Two recent machine learning methods for time series

Alfredo Cuesta Infante
alfredo.cuesta@urjc.es
School of Computer Science, URJC (Spain)

Workshop on Time Series @ Universidad de Zaragoza
March 30, 2017

Agenda

Markov Switching Copula Models

Long-Short Term Memories for Time Series

Markov Switching Copula Models

Long-Short Term Memories for Time Series

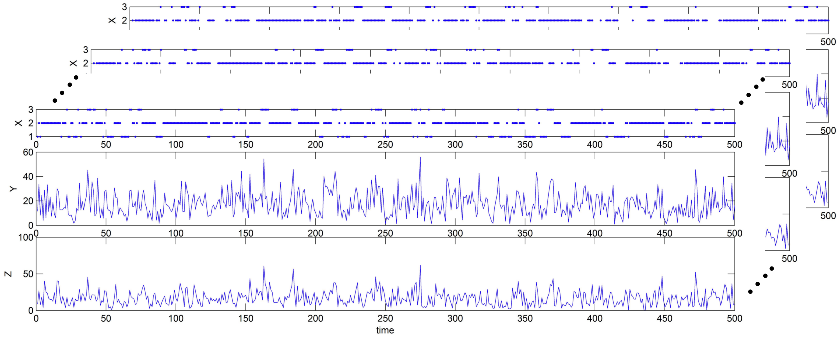
Motivation

Longitudinal data

- Multiple variables repeatedly measured over long periods of time.

  e.g. On-line education platforms where students interaction is monitored over time → predict drop out, sequential clustering.

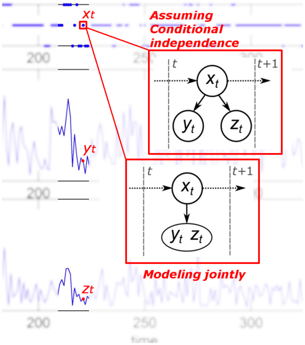
  e.g. Life signals such as ABP or ECG from patients in Intensive Care Units (ICU) are recorded continuously → predict acute hipotensive episodes.



Motivation

Generative Switching Models

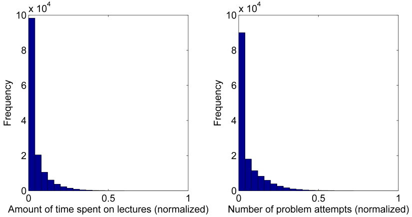
- A modeling paradigm in which observations are sampled from a distribution that is chosen from a range according to a latent variable, or hidden state, at each time slice t.
- The **goal** is to construct the joint distribution of states and observations.
- The problem is more difficult if there are two or more observations at a time, a.k.a. *covariates*.
- The simplest approach is to assume conditional independence.
- Otherwise, continuous covariates are usually modeled jointly as multivariate normal (MVN), or...
- Dynamic Bayesian Networks (DBN) after being discretized.
- **Real problems are not MVN...**



Motivation example

EdX students sequential clustering

- 2000 students and 2 distinctive features:
  - Amount of time spent viewing lecture videos
  - Number of problems attempted
- The distribution of each covariate is clearly non-normal



!! **We can do better with Copula functions.**

- The copula density function is what we need to reconstruct the joint PDF given the independence PDF.

$$f(x_1, x_2, \dots x_n) = c\left(F_1(x_1), F_2(x_2), \dots, F_n(x_n)\right) \cdot \prod_{i=1}^{n} f_i(x_i)$$

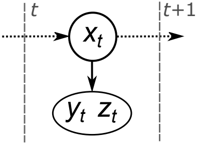
Learning Markov Switching Copula Models (MSCM)

MSCM proposed

- **Individual behaviour:** Weibull distributions
- **Dependence structure:** Gaussian copula
- **Learning:** Estimate both transition and observation PDFs for every possible state.

Parameters

- One PDF across the initial state
- One Transtion matrix
- One Weibull PDF for each covariate and each state
- One covariance matrix for each state



x_t	y_t	z_t	Copula
1	$p_y(\cdot \mid \ell_y^{(1)}, s_y^{(1)})$	$p_z(\cdot \mid \ell_z^{(1)}, s_z^{(1)})$	$c(\cdot, \cdot \mid \Sigma^{(1)})$
2	$p_y(\cdot \mid \ell_y^{(2)}, s_y^{(2)})$	$p_z(\cdot \mid \ell_z^{(2)}, s_z^{(2)})$	$c(\cdot, \cdot \mid \Sigma^{(2)})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
N_x	$p_y(\cdot \mid \ell_y^{(N_x)}, s_y^{(N_x)})$	$p_z(\cdot \mid \ell_z^{(N_x)}, s_z^{(N_x)})$	$c(\cdot, \cdot \mid \Sigma^{(N_x)})$

