Clinical Time Series Prediction with a Hierarchical Dynamical System

Zitao Liu and Milos Hauskrecht

Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, USA
{ztliu,milos}@cs.pitt.edu

Abstract. In this work we develop and test a novel hierarchical framework for modeling and learning multivariate clinical time series data. Our framework combines two modeling approaches: Linear Dynamical Systems (LDS) and Gaussian Processes (GP), and is capable to model and work with time series of varied length and with irregularly sampled observations. We test our framework on the problem of learning clinical time series data from the complete blood count panel, and show that our framework outperforms alternative time series models in terms of its predictive accuracy.

Keywords: Time Series, Gaussian Processes, Linear Dynamical System.

1 Introduction

The problem of modeling of clinical time series comes with a number of challenges [1]. First, time series for the different patients admitted to hospital are hard to align; they start at different times with respect to the disease, and they may vary in length depending on the span of patient's hospitalization. Second, observations corresponding to the different laboratory tests are collected at different times, and the time elapsed between two consecutive observations may vary. The challenge is to build models and algorithms that are both accurate and flexible enough to represent such time series.

We propose and develop (1) a hierarchical dynamical system model to represent the clinical time series data, and (2) algorithms that can (a) learn the model efficiently from observational data, and (b) support predictive inferences. Our model is built by combining two machine learning frameworks used frequently for modeling dynamical systems in statistical machine learning: the Linear Dynamical System (LDS) [2] and the Gaussian Process (GP) model [3]. LDS, or Kalman filter [2] is the most frequently used approach for modeling dynamical systems in practice. The model comes with numerous computational advantages and well-understood algorithms for both model learning and model inference. The model defines a state-space process with linear transitions between two consecutive states taken at discrete time points. Hence it assumes some predefined and fixed time discretization. However, observations in clinical time series are often spaced irregularly in time. To address this problem we extend the linear dynamical system with a secondary (lower-level) Gaussian process defined over time

N. Peek, R. Marín Morales, and M. Peleg (Eds.): AIME 2013, LNAI 7885, pp. 227–237, 2013.

windows, instead of fixed-time points. The parameters of the Gaussian process are controlled by the higher-level LDS. The advantage of the Gaussian process is that observations are treated as the function of time and can be defined for an arbitrary observation sequence. This extension gives us the flexibility needed to model time series with observations sampled unevenly in time.

We experiment with and test the new model on clinical time series prediction problem. More specifically, we define, build and run the model for six common blood tests from the complete blood count panel. Our results show that the model leads to a more accurate predictive performance than existing time series models. In addition, we show our model is more robust and performs well when the number of patients and observations used to train the model is small.

Our paper is organized as follows. In Section 2, we review the basics of linear dynamical systems and Gaussian process models. In Section 3 we describe the time series prediction problem we want to solve and present the new hierarchical framework that combines the two models. We present experimental results that consist of predicting future values for six common lab tests and compare our method to alternative modeling approaches in Section 4. In Section 5, we summarize the work and outline future model extensions.

2 Background

2.1 Linear Dynamical System (LDS)

The Linear Dynamical System (LDS) is a classic and widely used real-valued time series model. An LDS on variables $\mathbf{z}_{1:T}$, $\mathbf{y}_{1:T}$ is defined using the following two equations:

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + \mathbf{e}_t; \qquad \mathbf{y}_t = C\mathbf{z}_t + \mathbf{v}_t \tag{1}$$

where $t \in \{1, \dots, T\}$ is the discrete time index; $\mathbf{z}_1$ is the initial state distribution with mean $\boldsymbol{\pi}_1$ and covariance matrix V_1, $\mathbf{z}_1 \sim \mathcal{N}(\boldsymbol{\pi}_1, V_1)$, $\mathbf{z}_t$ are the hidden states defining the process that are generated with the help of the transition matrix A with the independent zero mean noise $\mathbf{e}_t$, $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, Q)$; and $\mathbf{y}_t$ are the observations generated by the emission matrix C and independent variant noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, R)$. The LDS is characterized by the state transition probability $p(\mathbf{z}_t|\mathbf{z}_{t-1})$, where $p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(A\mathbf{z}_{t-1}, Q)$, and the state-observation probability $p(\mathbf{y}_t|\mathbf{z}_t) = \mathcal{N}(C\mathbf{z}_t, R)$. The complete set of the LDS parameters is $\Lambda = \{A, C, Q, R, \boldsymbol{\pi}_1, V_1\}$. The parameters can be learned from the observed data sequences using the Expectation-Maximization (EM) algorithm.

