Gaussian Process Multi-Dynamical Models for Human Activity Recognition

Priyanshu Agrawal
School of Computing
University of Connecticut
Storrs, CT 06269
priyanshu.agrawal@uconn.edu

Abstract

Human activity recognition is essential for effective human-robot collaboration. In order for a robot and human to work collaboratively, the robot needs to infer intention and motion of the human. Current approaches often rely on deep learning techniques that require extensive training data, which may not be available for specific activities. We propose a data-efficient method for human activity recognition, capable of learning from only a few activity demonstrations. Our approach operates on time-series data of human joints, which can be obtained from specialized sensors or pre-trained vision models. We use Gaussian Process Dynamical Models to learn activity-specific behaviors, extending them to learn a unique dynamics model for each activity in a shared latent space. Then, we apply particle filters to estimate the most likely activity given new observations. This technique achieves strong results on walking and running trials from the CMU Motion Capture Database and shows potential to be applied to more complex datasets. Code is available at https://github.com/Priyanshu4/gpmdm.

1 Introduction

Human activity recognition is the problem of identifying the activity or task a human is performing based on wearable sensors or video. In this work, we focus specifically on human-activity recognition using three dimensional skeletal joint data. This data contains the angles of all joints in the human body at each frame. In the past, such data was only available through expensive motion capture systems [1]. However, recent work in computer vision has made progress on extracting this data from monocular cameras [2]. It is also possible to extract these estimates from depth cameras, which are commonly used in robotics applications. Assuming that we have a real-time pose extraction technique, collaborative robots can directly use human joint data to predict human motion and infer actions.

In some settings, like a collaborative manufacturing process with specific assembly tasks, data is highly limited. Therefore, using joint angles derived from a pre-trained model may be preferable to training a computer vision model, which requires significantly more training data. There are a variety of existing approaches to human action recognition from skeletal data, including spatio-temporal deep learning based approaches [3]–[5], standard classifiers applied to single poses or sequences with special feature extraction [6]–[8] and probabilistic models such as HMMs [9], [10].

In contrast, the proposed approach is based on Gaussian Process Dynamical Models (GPDM) [11]. This a non-parametric Bayesian model which require less training data than deep learning models and provides probabilistic predictions. Uniquely, a GPDM based approach learns dynamics models of the activities, meaning that it can be applied to motion prediction in addition to classification. In fact, GPDMs were originally developed to generate realistic human motions. There is previous work related to GPDMs for human activity recognition, such as [12], but this requires fitting a GPDM at inference time. Instead, our approach applies particle filtering to estimate class probabilities at potentially real-time rates.

Specifically, this work contributes the following:

1. An extension of GPDMs to learn dynamics for multiple classes in a shared latent space.
2. A particle filter to estimate latent states and class probabilities in this model.

2 Related Work

2.1 Gaussian Process Dynamical Models

Gaussian Process Dynamical Models (GPDMs), proposed in [11], are an extension of Gaussian Process Latent Variable Model (GPLVMs) to include temporal dynamics in the latent space. Standard GPLVMs learn a latent representation of the training data and a Gaussian process which maps the latent space value to the observation space. GPDMs also learn a Gaussian process which maps a latent space value to the latent space value at the next timestep. Specifically, they assume the latent dynamics x follow the form $x_{t+1} = f(x_t, A) + n_x$, and that the observation space y has the form $y_t = g(y_t, B) + n_y$. Here, f and g are parametrized by matrices A and B. The variable n corresponds to Gaussian noise. They consider f and g to be non-linear functions where they are linear combinations of basis functions. They are able to marginalize out the specific forms of f and g, allowing them to find a closed form expression for likelihood of the latent values and the likelihood of the training observations given the latent values. To train the model, they then apply numerical optimization (i.e. gradient descent) to minimize the negative log likelihood. Once the GPDM is trained, we have the ability to predict and covariance of a Gaussian distribution in the observation space given a latent state. We also have the ability to predict mean and covariance of a Gaussian distribution in latent space given the previous latent state.

