Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

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

Action classification using a discriminative multilevel HDP-HMM

Natraj Raman*, S.J. Maybank

Department of Computer Science and Information Systems, Birkbeck, University of London, London, UK

ARTICLE INFO

Article history:
Received 13 May 2014
Received in revised form
16 October 2014
Accepted 4 December 2014

Keywords:
Action classification
Depth image sequences
HDP-HMM
Discriminative classification
Slice sampling

ABSTRACT

We classify human actions occurring in depth image sequences using features based on skeletal joint positions. The action classes are represented by a multi-level Hierarchical Dirichlet Process-Hidden Markov Model (HDP-HMM). The non-parametric HDP-HMM allows the inference of hidden states automatically from training data. The model parameters of each class are formulated as transformations from a shared base distribution, thus promoting the use of unlabelled examples during training and borrowing information across action classes. Further, the parameters are learnt in a *discriminative* way. We use a normalized gamma process representation of HDP and margin based likelihood functions for this purpose. We sample parameters from the complex posterior distribution induced by our discriminative likelihood function using elliptical slice sampling. Experiments with two different datasets show that action class models learnt using our technique produce good classification results.

1. Introduction

Recognizing actions that occur in videos has applications in diverse areas such as smart surveillance, search and retrieval of video sequences and human computer interaction. Depth sensors such as Kinect, with inbuilt human motion capturing techniques, provide estimates of a human skeleton's 3D joint positions over time [1]. High level actions can be inferred from these joint positions. However, robust and accurate inference is still a problem.

Given a sequence of 3D joint positions, a state space model such as a Hidden Markov Model (HMM) is a natural way to represent an action class. Recall that in an HMM [2], a sequence of discrete state variables are linked in a Markov chain by a state transition matrix and each observation in the sequence is drawn independently from a distribution conditioned on the state. The HMM model parameters (viz. the state transition matrix and the state specific observation density) corresponding to each class can be learnt from prototypes belonging to the class. The prediction of a new input's class is obtained from the class conditional posterior densities.

In classical parametric HMMs, the number of states must be specified a-priori. In many applications this number is not known in advance. A typical ad hoc procedure is to carry out training using different choices for the number of states and then apply a model selection criterion to find the best result. Instead it is preferable to estimate the correct number of states automatically from data. Further, when training the HMM models, the focus is on *explaining* the examples of a particular class rather than *discriminating* them from other classes. Often this does not produce good classification results [3].

Non parametric Bayesian methods such as mixture modelling based on the Dirichlet Process (DP) estimate the number of mixture components automatically from data. The Hierarchical Dirichlet Process (HDP), a mixed membership model for groups of data, allows mixture components to be shared across the groups albeit with group specific mixture proportions [4]. It uses a set of DP priors – one for each group – with these DP priors linked through a base DP in order to share the mixture components across groups. The HDP-HMM is a non-parametric variant of the classical HMM that allows an unbounded set of states with one mixture component corresponding to each state. Each state (group) in a HDP-HMM thus has state specific transition probabilities (mixture proportions) but the atoms are shared across the states (groups).

It would be straight forward to use separate HDP-HMMs for each action class and train them individually. However, this would prohibit sharing training examples across the action classes. To see the merit of sharing examples, consider that an action is a sequence of poses. It is quite likely that a set of actions have many similar poses between them with possibly a few poses unique to an action. In fact, for actions such as 'stand-up' and 'sit-down' or 'push' and 'pull', the set of poses may be identical with only the temporal order of pose sequences differing. What necessarily differentiates one action from another are the transition probabilities of the poses. If a particular pose is absent from an action class then there is a low probability of transition to the state for that pose. In our work, we use a single HDP-HMM to model

* Correspondence to: Department of Computer Science and Information Systems, Birkbeck, University of London, Malet St, London WC1E7HX, UK.
Tel.: +442076316700.
E-mail addresses: nraman01@dcs.bbk.ac.uk (N. Raman), sjmaybank@dcs.bbk.ac.uk (S.J. Maybank).

http://dx.doi.org/10.1016/j.neucom.2014.12.009
0925-2312/

all the action classes, but with an additional class specific hierarchical level that accounts for differences in the state transition probabilities among the classes (Fig. 1).

As outlined earlier, in a HDP-HMM the mixture components are shared across the hierarchical levels. It would be more flexible to allow the mixture components of an action class to vary *slightly* from the other classes; i.e. we seek a class specific transformation of the shared mixture component parameters so that we can better discriminate the classes. Note that this is different from using individually trained HDP-HMM model where the component parameters are unshared among the classes. We assume the mixture components are distributed as a Gaussian, and use class specific affine transformation of the Gaussian distribution parameters (mean and covariance) in our work here.

The HDP-HMM based classification approach described above defines a joint distribution of the input data and class labels to train the classifier. This *generative* model allows the augmentation of the labelled training examples with unlabelled examples and thus provides a framework for semi-supervised learning. In contrast, a *discriminative* model uses conditional distribution of the class labels given the input data to train the classifier. This approach often produces good classification results [5]. For example, Support Vector Machines (SVMs) use a margin based likelihood that maximizes the distances of the feature vectors to the classification boundary while minimizing the empirical error rate on the training set. Inspired by this, we incorporate a margin based term in the likelihood function used in HDP-HMM training. The inclusion of this discriminative term in the otherwise generative model, compensates for model mis-specification and leads to better classification results.

Incorporation of a discriminative term into the HDP-HMM model makes the posterior sampling less straight-forward. The HDP model construction as such has no provision for including an additional term for the mixing proportions. For the mixture components with Gaussian distribution parameters, the prior is not any more of the same form as the likelihood and hence is not conjugate. We use a normalized gamma process formulation [6] of the HDP that allows scaling the mixing proportions of a DP through a discriminative weighting term. Slice sampling [7] based techniques allow sampling from any likelihood function, not necessarily a normalized density function. Specifically, we place a Gaussian prior on the parameters and use Elliptical Slice Sampling [8] to efficiently sample the posterior.

We perform our experiments on two different datasets to classify actions based on the above discriminative two level HDP-HMM. Both data sets have annotated 3D joint positions estimated from a depth sensor. We use relative joint positions based on a pre-defined skeleton hierarchy as features. We also show our results for features obtained by projecting the relative joint positions into three orthogonal Cartesian planes and employing a histogram based representation.

This paper is organized as follows: in Section 2 we review related research and provide relevant background on HDP-HMM in Section 3. We detail our model construction in Section 4 and discuss the discriminative aspect in Section 5. Posterior inference is presented in Section 6 and the results of the experiments are shown in Section 7. Section 8 is a conclusion.

2. Related research

Ref. [9] provides a survey of research methods in action analysis and discusses the methodologies used for recognizing simple actions and high level activities. A review of depth image based algorithms for action analysis can be found in [10]. Ref. [1] provides the state-of-the-art method to extract joint positions from depth images captured by an infrared sensor.

In [11], each joint position is associated with a local occupancy feature and the actions are characterized using a subset of these joint positions called *actionlet*. A Fourier temporal pyramid is used to capture the temporal structure and discriminative weights are learnt using a multiple kernel learning method. Ref. [12] uses histograms of 3D joint locations (HOJ3D) as a compact representation for postures in order to demonstrate view invariant recognition. Linear Discriminant Analysis (LDA) is applied to the HOJ3D features. The resulting low dimensional features are clustered into visual words that are modelled

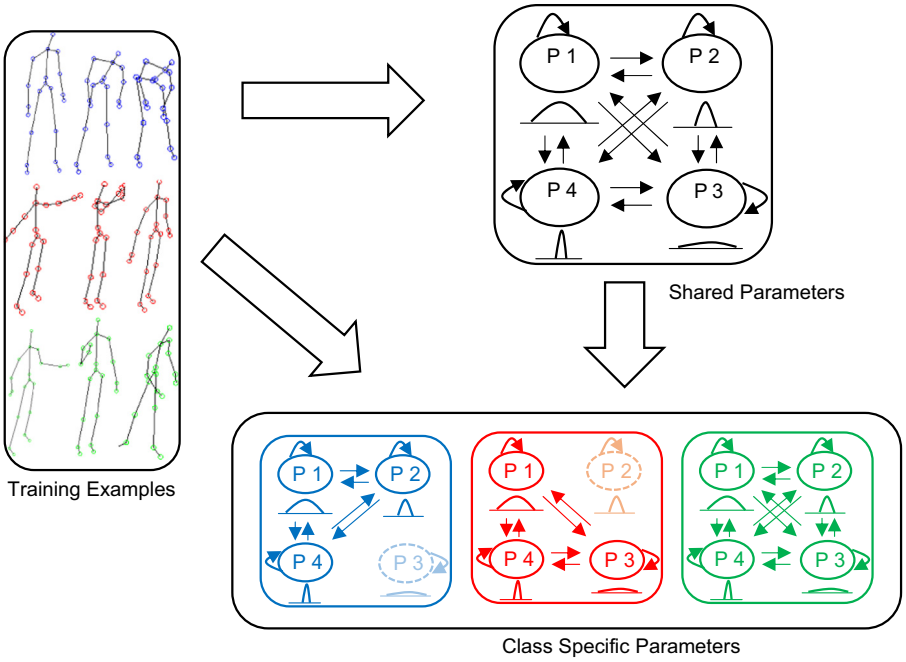


Fig. 1. Overview of action classification—training examples contain joint position sequences from different action classes. The examples from all these action classes are combined in order to infer the *shared* pose transitions and pose definitions. P1, P2, P3 and P4 represent the various poses (states) and each pose is defined through a distribution. The action class specific transitions and definitions are inferred as *transformations* of this shared representation. Pose P3 may be absent in the first action class and hence there is a low probability of transition to it.

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

by a discrete HMM. In [13], 3D trajectories of the joint positions are represented using a 2D trajectory based descriptor called histogram of oriented displacements (HOD) that is scale and speed invariant. A temporal pyramid is used to model the trajectories over time. The above works have mainly focused on computing appropriate features from the joint positions in order to classify the actions. In contrast, our focus in this paper is on a general classification mechanism for observation sequences in which training examples are shared among different classes and discriminatory model parameters are learnt. This allows our method to be applied to other sequence classification problems.

