The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

An Adversarial Hierarchical Hidden Markov Model for Human Pose Modeling and Generation

Rui Zhao, Qiang Ji
Rensselaer Polytechnic Institute
Troy NY, USA
{zhaor,jiq}@rpi.edu

Abstract

We propose a hierarchical extension to hidden Markov model (HMM) under the Bayesian framework to overcome its limited model capacity. The model parameters are treated as random variables whose distributions are governed by hyperparameters. Therefore the variation in data can be modeled at both instance level and distribution level. We derive a novel learning method for estimating the parameters and hyperparameters of our model based on adversarial learning framework, which has shown promising results in generating photorealistic images and videos. We demonstrate the benefit of the proposed method on human motion capture data through comparison with both state-of-the-art methods and the same model that is learned by maximizing likelihood. The first experiment on reconstruction shows the model's capability of generalizing to novel testing data. The second experiment on synthesis shows the model's capability of generating realistic and diverse data.

Introduction

In recent years, generative dynamic model has attracted a lot of attention due to its potential of learning representation from unlabeled sequential data as well as its capability of data generation. (Gan et al. 2015; Srivastava, Mansimov, and Salakhudinov 2015; Mittelman et al. 2014; Xue et al. 2016; Walker et al. 2016). Sequential data introduce additional challenge for modeling due to temporal dependencies and significant intra-class variation. Consider human action as an example. Even though the underlying dynamic pattern remains similar for the same type of action, the actual pose and speed vary for different people. Even if the same person performs the action repeatedly, there will be noticeable difference. This motivates us to design a probabilistic dynamic model that not only can capture the consistent dynamic pattern across different data instances, but also can accommodate the variation therein.

Widely used dynamic model like HMM models dynamic process through transition among different discrete states. In order to encode N bits of information, HMM needs 2^N number of states. Therefore, the model complexity increases exponentially with the model capacity. Linear dynamic system (LDS) uses continuous states to capture dynamics, which avoids exponential increase of model complexity. However, LDS assumes the underlying dynamics can be described by a linear model, which may not be sufficient for case like human motion data. On the other hand, more complex model such as recurrent neural networks (RNN) based deep models often has exceedingly large number of parameters. Without sufficiently large amount of data or careful regularization, training such model is prone to overfit. In addition, the model is deterministic. Simply reducing the model complexity compromises the capability of capturing randomness and variation presented in data. We instead propose a hierarchical HMM (HHMM), which extends the shallow HMM leveraging on Bayesian framework. The proposed HHMM allows model parameters vary as random variables among data instances. Given the same amount of parameters, HHMM has a much larger capacity compared to HMM. Besides, HHMM retains the inference method available in HMM, allowing us to do various inference tasks efficiently. Finally, as a probabilistic generative model, HHMM can capture spatio-temporal dependencies in dynamic process and modeling variations in a principled way.

As for model learning, maximum likelihood estimate (MLE) has been the de facto learning objective for probabilistic generative models. Despite its wide adoption, MLE tends to fit a diffused distribution on data (Theis, Oord, and Bethge 2015). For static image synthesis, the results often look blurred. Recently, adversarial learning has emerged to be a popular learning criteria for learning generative models. Variants of generative adversarial networks (GAN) (Goodfellow et al. 2014; Radford, Metz, and Chintala 2015; Reed et al. 2016; Nowozin, Cseke, and Tomioka 2016; Nguyen et al. 2017) show promising results in generating both sharp and realistic-looking images of face, object and indoor/outdoor scene. There is also an increasing interest in extending the framework for dynamic data (Vondrick, Pirsiavash, and Torralba 2016; Saito, Matsumoto, and Saito 2017; Tulyakov et al. 2017). In this work, we explore the idea of training the proposed HHMM using adversarial objective, which has two major benefits. First, it bypasses the intractable objective of MLE in hierarchical model where the integration over parameters introduces additional dependencies among random variables. Second, it aims at learning a model that can generate realistic-looking data. Following the adversarial learning framework, we introduce a separate discriminative dynamic model to guide the learning of HHMM, which serves

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

as the generator. While the generator tries to generate data that looks as realistic as possible. The discriminator tries to classify the generated data as fake. The two models compete against each other in order to reach an equilibrium. We derive a gradient ascent based optimization method for updating parameters of both models. To the best of our knowledge, this is the first work that exploit adversarial learning on modeling dynamic data with fully probabilistic generator and discriminator.

Related work

Probabilistic dynamic models HMM and its variants (Rabiner 1989; Fine, Singer, and Tishby 1998; Brand, Oliver, and Pentland 1997; Ghahramani, Jordan, and Smyth 1997; Yu 2010) are widely used to model sequential data, where dynamics change according to transition among different discrete states. The observations are then emitted from a state-dependent distribution. The state can also be continuous as modeled in LDS, which is also known as Kalman filters (Kalman and others 1960). In a more general formulation, both HMM and LDS can be considered as special variants of dynamic Bayesian networks (DBN) (Murphy 2002). Our model expands the model capacity through the hierarchical structure instead of increasing complexity, which is ineffective for HMM. With enhanced model capacity, our model can better accommodate variation and nonlinearity of the dynamics. Another major type of dynamic model consists of undirected graphical models such as temporal extension of restricted Boltzmann machine (RBM) (Taylor, Hinton, and Roweis 2006; Sutskever and Hinton 2007; Mittelman et al. 2014) and dynamic conditional random field (DCRF) (Sutton, McCallum, and Rohanimanesh 2007; Tang, Fei-Fei, and Koller 2012). While RBM can capture non-linearity and expand capacity through vectorized hidden states, the learning requires approximation to intractable partition function and the choice of hidden state dimension may not be trivial. DCRF model is trained discriminatively given class labels and not suitable for data generation task.