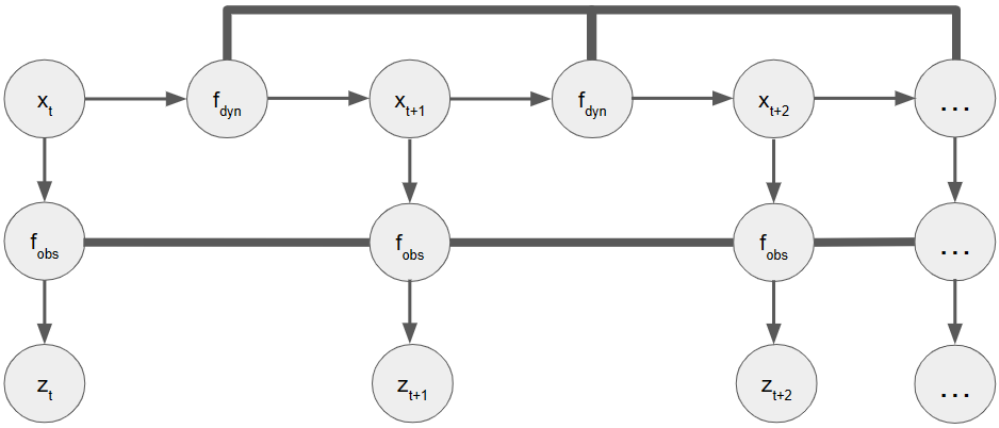

Figure 1: A probabilistic graphical model for GPDMs. Note that the solid lines represent that both functions are the exact same.

2.2 Gaussian Process Bayesian Filters

In [13], the authors show Gaussian process models can be integrated into Bayesian filters. Specifically, they cover the case where both the dynamics model and observation model are represented by Gaussian processes. They develop particle filter, extended Kalman filter and unscented Kalman filter algorithms that utilize Gaussian processes.

2.3 GPDMs for Human Activity Recognition

There are several other works using GPDMs for human activity recognition or related tasks. In [12], they train a seperate GPDM for each sequence of running, walking, jumping and swinging. Then, they apply an SVM for classification using the learned kernel hyperparameters as features. While this shows decent results, it requires fitting a GPDM to each new testing sample which is not practical in real-time.

In [10], they use an HMM to tell if a pedestrian is walking, stopping, starting or standing. Then, they use a GPDM (trained seperately for each class) to predict future motions. To estimate a new observation's latent state in the GPDM, they use the nearest neighbor in the training dataset as a starting point for gradient descent.

3 Methods

3.1 Learning Latent Representations and Dynamics Models

While standard GPDMs can only learn one observation mapping and one dynamics mapping, we modify GPDMs to learn multiple dynamics mappings. This enables us to learn a separate dynamics model for each activity but keep all activities in a shared latent space. This is done by modifying the dynamics mapping $\mathbf{K_X}$.

Consider the loss function used in training the GPDM, shown in equation 1.

$$\begin{aligned} &-\log p(\mathbf{Z} \mid \mathbf{X}) = -\log p(\mathbf{X} \mid \mathbf{Z}) - \log p(\mathbf{X}) \\ &= \underbrace{\frac{D}{2}\log|\mathbf{K_Z}| + \frac{1}{2}\operatorname{tr}\left(\mathbf{K_Z}^{-1}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\right)}_{\mathcal{L}_{\text{GPLVM}}} + \underbrace{\frac{D}{2}\log|\mathbf{K_X}| + \frac{1}{2}\operatorname{tr}\left(\mathbf{K_X}^{-1}\mathbf{X}_{2:T}\mathbf{X}_{2:T}^{\mathrm{T}}\right) - \log p\left(\mathbf{X}_1\right)}_{\mathcal{L}_{\text{Markov}}}. \end{aligned} \tag{1}$$

