Bayesian Hierarchical Dynamic Model for Human Action Recognition

Rui Zhao[1], Wanru Xu[2], Hui Su[1,3], Qiang Ji[1]
[1]RPI, [2]Beijing Jiaotong University, [3]IBM Research
{zhaorui.zju,bjtuxuwanru}@gmail.com, huisuibmres@us.ibm.com, qji@ecse.rpi.edu

Abstract

Human action recognition remains as a challenging task partially due to the presence of large variations in the execution of an action. To address this issue, we propose a probabilistic model called Hierarchical Dynamic Model (HDM). Leveraging on Bayesian framework, the model parameters are allowed to vary across different sequences of data, which increase the capacity of the model to adapt to intra-class variations on both spatial and temporal extent of actions. Meanwhile, the generative learning process allows the model to preserve the distinctive dynamic pattern for each action class. Through Bayesian inference, we are able to quantify the uncertainty of the classification, providing insight during the decision process. Compared to state-of-the-art methods, our method not only achieves competitive recognition performance within individual dataset but also shows better generalization capability across different datasets. Experiments conducted on data with missing values also show the robustness of the proposed method.

1. Introduction

Being able to recognize human action is crucial for understanding the intention of human. Over the past decades, numerous methods have been proposed to recognize human actions from visual inputs [53]. More recently, action recognition from 3D data becomes popular [1] with the availability of low-cost 3D sensing equipment and real-time 3D pose estimation technique [41, 30, 13]. Despite the significant progress made in this area, action recognition remains as one of the most challenging problems in computer vision partially due to significant variations caused by subject behavior, view change, occlusion, camera motion, cluttered background, *etc.* In particular, the difference of people's behavior in performing an action results in spatial and temporal intra-class variations. Even the same person may perform the same action differently. Such significant intra-class variation makes the inter-class difference vague.

In this paper, we address the issue of intra-class spatio-temporal variations for better action recognition. In addition, we provide a way of quantizing the uncertainty associated with classification, leveraging on Bayesian inference. We focus on the variations mainly caused by behavior difference rather than camera motion or occlusion and adopt the definition for such variations similar to [12]. The *spatial variation* is defined as body pose and appearance change when presenting a particular gesture. The *temporal variation* involves three factors: speed, duration and transition. Speed refers to the pace of executing an action. Duration represents the time spent in completing different phases of an action. Transition controls the change and order among different sub-actions. As an example, Figure 1 (left) shows skeleton joints of different subjects performing bowling action, which can be roughly divided into four phases including standing still, stepping forward, arm extending backward and leaning forward with arm extending forward. For spatial variation, different subjects stretch their arms and legs differently in both extent and orientation. For temporal variation, different subjects perform action using different orders of phases and spend different amount of time therein.

Our specific contributions are as follows. First of all, we propose the Hierarchical Dynamic Model (HDM), which is constructed to model different aspects of variations in a principled way. The temporal variation is handled in two aspects. First, we incorporate a probabilistic duration mechanism to allow flexible speed at each phase of an action. Second, the transitions among different phases of an action are modeled by transition probabilities among different hidden states. The spatial variation is modeled by probability distribution on observations at each individual frame. To further improve the capability of handling intra-class variation, we extend the model following Bayesian framework by allowing the parameters to vary across data, yielding a hierarchical structure. Secondly, we develop a learning algorithm to estimate the hyperparameters, which are usually treated as fixed in existing literatures. Furthermore, leveraging on Bayesian inference techniques, we propose a measure to quantize the uncertainty of the classification results. Finally, we conduct experiments on a variety of benchmark datasets to show the benefit of modeling variations and quantifying uncertainty for action recognition.

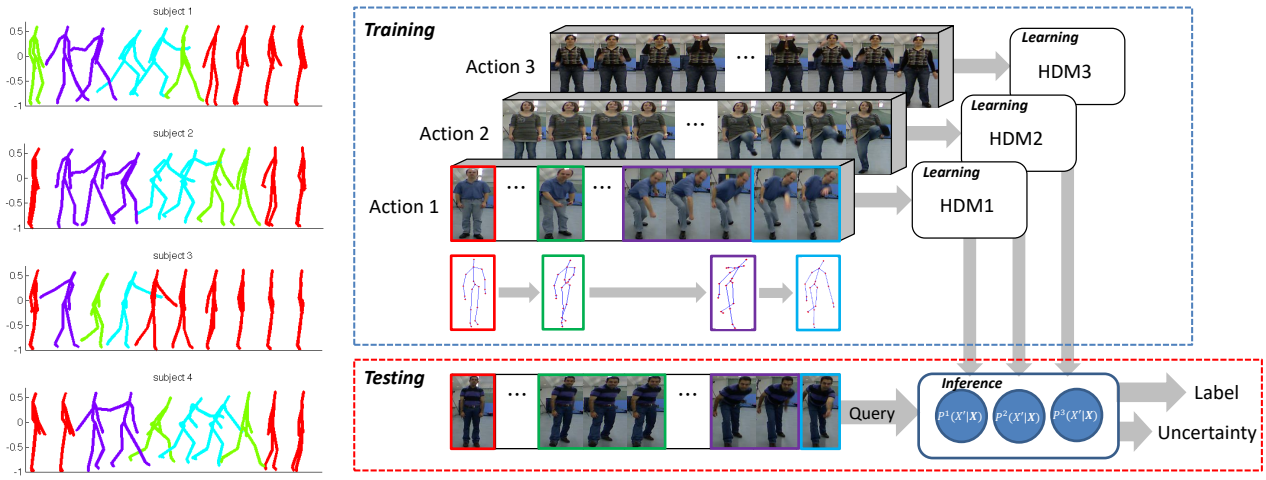

Figure 1. Left: Skeleton data examples from UTD-MHAD dataset [6] show spatial and temporal variations of different subjects performing the same action. All sequences have the same time scale. Different colors indicate different pose cluster assignments. Right: Overview of the recognition process. During training, we learn a set of models by fitting each model to its corresponding type of action (details in Section 3.2). During testing, the predictive likelihood computed by different models are used to determine class label and uncertainty (details in Section 3.3). Images are selected from Gaming 3D dataset [3]. (Best view in color)

We demonstrate our method has competitive classification performance, data efficiency, better generalization, and robustness to missing data.