Learning MSCM

Gibbs sampling

1. Sample the initial state, transition matrix, and Weibulls' scales.

Par.	Prior	Likelihood	Posterior $\propto$	
π	$p_D(\pi; \underline{1})$	$p_M(x_1; \pi)$	$p_D(\pi; \underline{1} + I(x_1))$	(a)
$\pi_{(i)}$	$p_D(\pi_{(i)}; \underline{1})$	$\prod_{t=1}^{T} p_M(x_{t+1}; \pi_{(i)})$	$p_D(\pi_{(i)}; \underline{1} + K(x_t = i))$	(b)
ℓ_i	$p_{IG}(\ell_i; \alpha, \beta)$	$\prod_{t=1}^{T} p_W(y_t; \ell_i, s_i)^{\mathbf{I}(x_t=i)}$	$p_{IG}(\ell_i; \alpha', \beta')$, $\alpha' = \alpha + n_i$, $\beta' = \beta + \sum_{t=1}^{T} \mathbf{I}(x_t = i) y_t^{s_i}$	(c)

2. Sample Weibulls' shapes

   BUT Weibull's shape has has not conjugate PDF.

3. Sample the covariance matrix

   BUT We cannot use Inverse Wishart directly because covariance matrix of any Gaussian bivariate copula is constrained to be $\frac{1}{12}\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$, where ρ is the correlation between the covariates of the copula.

4. Sample the Markov chain of states

Learning MSCM

Solution to Step 2: Sampling Weibull's shape

Par.	Prior	Likelihood	Posterior $\propto$	
s_i	$p_G(s_i;\zeta,\xi)$	$\prod_{t=1}^{T} p_W(y_t;\ell_i,s)^{\mathbf{I}(x_t=i)}$	$f_E \cdot p_G(s,\zeta',\xi')$, $\zeta' = \zeta + n_i$, $1/\xi' = \frac{1}{\xi} + \sum_{t=1}^{T} \ln y_t^{\mathbf{I}(x_t=i)}$, $f_E = \exp\left(-\frac{1}{\ell_i}\sum_{t=1}^{T} \mathbf{I}(x_t=i) y_t^{s_i}\right)$	(d)

!! Factor $f_E \in [0,1]$ and is monotone, so Posterior keeps the shape of p_G.

1. Find the interval where the posterior has its maximum.
2. Obtain an empirical CDF of the posterior.
3. Use its quantile function to sample from it.

Solution to Step 3: Sampling the covariance matrix

- Hoff (2007) proposes to use the Inverse Wishart *as usual*, and then, for the bivariate case, $\rho = \Sigma_{[1,2]}/\sqrt{\Sigma_{[1,1]}\Sigma_{[2,2]}}$

Step 4: Updating the hidden Markov chain

- $p(x_1 = i|\cdots) \propto \pi_i \cdot p_O(y_1, z_1|x_1 = i) \cdot p(y_{2:N_t}, z_{2:N_t}|x_1 = i)$
- $p(x_t = j|x_{t-1}=i,\cdots) \propto a_{ij} \cdot p_O(y_t, z_t|x_t=j) \cdot p(y_{t+1:N_t}, z_{t+1:N_t}|x_t = j)$

Experiment

Data set

- EdX students

Results

- The hidden state is the cluster estimated at every time slice.
- We assume 3 possible states.
- We compare the MSCM against a Multivariate Normal Markov Switching Model (MVN-MSM)

EdX students sequential clustering	
Model	Neg. Loglikelihood
MSCM	$\boxed{4.39 \times 10^3}$
MVN-MSM	7.62×10^3

Conclusions

- If we don't assume conditional independence,
  using copula functions give us great flexibility
- If only two covariates, there is a wide range of copula families,
  with different modeling properties.
- If there are more, gaussian copula is a good option.
  Also we can reduce dimensionality and go for the bivariate solution.

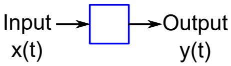
Markov Switching Copula Models

Long-Short Term Memories for Time Series

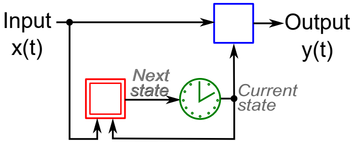
Dealing temporal data with Recurrent Neural Networks (RNN)

- The history of inputs is *coded* in the **state**.
- The current input and state are used to compute the output BUT ALSO the next state.
- Next state is used in the next time step.
- We can see the state as a *carry* that goes from one classic neuron to the next in an array of them.

