysis of the proposed HULM, using combined features, is given in Table IV, where accuracy (ACC), recall (RC), precision (PR), F1, AUC measures, and number of positive samples are given for each AU.

VI. CONCLUSION

In this paper, we presented the HULM, a new model for the single-label classification of time series. The model is similar in structure to the popular hidden CRF model, but it employs binary stochastic hidden units instead of multinomial hidden units between the data and label. As a result, the HULM can model exponentially more latent states than a hidden CRF with the same number of parameters. The results of our experiments with HULM on several real-world data sets show that this may result in improved performance on challenging time-series classification tasks. In particular, the HULM performs very competitively on complex computer-vision problems, such as facial expression recognition.

In future work, we aim to explore more complex variants of our HULM. In particular, we intend to study variants of the model in which the simple first-order Markov chains on the hidden units are replaced by more powerful, higher order temporal connections. Specifically, we intend to implement the higher order chains via a similar factorization as used in neural autoregressive distribution estimators [38]. The resulting models will likely have longer temporal memory than our current model, which will likely lead to stronger performance on complex time-series classification tasks. A second direction for

future work we intend to explore is an extension of our model to multi-task learning. Specifically, we will explore multi-task learning scenarios in which sequence labeling and time-series classification is performed simultaneously (for instance, simultaneous recognition of short-term actions and long-term activities, or simultaneous optical character recognition and word classification). By performing sequence labeling and time-series classification based on the same latent features, the performance on both tasks may be improved, because information is shared in the latent features.

REFERENCES

[1] A. Quattoni, S. Wang, L. P. Morency, M. Collins, and T. Darrell, "Hidden Conditional Random Fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1848–1852, Oct. 2007.

[2] S. B. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2006, pp. 1521–1527.

[3] Y. Wang and G. Mori, "Learning a discriminative hidden part model for human action recognition," in *Proc. NIPS*, 2008, pp. 1–8.

[4] H. Larochelle and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," in *Proc. ICML*, 2008, pp. 536–543.

[5] L. van der Maaten, M. Welling, and L. Saul, "Hidden-unit conditional random fields," in *Proc. Int. Conf. Artif. Intell, Statist.*, 2011, pp. 479–488.

[6] L. A. Jeni, A. Lõrincz, Z. Szabó, J. F. Cohn, and T. Kanade, "Spatio-temporal event classification using time-series kernel based structured sparsity," in *Proc. ECCV*, 2014, pp. 135–150.

[7] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[8] T. Jaakkola, M. Diekhans, and D. Haussler, "A discriminative framework for detecting remote protein homologies," *J. Comput. Biol.*, vol. 7, nos. 1–2, pp. 95–114, 2000.

[9] L. van der Maaten, "Learning discriminative Fisher kernels," in *Proc. ICML*, 2011, pp. 217–224.

[10] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *J. Mach. Learn. Res.*, vol. 5, pp. 819–844, Jul. 2004.

[11] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labelling sequence data," in *Proc. ICML*, 2001, pp. 282–289.

[12] L.-P. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *Proc. CVPR*, Jun. 2007, pp. 1–8.

[13] J. Peng, L. Bo, and J. Xu, "Conditional neural fields," in *Proc. NIPS*, Dec. 2009, pp. 1419–1427.

[14] T.-M.-T. Do and T. Artières, "Neural conditional random fields," in *Proc. 13th Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 9. May 2010, pp. 177–184.

[15] K. Bousmalis, S. Zafeiriou, L. Morency, and M. Pantic, "Infinite hidden conditional random fields for human behavior analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 170–177, Jan. 2013.

[16] K. Bousmalis, S. Zafeiriou, L.-P. Morency, M. Pantic, and Z. Ghahramani, "Variational hidden conditional random fields with coupled Dirichlet process mixtures," in *Proc. ECML PKDD*, 2013, pp. 531–547.

[17] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[18] M. Welling, M. Rosen-Zvi, and G. Hinton, "Exponential family harmoniums with an application to information retrieval," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 17. 2004, pp. 1481–1488.

[19] S. Ray and D. Ren, "On the upper bound of the number of modes of a multivariate normal mixture," *J. Multivariate Anal.*, vol. 108, pp. 41–52, Jul. 2012.

[20] G. Montúfar and J. Morton, "When does a mixture of products contain a product of mixtures?" *SIAM J. Discrete Math.*, vol. 29, no. 1, pp. 321–347, 2015.

[21] Y. Chen *et al.*, (Jul. 2015). *The UCR Time Series Classification Archive*. [Online]. Available: www.cs.ucr.edu/~eamonn/time_series_data/

[22] B. Williams, M. Toussaint, and A. J. Storkey, "Modelling motion primitives and their timing in biologically executed movements," in *Proc. NIPS*, 2008, pp. 1609–1616.

[23] N. Hammami and M. Bedda, "Improved tree model for Arabic speech recognition," in *Proc. Int. Conf. Comput. Sci. Inf. Technol.*, Jul. 2010, pp. 521–526.

[24] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. CVPR Workshops*, Jun. 2010, pp. 94–101.