2. Related Work

Modeling spatial and temporal variations: To account for spatial variation with known dynamic pattern, parametric HMM [54] and parametric switching LDS [37] are proposed by associating the observation probability with a global parameter. More flexible dynamic model with non-parametric observation model is also proposed in [12]. Despite better flexibility, it is difficult to generalize to poses that deviate from the training data due to variation. Our approach instead uses parametric distribution for pose features and leverages on the hierarchical extension for better generalization. To handle speed variation, Hidden semi-Markov Model (HSMM) [56] and its variants [11, 34, 33] are proposed to explicitly model the lasting time of hidden states. HSMM relaxes the Markov assumption of state transition in HMM and thus allows more flexible modeling of the dynamic process. Besides extending HSMM with structure, Bayesian extension of HSMM has been proposed in [15, 19] to further increase the modeling capacity. Another line of work tries to handle temporal variation through constructing a time-invariant representation of data. For instance, variants of temporal warping methods are used to handle recognition under speed variation [31, 46, 49]. Aggregate features extracted from different temporal scales are explored in [44, 50, 24], which can achieve certain temporal invariant representation. But its temporal granularities are manually decided. Our approach focuses on modeling dynamics of human action. We further improve the intra-class variation modeling capacity of HSMM by leveraging on Bayesian framework. Compared to existing work, we allow all the parameters to vary as random variables to account for spatial and temporal variations simultaneously. Furthermore, compared to previous work with fixed hyperparameters, we develop learning algorithm for hyperparameters estimation. The benefit of such extension is two-fold. First, the hierarchical structure allows the parameters to change across different data, while still sharing the property through prior distribution learned from all the within-class data. Second, the prior can regularize model complexity. Subject to the prior distribution, the model parameters can adapt to data variations without increasing model complexity, which helps avoid overfitting.

Action recognition frameworks: It is popular to adopt a discriminative framework for action recognition task, such as conditional random field (CRF) [23] and its extensions [39, 52, 27, 44]. Discriminative approaches mainly focus on modeling the conditional distribution of class labels in order to classify different classes. So it lacks the capability to model data distribution, which limits the use of discriminative model to classification only. Recently, deep learning framework becomes more popular as it can learn useful representation automatically. Typical approaches either use deep models to extract features to supply classifier learning [17, 55, 28] or combine variants of CNN and RNN to perform end-to-end learning [9, 43, 16, 40, 21, 42, 45]. It has been shown that modeling spatial and temporal dynamics is helpful for recognition [25, 10]. However, deep models rely on increasing model complexity to handle variations. It is prone to overfitting especially with limited data, thus proper regularization is essential [29, 59]. Joshi *et al.* [20]

proposed a Bayesian NN to better handle subject-dependent variation. We choose to use a generative model primarily due to its capability of capturing the data distributions subject to spatial and temporal variations. Furthermore, generative model can handle data with missing values. Compared to deep learning approach, HDM requires less training data and is less likely to overfit due to prior on parameters. It is also easier to train with very few model parameters to be tuned. Furthermore, the use of Bayesian inference allows us to quantify the uncertainty of the prediction to avoid overly confident but potentially incorrect predictions [22].

3. Methods

In this section, we introduce our methods, starting with a description of the model. Then we introduce learning and inference methods. We train one model for each type of action as illustrated by Figure 1 (right) and use the predictive likelihoods of the models for classification and uncertainty estimation.

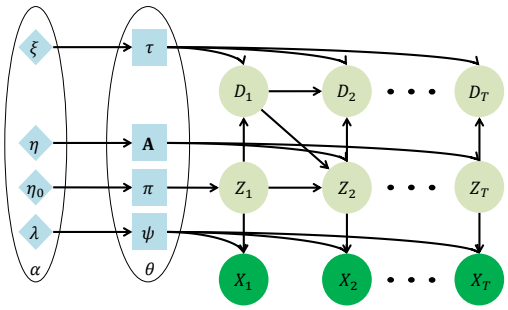

Figure 2. The topology of HDM.

3.1. Model description

Overview: Figure 2 shows the topology of our model. Random variables $\mathbf{X} = \{X_t \in \mathbb{R}^O\}_{t=1}^T$ represent a sequence of observations, where O is the dimension of each observation. $\mathbf{Z} = \{Z_t \in \{1, ..., Q\}\}_{t=1}^T$ represent hidden states associated with observations, where Q is the number of hidden states. $\mathbf{D} = \{D_t \in \{1, ..., T\}\}_{t=1}^T$ represent duration of the state *e.g.* $D_t = d$ means state chain $\mathbf{Z}$ remains at current value for the next d time stamps. X_t is continuous and observed, while Z_t and D_t are discrete and hidden. T can be different for different sequences. The parameters are $\theta = \{\pi, \mathbf{A}, \tau, \psi\}$, which specify the conditional distributions of random variables. The hyperparameters are $\alpha = \{\eta_0, \eta, \xi, \lambda\}$, which specify the prior distributions of parameters. The joint distribution of random variables is as follows.

$$P(\mathbf{X}, \mathbf{Z}, \mathbf{D}) = P(Z_1)P(D_1|Z_1)\prod_{t=1}^{T} P(X_t|Z_t) \qquad (1)$$
$$\prod_{t=2}^{T}[P(Z_t|D_{t-1}, Z_{t-1})P(D_t|D_{t-1}, Z_t)]$$

We use Gaussian mixture for emission distribution, Poisson for duration distribution and multinomial for initial state and transition distribution. The prior distributions of parameters are assumed independent of each other and conjugate prior is used *i.e.* $P(\theta|\alpha) = P(\pi|\eta_0)P(\mathbf{A}|\eta)P(\tau|\xi)P(\psi|\lambda)$. The detailed parameterization is provided in the supplementary materials.

