Dynamical Systems as Feature Representations for Learning from Temporal Data

Peter Tino
School of Computer Science
The University of Birmingham
UK

Yuan Shen (NTU), Krasimira Tsaneva-Atanasova (University of Exeter)
Learning in the Model Space Framework

- Sparsely and irregularly sampled time series data
  → cannot use learning models operating on numerical vectors of fixed dim.
  → cannot use state space models, e.g. such as RNN (including LSTM), HMM, Kalman Filters etc.

- Each data item (in our case a time series) is represented by a model that ”explains” it

- Learning is formulated in the space of models (function space)
Inferential Model

- **Non-parametric model class** (e.g. state space model)
  - don’t know the generative mechanism behind the data
  - flexible enough to represent variety of data items
  - sufficiently constrained to avoid overfitting - e.g. ESN
  - need more data, more frequently sampled

- **Parametric model class** (e.g. parametrized pathway model)
  - naturally include prior knowledge on the generative mechanism behind the data
  - interpretability of the predictive mechanism
Issues

- The model class is usually too complex w.r.t. the amount of observed data.

- Point model estimate is insufficient. Need to treat model uncertainty in a consistent manner:
  - uncertainty due to noisy observations
  - uncertainty due to insufficient amount of observations

- Each time series will be represented through posterior over models.

- Possible mismatch of model uncertainty levels between training and testing phases
More Formally...

(Binary) classification task:
$N$ labelled univariate or multivariate time series
$\{(\mathcal{Y}^k, c^k) : k = 1, ..., N\}$.

We do not assume all time series are collected on a fixed, regular time grid.

Each time series $\mathcal{Y}^k$ is accompanied with observation times $t^k = \{t^k_i\}_{i=1}^{L_k}$ at which the observations $\mathbf{Y}^k = \{\mathbf{y}^k_i\}_{i=1}^{L_k}$ are collected.

Predict a label for a new time series $\mathcal{Y} = (\mathbf{Y}, \mathbf{t})$ of length $L$. 

Tino, Shen, Tsaneva-Atanasova () Dynamical Systems as Feature Representatio
**Continuous-time deterministic dynamical system** - mathematically represented as a multivariate Ordinary Differential Equation (ODE),

\[
\frac{dx_t}{dt} = f(x_t; \theta),
\]

\(x_t \in X \subset \mathbb{R}^D\) state vector at time \(t\)
parameters \(\theta\) (include initial state \(x_0\)).
Stochastic dynamical system - can be considered ODE driven by a multivariate random process parameterized by covariance matrix $\Sigma$.

Equivalent to adding a Gaussian noise to the drift.

Mathematically - multivariate Stochastic Differential Equation (SDE):

$$dx_t = f(x_t; \psi) \, dt + \Sigma \, db_t$$

vector $b_t$ collects $D$ independent standard Brownian motions.
Observations

Observations \( Y = \{ \mathbf{y}_1, \mathbf{y}_2, \ldots \} \), \( \mathbf{y}_t \in \mathbb{R}^d \) from the underlying dynamical system’s trajectory \( \mathbf{x}_t \) are obtained through a measurement function \( \mathbf{h} \):

\[
y_i = \mathbf{h}(\mathbf{x}_{t_i}) + \epsilon_{t_i} \quad \text{for} \quad i = 1, 2, \ldots
\]

Frequently, the readout \( \mathbf{h} \) is a set of indicator functions which specify a subset of state variables that are directly observed.

Observation noise \( \epsilon_t \) - often assumed i.i.d. Gaussian with zero mean and error covariance matrix \( \mathbf{R} \) (determined e.g. form prior knowledge or learned from the data).
Each Observation Sequence Represented as a Model

Given a time series \( \mathcal{Y} = \{(t_i, y_i)\}_{i=1}^L \), ML estimate of \( \theta \) - maximize

**ODE:**
\[
p(Y|\theta, t; R) = \prod_{i=1}^L \mathcal{N}(y_i|\mathbf{x}_t(\theta), t_i, R)
\]

**SDE:**
\[
p(Y|\theta, t, R) = \mathbb{E}_{\mathbf{x}_t|\theta} \left[ \prod_{i=1}^L \mathcal{N}(y_i|\mathbf{x}_t, t_i, R) \right]
\]

Ignores uncertainty around the model estimate. When only noisy and/or sparse data are available.

Any point estimate of the model parameter is not a sufficient representation of the partially observed dynamical system.
Each Sequence Represented as Model Posterior

likelihood: $p(Y|\theta, t, R)$  
prior: $p(\theta)$

$$p(\theta|Y, R) = p(\theta|Y, t, R) \propto p(Y|\theta, t, R) \cdot p(\theta)$$

In most cases, computation of the normalizing term is analytically not tractable. The posterior needs to be approximated e.g. using a finite grid in the parameter space or by sampling/variational methods.
LiMS Classifier

\[ p(c|\mathcal{Y}) = \int d\mathbf{x}_t \int d\theta \ p(c|\mathcal{Y}, \mathbf{x}_t, \theta) \ p(\mathbf{x}_t, \theta|\mathcal{Y}, \mathbf{R}). \]

**Key point of LiMS** - all the relevant information in \((\mathcal{Y}, \mathbf{x}_t, \theta)\) for the class label prediction can be collapsed into the model \(\theta\).