The advantage of the linear dynamical model is its simplicity. The disadvantages are its linearity which may prevent one from modeling more complex time series data, and the fact that the model is a discrete-time model with observations and predictions restricted to fixed time intervals.

2.2 Gaussian Process (GP) Regression

The Gaussian process is a nonparametric nonlinear Bayesian model popular in statistical machine learning [3]. The GP is best viewed as an extension of the multivariate Gaussian to infinite-sized collections of real-valued variables defining the distribution over random functions. A GP is represented by the mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$and the covariance function $K(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$, where $f(\mathbf{x})$ is the real-valued process. Since GP represents a Gaussian distribution over functions, it can be used to estimate the values of function f at an arbitrary position x_*. This application is referred to as Gaussian Process Regression [3]. The GP regression equations are:

$$\bar{f}_* = K(x_*, \mathbf{x}) \left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1} \mathbf{y} \tag{2}$$

$$Cov(f_*) = K(x_*, x_*) - K(x_*, \mathbf{x}) \left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1} K(x_*, \mathbf{x}) \tag{3}$$

where I is the identity matrix, $\mathbf{x}$ is the input vector and $\mathbf{y}$ is the output or target vector, $\bar{f}_*$ is the posterior function mean and $Cov(f_*)$ is the posterior covariance. With the right choice of the covariance function, the associated prediction uncertainty increases in regions away from observations, while it shrinks when it is close to observed data.

2.3 Gaussian Process Time Series Models

In time series modeling Gaussian processes are used primarily to capture the nonlinear dynamics and nonlinear relations among states and observations [4]. Briefly, the models assume that there exists a sequence of latent variables $\mathbf{z}_t$ that evolve over time through a Markovian process specified by a Gaussian process f and the observations $\mathbf{y}_t$ are generated by another Gaussian process u from corresponding latent variables: $\mathbf{z}_t = f(\mathbf{z}_{t-1}) + \mathbf{e}_t$; $\mathbf{y}_t = u(\mathbf{z}_t) + \mathbf{v}_t$. The notation $\mathbf{z}_t$, $\mathbf{y}_t$, $\mathbf{e}_t$, $\mathbf{v}_t$ is the same as the notation used in the previous section for LDS. Briefly, the LDS assumes linear dependencies among latent states and observations, while the Gaussian-process-based model replaces the linear dependences with more general non-linear functions f and u represented by Gaussian processes.

The existing GP-based dynamical models [5–8] come with a number of limitations. First, all of the existing models set GP's mean function to zero, which severely limits the functions one can represent. Second, similarly to LDS, they assume the observations are obtained at the same (regularly sampled) times, which is not true in the real world. Discretizing irregular sampled time series may introduce unnecessary inaccuracy and hence lower the model's accuracy and performance.

In this paper, we address the shortcomings of these methods by defining a hierarchical dynamical system that splits the process into a sequence of dependent local Gaussian processes and by using the LDS to capture the dependences between these local GPs. This is unlike [9] where local GPs are independent to each other. The local GPs' dependences naturally account for the transitions of mean functions and the irregular samples are handled by the local GPs.

3 Hierarchical Dynamical System

3.1 Problem Description

We define the time series prediction/regression function for clinical time series as: $g : \mathbf{Y}_{\text{obs}} \times t \rightarrow \hat{\mathbf{y}}$, where $\mathbf{Y}_{\text{obs}}$ is a sequence of past observation-time pairs $\mathbf{Y}_{\text{obs}} = (\mathbf{y}_i, t_i)_{i=1}^n$, such that, $0 < t_i < t_{i+1}$, $\mathbf{y}_i$ is a p-dimensional observation vector made at time (t_i), and n is the number of past observations; and $t > t_n$ is the time at which we would like to predict the observation $\hat{\mathbf{y}}$.