Modeling temporal variation: The temporal variation is modeled at two levels. First, at the random variable level, the hidden state chain $\mathbf{Z}$ models the transition dynamics among different statuses, specified by initial state distribution $P(Z_1)$ and state transition distribution $P(Z_t|D_{t-1}, Z_{t-1})$ in Eq. (1). The duration of each state, which is mainly determined by the speed of action, is explicitly specified by $\mathbf{D}$ with distribution $P(D_t|D_{t-1}, Z_t)$ in Eq. (1). Second, to model temporal variation at parameter level, instead of fixing one set of parameters for all the within-class data, we allow parameters $\{\pi, \mathbf{A}, \tau\}$ to vary as random variables across different sequences, whose distributions are specified by hyperparameters $\{\eta_0, \eta, \xi\}$. On one hand, the hierarchy can accommodate large intra-class temporal variations as each sequence has its own temporal parameters. On the other hand, the parameters share the same prior, which is learned from all the within-class data. Thus the overall within-class temporal dynamics is preserved.

Modeling spatial variation: Similar to temporal variation, spatial variation is also modeled at two levels. First, at the random variable level, the observation X_t describes the pose or appearance at a given time t, specified by emission distribution $P(X_t|Z_t)$ in Eq. (1). Second, the spatial parameters ψ are also treated as random variables whose distributions are specified by hyperparameters λ. Different from temporal parameters, we do not vary spatial parameters across different sequences to ensure the consistency of hidden state value. Such hierarchy allows for large variation without needing to increase the mixture number, which regularizes the model complexity and avoids overfitting. Furthermore, since the prior is learned from data and shared by all spatial parameters, the overall within-class spatial distribution is preserved.

Generalization of existing models: Our model can be considered as a generalization of several existing models. If we set all the hyperparameters as fixed, it can be considered as Bayesian HSMM [1]. If we take out all the hyperparameters, it degenerates into explicit duration HMM [14]. If we further set $D_t = 1$ for all t, it reduces to HMM.

3.2. Learning

The goal of learning is to estimate hyperparameters α using training data, which is considered as an empirical Bayesian method. We fit one model for one action class

[1] A special case of Bayesian HSMM which has been proposed in [15] only considered placing prior on duration parameters.

so that each model only captures intra-class variation in the corresponding class. The following learning process applies to model for each class. The maximum likelihood estimation is an initial attempt to estimate α, which requires the integration of both hidden variables and model parameters.

$$\alpha^* = \arg\max_{\alpha} \log P(\{\mathbf{X}_n\}|\alpha) \tag{2}$$
$$= \arg\max_{\alpha} \log \int_{\theta} \prod_{n} \sum_{\mathbf{Z}_n, \mathbf{D}_n} P(\mathbf{X}_n, \mathbf{Z}_n, \mathbf{D}_n|\theta) P(\theta|\alpha) d\theta$$

where n is the index of sequence. However, the integration over transition parameters introduces additional dependencies among hidden variables that are not directly linked together. Thus the efficient forward-backward type of inference can no longer be performed. For sequence with more than moderate length, the summation becomes intractable. To bypass the integration challenge, we instead estimate α as follows.

$$\alpha^* = \arg\max_{\alpha} \log \prod_{n} \sum_{\mathbf{Z}_n, \mathbf{D}_n} P(\mathbf{X}_n, \mathbf{Z}_n, \mathbf{D}_n|\theta^*) P(\theta^*|\alpha) \tag{3}$$

where θ^* is one particular choice of θ. It leads to an alternating estimation process between θ and α. First, we compute MAP estimation of θ given current estimate of α. The objective of the estimation is the same as Eq. (3), except that the target variable becomes θ.

$$\theta^* = \arg\max_{\theta} \sum_{n} \log \sum_{\mathbf{Z}_n, \mathbf{D}_n} P(\mathbf{X}_n, \mathbf{Z}_n, \mathbf{D}_n|\theta) + \log P(\theta|\alpha) \tag{4}$$

We solve Eq. (4) using EM [7] based algorithm, which we call MAP-EM. The details are provided in supplementary materials. Second, we compute estimate of α using Eq. (3) given current estimate θ^*. Since the hyperparameters are independent of random variables given θ^*. Eq. (3) reduces to computing MLE of α as follows.

$$\alpha^* = \arg\max_{\alpha} \log P(\theta^*|\alpha) \tag{5}$$

Solving Eq. (5) can be done for each individual hyperparameter separately. The details are provided in supplementary materials.

Algorithm 1 Learning HDM

Input: $\mathbf{X}_n$: observation sequences
Output: Hyperparameters α
1: Initialization of α, θ
2: **repeat**
3: Update θ by solving Eq. (4)
4: Update α by solving Eq. (5)
5: **until** convergence
6: **return** α

The above alternating process will generate a sequence of estimations of θ, α that increase the value of $\log P(\{\mathbf{X}_n\}, \theta|\alpha)$. In experiment, it often converges in a few iterations. To initialize α, we use values that produce uniform initial, transition, duration distribution and mixture weights. We initialize ψ based on the mean and covariance of data. To initialize θ for MAP-EM, we use K-means to cluster data and use cluster assignment as hidden state value, from which we can estimate the model parameters. For evaluation of convergence, we use the change of $\log P(\{\mathbf{X}_n\}, \theta|\alpha)$ between two consecutive iterations. Algorithm 1 summarizes the overall learning process.