\((\mathcal{Y}, \mathbf{x}_t, \theta) \rightarrow \theta\) and so \(p(c|\mathcal{Y}, \mathbf{x}_t, \theta) \rightarrow p(c|\theta)\).

Assume no additional relevant information for the classification could be extracted from observation noise or observation times.
LiMS Classifier

\[ p(c|\mathcal{Y}) = \int \! dx_t \int \! d\theta \ p(c|\theta) \ p(x_t, \theta|\mathcal{Y}, R) \]

\[ = \int \! d\theta \ p(c|\theta) \int \! dx_t \ p(x_t, \theta|\mathcal{Y}, R) \]

\[ = \int \! d\theta \ p(c|\theta) \ p(\theta|\mathcal{Y}, R) \]

\[ = \mathbb{E}_{\pi(\theta)}[p(c|\theta)] \]

\[ = q(c|\pi). \]

Classifier \( q(c|\pi) \) operates on posterior distributions \( \pi \), but is formulated based on classifier \( p(c|\theta) \) operating on individual models.
Training the Classifier

Training set: \( V = \left\{ (\pi_1(\theta), c_1), \ldots, (\pi_N(\theta), c_N) \right\} \)

Minimize the Cross Entropy Error

\[
E_{\pi}(w | V) = - \sum_{k=1}^{N} \log q(c_k | \pi_k(\theta)) \\
= - \sum_{k=1}^{N} \log \mathbb{E}_{\pi_k(\theta)}[p(c_k | \theta; w)]
\]

Using grid in the model space:

\[
\hat{E}_{\pi}(w | V) = - \sum_{k=1}^{N} \log \left( \sum_{n=1}^{N^G_{\theta}} \pi_k^n \cdot p(c_k | \theta_G^n; w) \right),
\]

where \( \pi_k^n \) denotes the normalized posterior weight on the \( n \)-th grid point for the \( k \)-th posterior.
Alternatives...

- Probability Product Kernel (PPK)

\[
K_{PPK}(\pi_1, \pi_2) = \int_{\Theta} \pi_1^\alpha(\theta) \cdot \pi_2^\alpha(\theta),
\]  

- Kernel Mean Embedding (KME).

\[
K_{KME}(\pi_1, \pi_2) = \langle \mu_{\pi_1}, \mu_{\pi_2} \rangle_H
= \int_{\Theta} d\theta \int_{\Theta} d\eta \ \pi_1(\theta) \cdot \pi_2(\eta) \cdot k(\theta, \eta),
\]

- A vector classifier (MAP) simply using the maximum a posteriori estimates of \( \theta \) as feature vectors;

- LiMS, PPK, KME and MAP classifiers can all be implemented within the framework of Kernel Logistic Regression.
GnRH signaling model - an example

- The model describes how the pulse-like Gonadotropin-releasing hormone (GnRH) signal regulates the transcription and secretion of Gonadotropin hormones (GSU);
- How the transcription and secretion of GSU respond to variation of GnRH signal’s pulse frequencies differs between normal and pathological conditions;
- In GnRH signaling model, this is controlled by the setting of two transcription factor parameters;
- The concentration of secreted GSU is irregularly and/or sparsely measured over a number of time periods at different pulse frequencies;
- GnRH signaling model is a system of 11 coupled ordinary differential equations (ODE).
GnRH signaling model (cont.)

3 compartments along the signalling pathway

driving input - GnRH signal

measured output - GSU
Two classes of GnRH signalling models

“normal” subjects - bell-shaped frequency-response relationship (Blue Diamonds)

“abnormal” subjects - simple frequency dependency of response (Red Disks).
Experiment Setup

200 labelled GnRH models for training and testing by randomly sampling 200 parameter vectors \( \theta_{\text{GnRH}} = (\log K_{d_{TF_1}}, \log K_{d_{TF_2}}, t_p) \) of the GnRH model.

Each of the parameters sampled from its Gaussian distribution truncated to the permissible range.

<table>
<thead>
<tr>
<th>parameter</th>
<th>mean</th>
<th>variance</th>
<th>lower bound</th>
<th>upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log K_{d_{TF_1}} )</td>
<td>-1.6</td>
<td>0.2</td>
<td>-2.0</td>
<td>0.2</td>
</tr>
<tr>
<td>( \log K_{d_{TF_2}} )</td>
<td>-1.1</td>
<td>0.2</td>
<td>-1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>( t_p )</td>
<td>7.5</td>
<td>0.8333</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>
For each trajectory $x_t$, generate 15 observation sets using different observation noise level $\sigma$ and inter-sample interval ($ISI$).