There is wide interest in the application of Bayesian non-parametric methods for vision problems. For example, [14] used a transformed Dirichlet Process to automatically learn the number of parts composing an object and the number of objects composing a scene. The model used is a single hierarchical level and the transformation parameters are applied only to the mixture components. In our work, we use a two-level HDP and extend the transformation to include mixture weights. Further our model parameters are learnt in a discriminative manner and provides a mechanism for sequence classification. In [15], unsupervised activity discovery is achieved using a beta process driven HMM to segment videos over time. While we use a similar idea of applying non-parametric frameworks for action recognition, our work differs vastly in the formulation (multi-level gamma process vs beta process) and application (supervised classification vs unsupervised pattern discovery). In [16], actions are segmented and clustered into behaviours using a hierarchical non parametric Bayesian model. This unsupervised approach uses a switching linear dynamic system to perform hierarchical clustering which is different from the parameter transformation based model that we use to perform classification.

Models based on HDP-HMM have been explored before. Ref. [17] provides a HDP-HMM based method for jointly segmenting and classifying actions with new action classes being discovered as they occur. The model used here is simply the HDP-HMM extended for streaming data by performing batch inferences. This is in contrast to the multi-level HDP-HMM with discriminatively learnt parameters that we use. Further, our method allows using unlabelled examples as part of the training procedure. In [32], a HDP-HMM based framework is used to detect abnormal activities. However, the HDP-HMM is used only to compute a feature vector and a one-class SVM is used to determine the decision boundary. We do not use any additional learning mechanism such as SVM and our method can be extended seamlessly to semi-supervised learning. Ref. [33] applies the sticky HDP-HMM proposed in [22] for error detection during a robotic assembly task. Our multi-level HDP-HMM that uses the normalized gamma process formulation is very different from this.

The application of large margin and other discriminative learning approaches [18] for training HMM is popular in the speech recognition literature. A survey of such approaches can be found in [19]. A discussion focusing on optimizing the HMM learning procedure for discriminative criteria such as Minimum Classification Error (MCE) and Maximum Mutual Information (MMI) can be found in [20]. In [21], a margin based approach for supervised topic models that minimized expected margin loss using Gibbs sampling methods is discussed. More recently, [34] proposes a discriminative multi-scale model based on SVM for predicting action classes from partially observed videos.

Although HDP-HMM has been used before for solving vision problems, using a *discriminative* training method for HDP-HMM has not been explored before. The merits of using margin based learning for classification is well discussed [20]. The HDP-HMM formulation as such doesn't provide any mechanism to learn model parameters discriminatively. Our approach of using a normalized gamma process formulation and the application of elliptical slice sampling provides a new technique for discriminative parameter learning in HDP-HMM. To the best of our knowledge, such a model has not been used before in the literature.

3. Background

In this section we provide relevant background on the classical HMM and it's non parametric variant HDP-HMM. For more details see [4,22].

3.1. Bayesian HMM

The classical HMM consists of an observation sequence $\{x_t\}_{t=1}^{T}$, $x_t \in \mathbb{R}^d$ and a corresponding hidden state sequence $\{z_t\}_{t=1}^{T}$, $z_t \in \{1,2\ldots K\}$. Here K is the number of hidden states. The hidden state sequence follows a first order Markov chain $z_t \perp z_{1:t-2} \mid z_{t-1}$ and the observations are conditionally independent given the current state i.e. $x_t \perp z_{1:t-1}, x_{1:t-1} \mid z_t$.

The probabilities of transitions between states are given by $\{\pi_{j,k}\}_{j=0,\ k=1}^{K}$ where $\pi_{j,k} = P(z_t = k | z_{t-1} = j)$ is the probability of transitioning to state k given the current state j and $\pi_{0,k} = P(z_1 = k)$ is the initial probability of state k. The observation distribution is parameterized as $P(x_t|z_t = k) \sim F(\theta_k)$ where θ_k are the natural parameters of the family F of distributions. Here we assume the observations are generated from a mixture of Gaussians, with one Gaussian distribution corresponding to each state. Hence $P(x_t|z_t = k) \sim \mathcal{N}(\mu_k, \Sigma_k)$ where $\mathcal{N}(\mu, \Sigma)$ is the normal distribution with mean μ and covariance Σ.

In this Bayesian approach it is necessary to introduce priors for all the parameters. Let H be the prior for θ or more specifically for the Gaussian mixtures let the mixture mean have a normal prior $\mu \sim \mathcal{N}(\mu_0, \Sigma_0)$ and let the covariance have an Inverse-Wishart prior $\Sigma \sim IW(\nu_0, \Delta_0)$. We can use a Dirichlet prior for the state transitions but we must ensure that the transitions out of different states are coupled. Hence let $\beta \sim Dir(\frac{\gamma}{K} \cdots \frac{\gamma}{K})$ and $\pi_j \sim Dir(\alpha\beta_1 \ldots \alpha\beta_K)$ where $\gamma, \alpha \in \mathbb{R}^+$ are the hyper priors, β_k is the probability of reaching state k and Dir is the Dirichlet distribution.

3.2. Stick breaking construction of DP

Draws from a Dirichlet Process $G_0 \sim DP(\gamma, H)$, where H is a base distribution and $\gamma \in \mathbb{R}^+$ a concentration parameter, are distributions G_0 containing values drawn from H with γ controlling the variability around H. The almost sure discreteness of measures drawn from a Dirichlet Process can be made explicit through the stick breaking construction and we can write

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}$$

$$\beta_k = \beta'_k \prod_{l<k} (1-\beta'_l) \quad \beta'_k \,\Big|\, \gamma \overset{iid}{\sim} Beta(1,\gamma) \quad \theta_k \,\Big|\, H \overset{iid}{\sim} H \tag{1}$$

where θ_k are the atoms drawn independently from the base distribution and β_k are the probabilities that define the mass on the atoms with $\sum_{k=1}^{\infty} \beta_k = 1$. It is common to write the probability measure $\beta = \{\beta_k\}_{k=1}^{\infty}$ obtained from (1) as $\beta \sim GEM(\gamma)$.

The DP is a useful nonparametric prior distribution for mixture models. If we interpret the β_k as a random probability measure on the set $\mathbb{Z}^+$, then we can write the generative story for an observation x_n sampled from a mixture model as

$$\beta|\gamma \sim GEM(\gamma) \quad \theta_k|H \sim H$$

$$z_n|\,\beta \sim \beta \quad x_n \mid z_n, \{\theta_k\}_{k=1}^{\infty} \sim F(\theta_{z_n}) \tag{2}$$

Here z_n is a latent variable that indicates the mixture component of the nth observation and F denotes the distribution family of the mixture component using θ as its parameter. Thus the DP can be

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

used to model a mixture with no upper bounds on the number of components. A component k has parameters θ_k and the probability that an observation is in the kth component is β_k.

3.3. Grouped data and HDP

The HDP is an extension of the DP to model grouped data. Each group is associated with a mixture model but all the groups share the same mixture components. Hence each group has a separate DP nonparametric prior and these DPs are linked through a different base DP.

As before, let G_0 be drawn from a Dirichlet Process. Let $\{G_j\}_{j=1}^{J}$ be the set of random distributions over J groups of data. Given the global measure G_0, the set of measures over the J groups are conditionally independent with

$$G_0 \mid \gamma, H \sim DP(\gamma, H) \quad G_j \mid \alpha, G_0 \sim DP(\alpha, G_0) \tag{3}$$

The global distribution G_0 contains values drawn from the base distribution H with γ controlling the variability. The jth group's distribution G_j contains values drawn from G_0 with α controlling the variability. The stick breaking construction of the global measure is same as (1) while the construction of the group specific measure can be formulated as

$$G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k}$$

$$\pi_{jk} = \pi'_{jk} \prod_{l<k} \left(1-\pi'_{jl}\right) \quad \pi'_{jk} \,\middle|\, \alpha, \beta \overset{iid}{\sim} Beta\left(\alpha\beta_k, \alpha(1-\sum_{l<k}\beta_l)\right) \tag{4}$$

The HDP is a useful nonparametric prior distribution for a mixture model set. If we interpret $\pi_j = \{\pi_{jk}\}_{k=1}^{\infty}$ as a random probability measure on the set $\mathbb{Z}^+$, then we can write the generative story for an observation x_{jn} belonging to the jth group as

$$\beta|\gamma \sim GEM(\gamma) \quad \pi_j|\alpha, \beta \sim DP(\alpha, \beta) \quad \theta_k|H \sim H$$
$$z_{jn}|\pi_j \sim \pi_j \quad x_{jn}|z_{jn}, \{\theta_k\}_{k=1}^{\infty} \sim F(\theta_{z_{jn}}) \tag{5}$$

Here z_{jn} is a latent variable that indicates the mixture component of the jth group's nth observation. For a component k, all the groups share the same parameters θ_k but the jth group uses π_{jk} proportion while the jth group uses $\pi_{j'k}$ proportion.

3.4. Non parametric HMM

As outlined in Section 3.1, there are K hidden states in the parametric HMM and a mixture component corresponding to each state. Given a state π_j, the jth row of the state transition matrix defines the mixing proportions. In a non-parametric HMM, the number of hidden states K is unbounded and the observations are now generated from an infinite mixture of components. Each state is associated with a (infinite) mixture model defining varying mixing proportions. In order to ensure that the transitions out of different states are coupled, the mixture models corresponding to the states must share the same mixture components (Fig. 2).

Thus the non-parametric HMM can be represented using a HDP —a set of (infinite) mixture models, capturing the state specific mixture proportions, with the mixture models linked through a global DP that ensures sharing the same mixture components. We can write the generative story for an observation x_{nt} sampled at time t from a HDP-HMM that uses Gaussian mixtures as

$$\begin{aligned} &\beta|\gamma \sim GEM(\gamma) && \pi_j \mid \alpha,\ \beta \sim DP(\alpha,\ \beta) \\ &\mu_k|\mu_0, \Sigma_0 \sim \mathcal{N}(\mu_0, \Sigma_0) && \Sigma_k|\nu_0, \Delta_0 \sim IW(\nu_0, \Delta_0) \\ &z_{nt}|z_{nt-1}, \pi_j \sim \pi_{z_{nt-1}} && x_{nt} \mid z_{nt}, \{\mu_k, \Sigma_k\}_{k=1}^{\infty} \sim \mathcal{N}(\mu_{z_{nt}}, \Sigma_{z_{nt}}) \end{aligned} \tag{6}$$

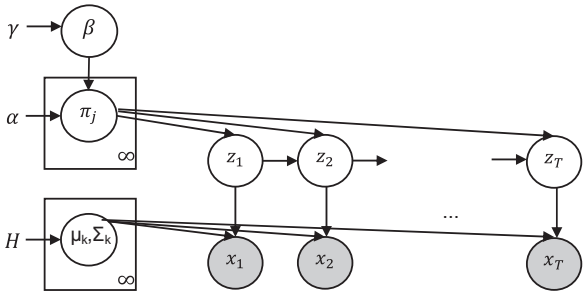


Fig. 2. Graphical representation of a HDP-HMM.