3.3. Inference

The goal of inference is to compute the posterior predictive likelihood of unseen data $\mathbf{X}$.

$$pl(\mathbf{X}|\alpha^*) \triangleq P(\mathbf{X}|\mathcal{D}, \alpha^*) \tag{6}$$
$$= \int_{\theta} \sum_{\mathbf{Z}, \mathbf{D}} P(\mathbf{X}, \mathbf{Z}, \mathbf{D}|\theta) P(\theta|\mathcal{D}, \alpha^*) d\theta$$

where $\mathcal{D} = \{\mathbf{X}_n\}$ is the set of training data. For the same reason discussed in Section 3.2, exact computation of Eq. (6) is intractable and approximate inference is needed. We use Monte Carlo estimation to approximate the integration by sampling θ from its posterior distribution.

$$pl(\mathbf{X}|\alpha^*) \approx \frac{1}{L} \sum_{l=1}^{L} \sum_{\mathbf{Z}, \mathbf{D}} P(\mathbf{X}, \mathbf{Z}, \mathbf{D}|\theta^{(l)}) \tag{7}$$

where $\theta^{(l)} \sim P(\theta|\mathcal{D}, \alpha^*)$ and L is the total number of samples. To generate samples of parameters from their posterior distributions, we consider two methods. The first one is structured mean-field variational inference [2], which finds an optimal variational distribution $q(\theta, \mathcal{H}|\phi) \triangleq q(\theta|\phi) q(\mathcal{H}|\phi)$ that maximizes a lower bound on $\log P(\mathcal{D}|\alpha^*)$. Here ϕ is the parameters of q and $\mathcal{H} = \{\mathbf{Z}_n, \mathbf{D}_n\}$ is the hidden states of all the training data $\mathcal{D}$. After we obtain optimal ϕ^*, parameter $\theta^{(l)}$ is then sampled from $q(\theta|\phi^*)$. The second one is blocked Gibbs sampling [19], which alternates the sampling between hidden state chain $\{\mathbf{Z}_n, \mathbf{D}_n\}$ and parameters θ. This process simulates a Markov chain whose stationary distribution converges to the true posterior distribution. Samples are collected after the burn-in period, which we determine by the change of log-likelihood of parameters. The inference algorithms are implemented using Pyhsmm [18] and BNT [32]. Given $\theta^{(l)}$, each term of summation in Eq. (7) can be computed using forward-recursion [57]. The same inference process is performed for each class model with hyperparameters learned in Section 3.2. The classification criterion is as follows.

$$y^* = \arg\max_{i} pl(\mathbf{X}|\alpha_i^*) \tag{8}$$

where the subscript i is the class index. The overall complexity is $O(KLQ^2T^2)$. In our experiments Q is usually between 10-20, whose value is determined by cross-validation. T is usually less than 200. K varies from 11 to 27. L is set to 100, which we found sufficient.

3.4. Uncertainty of classification

The use of Bayesian inference allows us to quantize the uncertainty of classification results. Specifically, we treat the class label y as a random variable that follows categorical distribution *i.e.* $y \sim Cat(\mathbf{p})$, where $\mathbf{p} = [p_1, ..., p_K]$ is a stochastic vector specifying the probability of the y being one of the K classes. For a sequence $\mathbf{X}$, we obtain $\mathbf{p}$ by normalizing the likelihood of different classes' model parameters evaluated on $\mathbf{X}$ *i.e.* $p_i^{(l)} = P(\mathbf{X}|\theta_i^{(l)}) / \sum_{j=1}^{K} P(\mathbf{X}|\theta_j^{(l)})$. To generate uncertainty measure, we first compute total covariance of y. Given the samples of parameters, the total covariance can be computed by Eq. (9). The proof is provided in supplementary materials.

$$V[y|\mathbf{X}] = E_\theta[V[y|\mathbf{X},\theta]] + V_\theta[E[y|\mathbf{X},\theta]] \quad (9)$$
$$\approx \frac{1}{L}\Sigma_{l=1}^{L} C_l + \frac{1}{L-1}\Sigma_{l=1}^{L}(\mathbf{p}_k - \bar{\mathbf{p}})(\mathbf{p}_k - \bar{\mathbf{p}})^T$$

where C_l is the covariance matrix of the categorical distribution corresponding to the l^{th} set of parameters. The entry of covariance can be computed by $C(i,j) = \delta(i,j)p_i - p_i p_j$. A similar decomposition of total variance is proposed in [22]. To obtain the uncertainty, we compute the trace of total covariance matrix *i.e.* $U(y) \triangleq \sum_i V[y|\mathbf{X}](i,i)$. The trace attains its minimum value 0 if and only if exactly one of the p_k equals to 1 and 0 otherwise. In such case, the prediction is absolutely certain. Our uncertainty measure indicates how confident the prediction is.

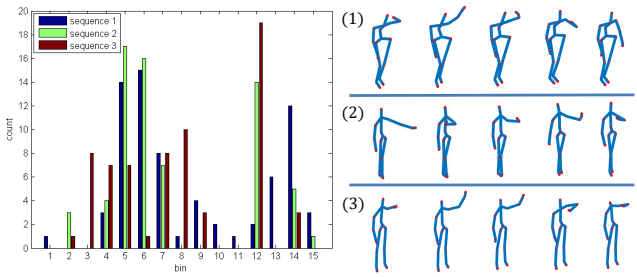

Figure 3. Left: Example histograms of right hand in high arm wave. Zero-count bin is pruned for compactness. Right: Actual waving action sequences from different datasets. (1) MSRA; (2) UTD; (3) G3D. (Best view in color)

4. Experiments

First, we perform a quantitative analysis of spatio-temporal variation on selected benchmark datasets. Second, we evaluate the performance of action recognition on individual dataset and compare with both baseline and state-of-the-art methods, followed by an uncertainty analysis. Third, we evaluate the generalization capability of our method across different datasets. Finally, we perform action recognition with missing observations[2].