More recently, models that combine probabilistic framework with deterministic model such as neural networks (NN) have been proposed. (Krishnan, Shalit, and Sontag 2015) proposed deep Kalman filters which used NN to parameterize transition and emission probability. (Johnson et al. 2016) used variational autoencoder to specify the emission distribution of switching LDS. (Gan et al. 2015) proposed deep temporal sigmoid belief network (TSBN), where the hidden node is binary and its conditional distribution is specified by sigmoid function. Variants of RNNs with additional stochastic nodes are introduced to improve the capability of modeling randomness (Bayer and Osendorfer 2014; Chung et al. 2015; Fraccaro et al. 2016). To better account for intra-class variation, (Wang, Fleet, and Hertzmann 2008) modeled dynamics using Gaussian process where the uncertainty is handled by marginalizing out parameter space imposed with Gaussian process prior. (Joshi et al. 2017) proposed a Bayesian NN which can adapt to subject dependent variation for action recognition. Deep learning based models typically require large amount of training data. For smaller dataset, careful regularization or other auxiliary techniques such as data augmentation, pre-train, drop-out, *etc.* are needed. In contrast, our HHMM has built-in regularization through the hyperparameters learned using all the intra-class data. It is less prone to overfitting. Besides, HHMM can handle missing data as the probabilistic inference can be carried out in absence of some observations. Furthermore, HHMM is easier to interpret as the nodes are associated with semantic meanings.

Learning methods of dynamic models Maximum likelihood learning is widely used to obtain point estimate of model parameters. For models with tractable likelihood function, numerical optimization techniques such as gradient ascent can be used to maximize likelihood function directly with respect to the parameters. In general, for model with hidden variables, whose values are always unknown during training, expectation maximization (EM) (Dempster, Laird, and Rubin 1977) is often used, which optimizes a tight lower bound of the model loglikelihood. Bayesian parameter estimation can also be used as an alternative to MLE in case when prior information on parameters need to be incorporated, resulting in maximum a posteriori (MAP) estimate. For instance, (Brand and Hertzmann 2000) introduced a prior on HMM parameters to encourage smaller cross entropy between specific stylistic motion model and generic motion model. In case the goal is to classify data into different categories, generative dynamic model can also be learned with discriminative criteria such as maximizing the conditional likelihood of being as one of the categories (Wang and Ji 2012). Our work provides another objective to learn generative dynamic models by adopting the adversarial learning framework. The generative model has to compete against another discriminative model in order to fit the data distribution well. An important difference of our method from existing adversarially learned dynamic models like TGAN (Saito, Matsumoto, and Saito 2017) is that, both our generator and discriminator are fully probabilistic models which explicitly model the variation of data distribution.

Methods

We first present the proposed dynamic model. Then we briefly review the adversarial learning framework and describe in details about the learning algorithm. Finally, we discuss the inference methods used for various tasks.

Bayesian Hierarchical HMM

We now describe the proposed HHMM, which models the dynamics and variation of data in two levels. First, the random variables capture spatial distribution and temporal evolution of dynamic data. Second, the parameters specifying the model are also treated as random variables with prior distributions. Notice that the term HHMM is first used in (Fine, Singer, and Tishby 1998), where the hierarchy is applied only on hidden or observed nodes with fixed parameters in order to model multi-scale structure in data. Our model constructs the hierarchy using Bayesian framework in order to handle large variation in data. Specifically, we define $\mathbf{X} = \{X_1, ..., X_T\}$ as the sequence of observations and $\mathbf{Z} = \{Z_1, ..., Z_T\}$ as the hidden state chain. The joint distribution of HHMM with

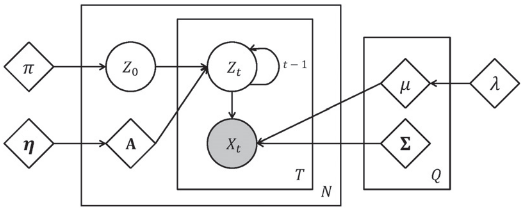


Figure 1: Topology of HHMM, where plate notation is used. T is the length of sequence. N is the number of sequences. Q is the number of hidden states. The self-edge of Z_t shows the temporal link from Z_{t-1} to Z_t. Circle-shaped nodes represent variables and diamond-shaped nodes represent parameters or hyperparameters.

first-order Markov assumption is given by

$$P(\mathbf{X}, \mathbf{Z}, \theta|\alpha) = P(Z_1|\pi) \prod_{t=2}^{T} P(Z_t|Z_{t-1}, \mathbf{A}) \tag{1}$$
$$\prod_{t=1}^{T} P(X_t|Z_t, \mu, \Sigma) P(\mathbf{A}|\eta) P(\mu|\lambda)$$

where π is a stochastic vector specifying the initial state distribution *i.e.* $P(Z_1 = i) = \pi_i$. $\mathbf{A}$ is a stochastic matrix where the ith row specifies the probability of transiting from state i to other states *i.e.* $P(Z_t = j|Z_{t-1} = i) = A_{ij}$. μ and Σ are the emission distribution parameters. We use Gaussian distribution as the observations are continuous *i.e.* $P(X_t|Z_t = i) = \mathcal{N}(\mu_i, \Sigma_i)$. Diagonal covariance matrix is also assumed. $\theta = \{\mathbf{A}, \mu\}$ and $\hat{\theta} = \{\pi, \Sigma\}$ are the model parameters and $\hat{\alpha} = \{\eta, \lambda\}$ are the model hyperparameters. We denote $\alpha = \{\hat{\alpha}, \hat{\theta}\} = \{\eta, \lambda, \pi, \Sigma\}$ as the augmented set of hyperparameters by including $\hat{\theta}$. The model topology is shown in Figure 1.