4. Model

We are given i.i.d training data $X = \{x^n\}_{n=1}^{N}$, $Y = \{y^n\}_{n=1}^{N}$, where $x^n = x_1^n \ldots x_T^n$ is an observation sequence and $y^n \in \{1 \ldots C\}$ its corresponding action class. An observation $x_t \in \mathbb{R}^d$ consists of features extracted from an image sequence at time-step t. We defer discussion on the features to Section 7. Let the set of all model parameters be θ. Our objective is classification, where given a new test observation sequence $\hat{x}$, we have to predict its corresponding action class $\hat{c}$. A suitable prediction is $\hat{c} = \text{argmax}_c p(c \mid \hat{x}, X, Y)$. The pdf $p(c \mid \hat{x}, X, Y)$ can be written in the form

$$p(c \mid \hat{x}, X, Y) = \int p(c \mid \hat{x}, \theta)\, p(\theta \mid X, Y)\, d\theta \tag{7}$$

4.1. Two level HDP

If we represent each action class by a separate HDP-HMM as outlined in Section 3, then $\theta^c = \{\beta^c, \pi^c, \mu^c_{1..\infty}, \Sigma^c_{1..\infty}\}$ are the parameters for class c with $\theta = \{\theta^c\}_{c=1}^{C}$ and $\gamma, \alpha, \mu_0, \Sigma_0, \nu_0, \Delta_0$ the hyper parameters. It would be straight forward to estimate the posterior density of parameters $p(\theta \mid X, Y)$ if each HDP-HMM model is trained separately i.e. we can define a class conditional density $p(x \mid c)$ for each class and estimate the posterior from

$$p(\theta^c \mid X, Y) = p(\theta^c) \prod_{n:y^n=c} p(x^n \mid \theta^c) p(c) \tag{8}$$

However, in this approach we do not make use of training examples from other classes while learning the parameters of a class. As noted in Section 1, many actions contain similar poses and it is useful to incorporate pose information from other classes during training. Specifically, the inclusion of additional observations for a similar pose benefits estimation of the Gaussian mixture parameters. The state transition parameters must continue to be different for each action class since it is these parameters that necessarily distinguish the actions.

Instead of separate HDP-HMMs, we define a single HDP-HMM for all the action classes albeit with an extra level that is class specific i.e. in addition to the global distribution G_0 and the state specific distributions G_j, we now have class specific distributions G_j^c for every state.

$$G_0 \mid \gamma, H \sim DP(\gamma, H) \quad G_j \mid \alpha, G_0 \sim DP(\alpha, G_0) \quad G_j^c \mid \lambda, G_j \sim DP(\lambda, G_j) \tag{9}$$

Just as the G_js are conditionally independent given G_0, the G_j^cs are conditionally independent given G_j. All the classes for a given state share the same subset of atoms but the proportions of these atoms will differ for each class determined by the concentration parameter λ. The varying atom proportions induce differences in state transition probabilities between action classes and ensure that classification can be performed. The stick breaking construction for the additional class

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

specific measure can be formulated as

$$G_j^c = \sum_{k=1}^{\infty} \varphi_{jk}^c \delta_{\theta_k}$$
$$\varphi_{jk}^c = \varphi_{jk}^{c\prime} \prod_{l<k} \left(1-\varphi_{jl}^{c\prime}\right) \quad \varphi_{jk}^{c\prime} \Big| \lambda, \pi_j \overset{iid}{\sim} Beta\left(\lambda\pi_{jk}, \lambda(1-\sum_{l<k}\pi_{jl})\right) \tag{10}$$

Similar to β and π_j, if we interpret $\varphi_j^c = \{\varphi_{jk}^c\}_{k=1}^{\infty}$ as a random probability measure on the set $\mathbb{Z}^+$, we can write the generative story for an observation x_t^n belonging to class c sampled at time t from the two level HDP-HMM that uses Gaussian mixtures as

$$\begin{aligned}
&\beta|\gamma \sim GEM(\gamma) && \pi_j|\alpha,\beta \sim DP(\alpha,\beta) && \varphi_j^c|\lambda,\ \pi_j \sim DP(\lambda,\ \pi_j)\\
&\mu_k|\mu_0,\Sigma_0 \sim \mathcal{N}(\mu_0,\Sigma_0) && && \Sigma_k|\nu_0,\Delta_0 \sim IW(\nu_0,\Delta_0)\\
&z_t^n|z_{t-1}^n, y^n = c, \left\{\varphi_j^c\right\}_{j=1,c=1}^{\infty,C} \sim \varphi_{z_{t-1}^n}^c && && \\
&x_t^n|z_t^n, \{\mu_k,\Sigma_k\}_{k=1}^{\infty} \sim \mathcal{N}(\mu_{z_t^n},\Sigma_{z_t^n}) && &&
\end{aligned} \tag{11}$$

Typically restrictions on Λ would have to be enforced for computational tractability. A useful simplification is to set Λ equal to the identity matrix. This is equivalent to restricting the transformations to a translation of the Gaussian mean by ρ. We can also restrict Λ to be diagonal, to account for scaling.

We introduce class specific transformations based on (13) to the Gaussian mixture component parameters. Let the transformation variable responsible for shifting the mean have a zero mean normal prior i.e. $\rho \sim \mathcal{N}(0,\Omega_0)$. We focus only on scale transformations and assume Λ is diagonal. Effectively the scale transform variable is now a vector and we assign independent log normal prior for each element i.e. $\log(\Lambda_j) \sim \mathcal{N}(\vartheta_0,\sigma_0)$. An observation x_t^n belonging to class c sampled at time t from the two level HDP-HMM that uses Gaussian mixtures with transformed parameters is generated as

$$\begin{aligned}
&\beta|\gamma \sim GEM(\gamma) && \pi_j|\ \alpha,\beta \sim DP(\alpha,\beta) && \varphi_j^c|\ \lambda,\ \pi_j \sim DP(\lambda,\ \pi_j)\\
&\mu_k|\mu_0,\Sigma_0 \sim \mathcal{N}(\mu_0,\Sigma_0) && && \Sigma_k|\nu_0,\Delta_0 \sim IW(\nu_0,\Delta_0)\\
&\rho_k^c|\Omega_0 \sim \mathcal{N}(0,\Omega_0) && && \log(\Lambda_{jk}^c)|\ \vartheta_0,\sigma_0 \sim \mathcal{N}(\vartheta_0,\sigma_0)\\
&z_t^n|z_{t-1}^n, y^n = c, \left\{\varphi_j^c\right\}_{j=1,c=1}^{\infty,C} \sim \varphi_{z_{t-1}^n}^c && && \\
&x_t^n|z_t^n, y^n = c, \{\mu_k,\Sigma_k\}_{k=1}^{\infty}, \{\rho_k^c,\Lambda_k^c\}_{k=1,c=1}^{\infty,C} \sim \mathcal{N}(\Lambda_{z_t^n}^c\mu_{z_t^n} + \rho_{z_t^n}^c, \Lambda_{z_t^n}^c\Sigma_{z_t^n}\Lambda_{z_t^n}^{cT}) && &&
\end{aligned} \tag{15}$$

Consequently, for the two level HDP-HMM, the set of all model parameters is $\theta = \{\beta,\ \pi,\ \varphi^{1..C}, \mu_{1..\infty}, \Sigma_{1..\infty}\}$ with $\gamma,\alpha,\mu_0,\Sigma_0,\nu_0,\Delta_0,\lambda$ being the hyper parameters.

4.2. Transformed HDP parameters

As explained in Section 3.3, in a HDP the same atoms are used by the different groups i.e. the component parameters θ_k remain the same in all G_j (and G_j^c in case of an additional level). This is less flexible than allowing the parameters to vary across the groups. As an example, a squat pose encountered during the course of an action might *mostly* look the same across action classes such as sit up, sit down and pick-up while it may *slightly* vary for pick-up class. In this case, it would be useful to capture the deviation from the standard squat pose for this pick-up action class—i.e. we wish to introduce a transformation of the parameters from its canonical form [14].

Let $\tau(u;\phi)$ denote the transformation of a parameter vector u. In order to express the transformations through a change of observation coordinates, let us impose the restriction that there exist a complementary transformation $\tau'(v;\phi)$ of an observation v such that

$$f(v \mid \tau(u;\phi)) \propto f(\tau'(v;\phi) \mid u) \tag{12}$$

where f is a density function. The existence of τ' satisfying (12) is useful during inference. In our work, we consider the affine transformation of the Gaussian distribution parameters mean μ and covariance Σ. Let ρ be a vector and Λ be an invertible matrix. The transforms

$$\tau(\mu,\Sigma;\rho,\Lambda) = (\Lambda\mu+\rho,\ \Lambda\Sigma\Lambda^T) \quad \tau'(v;\rho,\Lambda) = \Lambda^{-1}(v-\rho) \tag{13}$$

ensure that the covariance matrix is positive (semi) definite and we have

$$\mathcal{N}\left(v;\ \Lambda\mu+\rho,\ \Lambda\Sigma\Lambda^T\right) \propto \mathcal{N}(\Lambda^{-1}(v-\rho);\ \mu,\Sigma) \tag{14}$$

Inclusion of the class specific transforms can be interpreted as an extension of the parameter space. The global measure is now being drawn from $G_0 \sim DP(\gamma, H_s \times H_1 \ldots \times \ldots H_C)$, where H_s is a base distribution for parameters that are shared across the classes while $H_1,\ldots,H_C$ are class specific. During inference, the posterior distributions for the shared parameters do not depend upon the class labels unlike the class specific parameters. With the augmentation of transform variables, the set of all model parameters is $\theta = \{\beta,\ \pi,\ \varphi^{1..C}, \mu_{1..\infty}, \Sigma_{1..\infty}, \rho_{1..\infty}^{1..C}, \Lambda_{1..\infty}^{1..C}\}$ and $\gamma,\alpha,\mu_0,\Sigma_0,\ \nu_0,\Delta_0,\lambda,\ \Omega_0,\vartheta_0,\sigma_0$ are the hyper parameters (Fig. 3). A summary of notations used can be found in Table 1.