The portion of the loss due to $-\log p(\mathbf{Z} \mid \mathbf{X})$ is only related to the observation mapping and is present in GPLVMs. On the other hand, the portion of the loss due to $-\log p(\mathbf{X})$ is unique to the Markovian dynamics of GPDMs. Furthermore, note that the kernel $\mathbf{K_X}$ is responsible for the dynamics mapping. Therefore, by setting to zero all of the values of $\mathbf{K_X}$ which correspond to two samples of different classes, we can completely de-correlate the dynamics mapping from separate classes. As a result, our training process will learn a single shared latent space but a separate dynamics model for each class. During prediction for a given class, we use an $\mathbf{X}$ kernel matrix which contains zeros in all locations corresponding to another class. The GPLVM portion of the loss is left completely unaffected, as the matrix $\mathbf{K_Z}$ is never modified. I refer to this as a Gaussian Process Multi-Dynamical Model, or GPMDM.

This idea was inspired by [14], in which they decorrelate samples from different classes in the $\mathbf{K_Z}$ matrix but leave the $\mathbf{K_X}$ matrix unaffected. However, they are addressing a different problem and their goal is to separate two classes in latent space. In my case, I am interested in a shared latent space but unique dynamics models. A shared latent space is useful because it makes it simpler for a particle filter to estimate the latent state of the human. Unique dynamics models are necessary to distinguish between the different tasks.

To train my model, I utilize gradient descent, specifically an Adam optimizer. All hyperparameters of the model, including lengthscales and noise parameters are optimized using gradient descent. The kernel used is linear + radial basis function, the same kernel used in the original GPDM paper [11].

3.2 Particle Filter for Real-Time Classification

GPDMs do not provide a way to estimate the latent state of a new observation. Therefore, we need a new way of estimating both the latent state and class in our GPMDM model. In order to do so, we use a particle filter. To begin, we randomly select particles from the training data. Each particle i corresponds to one possible state of the human at time step k: $\langle x_k^{[i]}, c_k^{[i]}, w_k^{[i]} \rangle$, where x is the latent state, c is the class, and w is the weight.

In addition to having the GPMDM model, our particle filter requires us to have a Markovian transition matrix between classes. The transition matrix gives the probability of changing from each class to every another at each time step. Such a transition matrix can easily be computed from the training data or manually chosen by a domain expert.

At each time step, we update every particle. We sample each particles new class from the categorical distribution given by the class transition matrix. We sample each particles new latent state from the Gaussian distribution given by the corresponding dynamics model for the class of that particle. Lastly, we update the weights of each particle based on the likelihood that the particle generated the observation. This likelihood is computed from the observation Gaussian process. This algorithm is similar to that developed in [13], however, it also incorporates class information. Details are provided in algorithm 1.

The weights, latent states and classes of the particles give us a probability distribution over the latent states and classes. We can compute the mean and variance of the latent state distribution with weighted averages. We can compute the probability distribution over classes by taking the weighted mean of the particle classes. In my testing, I used the highest probability class as the chosen classification.

Algorithm 1: Update Particle Filter: `GPMDM-PF_Update`

Input: $\mathcal{X}_{k-1}$ (previous particle set), z_k (new observation)
Output: $\mathcal{X}_k$ (updated particle set)
Models: T (class transition matrix), GP_{dyn} (GP dynamics model for each class), GP_{obs} (GP observation model)
$\mathcal{X}_k \leftarrow \emptyset$;
for $m = 1$ **to** M **do**
    Sample new class $c_k^{[m]}$ for particle m from the categorical $\mathbf{T} * c_{k-1}^{[m]}$;
    Propagate latent state: $x_k^{[m]} \sim \mathcal{N}(\mu_{\text{dyn}}(x_{k-1}^{[m]}), \Sigma_{\text{dyn}}(x_{k-1}^{[m]})$ from GP_{dyn} for class $c_k^{[m]}$;
    Update weight: $w_k^{[m]} \propto \mathcal{N}(z_k; \mu_{\text{obs}}(x_k^{[m]}), \Sigma_{\text{obs}}(x_k^{[m]}))$ from GP_{obs};
end
Normalize weights: $w_k^{[m]} \leftarrow \frac{w_k^{[m]}}{\sum_{i=1}^{M} w_k^{[i]}}$;
for $m = 1$ **to** M **do**
    Resample particle i with probability $\propto w_k^{[i]}$;
    Add particle $\langle x_k^{[i]}, c_k^{[i]}, w_k^{[i]} \rangle$ to $\mathcal{X}_k$;
end
return $\mathcal{X}_k$;