We use conjugate prior for θ. Specifically, we use a Dirichlet prior on the transition parameter $\mathbf{A}$ with hyperparameter η and a Normal prior on the emission mean μ with hyperparameter $\{\mu_0, \Sigma_0\}$.

$$P(\mathbf{A}_{i:}|\eta_i) \propto \prod_{j=1}^{Q} A_{ij}^{\eta_{ij}-1}$$
$$P(\mu_i|\lambda) \propto \exp\big(-\frac{1}{2}(\mu_i - \mu_{i0})^T \Sigma_{i0}^{-1} (\mu_i - \mu_{i0})\big)$$

where $i = 1, ..., Q, \eta_{ij} > 0, \mu_{i0} \in \mathbb{R}^O, \Sigma_{i0} \in \mathbb{R}^{O\times O}$. Q is the number of hidden states and O is the dimension of data. The benefit of using hierarchical model can be seen from its structure. Under the same model complexity *i.e.* same number of hidden states and dimension of data, parameters in HHMM can further vary according to each data instance. Thus HHMM has increased modeling capacity compared to HMM, which is crucial for modeling data variation.

Adversarial learning of HHMM

The adversarial learning approach utilizes a novel objective to train a generative model G by introducing another discriminative model D. Intuitively, G is aimed at generating samples that resemble real data distribution. D tries to differentiate whether a sample is from real data or generated by G. When both G and D are parameterized by neural networks, it yields the GAN (Goodfellow et al. 2014). Leveraging on the adversarial learning framework, we develop a method for learning HHMM, which we use as the generator. The choice of discriminator is a pair of HMMs that are trained with discriminative objective. We describe the overall optimization formulation first, followed by detailed discussion on generator and discriminator learning. We introduce an additional binary variable y associated with $\mathbf{X}$ to indicate whether $\mathbf{X}$ is real $(y = 1)$ or fake $(y = -1)$. The overall optimization objective is defined by Eq. (2).

$$\min_{\alpha} \max_{\phi} \mathbb{E}_{\mathbf{X}\sim P_{data}(\mathbf{X})}[\log D(\mathbf{X}|\phi)] \tag{2}$$
$$+\mathbb{E}_{\mathbf{X}\sim P_G(\mathbf{X}|\alpha)}[\log(1 - D(\mathbf{X}|\phi))]$$

where $D(\mathbf{X}|\phi) \triangleq P_D(y = 1|\mathbf{X}, \phi)$ is the output of discriminator specifying the probability of $\mathbf{X}$ being real data and ϕ is the parameters of discriminator. $P_{data}(\mathbf{X})$ is the real data distribution. $P_G(\mathbf{X}|\alpha)$ is the likelihood of α on $\mathbf{X}$ generated from G. Compared to GAN, the use of probabilistic generative model directly specify distribution $\mathbf{X}$, where the randomness and dependency is encoded through the latent variables. The goal of learning is to estimate α and ϕ. The optimization uses alternating strategy where we optimize one model while holding the other as fixed at each iteration.

Generator We now discuss in details about generator learning, which is HHMM in our case. The benefit of using a probabilistic dynamic model is that we can model data variation and randomness in a principled way. In addition, we can generate different length of sequences. Finally, we can evaluate data likelihood using learned model as described later in inference. When optimizing α in Eq. (2), we hold ϕ fixed. The same approximate objective as in (Goodfellow et al. 2014) is also used. This results in the following objective.

$$\max_{\alpha} L_G(\alpha) \triangleq \mathbb{E}_{\mathbf{X}\sim P_G(\mathbf{X}|\alpha)}[\log D(\mathbf{X}|\phi)] \tag{3}$$
$$\approx \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{1}{MN} \log D(\mathbf{X}_{ij}|\phi),$$
$$\theta_i \sim P(\theta|\hat{\alpha}), \mathbf{X}_{ij} \sim P(\mathbf{X}|\theta_i, \hat{\theta})$$

However, the sample-based approximation no longer explicitly depends on α. We use the identity that $\nabla_X f(X) = f(X)\nabla_X \log f(X)$ to derive an unbiased estimate of gradient of $L_G(\alpha)$ by directly taking derivative of Eq. (3), where similar strategy is also used in (Williams 1992).

$$\frac{\partial L_G(\alpha)}{\partial \hat{\alpha}} \approx \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\log D(\mathbf{X}_{ij}|\phi)}{MN} \frac{\partial \log P(\theta_i|\hat{\alpha})}{\partial \hat{\alpha}} \tag{4}$$

$$\frac{\partial L_G(\alpha)}{\partial \hat{\theta}} \approx \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\log D(\mathbf{X}_{ij}|\phi)}{MN} \frac{\partial \log P(\mathbf{X}_{ij}|\theta_i, \hat{\theta})}{\partial \hat{\theta}} \tag{5}$$

where $\theta_i \sim P(\theta|\hat{\alpha}), \mathbf{X}_{ij} \sim P(\mathbf{X}|\theta_i, \hat{\theta})$. In Eq. (4), the partial derivative is taken by the prior distribution of parameters, which has an analytical form given our parameterization. In Eq. (5), the partial derivative corresponds to the gradient of loglikelihood of HMM, which can be computed by exploiting the chain structure of hidden states as described in (Cappé, Buchoux, and Moulines 1998). SGD with RMSProp (Tieleman and Hinton 2012) for adaptive gradient magnitude is performed to update α. We also reparameterize $\kappa = \log \sigma$, where σ^2 is the diagonal entries of Σ, which is assumed to be diagonal. Intuitively speaking, given a fixed D, samples with $D(\mathbf{X}_{ij}|\phi) \to 0$ will be weighted heavily to encourage improvement. Samples with $D(\mathbf{X}_{ij}|\phi) \to 1$ have $\frac{\partial L_G(\alpha)}{\partial \alpha} \to 0$, thus contribute little to the update.