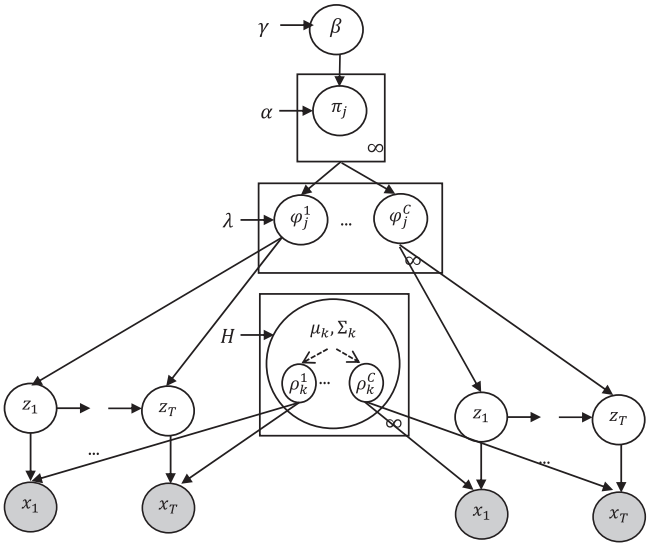


Fig. 3. Graphical representation of a two level HDP-HMM with transformed parameters—the observations on the left side belong to class '1' while those on the right side belong to class 'C'.

Table 1
Summary of notations.

General	
x^n	The nth training example sequence
y^n	The class that the nth training example belongs to
x_t	Observation at time instant t
z_t	Hidden state at time instant t
θ	Set of all model parameters
HDP-HMM	
β_k	Probability of transitioning to state k
π_{jk}	Probability of transitioning to state k given state j
μ_k	Mean of Gaussian distribution corresponding to component k
Σ_k	The covariance of Gaussian distribution corresponding to component k
γ	Hyper-prior for β
α	Hyper-prior for π
μ_0, Σ_0	Hyper-prior for μ
ν_0, Δ_0	Hyper-prior for Σ
Two level HDP-HMM	
φ^c_{jk}	Probability of transitioning to state k given state j for class c
λ	Hyper-prior for φ
Transformed HDP-HMM parameters	
ρ^c_k	Parameter for shifting mean μ_k for class c
Λ^c_k	Parameter for scaling covariance Σ_k for class c
ω^c_{jk}	Parameter used for scaling φ^c_{jk} for class c
Ω_0	Hyper-prior for sampling ρ
ϑ_0, σ_0	Hyper-prior for Λ
ε_0	Hyper-prior for ω
Posterior inference	
θ^c	Set of model parameters for class c
$\theta^{\setminus c}$	Set of model parameters excluding class c
θ^s	Set of model parameters shared for all the classes
L	Upper bound on the number of HMM states
n^c_{jk}	Number of transitions from state j to k for class c
n_{jk}	Number of transitions from state j to k for all the classes
$\mathcal{X}^c_k$	Set of observations from class c assigned to state k
ξ_0	Prior controlling importance of discriminative term
ζ_0	Prior controlling the distance between distributions

4.3. Chinese restaurant process metaphor

The mixture components generated by DP and its extensions described here can be better understood using a metaphor [4]. We have a restaurant with unbounded number of tables. A customer entering the restaurant either selects an unoccupied table with certain probability or selects an occupied table with a probability proportional to the number of customers already seated at the table. In this DP metaphor, the tables correspond to mixture components, the dish served at a table to the component parameters and the customers are observations.

In the HDP analogue, we have multiple restaurants with a single menu i.e. a restaurant franchise. The tables in the restaurants serve dishes from the shared menu and multiple tables in multiple restaurants can serve the same dish. A customer entering a given restaurant selects a table in proportion to the number of customers already seated in that restaurant's tables but can also select a new table. Each table is assigned a dish in proportion to the number of tables across the franchise serving that dish but a new dish can also be ordered. In this HDP metaphor, a restaurant correspond to a (data) group or in the case of HDP-HMM a state.

In the HDP extended to a second level, each restaurant in the franchise has sections viz. family, kids and adults section. There is still a single menu across the sections and the restaurants. Given the customer's preferred section, the customer entering a given restaurant selects a table in proportion to the number of customers already seated in the tables of that section of the restaurant. He can also select a new table in that section. Each table is now assigned a dish in proportion to the number of tables across the sections, across the franchise serving that dish. In this two-level HDP metaphor, the sections correspond to the action classes.

In the case of two-level HDP-HMM with transformed parameters, each dish now contains a base part and a flavouring part. A dish contains flavours for every section viz. spicy flavour for family, bland for kids and hot for adults. A dish served at a table in a given section (of any restaurant in the franchise) has its base part seasoned according to that section's flavour. In this metaphor, the flavours correspond to the class specific transform parameters while the base part correspond to parameters shared across the classes.

5. Discriminative learning

In the two level HDP-HMM with transformed parameters described above, let the model parameters specific to a class c be $\theta^c = \{\varphi^c, \rho^c_{1..\infty}, \Lambda^c_{1..\infty}\}$ and the shared parameters across the classes be $\theta^s = \{\beta, \pi, \mu_{1..\infty}, \Sigma_{1..\infty}\}$. Note that $\theta = \theta^s \cup \{\theta^c\}_{c=1}^{C}$. We will typically apply Gibbs sampling and use a very similar form to (8) to sample the class specific posterior.

$$p(\theta^c|X, Y, \theta^s) \propto p(\theta^c) \prod_{n:y^n = c} p(x^n|\theta^c, \theta^s) \tag{16}$$

The joint distribution over the inputs and labels $p(x, c \mid \theta^c)$ is used in this formulation. This type of learning is intended to best *explain* the training examples belonging to a class. In the asymptotic limit of infinite training examples and the distribution specified by the model being identical to the true distribution of data, it is a very effective way of learning. However, this *generative* model with its parameters learnt as above often produces poor classification results. In real world, the specified model is typically

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

inaccurate and we need to compensate for the model mis-specification.

In contrast, large margin based training used in discriminative learning methods often produces good classification results. The empirical error rate on the training data is balanced against the generalization of the test data. The tolerance to mismatch between training and test data is due to the classifier decision boundary being well separated from the classes—i.e. the decision boundary has a large margin to the training examples. Since the class conditional data likelihood is used during prediction in the generative model above, the classifier margin is a function of the model parameters and adjusting the parameters alters the margins.

There is an implicit assumption in (16) that the parameters of a class are (conditionally) independent of the parameters of other classes i.e. $\theta^c \perp \theta^{\setminus c} \mid \theta^s$. Let us relax this assumption and consider a slightly different formulation.

$$p(\theta^c|X,Y,\theta^s,\theta^{\setminus c}) \propto p(\theta^c) \times p\left(\theta^{\setminus c}|\theta^c,X,Y,\theta^s\right) \times \prod_{n:y^n=c} p(x^n|\theta^c,\theta^s) \quad (17)$$

Here we have made use of the Bayes theorem product rule for $p(\theta^c \mid X, \theta^{\setminus c})$. The introduction of the second term $p(\theta^{\setminus c} \mid \theta^c, X)$, referred henceforth as the discriminative term, offers more flexibility. For example, we can use this term during inference to minimize classification error on the training set and introduce margin constraints. This discriminative term compensates for the model mis-specification and improves classification results.

5.1. Scaled HDP and normalized gamma process

The HDP and its stick breaking construction does not provide any mechanism for influencing the per-group component proportions through additional factors. This makes incorporation of the discriminant during inference for φ^c tricky. An alternative construction for the last level in the two-level HDP in (10) is

$$G_j^c = \sum_{k=1}^{\infty} \frac{\varphi_{jk}^c}{\sum_{k'=1}^{\infty}\varphi_{jk'}^c}\delta_{\theta_k} \quad \varphi_{jk}^c \Big| \lambda,\pi_j \underset{\sim}{iid} Gamma(\lambda\pi_{jk},1) \quad (18)$$

A Dirichlet distributed vector can be generated by independently drawing from a gamma distribution and normalizing the values. Its infinite extension relates to this normalized gamma process construction. The representation in (18) as such does not allow using an additional factor. Let each component be associated with a latent location and let the group specific distribution of the HDP be formed by scaling the probabilities of an intermediate distribution. More specifically, let us modify the last level in the two-level HDP described in (9) as

$$G_j^{c\prime}|\lambda,G_j \sim DP(\lambda,G_j) \quad G_j^c|G_j^{c\prime},\omega_j^c \propto G_j^{c\prime} \times e^{\omega_j^c} \quad (19)$$

here $G_j^{c\prime}$ is an intermediate distribution for the existing parameters and ω_j^c is a scaling factor that depends on the latent location. Based on this *scaled* HDP structure, we can make use of the second variable of the gamma distribution and draw the class specific component proportions as

$$\varphi_{jk}^c \Big| \lambda,\pi_j,\omega_j^c \underset{\sim}{iid} Gamma\left(\lambda\pi_{jk},e^{-\omega_{jk}^c}\right) \quad (20)$$

The derivation of (20) follows from the property that if $y \sim Gamma(a,1)$ and is scaled by $b>0$ to produce $z=by$, then $z \sim Gamma(a,b^{-1})$. We refer the readers to [6] for a detailed discussion on this construction. In our case, this additional scaling factor allows incorporating the discriminative term. During inference, we have to draw ω_{jk}^c in such a way that the posterior φ_{jk}^c is primed for classification.

5.2. Elliptical slice sampling

We cannot use conjugate priors for the transform parameters $\rho_{1..\infty}^c, \Lambda_{1..\infty}^c$ because of the presence of the discriminative term. Hence there is no closed form solution for posterior inference of these parameters. In the absence of an analytical update we can resort to a Metropolis step, but it is necessary to find a proposal distribution. Complex tuning may also be required.

Slice sampling [7] methods provide an alternate solution for sampling from a pdf when the pdf is known up to a scale factor. The main idea is to sample points uniformly from a region under the true density curve and then use the sample points based on the horizontal coordinates. Let ϕ be a random variable from which we wish to draw samples and let f be a function proportional to the density of ϕ. Let ϕ^i be the current sample. In slice sampling, we first draw an auxiliary variable $u \sim \mathbb{U}[0,f(\phi^i)]$ that defines a horizontal slice. We then define a bracket (interval) around the current sample $B(\phi^i)$ and draw the new sample $\phi^{i+1} \sim \left\{\phi' \in B\left(\phi^i\right) : u < f(\phi')\right\}$.

The challenge in slice sampling is to define the bracket containing the current value from which the new value will be drawn. This is especially difficult if ϕ takes values in a high dimensional space, as in our work. If the density function for ϕ is a product of a likelihood function and a zero mean Gaussian prior, then Elliptical Slice sampling [8] provides a better sampling mechanism. Here a full ellipse is defined passing through the current sample and the brackets are determined by shrinking an angle variable.