4.1. Action datasets and feature extraction

Our experiments involve four benchmark action recognition datasets, where all datasets involve multiple subjects and action types ranging from hand movement to whole body movement. Specifically, **MSR Action3D (MSRA)** [26] includes 567 sequences from 20 types of action. **UTD-MHAD (UTD)** [6] includes 861 sequences from 27 types of actions. **Gaming 3D (G3D)** [3] consists of 600 sequences of 20 action types. **UPenn Action (Penn)** [58] contains 2326 RGB videos of 15 types of sports. We select a subset of 1650 videos from 11 actions, excluding 4 actions with large portion of missing body annotations due to occlusion. In all datasets, only skeleton is used for action recognition. The location and size of skeletons are normalized to ensure translation and scale invariance. Besides position, the motion is also extracted by computing the difference between consecutive frames for every pair of joints. Similar representation is adopted in [1, 3, 51]. The raw feature dimension is 266 per frame for 3D data and 117 per frame for 2D data. We further perform PCA for position and motion feature separately and retain 95% energy for each type of features at each frame. Finally, the two features are concatenated.

4.2. Spatial and temporal variation analysis

We first introduce a quantitative measure of intra-class variation based on a histogram representation of action sequence. We divide the 3D space into $5 \times 5 \times 5$ grids with equal volume. Then for each joint in each sequence, we construct a histogram whose number of bins is equal to the number of spatial grids. The bin value equals to the number of times when the joint position occupies the grid. We keep the bin value unnormalized so that it depends on both the spatial pose and the temporal pace. Figure 3 shows an example of obtained histogram for different sequences of the same action and the same joint. All three histograms show a bi-modal distribution. However, the specific bin counts are very different due to position and speed variation of the hand joint. After computing histograms, we compute the standard deviation of each bin value over all the sequences and sum over all the bins, yielding total variation. Finally, the total variation is averaged over all the joints as the final variation score. Such metric satisfies the following properties. First, if all the sequences are identical, the metric attains its minimum value 0. Second, the metric increases as the intra-class variation increases.

Figure 4 shows the measured variation scores for different actions in different datasets. In addition, we evaluate the variation score on combined dataset, where the same

[2]Code available at http://bit.ly/BayesianHDM

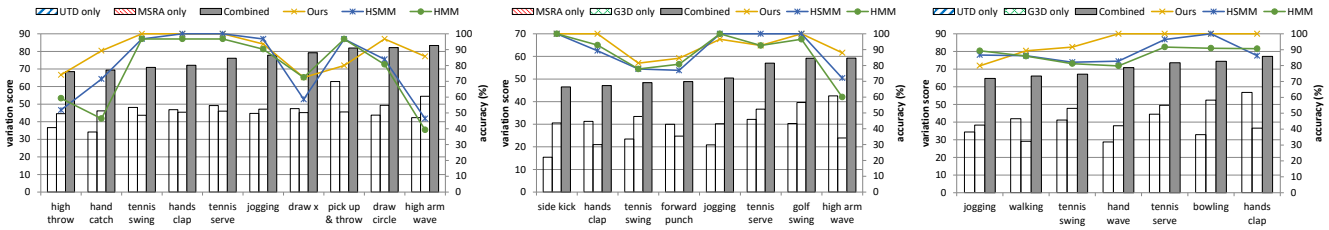

Figure 4. Variation score and the corresponding classification accuracy, which are obtained by training and testing on combined dataset. The details are referred to Section 4.5. Left: UTD and MSRA. Middle: MSRA and G3D. Right: G3D and UTD. (Best view in color)

pre-processing is applied on combined dataset for scale and translation invariance. From the figure we observe that action involves larger extent of whole body movement tends to have larger variations *e.g.* golf swing and bowling. Action with ambiguous explanation also has large variations *e.g.* high wave. For each action, the combined dataset has larger variation score than each individual dataset. We also draw class-wise classification accuracy obtained on the combined dataset. The classification details are discussed in Section 4.5. In general, our method performs better than the baseline methods especially on actions with larger intra-class variation score. This shows the benefit of explicit modeling of intra-class variations.

Table 1. Compare recognition accuracy (%) on different datasets with different baseline models.

Model	MSRA	UTD	G3D	Penn	Avg.
HMM	67.8	82.8	68.1	82.3	75.3
HSMM	66.3	82.3	77.5	78.9	76.3
LSTM	74.7	77.0	82.2	90.3	81.1
HCRF	70.7	74.2	79.0	86.3	77.6
HDM-PI	70.3	84.4	79.4	89.8	81.0
HDM-PL	80.6	90.2	87.7	91.6	87.5
HDM-BV	82.1	91.4	87.7	90.8	88.0
HDM-BG	**86.1**	**92.8**	**92.0**	**93.4**	**91.1**

4.3. Individual dataset experiments

For individual dataset experiments, training-testing split follows convention suggested by dataset authors. We conduct an ablation study by comparing our models with different simplified models. For our model, we consider four variants depending on how the inference is performed. The first two are based on point estimate of parameters. The MAP estimation of the parameters is obtained during learning and the predictive likelihood is simply computed as the likelihood of the MAP parameters. For PI, the initial values of hyperparameters are used. For PL, the learned hyperparameters are used. The last two variants use Bayesian inference, where the predictive likelihood is computed following Section 3.3 using either variational inference (BV) or Gibbs sampling (BG). Based on the results in Table 1, we have following observations. First, compared with non-hierarchical baseline HMM and HSMM, HDM achieves consistent improvement. Furthermore, HDM is superior to both HCRF and LSTM, which do not explicitly consider data variations. These results demonstrate the benefit of modeling spatial and temporal variations. Second, comparing the two point estimate approaches, using learned hyperparameters improves accuracy by 6.5%. This demonstrates the benefit of learning hyperparameters. Third, compared to point estimate, Bayesian inference improves performance by 0.5% (BV) and 3.3% (BG). This shows that by averaging out the model uncertainty in inference, we can improve the prediction. While variational inference is easier to determine the convergence of approximation, the quality of the approximation may not be optimal. Gibbs sampling on the other hand can converge to true posterior provided with enough sampling iterations and proper determination of mixing condition. In our experiment, we observe the log-likelihood of correct model obtained by Gibbs sampling is usually higher than variational inference, which is also consistent with its performance in classification. For the remaining experiments, we report the results of HDM-BG.