Discriminator Our discriminator consists of a pair of HMMs with parameters specified as ϕ^+ and ϕ^- respectively. The use of dynamic model based discriminator is largely motivated by the needs to work with sequential data. To differentiate whether a motion sequence looks realistic or not, the discriminator should be able to recognize the underlying motion pattern subject to variation. In addition, dynamic discriminator also can accommodate sequences with different length. Specifically, the output of discriminator is defined as follows.

$$D(\mathbf{X}|\phi) = \frac{P(\mathbf{X}|\phi^+)}{P(\mathbf{X}|\phi^+) + P(\mathbf{X}|\phi^-)\frac{P(y=-1)}{P(y=1)}} \tag{6}$$

where $P(y)$ is the prior probability of the labels. Since we choose the same number of real and fake samples at each update, we can assume uniform distribution of labels, namely $P(y = 1) = P(y = -1) = 1/2$. $P(\mathbf{X}|\phi^+)$ and $P(\mathbf{X}|\phi^-)$ are the likelihoods of ϕ^+ and ϕ^- evaluated on $\mathbf{X}$ respectively. The two HMMs are trained discriminatively under the objective of Eq. (2) with α holding fixed. Specifically, given a set of M randomly generated samples $\{\mathbf{X}_j^-\}$ from generator and a set of M randomly selected real data samples $\{\mathbf{X}_j^+\}$, the objective of learning ϕ is equivalent to the negative cross-entropy loss as follows.

$$\begin{aligned}\max_{\phi} L_D(\phi) &\triangleq \mathbb{E}_{\mathbf{X}\sim P_{data}(\mathbf{X})}[\log D(\mathbf{X}|\phi)] \\ &+ \mathbb{E}_{\mathbf{X}\sim P_G(\mathbf{X}|\alpha)}[\log(1 - D(\mathbf{X}|\phi))] \\ &\approx \frac{1}{M}\sum_{j=1}^{M} \log D(\mathbf{X}_j^+|\phi) + \log(1 - D(\mathbf{X}_j^-|\phi))\end{aligned} \tag{7}$$

By substituting Eq. (6) to Eq. (7), we can compute the gradient of $L_D(\phi)$ with respect to ϕ.

$$\begin{aligned}\frac{\partial L_D(\phi)}{\partial \phi^+} \approx \frac{1}{M}\sum_{j=1}^{M}[&\frac{P(\mathbf{X}_j^+|\phi^-)}{s(\mathbf{X}_j^+)}\frac{\partial \log P(\mathbf{X}_j^+|\phi^+)}{\partial \phi^+} \\ &- \frac{P(\mathbf{X}_j^-|\phi^+)}{s(\mathbf{X}_j^-)}\frac{\partial \log P(\mathbf{X}_j^-|\phi^+)}{\partial \phi^+}]\end{aligned} \tag{8}$$

$$\begin{aligned}\frac{\partial L_D(\phi)}{\partial \phi^-} \approx \frac{1}{M}\sum_{j=1}^{M}[&\frac{P(\mathbf{X}_j^-|\phi^+)}{s(\mathbf{X}_j^-)}\frac{\partial \log P(\mathbf{X}_j^-|\phi^-)}{\partial \phi^-} \\ &- \frac{P(\mathbf{X}_j^+|\phi^-)}{s(\mathbf{X}_j^+)}\frac{\partial \log P(\mathbf{X}_j^+|\phi^-)}{\partial \phi^-}]\end{aligned} \tag{9}$$

where $s(\mathbf{X}) = P(\mathbf{X}|\phi^+) + P(\mathbf{X}|\phi^-)$. Again, $\frac{\partial \log P(\mathbf{X}|\phi^+)}{\partial \phi^+}$ and $\frac{\partial \log P(\mathbf{X}|\phi^-)}{\partial \phi^-}$ are gradients of loglikelihood of the two HMMs respectively, where analytical form is available as described in generator update. The overall algorithm is summarized as Algorithm 1.

Algorithm 1 Adversarial learning of HHMM

```
Require: {$\mathbf{X}$}: real dataset. $Q$: number of hidden states. $M$: number of samples. $N$: number of parameter sets. $k$: update step for $\phi$. $l$: update step for $\alpha$.
Ensure: Generator $\alpha$. Discriminator $\phi$.
Initialization of $\alpha, \phi$
repeat
    for $k$ steps do
        Draw $M$ samples from both $P_G$ and real dataset.
        Update discriminator $\phi$ using RMSProp with gradient defined by Eq. (8) and Eq. (9).
    end for
    for $l$ steps do
        Draw $N$ samples of $\theta$. For each $\theta$, draw $M$ samples.
        Update generator $\alpha$ using RMSProp with gradient defined by Eq. (4) and Eq. (5).
    end for
until convergence or reach maximum iteration number
return $\alpha$
```