4 Results

4.1 Dataset

To test our approach, we use the CMU Motion Capture Database [1]. Specifically, we select 31 walking trials and 27 running trials from the database. We use one-third of these trials as a training set and the remaining two-thirds are our test set. In total, there are 19 trials in our training set and 39 trials in our testing. Each trial is about 2 to 5 seconds. There are 10 unique subjects across all trials. Note that the train set was selected randomly, however, it was done in a way such that at minimum, there was at least one example of every subject in the training set. This was necessary because the dataset is almost entirely composed by mocap database subjects 7 and 8 for walking. For running, it is mostly composed by subjects 9, 16 and 35. It just happened to be that these people did most of the walking and running trials in the database. Therefore, to ensure some diversity in the training set, we required that all subjects in the test set had at least one training example.

Another important note about our dataset is that it contains no examples of switching between actions. Our particle is designed to handle such situations, but none were present in the data. Regardless, we used a transition matrix corresponding to 10% chance of switching at each frame. Furthermore, while the database provides 120 frames per second, all the data that we used in our experiments was downsampled to 30 frames per second. Lastly, while the dataset contains 62 degrees of freedom per skeleton, we ignore joint angles corresponding to fingers, hands, wrist and neck, reducing the degrees of freedom to only 35 to reduce computational cost.

4.2 Training

Before training, the latent states were initialized using PCA. The GPMDM used had 4 latent dimensions. The model was trained for 500 iterations using an Adam optimizer with a 0.01 learning rate. On a consumer laptop CPU from 2017, the model took around 45 minutes to train.

Figure 2 shows the learned latent embeddings of the training data. Since there is no examples of switching between walking and running, there is no overlap or connection between the two classes of data in the latent space. It is not required by the model to have classes be separated in latent space, however, it occurred in this case, indicating the simplicity of the dataset.

4.3 Classification Performance

Our model achieves very strong classification results on the dataset. This is in part due to the simplicity of the dataset, but also indicates potential for the approach to scale to more complex datasets.

On a per trial basis, only 1 trial out of the 39 was misclassified. This corresponded to an accuracy of 0.974 and an F1-Score of 0.976. A trial was considered correct if more than 50% of its frames were correctly classified. In the

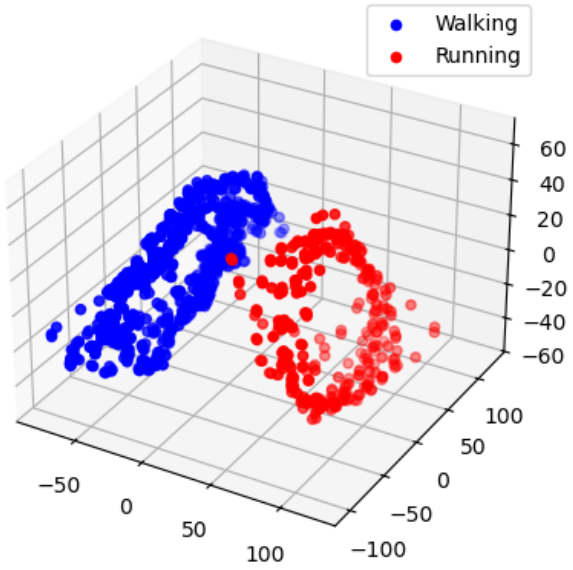

Figure 2: The learned latent representation of the walking and running training data. Each data point represents a specific human pose.

trials that were classified correctly, almost all their frames were classified correctly. In the one trial that was classified incorrectly, almost none of its frames were classified correctly.

