LINMA 2120 - Seminars in Applied Mathematics
High dimensional time-series prediction with missing values

Quentin Lété

November 2017

1 Introduction

Let us first give a definition of time series to explain the formal context in which we work.

Definition 1. Time series A time series is a sequence of data indexed by the time.

A time series is thus a realization of a stochastic process which can be seen as a family of random variables indexed by the time.

We also define here a property of a stochastic process that will be useful later : the Markow property.

Definition 2. Markov property We say that $(Y_t)_{t\geq 0}$ is a Markov chain if for all $t > 0$

$$\pi(y_t|y_{1:t-1}) = \pi(y_t|y_{t-1})$$

where π is the probability.

This means that all the information about the past is completely carried out in the Y_t at each step. With Markov chains, the joint distribution of observations take a fairly simple form :

$$\pi(y_{1:t}) = \pi(y_1) \cdot \prod_{j=2}^{2} \pi(y_j|y_{j-1})$$

Classical methods

In this section, we present classical methods used for time series forecasting. These classical methods include AR (autoregressive) models and DLM (dynamic linear models) and are used for prediction. Let us explain sequentially these models.

Autoregressive models

As explained in [1], the idea of this model is to represent each observation as a noisy linear combination of previous observations. Formally, if X_t is the measurement at time t, the AR(p) model prametrized by the lag p and the coefficient vector $\alpha \in \mathbb{R}^p$ can be written as

$$X_t = \sum_{k=1}^{p} \alpha_k X_{t-k} + \epsilon_t$$

where $\{\epsilon_t\}_t \in \mathbb{Z}$ is a white noise.

This moedel is motivated by a thorem due to Wold that states that a stationary process can be represented by a MA(∞) model, that is

$$X_t = \sum_{i=1}^{\infty} \beta_i \epsilon_{t-i} + \epsilon_t$$

where $\sum_{i=1}^{\infty} \beta_i < \infty$ and $\{\epsilon_t\}_{t \in \mathbb{Z}}$ have zero mean and equal variance. And if $\{X_t\}_t^{\infty}$ is invertible, we also have that it can be represented by an AR(∞) model, that is

$$X_t = \sum_{i=1}^{\infty} \alpha_i X_{t-i} + \epsilon_t$$

With this theorem, it seems natural to use AR(p) models for prediction.

Dynamic Linear Models

DLM is a simpler model of a more general framework called state space models.

Definition 3. State space model A state space model consist of two time series : a $\mathbb{R}^p$-valued time series $\{\theta_t\}$ and a a $\mathbb{R}^m$-valued time series $\{Y_t\}$ satisfying the following assumptions :

- (θ_t) is a Markov chain
- Conditionnaly on (θ_t), the Y_t's are independent and the Y_t's depends only on (θ_t)

The idea inder this model is that the seris Y_t is determined by a latent process θ_t. The θ_t's usually represent all the observable physical variables that have an influence on the variable of interest. For instance, if we try to predict the price of electricity in the future, the latent variables could be the temperature, the wind speed, the sunshine, ... Note that this (θ_t) is assumed to be a Markov chain.

In general, a state space model consists in two equations : an **obvservation equation** which gives Y_t in function of θ_t at each step and an **evolution equation** which gives θ_t in function of θ_{t-1}. Both equations are pertubed by noise. This can be written as

$$Y_t = f_t(\theta_t, v_t)$$
$$\theta_t = g_t(\theta_{t-1}, w_t)$$

To be complete we also have to specify the prior distribution for θ_0.

With that general state space model defined, it is possible to define the DLM which is a special case of it with linear functions and Gaussian noise.

Definition 4. DLM A dynamic linear model (DLM) is specified by a Normal prior distribution for the p-dimensional state vector at time t = 0,

$$\theta_0 \sim \mathcal{N}_p(m_0, C_0)$$

together with a pair of equations for each time $t \geq 1$,

$$Y_t = F_t\theta_t + v_t, \qquad v_t \sim \mathcal{N}_m(0, V_t)$$
$$\theta_t = G_t\theta_{t-1} + w_t, \qquad v_t \sim \mathcal{N}_p(0, W_t)$$

where F_t and G_t are known matrices of order $m \times p$ and $p \times p$ respectively.

Forecasting

Based on this model, how can we predict the future observation ? The problem is solved by use of the **Kalman filter**. This filter is based on the idea that, due to the markovian property of the state vector and the assumption of conditional independence of the observations, the density of the state and the observation knowing their previous values can be computed recursively. Moreover, because the equations are linear and the noise and prior distribution are Gaussian, it is possible to prove that the random vector $(\theta_0, ..., \theta_t, Y_0, ..., Y_t)$ follows a mutlivariate Gaussian distribution. Therefore, only the mean and covariance matrix has to be computed. The Kalman filter gives the recursion equation for them :