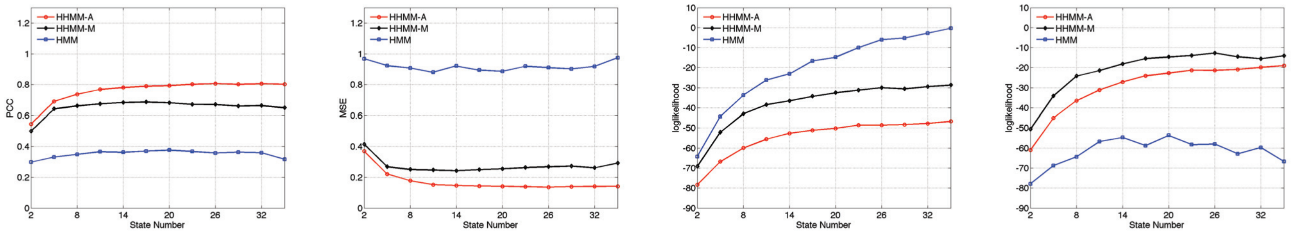
Inference

We describe our methods on three inference problems associated with HHMM when applied to different data analysis applications as described later in experiments.

Data synthesis One of the major applications for generative model is to automatically synthesize data. The potential use of synthetic motion data is to supply the training of deep learning models for tasks like action recognition. We use ancestral sampling based approach to generate synthetic motion data. Specifically, we first sample parameter $\mathbf{A}, \mu$ from their corresponding prior distribution given learned hyperparameters *i.e.* $\mathbf{A} \sim P(\mathbf{A}|\eta), \mu \sim P(\mu|\mu_0, \Sigma_0)$. Second, we sample hidden state chain given sampled parameters $\mathbf{A}$ and learned parameters π *i.e.* $Z_1 \sim P(Z_1|\pi), Z_t \sim P(Z_t|Z_{t-1}, \mathbf{A})$. Finally, we compute the most likely observation sequence $X_1, ..., X_T$ conditioned on $Z_1, ..., Z_T$ and parameters μ, Σ. Due to the model structure, observed nodes are independent with each other given the hidden states. A naive solution that maximizes the conditional likelihood $P(\mathbf{X}|\mathbf{Z})$ yields mean value of the corresponding Gaussian at each frame *i.e.* $X_t = \mu_{Z_t}$. For motion capture data, this results non-smooth change between different poses. We alleviate this issue by augmenting features X_t to include information with both first order *i.e.* position and second order *i.e.* speed as suggested in (Brand 1999), where the speed is computed as the difference of consecutive position change. Then we solve the following inference problem.

$$\max_{\mathbf{X}} \log P(\tilde{\mathbf{X}}|\mathbf{Z}) = \sum_t \log N(\tilde{X}_t|\mu_{Z_t}, \Sigma_{Z_t}) \tag{10}$$

where $\mathbf{X} = \{X_t\}, \tilde{\mathbf{X}} = \{\tilde{X}_t\}, \tilde{X}_t = [X_t, X_t - X_{t-1}]$. Eq. (10) is a quadratic system with respect to $\mathbf{X}$, where



(a) PCC (b) MSE (c) Loglikelihood on training data (d) Loglikelihood on testing data

Figure 2: Reconstruction experiment results on Berkeley's jumping action versus change of number of hidden states. HHMM-A refers to adversarial learning variant and HHMM-M refers to maximum likelihood learning variant. (Best view in color)

closed-form solution can be obtained by setting the derivative to zero.

Reconstruction The goal of reconstruction is to generate a novel sequence that resembles the input sequence. The reconstruction process evaluates the capability of model capturing the dynamics of sequential data. Since the hidden state chain is the primary source that encodes the dynamic change of the data, we first infer the most probable configuration of state chain. We compute the MAP estimate θ^* by solving the following problem using MAP-EM (Gauvain and Lee 1994), where the E-step has complexity $O(Q^2T)$ and M-step has closed-form solution.

$$\theta^* = \arg\max_{\theta} \log \sum_{\mathbf{Z}} P(\mathbf{X}, \mathbf{Z}|\theta, \hat{\theta}) + \log P(\theta|\hat{\alpha}) \quad (11)$$

Then we perform Viterbi decoding algorithm (Rabiner 1989) on observed testing sequence given θ^*. Finally, we compute the most likely observations given the decoded states in the same way as described in data synthesis.

Compute data likelihood The marginal likelihood of the model evaluated on data $\mathbf{X}$ is defined as follows.

$$\begin{aligned} llh(\mathbf{X}) &= \log P_G(\mathbf{X}|\alpha) & (12)\\ &= \log \int_{\theta} \sum_{\mathbf{Z}} P(\mathbf{X}, \mathbf{Z}|\theta, \hat{\theta}) P(\theta|\hat{\alpha}) d\theta \end{aligned}$$

Exact computation of Eq. (12) is intractable due to the integration over θ introduces additional dependencies among $\mathbf{Z}$. We use the following approximation.

$$llh(\mathbf{X}) \approx \hat{llh}(\mathbf{X}) = \log \sum_{\mathbf{Z}} P(\mathbf{X}, \mathbf{Z}|\theta^*, \hat{\theta}) \quad (13)$$

where θ^* is defined by Eq. (11). Then Eq. (13) can be computed using forward-backward algorithm (Rabiner 1989).

Experiments

We evaluate the model on two tasks related to motion capture data analysis. For each type of real motion capture data, we fit one model to capture the specific dynamic process of the action. We first quantitatively evaluate the model capability in capturing dynamics through reconstruction experiments. Then we show the learned model can be used to synthesize novel motion data with different intra-class variation with both quantitative and qualitative results.