Typically, the prediction function assumes that the values (observations) are given at regularly sampled discrete-time points, which means $t_{i+1} - t_i = L$, where L is a constant reflecting the data-sampling interval. However, in this work we assume our observations are sampled irregularly from some process which is more general and more common in real-life settings. For example, in the clinical domain, observations that correspond to lab test values for a patient during his or her hospital period are often recorded irregularly due to the different patients' health conditions or the different sample-collection times.

3.2 Hierarchical Dynamical System (HDS)

In this section, we develop a two-layer hierarchical dynamical model that lets us represent the entire time series information in a more flexible manner. The key structure of the model is shown in Figure 1. Briefly, the model consists of two hierarchically related processes: the Gaussian process and the linear dynamical system. The Gaussian process is restricted to a time window of a finite duration and it is used to represent time series and its changes for shorter time spans. Longer-term process changes are modeled and controlled by the linear dynamical process. In the first layer, which is shown in a dashed line box, we transform the entire irregular time series data into m windows $\{w_i\}_1^m$ using a predefined window size and a predefined overlap size. Each window w_i in Figure 1 relates observations $\{y_i^1, y_i^2, \ldots, y_i^{N_i}\}$ using the same window-specific GP and N_i is the number of observations in window w_i. Hence, instead of using a single GP we capture the variation in the entire time series by using many different window-specific Gaussian processes. We denote the GP for window w_i as GP_i.

The sequence of window-specific GP_is (more precisely their parameters) is linked together using the higher-level LDS (in layer 2). That is, the LDS represents the dynamics and changes of parameters γ_i defining individual GP_i. We assume the different GP_i share the same covariance function parameter Θ. The parameters γ_i that define the mean of the window-specific GP_i are unobserved (hidden). In the learning phase, we estimate γ_i measurements $\{y_i^1, y_i^2, \ldots, y_i^{N_i}\}$ observed in window w_i. Meanwhile, in the prediction phase, γ_i is controlled by the LDS and evolves in time. The LDS is defined as:

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + \mathbf{e}_t; \gamma_t = C\mathbf{z}_t + \mathbf{v}_t \tag{4}$$

where parameters $\{\gamma_i\}$ act like observations, and $\mathbf{e}_t$ and $\mathbf{v}_t$ are zero-mean normally distributed random variables with covariance matrices Q and R

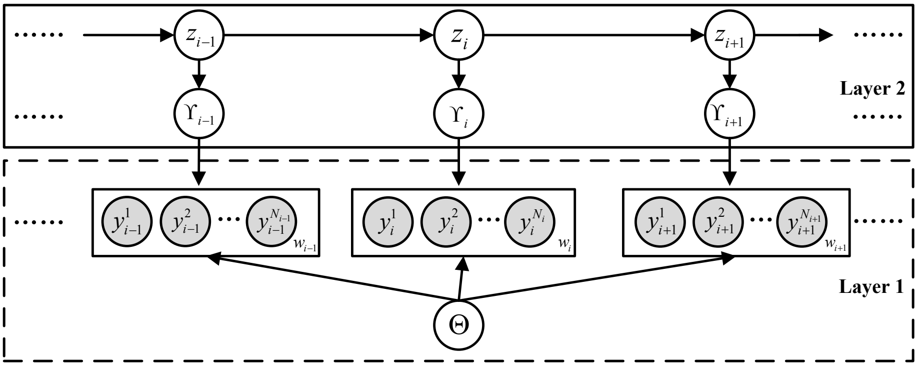
Layer 2
Layer 1
z_{i-1}
z_i
z_{i+1}
Υ_{i-1}
Υ_i
Υ_{i+1}
y_{i-1}^1 y_{i-1}^2 $y_{i-1}^{N_{i-1}}$
w_{i-1}
y_i^1 y_i^2 $y_i^{N_i}$
w_i
y_{i+1}^1 y_{i+1}^2 $y_{i+1}^{N_{i+1}}$
w_{i+1}
Θ

Fig. 1. Graphical illustration of our hierarchical dynamical model combining the Gaussian process and the linear dynamical system

respectively. Similarly to regular LDS introduced earlier in Section 2.1, $\boldsymbol{\pi}_1$ and V_1 are the initial state mean and variance.