Table 2. Compare recognition accuracy (%) with state-of-the-art.

MSRA		UTD	
Method	Acc.	Method	Acc.
AS[38]	83.5	Fusion[6]	79.1
AL[48]	88.2	DMM[4]	84.2
MT[8]	**92.0**	CNN[51]	87.9
HDM	86.1	HDM	**92.8**
G3D		**Penn**	
Method	Acc.	Method	Acc.
LRBM[35]	90.5	Actemes[58]	86.5
R3DG[47]	91.1	AOG[36]	84.8
CNN[51]	**96.0**	JDD[5]	93.2
HDM	92.0	HDM	**93.4**

Then we compare the performance of our method with state-of-the-art methods. The average recognition accuracy is shown in Table 2. Compared to feature based methods, we achieve 4.9% improvement on UTD. For G3D, our model is better than both model based approach [35] and skeleton feature based approach [47]. Another approach [51] requires dataset-dependent encoding of features, while we use the same data processing for all datasets. In Penn dataset, we outperform methods based on pose features [58, 36] and we are slightly better than appearance feature

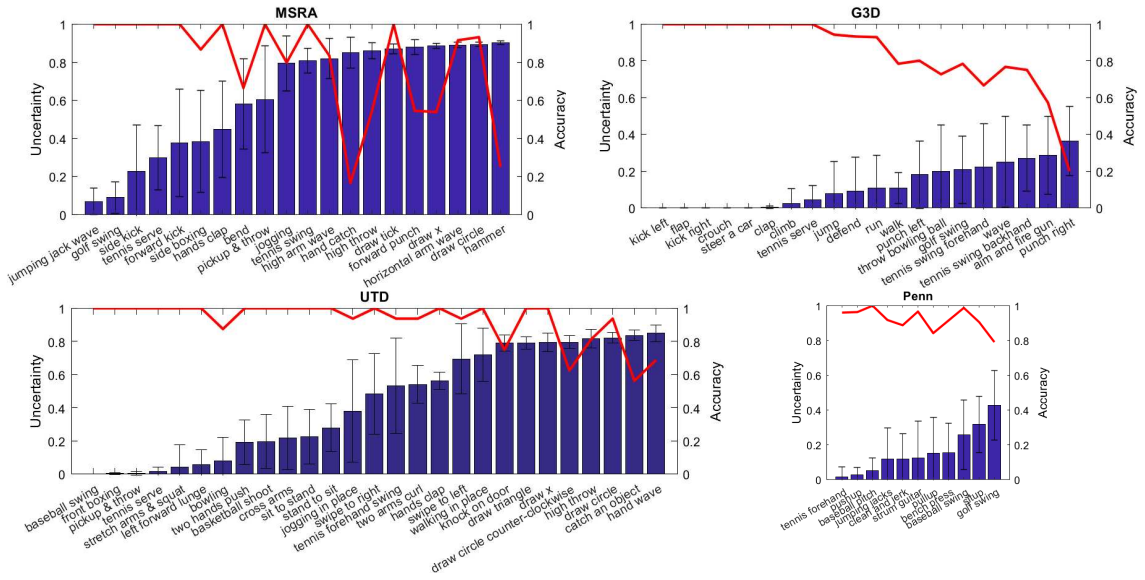

Figure 5. Class-wise uncertainty in different datasets, where standard deviation is indicated by the error bar. The curve corresponds to class-wise accuracy. The Pearson correlation coefficients between the two are MSRA:-0.5811, UTD:-0.5723, G3D:-0.8999, Penn:-0.6215.

based method [5], which used more information than ours. On MSRA, the performance gap between ours and [8] is mainly due to use of a sophisticated encoding of skeleton features, which we plan to explore as future work. We use the same kinematic features for all datasets without heavily engineering the features. Overall, these results demonstrate that by capturing intra-class variations, our model achieves competitive recognition performance on various datasets.

4.4. Uncertainty analysis

First, we verify the validity of the proposed uncertainty measure as defined in Section 3.4. We compute the error rate of different portions of data ranging from the most certain to the least certain. The curve in Figure 6 shows that the uncertainty correlates well with the error rate. For example, in MSRA, when we select the 30% of data with lowest uncertainty, the error rate is 0. When we expand the portion to 50%, the error rate increases to 8%. We also visualize data and corresponding class probability with different uncertainty values in Figure 6. For low uncertainty data we see the probability value is almost peak at the correct class. While for data with high uncertainty at the upper right corner, we see a diffused and low probability value.