Datasets: CMU Motion capture database (CMU) contains a diverse collection of human motion data captured by commercial motion capture system. Up to date, there are 2605 sequences in 6 major categories and 23 subcategories collected from 112 different subjects. We select a subset of the database to train our model including actions of walking, running and boxing from 31 subjects with averaging 101 sequences per action. **UC Berkeley MHAD** (Ofli et al. 2013) contains motion data collected by multiple modalities. We only use the motion capture data. There are 12 subjects perform 11 type of actions and each action is repeated 5 times, yielding large intra-class variation. We select three actions for experiments, namely jumping in place, jumping Jack and boxing, which involve substantial whole body movement.

Preprocessing: We subtract the root joint location of each frame to make the skeleton pose invariant to position change. We further convert the rotation angles to exponential map representation in the same way as (Taylor, Hinton, and Roweis 2006), which makes the skeleton pose invariant to the orientation against gravitational vertical. We exclude features that are mostly constant (standard deviation < 0.5), resulting 53 and 60 feature dimension per frame respectively on CMU and Berkeley datasets. The feature dimension is then doubled by including speed feature obtained as the difference of consecutive frames along each feature dimension. All features are scaled to have standard deviation 1 within each dimension. Finally, we divide the original sequences into overlapping segments of the same length for simplicity so that the model likelihood on different data is unaffected by the sequence length, though our model can take sequence input with different length. The preprocessed data is then used to train HHMM and other compared methods. We evaluate performance on feature space for all methods.

Implementation: For Algorithm 1, we use $k = 1, l = 1, M = 10, N = 100$. RMSProp decay is 0.9 and perturbation is 10^{-6}. The learning rate for generator is 10^{-3} and for discriminator is 10^{-4}. The maximum number of epochs is set to 100. To initialize α, we use K-means to cluster observations and use cluster assignment as hidden state value, from which we can estimate the model parameters and hyperparameters. To initialize ϕ^+, we use MLE of the first batch of real and synthetic data. ϕ^- is set equal to ϕ^+. Our Matlab code runs on a PC with 3.4GHz CPU and 8GB RAM. The average training time per class is 1.3 hour on CMU dataset and 1.9 hour on Berkeley dataset.

Data reconstruction

In this experiment, we demonstrate the learned HHMM have large capacity to handle intra-class variation in motion capture data. For each action category, we divide the data into 4 folds with each fold containing distinct subjects. Reconstruction is performed in a cross-fold manner meaning each fold is used as testing data once with remaining folds as training data. We report the average results over all folds and all input dimensions.

Quantitative metrics: We use Pearson correlation coefficient (PCC) and mean squared error (MSE) computed in feature space between reconstructed and actual values. PCC measures how well the prediction can capture the trend of motion change. PCC is a number between 0 and 1 and the larger the better. MSE is a positive number measuring the deviation between reconstructed and actual value and the smaller the better. We also report approximate loglikelihood of model evaluated on reconstructed data.

First, we compare with two baselines namely HMM and HHMM that are both learned by maximizing likelihood. While MLE of HMM is done using EM, MLE of HHMM is intractable. We approximate the MLE through a two-step optimization process as described in inference method. We vary the hidden state number of all the methods and evaluate their performance as shown in Figure 2.

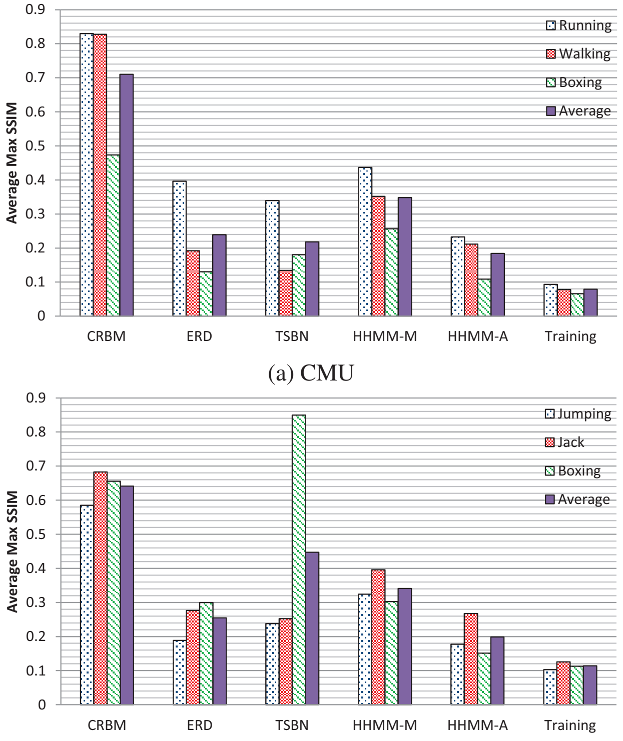
We observe that both variants of HHMM consistently outperforms HMM in PCC and MSE across different state numbers. In addition, when the state number is small, increasing the value helps both methods. As the value keeps increasing, HHMM performance reaches a plateau and HMM performance starts to drop, showing symptom of overfitting to training data. The overfitting of HMM becomes more clear when looking at the likelihoods, which drop significantly from training data to testing data. This shows that compared to non-hierarchical counter-part, HHMM has a larger capacity, which allows the model to adapt to novel data and less prone to overfitting. Comparison between two variants shows that HHMM-M consistently achieves higher likelihood on training data across actions and datasets than HHMM-A. This is consistent with the maximizing likelihood objective of HHMM-M. On testing data, the likelihood gap between HHMM-M and HHMM-A becomes smaller. For PCC and MSE, HHMM-A consistently outperforms HHMM-M. Overall, these results show that the adversarially learned HHMM can generalize better to novel data by capturing dynamic data distribution well.