All observation sets are then grouped into 5 collections $G^a, a = 1, \ldots, 5$. $G^1$ and $G^5$ induce maximal and minimal model uncertainty, respectively.

Classifier trained on $G^a$ and tested on $G^b$ yields accuracy $\text{Acc}^{(a),b}$. 
Results

Accuracies $\text{Acc}^{(a), a}$, $a = 1, 2, ..., 5$.

Upper-left panel: LiMS
Upper-right panel: KME
Lower-left panel: PPK
Lower-right: MAP
Results (cont.)

LiMS framework

<table>
<thead>
<tr>
<th>Kernel width $\rho$</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tempering parameter $\alpha$</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>2^-5</td>
<td></td>
</tr>
<tr>
<td>2^-4</td>
<td></td>
</tr>
<tr>
<td>2^-3</td>
<td></td>
</tr>
<tr>
<td>2^-2</td>
<td></td>
</tr>
<tr>
<td>2^-1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Tino, Shen, Tsaneva-Atanasova

Dynamical Systems as Feature Representations for Learning from Temporal Data
Results (cont)

Figure: Four training sets $G^1,\ldots, G^4$: ideal case $\text{Acc}^{(5),5}$ (Black □), $\text{Acc}^{(a),5}$ (Red △), and $\text{Acc}^{(a),a}$ (Black ◊).
Results (cont)

Table: **Mean classification accuracy** (± standard deviation). Classifiers trained on $G^5$ and tested on $G^b$, $b = 1, ..., 5$. Bold font - LiMS classifier outperforms the corresponding classifier with statistical significance ($p < 0.05$, one-sided sign rank test).

<table>
<thead>
<tr>
<th>Classifier</th>
<th>$G^1$</th>
<th>$G^2$</th>
<th>$G^3$</th>
<th>$G^4$</th>
<th>$G^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiMS</td>
<td>0.55 ± 0.03</td>
<td>0.81 ± 0.04</td>
<td>0.90 ± 0.02</td>
<td>0.90 ± 0.01</td>
<td>0.91 ± 0.02</td>
</tr>
<tr>
<td>KME</td>
<td>0.53 ± 0.03</td>
<td>0.52 ± 0.01</td>
<td>0.80 ± 0.05</td>
<td>0.87 ± 0.03</td>
<td>0.90 ± 0.03</td>
</tr>
<tr>
<td>PPK</td>
<td>0.59 ± 0.06</td>
<td>0.68 ± 0.10</td>
<td>0.66 ± 0.11</td>
<td>0.64 ± 0.07</td>
<td>0.88 ± 0.04</td>
</tr>
<tr>
<td>MAP</td>
<td>0.50 ± 0.09</td>
<td>0.70 ± 0.08</td>
<td>0.87 ± 0.02</td>
<td>0.88 ± 0.02</td>
<td>0.90 ± 0.02</td>
</tr>
</tbody>
</table>
Results (cont.)

Table: Mean classification accuracy (± standard deviation). Classifiers trained on data sets consisting of the time series from all of $G^1...G^5$ and tested on $G^b$, $b = 1, ..., 5$. Bold font - LiMS classifier outperforms the corresponding classifier with statistical significance ($p < 0.05$, one-sided sign rank test).

<table>
<thead>
<tr>
<th>Classifier</th>
<th>$G^1$</th>
<th>$G^2$</th>
<th>$G^3$</th>
<th>$G^4$</th>
<th>$G^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiMS</td>
<td>0.51 ± 0.02</td>
<td>0.69 ± 0.01</td>
<td>0.83 ± 0.01</td>
<td>0.85 ± 0.02</td>
<td>0.87 ± 0.04</td>
</tr>
<tr>
<td>KME</td>
<td>0.51 ± 0.03</td>
<td>0.66 ± 0.02</td>
<td>0.80 ± 0.02</td>
<td>0.81 ± 0.03</td>
<td>0.80 ± 0.04</td>
</tr>
<tr>
<td>PPK</td>
<td>0.51 ± 0.03</td>
<td>0.63 ± 0.03</td>
<td>0.71 ± 0.06</td>
<td>0.64 ± 0.07</td>
<td>0.60 ± 0.06</td>
</tr>
<tr>
<td>MAP</td>
<td>0.49 ± 0.02</td>
<td>0.63 ± 0.03</td>
<td>0.77 ± 0.04</td>
<td>0.80 ± 0.03</td>
<td>0.85 ± 0.04</td>
</tr>
</tbody>
</table>
Conclusion

- Compared to PPK, KME, and MAP, the performance of LiMS classifier is quite robust with respect to kernel width parameter variation.

- LiMS classifier outperforms other three classifiers in the case of high model uncertainty levels.

- LiMS classifier is robust to the overall mismatch of model uncertainty levels between the training and testing phase.
Interested?