[25] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Proc. CVPR*, Jun. 2010, pp. 9–14.

[26] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. CVPR*, Jun. 2012, pp. 1290–1297.

[27] L. van der Maaten and E. Hendriks, "Action unit classification using active appearance models and conditional random fields," *Cognit. Process.*, vol. 13, no. 2, pp. 507–518, 2012.

[28] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. CVPR*, 2011, pp. 1297–1304.

[29] K. Bousmalis. *Hidden Conditional Random Fields Implementation*. [Online]. Available: http://www.doc.ic.ac.uk/~kb709/software.shtml

[30] P.-F. Marteau, and S. Gibet, "On recursive edit distance kernels with application to time series classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1121–1133, Jun. 2015.

[31] P. Marteau, S. Gibet, and C. Reverdy, "Down-sampling coupled to elastic kernel machines for efficient recognition of isolated gestures," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2014, pp. 363–368.

[32] S. Koelstra, M. Pantic, and I. Patras, "A dynamic texture-based approach to recognition of facial actions and their temporal models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 1940–1954, Nov. 2010.

[33] M. F. Valstar and M. Pantic, "Fully automatic recognition of the temporal phases of facial actions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 28–43, Feb. 2012.

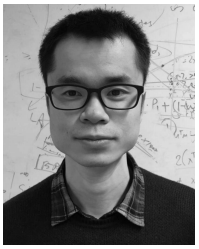
[34] Y. Li, J. Chen, Y. Zhao, and Q. Ji, "Data-free prior model for facial action unit recognition," *IEEE Trans. Affect. Comput.*, vol. 4, no. 2, pp. 127–141, Apr. 2013.

[35] Y. Li, S. Wang, Y. Zhao, and Q. Ji, "Simultaneous facial feature tracking and facial expression recognition," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2559–2573, Jul. 2013.

[36] X. Ding, V. Chu, F. De la Torre, J. F. Cohn, and Q. Wang, "Facial action unit event detection by cascade of tasks," in *Proc. ICCV*, 2013, pp. 2400–2407.

[37] X. Zhang, M. H. Mahoor, S. M. Mavadati, and J. F. Cohn, "A l_p-norm MTMKL framework for simultaneous detection of multiple facial action units," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 1104–1111.

[38] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator," *J. Mach. Learn. Res.*, vol. 15, pp. 29–37, 2011.



Wenjie Pei received the B.S. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, the M.Sc. degree in computer graphics and visualization from Zhejiang University, Hangzhou, China, in 2011, and the M.Sc. degree in computer science specialized in data mining from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2013. He is currently pursuing the Ph.D. degree with the Pattern Recognition and Bioinformatics Group, Delft University of Technology, Delft, The Netherlands, Co-Supervised by D. M. J. Tax and L. van der Maaten.

In 2016, he was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA. His current research interests include time series classification, time series similarity embedding learning, and time-series related applications.

Hamdi Dibeklioğlu (S'08–M'15) received the M.Sc. degree from Boğaziçi University, Istanbul, Turkey, in 2008, and the Ph.D. degree from the University of Amsterdam, Amsterdam, The Netherlands, in 2014.

He was a Visiting Researcher with Carnegie Mellon University, Pittsburgh, PA, USA, the University of Pittsburgh, Pittsburgh, and the Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Post-Doctoral Researcher with the Pattern Recognition and Bioinformatics Group, Delft University of Technology, Delft, The Netherlands. His current research interests include affective computing, intelligent human–computer interaction, pattern recognition, and computer vision.

Dr. Dibeklioğlu was a Co-Chair of the Netherlands Conference on Computer Vision 2015, and a Local Arrangements Co-chair of the European Conference on Computer Vision 2016. He served on the Local Organization Committee of the eNTERFACE Workshop on Multimodal Interfaces, in 2007 and 2010.

David M. J. Tax received the M.Sc. degree in physics from the Radboud University in Nijmegen, The Netherlands, in 1996, with his thesis Learning of Structure by Many-take-all Neural Networks. In 2001, he received the Ph.D. degree with the thesis One class Classification from the Delft University of Technology, Delft, The Netherlands, under the supervision of Dr. R. P. W. Duin.

He was a MarieCurie Fellow with the Intelligent Data Analysis Group, Berlin. He is currently an Assistant Professor with the Pattern Recognition Laboratory, Delft University of Technology. His current research interests include the learning and development of detection algorithms and (one-class) classifiers that optimize alternative performance criteria, such as ordering criteria using the area under the ROC curve or a precision-recall graph. Furthermore, the problems concerning the representation of data, multiple instance learning, simple and elegant classifiers, and the fair evaluation of methods have focus.

Laurens van der Maaten received the Ph.D. degree from Tilburg University, Tilburg, The Netherlands.

He was a Post-Doctoral Researcher with the University of California at San Diego, La Jolla, CA, USA. He was a Visiting Ph.D. Student with the University of Toronto, Toronto, ON, Canada. He is currently an Assistant Professor in computer vision and machine learning with the Delft University of Technology, Delft, The Netherlands. His current research interests include time series models, computer vision, dimensionality reduction, and classifier regularization.