Let $L(\phi)=p(.|\phi)$ be a likelihood function and $p_0=\mathcal{N}(\phi;0,\Sigma)$ be a multivariate normal prior with $f(\phi) \propto L(\phi)p_0$ the density function from which we wish to draw samples using Elliptical slice sampling. Similar to slice sampling, an auxiliary variable $u \sim \mathbb{U}[0,f(\phi^i)]$ is drawn first. We then draw $\nu \sim N(0,\Sigma)$ that defines an ellipse centered at the origin. An angle $\psi \sim \mathbb{U}[0,2\pi]$ determines the bracket and a new location is computed as $\phi' = \nu \sin\psi + \phi^i \cos\psi$. If $u < f(\phi')$, then ϕ' is accepted as the new sample ϕ^{i+1}; otherwise ψ is shrunk to determine a new location.

Since the angles are shrunk exponentially and the states considered for an update lie within a two dimensional plane, this technique provides an efficient mechanism for sampling high dimensional variables. We use Elliptical slice sampling for inferring the transform parameters $\rho_{1..\infty}^c, \Lambda_{1..\infty}^c$ from the density function defined in (17).

6. Posterior inference

The central computation problem is posterior inference for the parameters. Since it is intractable to compute the exact posterior, we will resort to Markov Chain Monte Carlo (MCMC) techniques to draw posterior samples from $p(\theta \mid X,Y)$. Recall that we have the shared parameters $\theta^s=\{\beta,\pi,\mu_{1..\infty},\Sigma_{1..\infty}\}$ and the class specific parameters $\theta^c=\{\varphi^c,\rho_{1..\infty}^c,\Lambda_{1..\infty}^c\}$ with $\gamma,\alpha,\mu_0,\Sigma_0,\nu_0,\Delta_0,\lambda,\Omega_0,\vartheta_0,\sigma_0$ as the hyper parameters. We can apply Gibbs sampling and sample the shared parameters θ^s first and then given θ^s, we can draw samples for each class one by one.

6.1. Truncated approximation

For sampling the HDP-HMM parameters, one option would be to marginalize over the infinite state transition distributions π and component parameters (μ,Σ) and sequentially sample the hidden states z_t. Unfortunately this technique, referred as *direct assignment* or *collapsed* sampler, exhibits slow mixing rates because the HMM states are temporally coupled.

A better technique is to *block sample* the hidden state sequence z_t using the standard HMM forward–backward algorithm [2]. In this *uncollapsed* sampler the state transition distributions and

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

component parameters are explicitly instantiated. In order to take account of the fact that there is no upper bound on the number of states and the corresponding parameters, we can employ slice sampling techniques [23,24] or use truncated approximations [25]. In almost sure truncations, for a given number L the stick breaking construction is discarded for $L+1, L+2 \ldots \infty$ by setting $\beta'_L = 1$ in (1). An alternative technique is to consider a *weak limit approximation* to DP and set

$$GEM(\gamma) \triangleq Dir(\frac{\gamma}{L}, \ldots \frac{\gamma}{L}) \tag{21}$$

here L is an upper bound on the number of components and as $L \to \infty$, the marginal distribution of this finite model approaches the DP. We use this weak limit approximation for its computational efficiency and consequently in (15) we have

$$\beta \Big| \gamma \sim Dir(\tfrac{\gamma}{L}, \ldots \tfrac{\gamma}{L}) \pi_j | \alpha, \beta \sim Dir(\alpha\beta_1, \ldots \alpha\beta_L) \varphi^c_j | \lambda, \pi_j \sim Dir(\lambda\pi_{j1}, \ldots \lambda\pi_{jL}) \tag{22}$$

Note that this is different from the classical parametric HMM with finite Dirichlet priors. The prior induced by HDP leads to a subset of L possible states with L being usually set to a large number. Given this truncated approximation, the standard forward–backward algorithm can be employed to sample the hidden state sequences.

6.2. Sampling state transitions

The sampler is initialized by drawing the initial value of the parameters from their respective priors. For a training example x^n whose $y^n = c$, given the state transitions $\{\varphi^c\}_{j=0,k=1}^{L,L}$, the component means $\{\Lambda^c_k \mu_k + \rho^c_k\}_{k=1}^{L}$ and the covariances $\{\Lambda^c_k \Sigma_k \Lambda^{cT}_k\}_{k=1}^{L}$, the hidden state sequence is sampled from

$$p(z^n_t = k) \propto \varphi^c_{z^n_{t-1}k}\, m_{t+1,t}(k) \mathcal{N}(x^n_t;\, \Lambda^c_k \mu_k + \rho^c_k,\, \Lambda^c_k \Sigma_k \Lambda^{cT}_k) \tag{23}$$

here $m_{t,t-1}(k)$ is the HMM backward message that is passed from z^n_t to z^n_{t-1} and is determined recursively as

$$m_{t,t-1}(k) = \sum_{j=1}^{L} \varphi^c_{kj}\; m_{t+1,t}(j) \mathcal{N}(x^n_t; \Lambda^c_j \mu_j + \rho^c_j,\, \Lambda^c_j \Sigma_j \Lambda^{cT}_j) \tag{24}$$

Let $n^c \in \mathbb{Z}^{L+1 \times L}$ be a matrix of counts computed from the sampled hidden state sequences with n^c_{jk} being the number of transitions from states j to k for class c. We use the notation $n^c_{\cdot jk}$ to denote the number of transitions from j to k for all the classes and $n^c_{\cdot \cdot k}$ to denote the number of transitions to k. The scaling factor ω^c_j in (19) is used as the discriminative term and we set it as

$$\omega^c_{jk} = \frac{1}{\varepsilon_0} \left[\frac{n^c_{jk} - n'_{jk} + D}{\sum_{k'} n^c_{jk'} - n'_{jk'} + D} \right] \tag{25}$$

Intuitively, the weight for a state k will be higher if there are fewer transitions to this state from classes other than c. Here ε_0 is a prior that controls the importance of the scaling factor and D is a sufficiently large constant to ensure that the scaling factor is positive. We now proceed to sample the posteriors as

$$\begin{aligned} &\beta \mid \gamma, \overline{m} \sim Dir(\tfrac{\gamma}{L} + \overline{m}_{\cdot 1}, \ldots \tfrac{\gamma}{L} + \overline{m}_{\cdot L}) \\ &\pi_j \\ &\quad \mid \alpha,\, \beta, \overline{n} \sim\; Dir(\alpha\beta_1 + \overline{n}_{j1}, \ldots \alpha\beta_L + \overline{n}_{jL}) \varphi^{c\prime}_{jk} \\ &\quad \mid \lambda, \pi_j, \omega^c_j, n^c \;\sim\; Gamma\left(\lambda\pi_{jk} + n^c_{jk}, e^{-\omega^c_{jk}}\right) \quad \varphi^c_{jk} = \frac{\varphi^{c\prime}_{jk}}{\sum_{k'=1}^{L} \varphi^{c\prime}_{jk'}} \end{aligned} \tag{26}$$

Here $\overline{m}$, $\overline{n}$ are auxiliary count matrices that are sampled from the class specific matrices n^c. In the Chinese restaurant metaphor, these matrices correspond to the number of tables across the franchise serving a dish and the number of tables across sections in a restaurant serving a dish. These auxiliary matrices and the hyper parameters γ, α, λ are sampled in the standard way as outlined in [4].

6.3. Sampling component parameters

We sample the shared parameters first and then the class specific parameters. Further we proceed by sampling posteriors one component at a time. Let the set of observations belonging to class c and assigned to hidden state k be $\mathcal{X}^c_k = \{x^n_t \in X : z^n_t = k \wedge y^n = c\}$ with $\mathcal{X}_k = \{\mathcal{X}^c_k\}_{c=1}^{C}$. For the mean and covariance parameters that are shared across the classes, we have conjugate priors and the posteriors can be computed using the standard closed form updates as

$$\Sigma_k \mid \nu_0, \Delta_0, \mu_k, \mathcal{X}_k \;\sim\; IW(\overline{\nu}_k, \overline{\nu}_k \overline{\Delta}_k)$$

$$\mu_k \mid \mu_0, \Sigma_0 \Sigma_k, \mathcal{X}_k \;\sim\; \mathcal{N}(\overline{\mu}_k, \overline{\Sigma}_k)$$

where

$$\begin{aligned} &\overline{\nu}_k = \nu_0 + |\mathcal{X}_k| \\ &\overline{\nu}_k \overline{\Delta}_k = \nu_0 \Delta_0 + \sum_{x_n \in \mathcal{X}_k} (x_n - \mu_k)(x_n - \mu_k)^T \\ &\overline{\Sigma}_k = (\Sigma_0{}^{-1} + |\mathcal{X}_k| \Sigma_k{}^{-1})^{-1} \\ &\overline{\mu}_k = \overline{\Sigma}_k \left(\Sigma_0{}^{-1} \mu_0 + \Sigma_k \sum_{x_n \in \mathcal{X}_k} x_n \right) \end{aligned} \tag{27}$$

For the transform parameters, we have to sample posterior from (17) after defining the form of $p(\theta^{\setminus c} \mid \theta^c, X, \theta^s)$. There are several choices for the discriminative term and one option is to set it based on distance between the distributions of component parameters. If the distribution distances are large, the parameters are well separated and this will result in a larger margin for the classifier decision boundary. For the state k of class c whose transform parameters need to be sampled, we set

$$\begin{aligned} p(\theta^{\setminus c} | \theta^c, \theta^s) = \prod_{c' \in \setminus c} \prod_{k'=1}^{L} \exp\{ -\xi_0 \max \\ \times \left(0,\;\; \zeta_0 - D\left(\mathcal{N}\left(\overline{\mu}^c_k, \overline{\Sigma}^c_k\right) || \mathcal{N}\left(\overline{\mu}^{c\prime}_{k'}, \overline{\Sigma}^{c\prime}_{k'}\right)\right)\right) \} \end{aligned}$$

$$\text{where } \overline{\mu}^c_k = \Lambda^c_k \mu_k + \rho^c_k \; \overline{\Sigma}^c_k = \Lambda^c_k \Sigma_k \Lambda^{cT}_k \tag{28}$$

Here $D(P||Q)$ measures the similarity between two distributions P and Q, ζ_0 is a prior that specifies the minimum separation distance and ξ_0 is a constant that controls the overall importance of the discriminative term. Since we have normal distributions in our case, we can use Hellinger or Bhattacharya distance as a similarity measure. Intuitively, we compare the distribution of a component k from class c that we wish to sample to all the competing classes and their corresponding components. If the distance is lesser than a pre-specified minimum separation, then the pdf value will be lower and perhaps the sample is inappropriate. The discriminative term specified in (28) is computationally simple since it does not involve the training examples and instead uses the sufficient statistics.