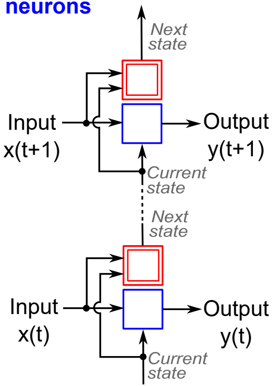
Classic Neuron



Recurrent Neuron

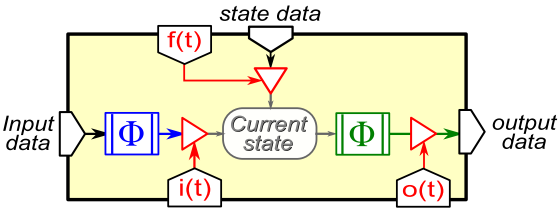


Recurrent Neuron unfolded as classic neurons



Long-Short Term Memories (LSTM)

- **x** **Problem with RNN** Data vanishes as goes through the array.
  $\rightarrow$ Very old data is *forgotten*.
- ✓ At every time step t, LSTM neurons control what to:
  - admit $\rightarrow$ **input** gate $i(t)$
  - forget $\rightarrow$ **forget** gate $f(t)$
  - transmit $\rightarrow$ **output** gate $o(t)$
- Input and Forget gates modify the state
- Output gate modify the result.
- In addition:
  - Input data at time t, $\mathbf{X}(t)$, is transformed into $\Phi(\mathbf{X}) \in [-1, 1]$.
  - Output data, $y(t)$ comes from the current state, also transformed with Φ.
  - In both, $\Phi = \tanh$



- These gates are indeed functions that depend on current and past data.

Long-Short Term Memories (LSTM)

Input to LSTM

- The input at time t is concatenated with the output at time $t-1$

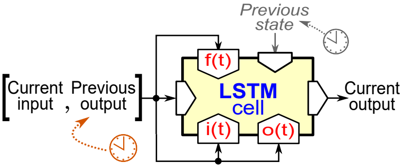
$$\mathbf{X}(t) = [x(t), y(t-1)]$$

Current state

- It combines the transformed input $\Phi(\mathbf{X})$ weighted by the input gate, with the previous state weighted by the forget gate.

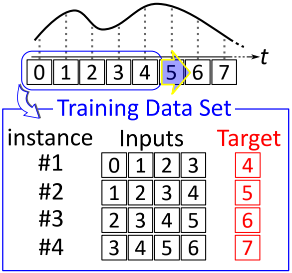
Output of the LSTM

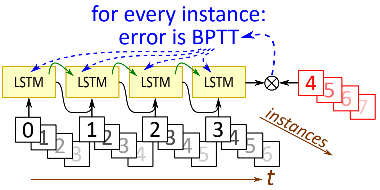
- Current state transformed with Φ, and then weighted by the output gate.



Using LSTM in time series

- Forecasting can be seen as a regression problem in which we use a sliding window of size T that creates instances for training.
  - $T-1$ values of the time series as input
  - Value at $t=T$ as output
- We feed a LSTM neuron with one instance after another and compare the outcome with the target.
  - NOTICE that every instance has $T-1$ values, and goes into the LSTM sequentially.
  - HOWEVER, it is easier to understand if we unfold the LSTM.
- The error is backpropagated thought time (BPTT)





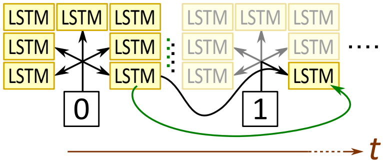
Using LSTM in time series

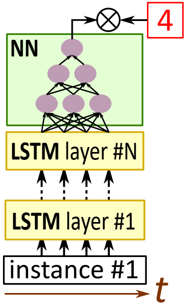
Getting bigger

- We can use many different LSTM cell for each time step.
- The intuition is that every LSTM will learn something different

Getting deeper

- Arrange a number of LSTM cells in a *layer*
- Then stack N layers
- Finally use a dense neural network to produce the outcome for each instance.





Conclusions

- HMM, and switching variants, have some limitations
  - ✗ Usually we have to assume a discrete, and small, number of states.
  - ✗ If there are N different states, the transition matrix has size N^2.
  - ✗ The current state depends on the previous. It is possible to extend, but is a pain in the neck.
- LSTMs can let the information pass through cells ans time. Hence long term dependencies are better captured.
- There is a growing interest in this area, mainly due to:
  - ✓ Excellent results
  - ✓ Many open source libraries.

One of the most relevant and successful is TensorFlow, by Google !

Thank you

Two recent machine learning methods for time series

Alfredo Cuesta Infante

Univ. Rey Juan Carlos, Spain
alfredo.cuesta@urjc.es

Universidad
Rey Juan Carlos

Workshop on Time Series

Univ. Zaragoza – March. 30th, 2017