3.3 Learning

The parameters of our hierarchical dynamical model are learnt in the two steps. In step 1 we learn the covariance function Θ. In step 2, we first estimate parameters $\{\gamma_i\}$ and then use them to learn the parameters of linear dynamical System in layer 2.

Step 1. Since GPs share the same covariance function, we use the entire time series data to make the estimation of the parameters Θ in the covariance function. We set Θ by maximizing the likelihood and seek the partial derivatives of the likelihood with respect to each parameter θ_i in Θ.

$$\frac{\partial}{\partial \theta_i} \log p(\mathbf{Y}|\Theta) = -\frac{1}{2}\mathrm{Tr}\left[K^{-1}\frac{\partial K}{\partial \theta_i}\right] + \frac{1}{2}\mathbf{Y}^T K^{-1}\frac{\partial K}{\partial \theta_i}K^{-1}\mathbf{Y} \tag{5}$$

where K is covariance matrix for all training data.

Step 2. We first estimate all parameters $\{\gamma_i\}$ representing the means of window-specific GPs from observations in windows $\{w_i\}$. In general, there are many different ways to estimate $\{\gamma_i\}$. Let h be a function used for estimating the mean of the Gaussian process from observations $\{y_i^j\}$. Examples of h can be $min, max, mean$ functions that return the minimum, maximum or mean observed value in the window. In our model, we use the value of the most recent observation in each window to estimate $\{\gamma_i\}$, that is, $\gamma_i = h(\{y_i^j\}) = y_i^{N_i}$. Once we get $\{\gamma_i\}$, we treat values of different $\{\gamma_i\}$ as the observations for the LDS (layer 2) in our hierarchical dynamical system. To learn the parameters of the LDS, we use the Expectation-Maximization(EM) learning algorithm to [10] iteratively re-estimate the parameters $\Lambda = \{A, C, Q, R, \boldsymbol{\pi}_1, V_1\}$ defining the LDS [11]. The learning algorithm is summarized in Algorithm 1.

Algorithm 1. Learning Hierarchical Dynamical Model

```
Do initialization and split the data into windows.
// learn $\Theta$, $\Theta = (\theta_1, \ldots, \theta_d)$
for $i = 1$ to $d$ do
    Get $\theta_i$ by any gradient based optimizer based on Eq.5
end for
// learn $\{A, C, Q, R, \boldsymbol{\pi}_1, V_1\}$
Learning LDS from the EM algorithm. See [11]
return $\Omega = \{\Theta, A, C, Q, R, \boldsymbol{\pi}_1, V_1\}$
```

3.4 Prediction

Once the hierarchical dynamical system is learned from the training data we would like to use it to support time series prediction on future time series. Given the initial observations $\mathbf{Y}_{obs}$ and an arbitrary time index t modeling a sequence of future times, our objective is to predict $\hat{\mathbf{y}}$ at time t.

To support the prediction inference, we need the following steps:

Step 1. Split $\mathbf{Y}_{\text{obs}}$ and t into windows.

Step 2. For windows that do not contain t, compute $\boldsymbol{\gamma}$s through mean estimation function h and feed them into Kalman Filter algorithms(See *Kalman_Filter* in Appendix A.1) to infer the most recent hidden state $\mathbf{z}_k$ where k is the index of the last window that does not contain t.

Step 3. Get $\boldsymbol{\gamma}_{k+1} = CA\mathbf{z}_k$ from $\mathbf{z}_{k+1} = A\mathbf{z}_k$ and $\boldsymbol{\gamma}_{k+1} = C\mathbf{z}_{k+1}$.

Step 4. If t is in window k+1 use observations $(\mathbf{y}_{k+1}, t_{k+1})$ in window k+1 and $\boldsymbol{\gamma}_{k+1}$ to make the prediction, where $\hat{\mathbf{y}} = \boldsymbol{\gamma}_{k+1} + K(t, t_{k+1})K^{-1}(t_{k+1}, t_{k+1})(\mathbf{y}_{k+1} - \boldsymbol{\gamma}_{k+1})$; otherwise find out the window index i where t belongs to. The prediction at t is $\hat{\mathbf{y}} = CA^{i-k}\mathbf{z}_k$.