Then we compare our method with several state-of-the-art dynamic generative models including GPDM (Wang, Fleet, and Hertzmann 2008), which is a non-parametric model, ERD (Fragkiadaki et al. 2015), which is an RNN/LSTM based method, and TSBN (Gan et al. 2015), which incorporates neural networks and graphical models. We set the hidden state number to 20 for HHMM throughout the remaining experiments. For other methods, we use author provided code to perform experiments. The results average over different actions are shown in Table 1.

Table 1: Reconstruction results of different methods averaged over different features and actions. Number in [] is standard deviation.

Dataset	CMU		Berkeley	
Metric	PCC	MSE	PCC	MSE
HMM	0.36[0.46]	1.12[1.54]	0.43[0.46]	0.87[1.95]
GPDM	0.70[0.24]	0.24[0.36]	0.47[0.35]	0.51[1.03]
ERD	0.66[0.34]	0.61[1.15]	0.75[0.30]	0.31[1.17]
TSBN	0.79[0.24]	0.27[0.92]	**0.81**[0.25]	0.18[0.64]
HHMM	**0.81**[0.22]	**0.20**[0.77]	**0.81**[0.26]	**0.12**[0.30]

On average, in CMU dataset, we achieve 2% absolute improvement in PCC compared to the second best TSBN and 0.04 absolute reduction in MSE compared to the second best GPDM. In Berkeley dataset, we achieve comparable performance in PCC compared to the second best TSBN and 6% improvement to ERD. We reduce MSE by 0.06 compared to the closest competitive TSBN. In both datasets, we outperform the baseline method by a large margin.



(a) CMU

(b) Berkeley

Figure 3: Average largest pairwise SSIM between synthetic motion sequences and real sequences from (a) CMU and (b) Berkeley datasets. (Best view in color)

Data synthesis

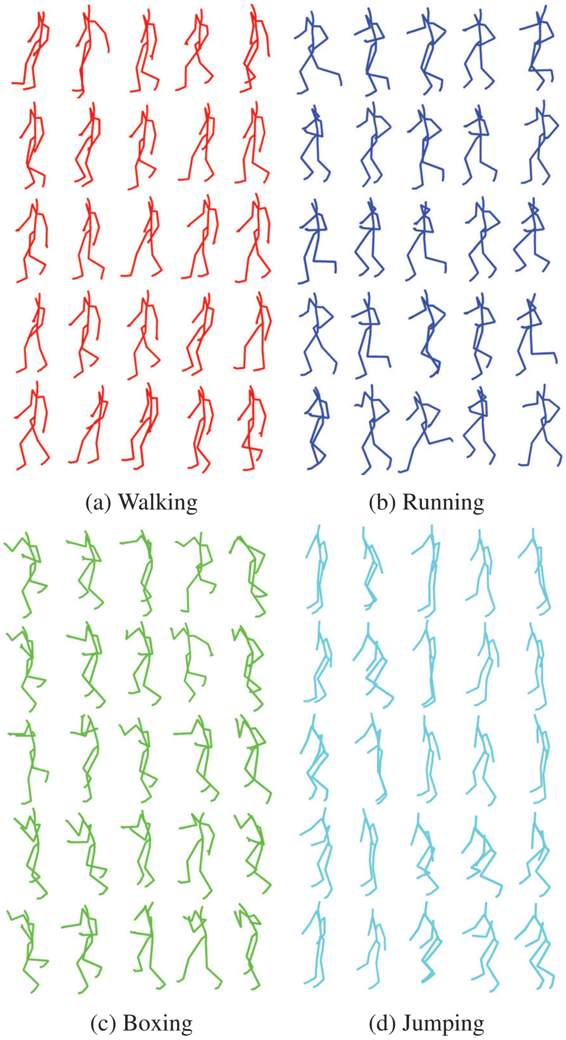
In this experiment, we demonstrate that adversarially learned HHMM can generate both realistic and diverse motion sequences. For each type of action, we train a model, which is then used to generate motion of the same type following the description in inference method.

Quantitative results: Sequential data brings additional challenge to quality evaluation due to large variation and dependency on both space and time. Motivated by the need to consider both fidelity and diversity of the generated sequential data, we adopt the structure similarity index (SSIM) (Wang et al. 2004) to evaluate synthesized data quality. SSIM is originally proposed for evaluating quality of a corrupted image against intact reference image. It is easy to compute and correlates well with perceptual quality of the image. It is a value between 0 and 1. The larger the value the more perceptually similar the images. (Odena, Olah, and Shlens 2017) adopted it for evaluating the overall image diversity generated by the GAN. To adapt SSIM for sequential data, we concatenate the features over time so that it can be viewed as an image, where each pixel in the image corresponds to a joint angle at a time. For each method, we generate 1000 sequences. For each sequence, we compute the pairwise SSIM against all the training sequences and choose the largest one. Finally, we use the average largest SSIM as measure of the diversity of the synthesized sequences. As a reference, we compute the pairwise SSIM among all the training sequences. The results are shown in Figure 3. For both datasets, the average training data SSIM is the lowest among all the results, indicating significant intra-class variation. Among different competing methods, HHMM-A achieves the lowest average SSIM. This shows adversially learned HHMM can generate the most diverse set of motion sequences. Comparing different action categories, a more complex action such as boxing usually achieves lower SSIM. From this point of view, HHMM is generating data consistent with the training set. For method producing high SSIM value *e.g.* TSBN on Berkeley's boxing, it indicates that the method overfits to some training data instances and fails to generalize diverse synthetic data.