Another option for the discriminative term is to use the likelihood of observations. The idea here is to ensure that the Gaussian pdf value of an observation from class c assigned to a component k is larger than the pdf value of competing classes and their

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

Input: Training observations with their corresponding class labels and hyper parameters
Output: Samples of posterior parameters

1. Sample the initial values $\beta, \pi, \mu_{1..L}, \Sigma_{1..L}, \varphi^{1..C}, \rho_{1..L}^{1..C}, \Lambda_{1..L}^{1..C}$ from their respective hyper parameters.
2. Sample hidden state sequences z_t^n using HMM forward backward algorithm as per (23).
3. For all classes, compute the matrix of counts n^c from the sampled hidden states.
4. For all classes and all states, determine the scaling factor ω_{jk}^c as per (25).
5. Sample the top level stick breaking weights β according to (26) using an auxiliary count matrix.
6. Sample the state specific stick breaking weights π for all states according to (26) using an auxiliary count matrix.
7. Sample the class specific stick breaking weight φ for all classes and all states according to (26).
8. For all components, sample the shared covariance Σ_k and then the mean μ_k as per (27).
9. For all classes and for all components, use (28) or (29) in (17) and sample the transform parameters ρ_k^c, Λ_k^c using Elliptical slice sampling.
10. Sample the hyper parameters.
11. Repeat from step (2) to collect more samples.

Fig. 4. Posterior inference algorithm.

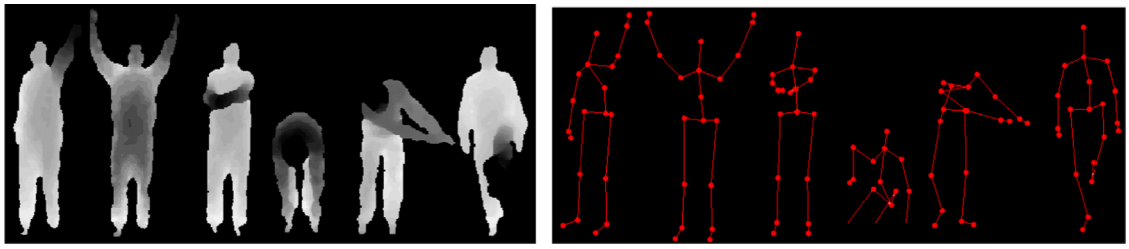


Fig. 5. Examples of actions from the MSR-Action3D dataset [26] *left*: depth images *right*: the corresponding joint positions.

corresponding components.

$$p(\theta'^c \mid \theta^c, \theta^s, X, Y) = \prod_{x_t^n \in X} \exp\left\{ -\xi_0 \max\left(0,\ \zeta_0 - \left(\mathcal{N}\left(x_t^n;\ \overline{\mu}_{z_t^n}^{y_n}, \overline{\Sigma}_{z_t^n}^{y_n}\right) - \max_{\substack{c' : y_n \neq c \\ k' = 1:L}} \mathcal{N}\left(x_t^n;\ \overline{\mu}_{k'}^{c'}, \overline{\Sigma}_{k'}^{c'}\right) \right) \right) \right\}$$

where

$$\overline{\mu}_k^c = \Lambda_k^c \mu_k + \rho_k^c$$
$$\overline{\Sigma}_k^c = \Lambda_k^c \Sigma_k \Lambda_k^{cT} \tag{29}$$

If we consider our model as a single component Gaussian instead of a HMM with Gaussian mixtures, then (29) encourages that the pdf value for the correct label must be greater than the pdf value of competing classes. The above discsriminative term can be treated as an approximation to the empirical error rate and ζ_0 offers the flexibility for a soft margin.

By plugging in (28) or (29) into (17), we get the posterior distribution for the transform parameters. We can sample $\Lambda_k^c \mid \rho_k^c, \mu_k, \Sigma_k$ and then $\rho_k^c \mid \Lambda_k^c, \mu_k, \Sigma_k$. Since the priors for both these variables are Gaussian distribution, we can use Elliptical slice sampling as specified in Section 5.2 for getting the posterior updates. Note that if we have a non-zero mean as Gaussian prior, we have to perform a shift to have zero mean. The inference algorithm is provided in Fig. 4.

7. Experiments

We conduct our experiments on the MSR Action3D [26] and UTKinect-Action [12] datasets. The datasets contain various actions performed by different subjects. However each action involves only one individual and there is no human-object interaction. All these datasets use an infrared camera to capture the depth image sequences. The datasets also contain annotated 3D joint positions of the subjects. These joint positions were estimated from the depth image sequence as explained in [1] and may have errors when there are occlusions. We work with these noisy joint positions (Fig. 5).

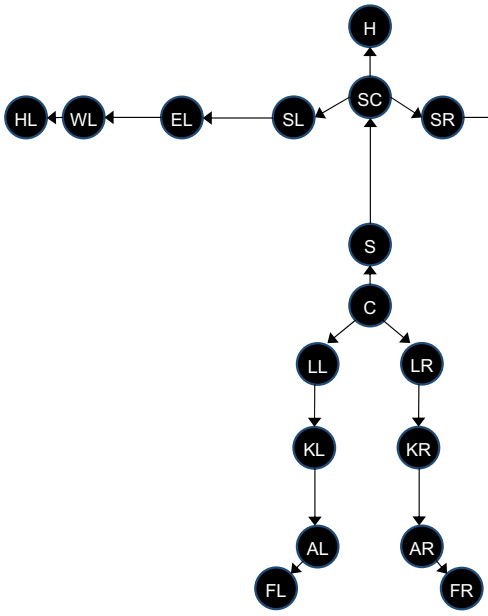
7.1. Joint position features

Each depth image contains $\{P_i\}_{i=1}^{20} \in (x, y, z)$ joint positions. We perform experiments on two types of features—one based on a subset of pairwise relative joint positions within a frame and another based on histogram of gradients that takes into account all combinations of joint positions and includes adjacent frames. For the first feature type, we determine 19 joint position pairs (P_i, P_j). The pairs are defined based on a pre-defined skeleton hierarchy as outlined in Fig. 6. The relative positions $P_i - P_j$ are used as features. Hence $x_t^n \in \mathbb{R}^{57}$. By using relative positions as features we ensure invariance to uniform translation of the body.

For the second feature type, we compute the relative position of a joint to all the other joints in the current frame and adjacent frames. Further, we use three 2D values instead of a single 3D value, representing the projection of a relative position on the orthogonal (xy, yz, xz) Cartesian planes i.e. for a joint i the features are

$$f_i(x,y) = \{(P_i^x - P_j^x,\ P_i^y - P_j^y) \quad \forall\ j \in P(t-1), P(t), P(t+1) \ \wedge\ i \neq j\}$$
$$f_i(y,z) = \{(P_i^y - P_j^y,\ P_i^z - P_j^z) \quad \forall\ j \in P(t-1), P(t), P(t+1) \ \wedge\ i \neq j\}$$
$$f_i(x,z) = \{(P_i^x - P_j^x,\ P_i^z - P_j^z) \quad \forall\ j \in P(t-1), P(t), P(t+1) \ \wedge\ i \neq j\} \tag{30}$$

where $P_i^x(t)$ is the x co-ordinate of the ith joint position at time t. We then assign the gradients $f_i(x,y)$ to a histogram of 8 bins based on the direction with the bin values being the gradient magnitude. This technique is very similar to the Histogram of Oriented



H	Head
SC	Shoulder Center
SL/SR	Shoulder Left/Right
EL/ER	Elbow Left/Right
WL/WR	Wrist Left/Right
S	Spine
C	Hip Center
HL/HR	Hand Left/Right
LL/LR	Leg Left/Right
KL/KR	Knee Left/Right
AL/AR	Ankle Left/Right
FL/FR	Feet Left/Right

Fig. 6. Skeleton Hierarchy used for defining joint position pairs [27]. The arrows indicate the parent–child joint pairs with Hip Center as the root joint.

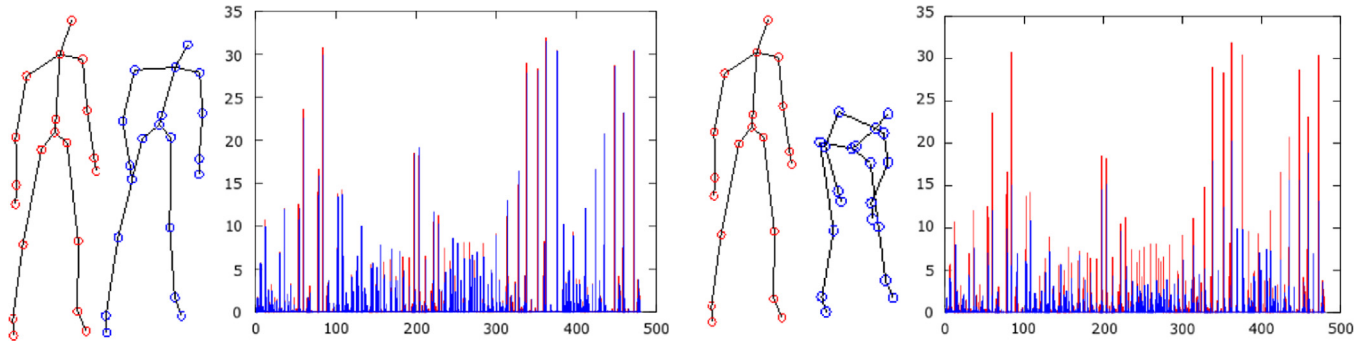


Fig. 7. Comparison of the HOG based descriptor—for two similar poses on the left, the corresponding descriptor values appear overlapped when compared with the dissimilar poses on the right.

Gradients (HOG) [28]. Repeating the step for $f_i(y,z)$ and $f_i(x,z)$ we now have 24 bins for each joint. Concatenating the bins for all the joints, we have a descriptor of length 20×24 for a frame and thus $x_t^n \in \mathbb{R}^{480}$. Finally we apply Principle Component Analysis (PCA) and use a subset of the Eigen vectors as features (Fig. 7).