Then we analyze the class-wise uncertainty by computing the mean and standard deviation of uncertainty within each class. Figure 5 plots the class-wise uncertainty and accuracy. We observe in general that the higher the uncertainty, the lower the accuracy. Actions only involving small extent of motion tend to have higher uncertainty. For instance, the top 5 uncertain actions in MSRA and UTD are all single-hand actions. Some actions have subtle difference such as 'high throw' and 'catch an object' in UTD. Some actions involve similar motion like 'hammer' and 'forward

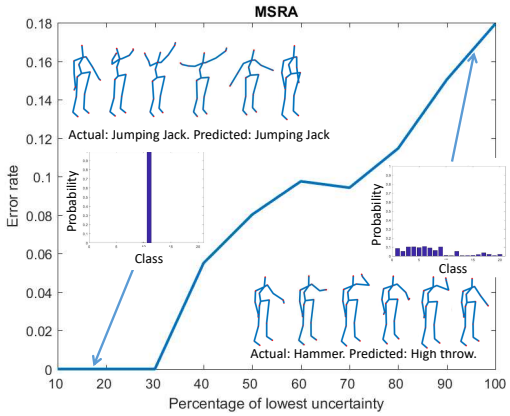

Figure 6. Classification error rate versus different portions of uncertainty values. (See Section 4.4 for details)

punch' in MSRA. More results are provided in supplementary materials. These results suggest that we should take uncertainty into consideration for classification decision. One future direction of this work is to incorporate the uncertainty during testing to automatically refine the model.

4.5. Multi-dataset experiments

To further demonstrate the capability of our model in generalizing across different subjects and trials. We perform two experiments involving multiple datasets including: A. MSRA; B. UTD; C. G3D. They share multiple action types in common.

In the first experiment, we train our model on combined dataset and test on each individual dataset with subjects that are not included in combined dataset. For the combined dataset, we expect significant intra-class variation. The results are shown in column 2-8 of Table 3. From the results, we observe that 1) HDM consistently outperforms

Table 3. Classification accuracy (%) on multi-dataset experiments. Results of other methods are obtained using original implementation. The number of shared actions for (A,B), (A,C), (B,C) and (A,B,C) is 10, 8, 7 and 5, respectively. The action names are shown in Figure 4.

Train	A,B		A,C		B,C		Avg.	B,C	A,C	A,B	Avg.
Test	A	B	A	C	B	C		A	B	C	
HSMM	73.7	82.5	89.0	87.0	91.0	83.2	84.4	65.3	61.9	42.5	56.5
DMM[4]	76.6	90.6	91.7	84.3	92.8	76.4	85.4	76.2	**86.3**	51.1	71.2
R3DG[47]	82.5	91.9	**93.6**	90.0	**97.3**	82.9	89.7	44.9	84.4	**72.7**	67.3
DLSTM[59]	83.9	**93.1**	88.1	87.0	82.9	80.9	86.0	70.8	85.0	38.9	64.9
HDM	**86.9**	91.9	**93.6**	**92.6**	**97.3**	**91.0**	**92.2**	**89.2**	75.0	61.2	**75.1**

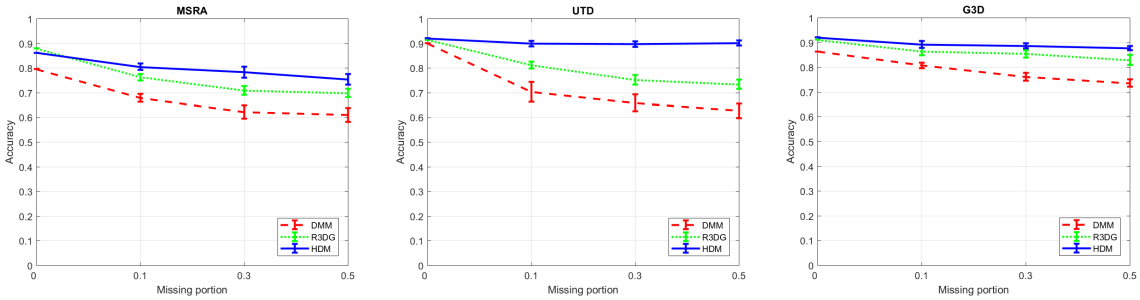

Figure 7. Average accuracy and standard deviation (error bar) under different portions of missing data.

HSMM, which lacks the improved capacity due to hierarchical structure. 2) HDM outperforms or achieves similar results as other three methods [4, 47, 59] in five out of six cases. On average, it outperforms all other methods with 2.5% improvement compared to the second best method, which uses sophisticated scheme to extract features. This demonstrates our model's generalization capability across datasets through capturing large intra-class variation.

In the second experiment, we train different models on two datasets and test on the remaining one dataset. This is a more challenging scenario, since the data collection settings in training and testing are very different. The results are reported in Table 3 column 9-12. HDM significantly outperforms non-hierarchical baseline HSMM with average improvement of 18.6%. HDM also outperforms [4], [47] and [59] by 3.9%, 7.8% and 10.2%. Although the absolute performance drops for all the methods, the relative improvement of our method compared to others becomes more significant than pairwise case. These results further demonstrate that HDM has enough capacity to absorb large variation. Thus, it can generalize better across different datasets.

4.6. Classification with missing data

One of the benefits using generative model is to handle missing data. In skeleton based action recognition, it is possible to have missing values in the observations, which are often caused by failure of tracking or occlusion. To demonstrate the robustness of the proposed approach in handling missing values, we conduct an experiment where the model is trained and tested on skeleton data with randomly missing values. To handle input with missing values, we compute likelihood $P(X_t|Z_t)$ using only observed part of X_t. For fair comparison, the same data with missing values are used by other methods. We repeat the classification 10 times and the results are shown in Figure 7. Our method achieves the smallest decrease in performance as missing portion increases. This shows that the combination of generative model with Bayesian inference maintains the robustness against missing values in data.

5. Conclusion

In this paper, we proposed a probabilistic hierarchical dynamic model to handle intra-class spatio-temporal variations for human action recognition. By treating model parameters as random variables with designated prior distributions, the model can better adapt to intra-class variations. An algorithm of learning hyperparameters is developed. The use of Bayesian inference not only improves the generalization of the model but also allows us to provide an uncertainty measure of the prediction, which provides a reference on the decision. Experiments conducted within individual and across multiple datasets show that the proposed HDM not only can capture the underlying dynamics of different actions but also possess enough capacity to allow large intra-class variations. Experiment with missing values also shows the robustness of the proposed method.