The prediction algorithm is summarized in Algorithm 2.

Algorithm 2. Prediction with the Hierarchical Dynamical System

```
Split $\mathbf{Y}_{\text{obs}}$ and $t$ into windows.
Find all $k$ windows that do not contain $t$.
Estimate $\{\gamma_i\}_{i=1}^{k}$ using mean estimation function $h$ in these $k$ windows.
// Compute $z_k$ by $Kalman\_Filter$ algorithm.(See Appendix A.1)
$\mathbf{z}_k = Kalman\_Filter(\{\gamma_i\}_{i=1}^{k}, A, C, R, Q, \pi_1, V_1)$
if $t$ is in window $k + 1$ then
    $\boldsymbol{\gamma}_{k+1} = C\mathbf{z}_{k+1} = CA\mathbf{z}_k$
    // Observations in window $k + 1$ are $(\mathbf{y}_{k+1}, t_{k+1})$
    $\hat{\mathbf{y}} = \boldsymbol{\gamma}_{k+1} + K(t, t_{k+1})K^{-1}(t_{k+1}, t_{k+1})(\mathbf{y}_{k+1} - \boldsymbol{\gamma}_{k+1})$
else
    $\hat{\mathbf{y}} = CA^{i-k}\mathbf{z}_k$
end if
return $\hat{\mathbf{y}}$
```

4 Experimental Results

We have tested our approach on time series data obtained from electronic health records of approximately 4,500 post-surgical cardiac patients stored in PCP database [12–14]. To test the performance of our prediction model, we randomly selected 1000 patients with the *Complete Blood Count*(CBC) panel test[1] whose hospitalization is longer than 10 days. We selected six tests from the CBC panel to learn the time series models, and applied them to time series prediction tasks. The six tests, their means and standard deviations are listed below:

- *White Blood Cell* (WBC) a count of the total number of white blood cells in a person's sample of blood. Mean: 11.98($\times 10^9$); standard deviation: 6.06.
- *Mean Corpuscular Hemoglobin* (MCH) is a calculation of the average amount of oxygen-carrying hemoglobin inside a red blood cell. Mean: 33.87(pg); standard deviation: 0.81.
- *Mean Corpuscular Hemoglobin Concentration* (MCHC) is a calculation of the average concentration of hemoglobin inside a red cell. Mean: 30.53(g/dL); standard deviation: 1.81.
- *Mean Corpuscular Volume* (MCV) is a measurement of the average size of patient's red blood cell. Mean: 90.12(fL); standard deviation: 4.56.
- *Platelet* (PLT) count is the number of platelets in a given volume of blood. Mean: 201.15($\times 10^9$); standard deviation: 126.01.
- *Red cell Distribution Width* (RDW) is a calculation of the variation in the size of red blood cells. Mean: 16.68(%); standard deviation: 2.64.

To evaluate the performance of our hierarchical dynamical system (HDS-GPLDS) approach we used the five-fold cross validation approach to split the examples into the training and testing sets, such that 200 examples formed the test data, and 800 training examples were used to vary the size of the training set from 100 to 800 in increments of 100 examples and reported the average results. Since the CBC panel is typically ordered once or just a few times a day, we used the default Gaussian process window size of seven days. The covariance function is the combination of the mean reverting and the periodic functions([15]). The mean reverting function forces the process to approach the long term mean, but at the same time permits temporary deviations from the mean corresponding to episodic events or complications. The periodic function can capture the fluctuation within the short period of time and keep the variation of different values within a reasonable range. The covariance function is defined as:

$$K(\mathbf{t}, \mathbf{t}') = \sigma_1 \exp(\alpha_1 |\mathbf{t} - \mathbf{t}'|) + \sigma_2 \exp(\alpha_2 \sin^2 \left[\frac{\omega}{2\pi} (\mathbf{t} - \mathbf{t}') \right]) \qquad (6)$$

where $\Theta = (\sigma_1, \alpha_1, \sigma_2, \alpha_2, \omega)$ are the parameters optimized from the data.

[1] CBC test is used as a broad screening test to check for such disorders as anemia, infection, and many other diseases.