On a per frame basis, the accuracy was 0.921 and F1-Score was 0.943. The vast majority of the incorrect frames are from the one misclassified trial. The particle filter was able to process around 10 to 15 frames per second on a laptop CPU. This indicates the ability to achieve real-time performance with additional compute.

Additionally, while the particle filter should in theory provide a probability distribution over classes for each frame, in practice, the probability distribution was always a 1 for one class and 0 for the other. This is because one particle typically has a significantly higher weight than all other particles.

	Actual Walk	**Actual Run**	**Total**
Predicted Walk	20	0	20
Predicted Run	1	18	19
Total	21	18	39

Table 1: Confusion Matrix for Trial Results

	Actual Walk	**Actual Run**	**Total**
Predicted Walk	1704	8	1712
Predicted Run	199	717	916
Total	1903	725	2628

Table 2: Confusion Matrix for Frame Results

5 Discussion

The model shows significant promise to scale to more complex datasets but also shows some potential for improvements. The model should be tested on a dataset that includes switching between actions in order to determine if the model will achieve strong performance even when the latent spaces for different classes overlap. Furthermore, the model should be tested for motion prediction in addition to activity recognition. The particle filter approach already provides an estimate of the current latent state, so by propagating this estimate forward in time we can generate predictions with little extra work. It is possible that the model can still achieve strong results on these more complex tasks, however, the number of particles may need to be increased.

Despite the strong results of the model, there is also much room for improvement. For instance, it may be worthwhile to explore inducing point based approaches to reduce the cost of using the model to make predictions at test time. This could enable a fully real-time model, even on larger datasets. Furthermore, the model doesn't provide an accurate estimate of uncertainty because one particle significantly outweighs the rest. This results in the model being overconfident in its prediction. It could be interesting to explore adaptive particle filtering or a smoothing of weights in order to create better probability estimates.

References

[1] Carnegie Mellon University Motion Capture Database, *Cmu motion capture database*, `http://mocap.cs.cmu.edu/`, Accessed: 2024-12-10, 2024.

[2] J. Gong, L. G. Foo, Z. Fan, Q. Ke, H. Rahmani, and J. Liu, "Diffpose: Toward more reliable 3d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 13 041–13 051.

[3] A. Franco, A. Magnani, and D. Maio, "A multimodal approach for human activity recognition based on skeleton and rgb data," *Pattern Recognition Letters*, vol. 131, pp. 293–299, 2020.

[4] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention lstm networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2018. DOI: 10.1109/TIP.2017.2785279.

[5] S. Cho, M. Maqbool, F. Liu, and H. Foroosh, "Self-attention network for skeleton-based human action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2020.

[6] H. Zhou, L. Wang, and D. Suter, "Human action recognition by feature-reduced gaussian process classification," *Pattern Recognition Letters*, vol. 30, no. 12, pp. 1059–1066, 2009, Image/video-based Pattern Analysis and HCI Applications, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2009.03.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865509000592.

[7] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2014.

[8] K. Ashwini, R. Amutha, and S. Aswin raj, "Skeletal data based activity recognition system," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 444–447. DOI: 10.1109/ICCSP48568.2020.9182132.

[9] L. Piyathilaka and S. Kodagoda, "Gaussian mixture based hmm for human daily activity recognition using 3d skeleton features," in *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, 2013, pp. 567–572. DOI: 10.1109/ICIEA.2013.6566433.

[10] R. Quintero Mínguez, I. Parra Alonso, D. Fernández-Llorca, and M. Á. Sotelo, "Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, 2019. DOI: 10.1109/TITS.2018.2836305.

[11] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 283–298, 2007.

[12] H. Jamalifar, V. Ghadakchi, and S. Kasaei, "3d human action recognition using gaussian processes dynamical models," in *6th International Symposium on Telecommunications (IST)*, 2012, pp. 1179–1183. DOI: 10.1109/ISTEL.2012.6483167.

[13] J. Ko and D. Fox, "Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3471–3476. DOI: 10.1109/IROS.2008.4651188.

[14] R. Sawata, T. Ogawa, and M. Haseyama, "Class-aware shared gaussian process dynamic model," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096743.