Acknowledgment

This work is partially supported by Cognitive Immersive Systems Laboratory (CISL), a collaboration between IBM and RPI, and also a center in IBM's AI Horizon Network. Xu is also supported by NSFC61672089 and partially supported by a CSC scholarship.

References

[1] Jake K Aggarwal and Lu Xia. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 2014.

[2] Matthew J Beal. *Variational algorithms for approximate Bayesian inference*. University of London, 2003.

[3] Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou. G3d: A gaming action dataset and real time action recognition evaluation framework. In *CVPR Workshop*, 2012.

[4] Mohammad Farhad Bulbul, Yunsheng Jiang, and Jinwen Ma. Dmms-based multiple features fusion for human action recognition. *International Journal of Multimedia Data Engineering and Management*, 2015.

[5] Congqi Cao, Yifan Zhang, Chunjie Zhang, and Hanqing Lu. Action recognition with joints-pooled 3d deep convolutional descriptors. In *IJCAI*, 2016.

[6] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *ICIP*, 2015.

[7] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1977.

[8] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *Cybernetics*, 2015.

[9] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[10] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.

[11] Thi Duong, Dinh Phung, Hung Bui, and Svetha Venkatesh. Efficient duration and hierarchical modeling for human activity recognition. *Artificial Intelligence*, 2009.

[12] Ahmed Elgammal, Vinay Shet, Yaser Yacoob, and Larry S Davis. Learning dynamics for exemplar-based gesture recognition. In *CVPR*, 2003.

[13] Haoshu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *AAAI*, 2018.

[14] Jack D Ferguson. Variable duration models for speech. In *Symposium on the Application of HMMs to Text and Speech*, 1980.

[15] Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda. A bayesian approach to hidden semi-markov model based speech synthesis. In *INTERSPEECH*, 2009.

[16] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *CVPR*, 2017.

[17] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 2013.

[18] Matthew J Johnson. *Bayesian time series models and scalable inference*. PhD thesis, MIT, 2014.

[19] Matthew J Johnson and Alan S Willsky. Bayesian nonparametric hidden semi-markov models. *JMLR*, 2013.

[20] Ajjen Joshi, Soumya Ghosh, Margrit Betke, Stan Sclaroff, and Hanspeter Pfister. Personalizing gesture recognition using hierarchical bayesian neural networks. In *CVPR*, 2017.

[21] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *CVPR*, 2017.

[22] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017.

[23] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[24] Zhengzhong Lan, Ming Lin, Xuanchong Li, Alex G Hauptmann, and Bhiksha Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *CVPR*, 2015.

[25] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*. IEEE, 2011.

[26] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *CVPR Workshop*, 2010.

[27] Ivan Lillo, Alvaro Soto, and Juan Niebles. Discriminative hierarchical modeling of spatio-temporally composable human activities. In *CVPR*, 2014.

[28] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *CVPR*, 2018.

[29] Behrooz Mahasseni and Sinisa Todorovic. Regularizing long short term memory with 3d human-skeleton sequences for action recognition. In *CVPR*, June 2016.

[30] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *TOG*, 2017.

[31] Meinard Müller and Tido Röder. Motion templates for automatic classification and retrieval of motion capture data. In *SIGGRAPH*, 2006.

[32] Kevin Murphy. The bayes net toolbox for matlab. *Computing science and statistics*, 2001.

[33] Pradeep Natarajan and Ramakant Nevatia. Coupled hidden semi markov models for activity recognition. In *WMVC*, 2007.

[34] Pradeep Natarajan and Ramakant Nevatia. Online, real-time tracking and recognition of human actions. In *WMVC*, 2008.

[35] Siqi Nie, Ziheng Wang, and Qiang Ji. A generative restricted boltzmann machine based method for high-dimensional motion data modeling. *CVIU*, 2015.

[36] Xiaohan Nie, Caiming Xiong, and Song-Chun Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[37] Sang Min Oh, James M Rehg, Tucker Balch, and Frank Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *IJCV*, 2008.

[38] Eshed Ohn-Bar and Mohan Trivedi. Joint angles similarities and hog2 for action recognition. In *CVPRW*, 2013.

[39] Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Morency Collins, and Trevor Darrell. Hidden conditional random fields. *PAMI*, 2007.

[40] Hossein Rahmani and Mohammed Bennamoun. Learning action recognition model from depth and skeleton videos. In *ICCV*, 2017.

[41] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 2013.

[42] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. *ECCV*, 2018.

[43] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, 2017.

[44] Yale Song, Louis-Philippe Morency, and Randall Davis. Action recognition by hierarchical sequence summarization. In *CVPR*, 2013.

[45] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *CVPR*, 2018.

[46] Ashok Veeraraghavan, Anuj Srivastava, Amit K Roy-Chowdhury, and Rama Chellappa. Rate-invariant recognition of humans and their activities. *TIP*, 2009.

[47] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, 2014.

[48] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.

[49] Jiang Wang and Ying Wu. Learning maximum margin temporal warping for action recognition. In *ICCV*, 2013.

[50] Limin Wang, Yu Qiao, and Xiaoou Tang. Latent hierarchical model of temporal structure for complex activity classification. *TIP*, 2014.

[51] Pichao Wang, Wanqing Li, Chuankun Li, and Yonghong Hou. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Systems*, 2018.

[52] Yang Wang and Greg Mori. Learning a discriminative hidden part model for human action recognition. In *NIPS*, 2009.

[53] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *CVIU*, 2011.

[54] Andrew D Wilson and Aaron F Bobick. Parametric hidden markov models for gesture recognition. *TPAMI*, 1999.

[55] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv*, 2018.

[56] Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 2010.

[57] Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *Signal processing letters*, 2003.

[58] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013.

[59] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, Xiaohui Xie, et al. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, 2016.