We compared the HDS-GPLDS predictions to four other methods:

- Linear dynamical system (LDS) trained on the entire time series with a three-hour period discretization. We applied Gaussian process interpolation to fill the missing values [16].
- Window-based linear dynamical system(WLDS). This model is different from the LDS model. It splits the time series first into windows the same way as HDS-GPLDS and, after that, it trains an LDS using the last observation in each window.
- Standard Gaussian Process regression (GP) with covariance function defined in Eq. 6.
- Nonlinear dynamical system using GPIL algorithm for inference and learning with the Gaussian kernel covariance function for both the transition and observation models [8].

We evaluated and compared the performances of the different methods by calculating the Root Mean Square Error (RMSE) on the test predictions. More specifically, the RMSE is defined as follows:

$$RMSE = \left[n^{-1} \sum_{i=1}^{n} |y_i - \hat{y}_i|^2 \right]^{1/2}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value and n is the number of data points. The results of RMSE on the six lab tests from the CBC panel (for gradually increasing training and test data sets as described above) are summarized in Figure 2.

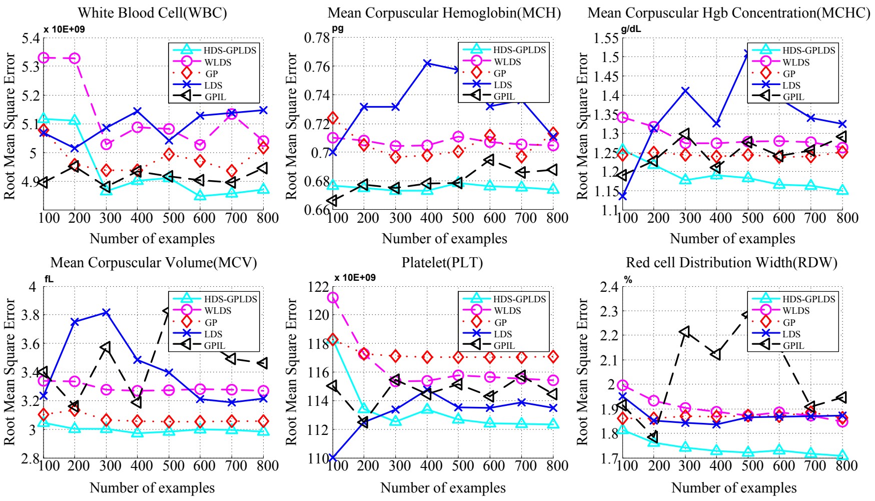

Fig. 2. Root Mean Square Error (RMSE) on CBC test samples

4.1 Discussion

The results of our experiments (Figure 2) show that our hierarchical dynamical system (HDS-GPLDS) outperforms all other methods in terms of prediction errors on all six CBC lab tests. One of the advantages of our method is that its prediction error is small even when it is trained on a small number of patients and observations. In the following we discuss the obtained results in more depth.

First, by comparing HDS-GPLDS, GP and LDS, GPIL, we can see that methods with continuous functions (HDS-GPLDS, GP) are much stable than discretized methods (LDS, GPIL). Their performances are gradually improved with the increase of training examples. The reason is that the discretized methods do require any discretization or interpolations for the irregular sampled dataset, which may bring inaccuracy in the training and prediction phases.

Second, HDS-GPLDS/GP outperform WLDS/LDS. The reasons are (1) the values from patients' tests are always around a normal range plus some variation. The combination of the mean-reverting and periodic functions captures this phenomenon: the mean reverting function forces the predicted values within a normal range and the periodic function allows the fluctuation and variation flexibility. However, WLDS and LDS cannot capture these variations due to the linearity embodied in their linear models. (2) WLDS and LDS solve the multi-step prediction problem by constructing a single model from past observations and predict future values iteratively. Since they use predictions from the past, both are very susceptible to the error accumulation: errors generated in the history are propagated into future predictions. In contrast to this the HDS-GPLDS/GP make the multi-step predictions directly and hence do not suffer this problem.

Third, comparing to other methods, HDS-GPLDS does not require a large number of training examples, and it can still perform well for small training data sizes. In contrast to this, error rates for other methods are initially high and they are decreased by a large amount when more examples are used to train them. Stable performance on small training data is very important in practice.