7.2. UTKinect-Action dataset

We show results from the UTKinect-Action [12] dataset for human actions *walk, sit-down, stand-up, pick-up, carry, throw, pull, wave* and *clap-hands*. All these actions were performed in indoor settings with each action collected from 10 subjects and repeated twice. For each action, 60% of examples are used for training and the rest for testing. We use features based on the pairwise relative joint positions as shown in Fig. 6 for this dataset.

Parametric HMM: We first train a classifier, independently for each class, based on classical HMM. The standard Baum–Welch Expectation Maximization algorithm [2] is used for learning the HMM parameters. Since the number of states must be specified apriori for parametric HMMs, different numbers of states for each class are tried during training. In the absence of priors, an additional clustering step with K-Means is performed to estimate the initial values of transition matrix and the mean and covariance parameters. During testing, we evaluate a test example against all the classes and select the class with the largest (log) likelihood as the predicted class. Our observed best classification accuracy was ***58.2%***. The summary of classification results for HMM is presented in Table 2.

Table 2
Classical parametric HMM classification results.

Number of states	**Accuracy (%)**	**Precision (%)** (Average across classes)	**Recall (%)** (Average across classes)
3	49.3	52.3	50.0
5	48.1	60.3	48.7
7	53.1	64.4	53.7
10	***58.2***	65.6	58.7
15	55.6	70.8	56.2

HDP-HMM: We also train a HDP-HMM classifier, independently for each class as before. We specify an upper bound on the number of states ($L = 20$) as explained in Section 6.1. The number of states is automatically learnt from the data for HDP-HMM unlike the parametric HMM. In Fig. 8 the total number of states for the different action classes in a sample collected during training is shown. In an equivalent parametric HMM, we will have to run a tedious and adhoc model selection step individually for each class since the optimum cardinality of states vary between classes.

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

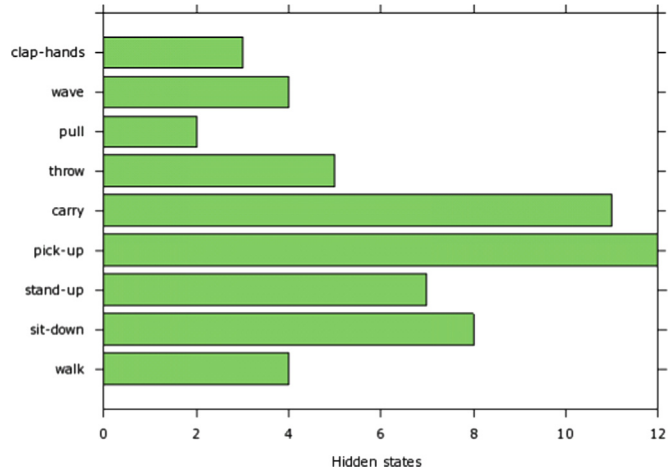


Fig. 8. The number of hidden states being active for different action classes in a sample collected during training. An active state is one in which at least one observation is assigned to this state.

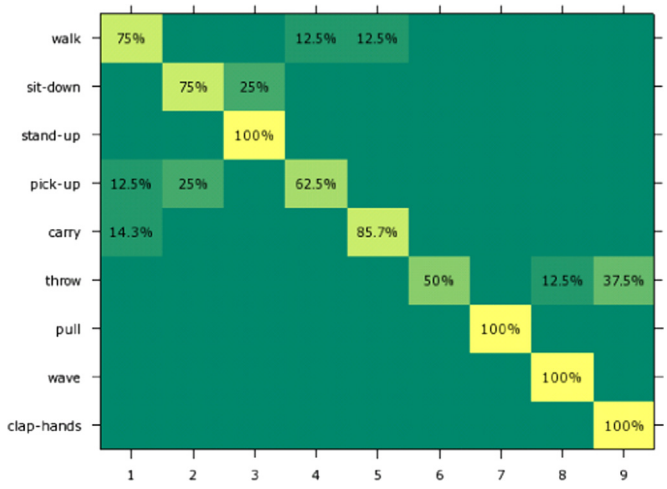


Fig. 9. Confusion matrix for classification results on UTKinect-Action dataset.

Table 3
HDP-HMM classification results.

Action	Precision (%)	Recall (%)
walk	100	87.5
sit-down	50	50
stand-up	66.6	100
pick-up	100	50
carry	77.7	100
throw	100	50
pull	100	100
wave	85.7	75
clap-hands	50	75

Table 4
Two-level HDP-HMM with generative learning classification results.

Action	Precision (%)	Recall (%)
walk	100	50
sit-down	63.6	87.5
stand-up	88.8	100
pick-up	80	50
carry	70	100
throw	100	50
pull	100	100
wave	75	75
clap-hands	58.3	87.5

This advantange of automatic state inference with HDP-HMM is reflected as an improved classification accuracy of ***76.1%***. The results are shown in Table 3.

Multi-level HDP-HMM with generative learning: We evaluate our results on the two-level HDP-HMM but exclude discriminative criterions. In this method, examples from all the classes are used during parameter estimation. Thus it allows sharing of parameters across classes and enables semi-supervised learning. In order to exclude the discriminative conditions for the state transitions, we simply set the scaling factor ω_j^c to zero. This is equivalent to sampling φ_{jk}^c (probability of transitioning to state k given we are in state j for a class c) as per Eq. (10) instead of (26). Similarly for the class specific transformation parameters, we set $p(\theta^{\setminus c})$ to be a constant in Eq. (17) thereby excluding the discriminative conditions. The classification results are shown in Table 4 and the accuracy is ***77.4%***. These results confirm that sharing of parameters across classes doesn't make the classification any worse. We interpret the lack of a big increase in accuracy when compared with HDP-HMM as an indication that there is a need for some additional discriminative condition. In addition, the smaller number of training examples in this dataset could have been a factor. Nevertheless, this technique provides a viable way to learn parameters in situations where we can incorporate unlabelled examples.

Multi-level HDP-HMM with discriminative learning: Finally, we evaluate our results on the two-level HDP-HMM including the discriminative conditions. The confusion matrix is shown in Fig. 9. It is evident from the confusion matrix that the recognition rate is good for most actions. However, there are a few mis-classifications for actions that involve very similar pose sequences. For example, some *sit-down* actions are classified as *stand-up*, *pick-up* actions as *sit-down* and *throw* actions as *clap-hands*. We report an overall classification accuracy of ***83.1%***.

7.3. MSR-Action3D dataset

We also conduct our experiments on the MSR Action3D [26] dataset. The dataset has 20 actions *high-arm-wave, horizontal-arm-wave, hammer, hand-catch, forward-punch, high-throw, draw-x, draw-tick, draw-circle, hand-clap, two-hand-wave, side-boxing, bend, forward-kick, side-kick, jogging, tennis-swing, tennis-serve, golf-swing, pickup-throw*. The actions were performed by 10 subjects and each one was repeated two or three times. Since some of the actions overlap, the actions are grouped into three sets as in [12,13] for performing classification. As before, we use 60% of examples for training and the rest for testing. We use features based on HOG descriptor as shown in (30) for this dataset. Using the multi-level HDP-HMM with discriminative learning, we report an overall classification accuracy of 81.2%, 78.1% and 90.6% for the three sets, respectively. The confusion matrix is shown in Fig. 10.

In this dataset as well, the classifier has incorrectly labelled few actions. These mis-classified actions involve very similar pose sequences. In particular, some *bend* actions are classified as *pickup-throw*, *hand-clap* actions as *tennis-serve* and *hand-catch* actions as *arm-wave*. The *draw-x*, *draw-tick* and *draw-circle* actions are challenging to classify and has less labelling accuracy compared to other actions.

Summary: A comparison of the classification results can be seen in Table 5. It is evident that the HDP-HMM improves classification accuracy significantly when compared with a parametric HMM. The multi-level HDP-HMM allows sharing the parameters across classes and it doesn't make the classification any worse.

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

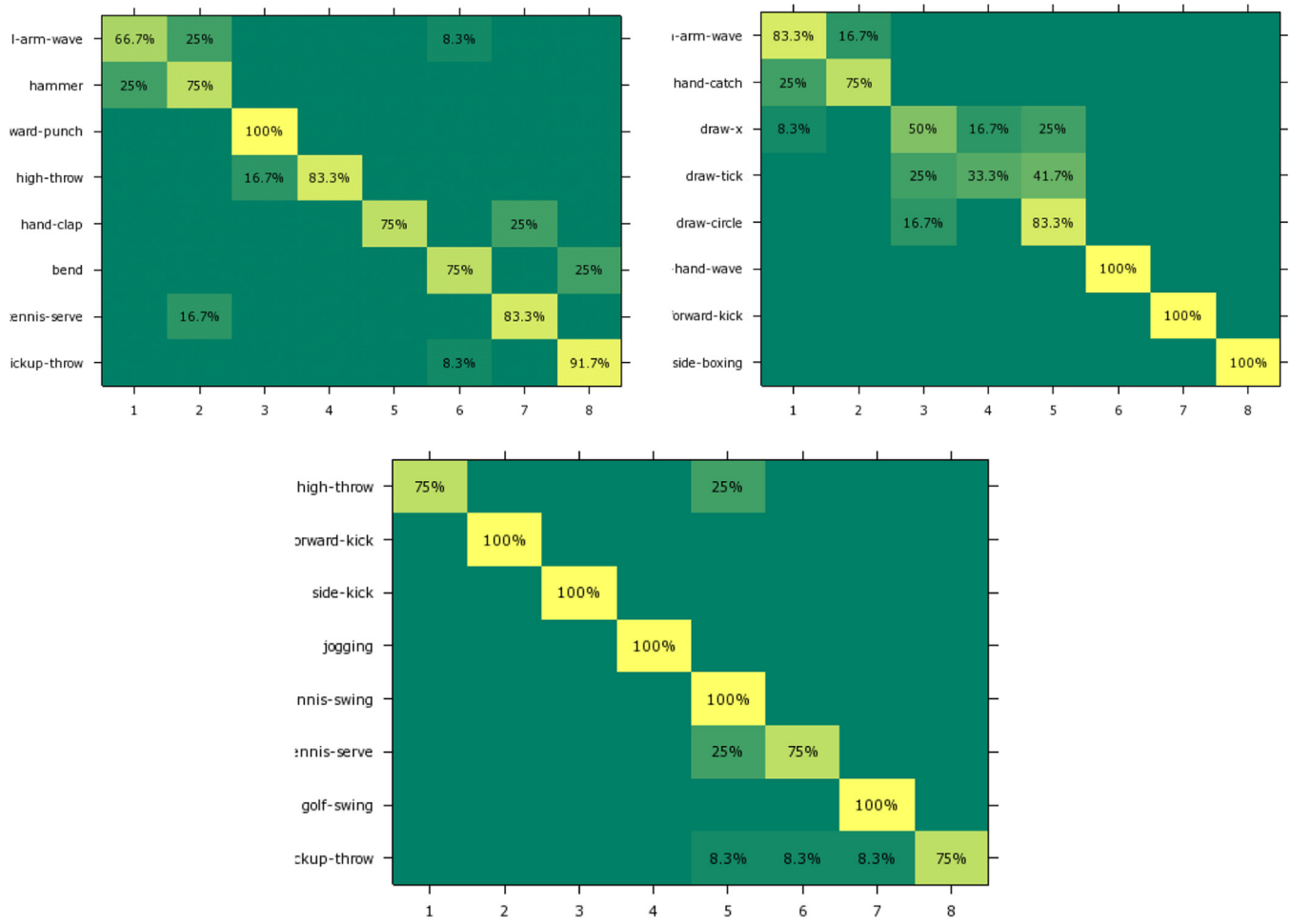


Fig. 10. Confusion matrix for classification results on MSR Action3D dataset— *top-left*: Actions horizontal-arm-wave, hammer, forward-punch, high-throw, hand-clap, bend, tennis-serve, pickup-throw, *top-right*: actions high-arm-wave, hand-catch, draw-x, draw-tick, draw-circle, two-hand-wave, forward-kick, side-boxing *bottom*: Actions high-throw, forward-kick, side-kick, jogging, tennis-swing, tennis-serve, golf-swing, pickup-throw.