Qualitative results: Figure 4 show some examples of synthetic sequences of different actions, where different rows show different samples drawn from the same motion category. Notice that the sampling process may not always generate meaningful motion in terms of the pose change since the random nature of hidden state transition. We use SSIM as a reference and select the generated sequence whose largest SSIM is above a threshold. On one hand, we are able to distinguish different motion sequences, indicating the data look physically meaningful and realistic. On the other hand, the sequences show various styles in motion, which show HHMM can generate different variations for the same action.

Conclusion

In this paper, we enhanced HMM through Bayesian hierarchical framework to improve the model capability in modeling dynamics under intra-class variation. We proposed a novel learning method that learns HHMM under adversarial objective, which has shown promising results in data generation applications compared to conventional maximum likelihood learning. Through both quantitative and qualitative evaluations, we showed the learned model can capture the dynamic process of human motion data well and can be used to generate realistic motion sequence with intra-class variation. For future work, we plan to introduce higher order dependency structure to better capture long-term dependency. We are also interested in training with different types of actions together instead of fitting one model at a time.


(a) Walking (b) Running

(c) Boxing (d) Jumping

Figure 4: Synthetic motion sequences. Each row is a uniformly downsampled skeletal sequence from one synthetic action. Different rows are different samples.

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 Cognitive Horizon Network (CHN).

References

Bayer, J., and Osendorfer, C. 2014. Learning stochastic recurrent networks. *arXiv*.

Brand, M., and Hertzmann, A. 2000. Style machines. In *SIGGRAPH*. ACM.

Brand, M.; Oliver, N.; and Pentland, A. 1997. Coupled hidden markov models for complex action recognition. In *CVPR*.

Brand, M. 1999. Voice puppetry. In *SIGGRAPH*.

Cappé, O.; Buchoux, V.; and Moulines, E. 1998. Quasi-newton method for maximum likelihood estimation of hidden markov models. In *ICASSP*.

Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A. C.; and Bengio, Y. 2015. A recurrent latent variable model for sequential data. In *NIPS*.

CMU. Cmu mocap database http://mocap.cs.cmu.edu/.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society*.

Fine, S.; Singer, Y.; and Tishby, N. 1998. The hierarchical hidden markov model: Analysis and applications. *Machine learning*.

Fraccaro, M.; Sønderby, S. K.; Paquet, U.; and Winther, O. 2016. Sequential neural models with stochastic layers. In *NIPS*.

Fragkiadaki, K.; Levine, S.; Felsen, P.; and Malik, J. 2015. Recurrent network models for human dynamics. In *ICCV*.

Gan, Z.; Li, C.; Henao, R.; Carlson, D. E.; and Carin, L. 2015. Deep temporal sigmoid belief networks for sequence modeling. In *NIPS*.

Gauvain, J.-L., and Lee, C.-H. 1994. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *TSAP*.

Ghahramani, Z.; Jordan, M. I.; and Smyth, P. 1997. Factorial hidden markov models. *Machine learning*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Johnson, M.; Duvenaud, D. K.; Wiltschko, A.; Adams, R. P.; and Datta, S. R. 2016. Composing graphical models with neural networks for structured representations and fast inference. In *NIPS*.

Joshi, A.; Ghosh, S.; Betke, M.; Sclaroff, S.; and Pfister, H. 2017. Personalizing gesture recognition using hierarchical bayesian neural networks. In *CVPR*.

Kalman, R. E., et al. 1960. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*.

Krishnan, R. G.; Shalit, U.; and Sontag, D. 2015. Deep kalman filters. *arXiv*.

Mittelman, R.; Kuipers, B.; Savarese, S.; and Lee, H. 2014. Structured recurrent temporal restricted boltzmann machines. In *ICML*.

Murphy, K. P. 2002. *Dynamic bayesian networks: representation, inference and learning*. Ph.D. Dissertation, University of California, Berkeley.

Nguyen, A.; Clune, J.; Bengio, Y.; Dosovitskiy, A.; and Yosinski, J. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *CVPR*.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier GANs. In *ICML*.

Ofli, F.; Chaudhry, R.; Kurillo, G.; Vidal, R.; and Bajcsy, R. 2013. Berkeley mhad: A comprehensive multimodal human action database. In *WACV*.

Rabiner, L. R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv*.

Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative adversarial text to image synthesis. In *ICML*.

Saito, M.; Matsumoto, E.; and Saito, S. 2017. Temporal generative adversarial nets with singular value clipping. In *ICCV*.

Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *ICML*.

Sutskever, I., and Hinton, G. 2007. Learning multilevel distributed representations for high-dimensional sequences. In *AISTATS*.

Sutton, C.; McCallum, A.; and Rohanimanesh, K. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *JMLR*.

Tang, K.; Fei-Fei, L.; and Koller, D. 2012. Learning latent temporal structure for complex event detection. In *CVPR*.

Taylor, G. W.; Hinton, G. E.; and Roweis, S. T. 2006. Modeling human motion using binary latent variables. In *NIPS*.

Theis, L.; Oord, A. v. d.; and Bethge, M. 2015. A note on the evaluation of generative models. *arXiv*.

Tieleman, T., and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2017. Mocogan: Decomposing motion and content for video generation. *arXiv*.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. In *NIPS*.

Walker, J.; Doersch, C.; Gupta, A.; and Hebert, M. 2016. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*.

Wang, X., and Ji, Q. 2012. Learning dynamic bayesian network discriminatively for human activity recognition. In *ICPR*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *TIP*.

Wang, J. M.; Fleet, D. J.; and Hertzmann, A. 2008. Gaussian process dynamical models for human motion. *TPAMI*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*.

Xue, T.; Wu, J.; Bouman, K.; and Freeman, B. 2016. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*.

Yu, S.-Z. 2010. Hidden semi-markov models. *Artificial Intelligence*.