Fourth, comparing GP and HDS-GPLDS, we can see that the HDS-GPLDS is much better than the GP approach. The results show that a single constant mean is not enough in the complex time series setting. The evolution of mean variables in the consequent windows is modeled by a linear dynamical system, which expresses a stronger descriptive ability. During the prediction phase, the predicted mean is used by the GP to make a more accurate prediction.

5 Conclusion

In this work, we have presented a novel two-layer hierarchical dynamical system for multi-step time series prediction. Comparing with the traditional linear state space systems and modern Gaussian process regression, this novel system is (1) more robustness to irregular sampling; (2) more robust for small training data sizes; (3) more accurate in making the long-term multi-step predictions. Experimental results on real world clinical data from electronic health records systems demonstrated that our prediction model achieves errors that are statistically significantly lower than errors of other state of the art approaches. The limitation of out current work is the analysis of univariate time-series model. In the future, we plan to study and model dependences among multiple time series, as well as, extensions to switching-state [17] and controlled [18, 19] dynamical systems.

Acknowledgments. This work was supported by grants 1R01LM010019-01A1 and 1R01GM088224-01 from the NIH. Its content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

1. Combi, C., Keravnou-Papailiou, E., Shahar, Y.: Temporal information systems in medicine. Springer Publishing Company (2010) (Incorporated)
2. Kalman, R.: Mathematical description of linear dynamical systems. Journal of the Society for Industrial & Applied Mathematics, Series A: Control 1, 152–192 (1963)
3. Rasmussen, C., Williams, C.: Gaussian processes for machine learning, vol. 1. MIT Press, Cambridge (2006)
4. Rasmussen, C., Kuss, M., et al.: Gaussian processes in reinforcement learning. Advances in Neural Information Processing Systems 16 (2004)
5. Deisenroth, M., Huber, M., Hanebeck, U.: Analytic moment-based gaussian process filtering. In: Proceedings of the 26th ICML, pp. 225–232. ACM (2009)
6. Wang, J., Fleet, D., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Transactions on PAMI 30, 283–298 (2008)
7. Ko, J., Fox, D.: Learning gp-bayes filters via gaussian process latent variable models. Autonomous Robots 30, 3–23 (2011)
8. Turner, R., Deisenroth, M., Rasmussen, C.: State-space inference and learning with gaussian processes. In: Proceedings of 13th AISTATS, vol. 9, pp. 868–875 (2010)
9. Nguyen-tuong, D., Peters, J.: Local gaussian process regression for real time online model learning and control. In: Advances in NIPS (2008)
10. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1–38 (1977)
11. Ghahramani, Z., Hinton, G.: Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, University of Totronto (1996)
12. Hauskrecht, M., Valko, M., Batal, I., Clermont, G., Visweswaran, S., Cooper, G.: Conditional outlier detection for clinical alerting. In: AMIA Annual Symposium Proceedings, vol. 2010, pp. 286–290. AMIA (2010)
13. Hauskrecht, M., Batal, I., Valko, M., Visweswaran, S., Cooper, G.F., Clermont, G.: Outlier detection for patient monitoring and alerting. Journal of Biomedical Informatics 46(1), 47–55 (2013)
14. Valko, M., Hauskrecht, M.: Feature importance analysis for patient management decisions. In: 13th International Congress on Medical Informatics, NIH Public Access, pp. 861–865 (2010)
15. Bibbona, E., Panfilo, G., Tavella, P.: The ornstein–uhlenbeck process as a model of a low pass filtered white noise. Metrologia 45, S117 (2008)
16. Gibbs, M., MacKay, D.: Efficient implementation of gaussian processes (1997)

17. Kim, C.J.: Dynamic linear models with markov-switching. Journal of Econometrics 60, 1–22 (1994)
18. Hauskrecht, M., Fraser, H.: Modeling treatment of ischemic heart disease with partially observable markov decision processes. In: Proceedings of the AMIA Symposium, pp. 538–542. American Medical Informatics Association (1998)
19. Kveton, B., Hauskrecht, M.: Solving factored mdps with exponential-family transition models. In: 16th International Conference on Automated Planning and Scheduling, pp. 114–120 (2006)