Table 5
Summary of classification results.

Method	**UTKinect-Action *(Accuracy %)***
Parametric HMM	58.2
HDP-HMM	76.1
Multi-level HDP-HMM (Generative learning)	77.4
Multi-level HDP-HMM (Discriminative learning)	***83.1***
Xia et al. [12]	90.9
Devanne et al. [30]	91.5
Slama et al. [31]	95.2
Method	**MSR-Action3D *(Accuracy %)***
Parametric HMM	48.7
HDP-HMM	75.3
Multi-level HDP-HMM (Generative learning)	76.8
Multi-level HDP-HMM (Discriminative learning)	***83.3***
Li et al. [26]	74.7
Xia et al. [12]	78.9
Yang et al. [29]	82.3

The introduction of discriminative conditions on the multi-level HDP-HMM has improved the classification results. Our classifier accuracy in the MSR-Action3D dataset is better than the other approaches in [26,12,29]. However, our accuracy is less when compared with [12,30,31] for the UT-Kinect dataset.

Discussion: In our training, we completely exclude 40% of the subjects and use the instances of these subjects as test examples. This makes classification more difficult than in the alternative arrangement, in which training samples for all the subjects are included and only specific samples for each subject are excluded.

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009

In [12,30,31] a Leave-One-Out-Cross-Validation method is used. In a particular iteration, they use only one observation sequence for testing and the rest of the observation sequences are used for training. This procedure is repeated to include all the observation sequences for testing and finally the average accuracy across iterations is reported. In our experiments, we completely separate the training and test examples and use a more challenging cross subject evaluation. This tests the variations of actions performed by different subjects in a more realistic manner. Additionally, our features (relative joint position pairs) are much simpler and generic when compared with the features used in [12]. Inspite of the limited number of training examples, our experiments prove the utility of using a multi-level HDP-HMM with discriminative learning for classification purposes.

8. Conclusion

We have proposed an action classification method based on a multi-level HDP-HMM that shares training examples across action classes. The non-parametric nature of HDP-HMM allows an unbounded number of states. The normalized gamma process representation of the HDPs last level and the usage of elliptical slice sampling has allowed the inference of the posterior parameters in a discriminative way. Our experiments demonstrate the utility of this approach. We intend to broaden the discriminative criterions and apply our technique to classify activities involving humans and objects.

References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake. Real-time human pose recognition in parts from single depth images, in:. CVPR, 2011.
[2] L. Rabiner, B.H. Juang, An introduction to hidden Markov models., IEEE ASSP Mag., 3 (1986) 4–16.
[3] Tien-ho Lin, Naftali Kaminski, Bar-Joseph Ziv, Alignment and classification of time series gene expression in clinical studies., Bioinformatics 24 (13) (2008) i147–i155.
[4] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, M. Blei David, Hierarchical dirichlet processes., J. Am. Stat. Assoc. 101 (2006) 476.
[5] A. Lasserre, Julia., Christopher M. Bishop, Thomas P. Minka. Principled hybrids of generative and discriminative models Comput. Vision Pattern Recognit. (2006).
[6] J. Paisley, C. Wang, D. Blei, The discrete infinite logistic normal distribution., Bayesian Anal. (2012).
[7] Radford M. Neal, Slice sampling., Ann. Stat. (2003) 705–741.
[8] Iain Murray, Ryan Prescott Adams, JC MacKay. David, Elliptical Slice Sampling., arXiv preprint arXiv 1001 (2009) 0175.
[9] J.K. Aggarwal, S. Michael Ryoo. Human activity analysis: a review ACM Comput. Surv. (CSUR) 43.3: 16 (2011).
[10] Han, Jungong, Ling Shao, Dong Xu, Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: a review IEEE Trans. Cybern. (2013).
[11] Wang, Jiang, Zicheng Liu, Ying Wu, Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras Comput. Vision Pattern Recognit. (CVPR), (2012).
[12] L. Xia, C.C. Chen, J.K. Aggarwal, View invariant human action recognition using histograms of 3d joints, in: IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), 2012.
[13] M.A. Gowayyed, M. Torki, E.M. Hussein, M. El-Saban, Histogram of oriented displacements (HOD): describing trajectories of human joints for action recognition, in: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. AAAI Press (2013).
[14] E.B. Sudderth, A. Torralba, W.T. Freeman, A. S. Willsky, Describing visual scenes using transformed objects and parts. Int. J. Comput. Vision (2008).
[15] Hughes, C. Michael, B. Erik Sudderth. Nonparametric Discovery of Activity Patterns from Video Collections Computer Vision and Pattern Recognition Workshops, 2012.
[16] Kooij. F.P. Julian, Gwenn Englebienne, M. Dariu, Gavrila. A Non-parametric Hierarchical Model to Discover Behavior Dynamics from Tracks Computer Vision–ECCV, 2012.
[17] Bargi, Ava, R.Y.D. Xu, Massimo Piccardi. An online HDP-HMM for joint action segmentation and classification in motion capture data, in: CVPRW, 2012.
[18] D. Yu, L. Deng, Large-margin discriminative training of hidden Markov models for speech recognition, in: International Conference on Semantic Computing, IEEE, 2007.
[19] Hui. Jiang, Discriminative training of HMMs for automatic speech recognition: a survey., Comput. Speech Lang. 24 (4) (2010) 589–608.
[20] Xiaodong He, Li Deng, Wu Chou, Discriminative Learning In Sequential Pattern Recognition, Signal Processing Magazine, IEEE (2008) 14–36.
[21] J. Zhu, N. Chen, H. Perkins, B. Zhang, Gibbs max-margin topic models with fast sampling algorithms, in: Proceedings of the 30th International Conference on Machine Learning (ICML), 2013.
[22] Fox, B. Emilys, B. Erik Sudderth, I. Michael Jordan, and S. Alan Willsky, An HDP-HMM for systems with state persistence, in: Proceedings of the 25th international conference on Machine learning. ACM, 2008.
[23] J. Van Gael, Y. Saatci, Y.W. Teh, Z. Ghahramani, Beam sampling for the infinite hidden Markov model, in:25th International Conference on Machine Learning, pp. 1088-1095. ACM, 2008.
[24] M. Kalli, J.E. Griffin, S.G. Walker, Slice sampling mixture models., Stat. Comput. 21 (1) (2011) 93–105.
[25] H. Ishwaran, Zarepour, Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models., Biometrika 87 (2) (2000) (s371-39).
[26] Li, Wanqing, Zhengyou Zhang, Zicheng Liu. Action Recognition Based on a Bag of 3d Points Computer Vision and Pattern Recognition Workshops (CVPRW), 2010.
[27] Microsoft Developer Network, Joint Orientation, Kinect for Windows Retrieved from ⟨http://msdn.microsoft.com/en-us/library/hh973073.aspx⟩.
[28] N. Dalal, B. Triggs. Histograms of oriented gradients for human detection, in: CVPR, 2005.
[29] X. Yang, Y. Tian, Eigenjoints-based Action Recognition Using Naive-bayes-nearest-neighbor., Computer Vision and Pattern Recognition Workshops (CVPRW) (2012).
[30] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A Del Bimbo, Space-time pose representation for 3d human action recognition, in: ICIAP (2013).
[31] R. Slama, H. Wannous, M. Daoudi, Grassmannian representation of motion depth for 3D human gesture and action recognition., in: ICPR (2014).
[32] D.H. Hu, X.X. Zhang, J. Yin, V.W. Zheng, Q. Yang, Abnormal activity recognition based on HDP-HMM models, in: IJCAI (2009).
[33] E. Di Lello, T. De Laet, H. Bruyninckx, Hierarchical dirichlet process hidden markov models for abnormality detection in robotic assembly, in: NIPS (2012).
[34] Y. Kong, D. Kit, Y. Fu, A discriminative model with multiple temporal scales for action prediction, in: ECCV (2014).

Natraj Raman is a Ph.D. student in Computer Vision at Birkbeck, University of London. He received a Bachelors degree in Computer Science and Engineering from Bharathiyar University and a Masters degree in Intelligent Information Systems from University of London. His research focuses on recognizing activities that occur in video sequences. His research interests include image processing, swarm optimization and machine learning.

Stephen J. Maybank received the B.A. degree in mathematics from King's College Cambridge in 1976 and the Ph.D. degree in computer science from Birkbeck College, University of London in 1988. He was a research scientist at GEC from 1980 to 1995, first at MCCS, Frimley, and then, from 1989, at the GEC Marconi Hirst Research Centre in London. In 1995, he became a lecturer in the Department of Computer Science at the University of Reading and, in 2004, he became a professor in the Department of Computer Science and Information Systems at Birkbeck College, University of London. His research interests include camera calibration, visual surveillance, tracking, filtering, applications of projective geometry to computer vision and applications of probability, statistics and information theory to computer vision. He is the author of more than 120 scientific publications and one book. He is a Fellow of the IEEE and a Fellow of the Royal Statistical Society. He received the Koenderink Prize in 2008.

Please cite this article as: N. Raman, S.J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.12.009