Theorem 1. *Let $\mathcal{D}_t$ be the information provided by the first t observations $Y_1, ..., Y_t$. Then, if*

$$\theta_{t-1} \sim \mathcal{N}(m_{t-1}, C_{t-1}),$$

where $t \geq 1$, then

(a) the one-step-ahead predictive density of θ_t given $\mathcal{D}_{t-1}$ is Gaussian with parameters

$$\begin{aligned} a_t &= E(\theta_t|\mathcal{D}_{t-1}) = G_t m_{t-1} \\ R_t &= Var(\theta_t|\mathcal{D}_{t-1}) = G_t C_{t-1} G_t' + W_t \end{aligned}$$

(b) the one-step-ahead predictive density of Y_t given $\mathcal{D}_{t-1}$ is Gaussian with parameters

$$\begin{aligned} f_t &= E(Y_t|\mathcal{D}_{t-1}) = F_t a_t \\ Q_t &= Var(Y_t|\mathcal{D}_{t-1}) = F_t R_t F_t' + V_t \end{aligned}$$

(c) the filtering density of θ_t given $\mathcal{D}_t$ is Gaussian with parameters

$$\begin{aligned} m_t &= E(\theta_t|\mathcal{D}_t) = a_t + R_t F_t' Q_t^{-1} e_t \\ C_t &= Var(\theta_t|\mathcal{D}_t) = R_t - R_t F_t' Q_t^{-1} F_t R_t \end{aligned}$$

where $e_t = Y_t - f_t$ is the forecast error.

This theorem gives us everything we need to predict the behaviour of the observations for a time series modeled by a DLM.

It can be seen that the complexity of this algorithm is $\mathcal{O}(kn^2T + k^3T)$. This rather high complexity can become prohibitive when dealing with high-dimensional data. For instance, it is noted in [2] that using a common package in *R* for DLM, problems with $n \geq 32$ cannot be solved.

Another problem with the two classical methods that we have described is that it is not clear how missing data would be dealed with. Yet, missing data are common in many application involving time series and can be due to occlusions or misfunction of sensor, among others.

We will now present the novel approach to make predictions about high-dimensional time series published in [2].

2 Temporal Regularized Matrix Factorization

This section aims to describe the method developed in [2].

Framework

The authors place themselves in a framework similar to the state space model defined in definition 3. The difference is that here θ is not supposed to respect the Markov property. Moreoveer, the observation equation is assumed to be linear. Formally, this can be described by the following equations :

$$\mathbf{y}_t = F\mathbf{x}_t + \eta_t \tag{1}$$

$$\mathbf{x}_t = M_\Theta(\{\mathbf{x}_{t-l} : l \in \mathcal{L}\}) + \epsilon_t \tag{2}$$

The vector noise η_t and ϵ_t are supposed to be Gaussian. These equations are parametrized by sets : $\mathcal{L}$ that contains the number of previous state vectors on wich $\mathbf{x}_t$ depends and Θ that contains the weights associated to each vector in time.

Matrix factorization formulation

The primary obsevation made by the authors is that when we stack all $\mathbf{x}_t$ in a matrix X and if we do the same with the $\mathbf{y}_t$ in the matrix Y, we can see that $Y \approx FX$. Of course, noise has been omitted in this equation and this is why it is not an equality.

With this observation, our problem becomes a problem of matrix factorization which has been widely studied in the past. For instance, this is a key element in some recommender systems and the use of matrix factorization in this context to perform machine learning has been used in the well-known Netflix prize in 2007. In the context of the Netflix prize, the incomplete matrix of ratings was decomposed into a matrix for the users and a matrix for the items. Those two matrices were learned by minimizing an mean square error on the data.

A similar approach can be used in our context. If we denote by $\mathbf{f}_i^\top$ the i$^{\text{th}}$ row of the matrix F and by Ω the set of observed entries, we could learn the X and F factors as

$$\min_{F,X} \sum_{(i,t)\in\Omega} (Y_{it} - \mathbf{f}_i^\top \mathbf{x}_t)^2 + \lambda_f \mathcal{R}_f(F) + \lambda_x \mathcal{R}_x(X)$$

where $\mathcal{R}_f(F)$ and $\mathcal{R}_x(X)$ are regularization factors on F and X respectively. These factors are important to prevent overfitting and to take into accounts possible dependecies between the different entries of the matrices.
The link that has just been established between the general framework and matrix factorization approaches is important but it still does not tell us how to use equation 2 to account for the temporal dependencies. We also need to ddescribe how forecasting could take place in this framework.
In classical MF applications like the one mentionned earlier for recommender systems, the regularization factors are often taken as the norm of the matrix (for example the Frobenius norm) to penalize high values of the larned elements. However, here, this choice would not be a good idea as it would not take into account the temporal dependecies that naturally exists in the $\mathbf{x}_t$s. In [2], the authors propose a choice of regularization factors which remdies that problem.

3 Temporal regularization

The goal in this section is to ddescribe a choice of regulization factor that is called $\mathcal{T}_M(X|\Theta)$ that will promote the relationship between the $\mathbf{x}_t$s given in (2).

References

[1] O. Anava, E. Hazan, and A. Zeevi, “Online time series prediction with missing data,” in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2191–2199. [Online]. Available: http://proceedings.mlr.press/v37/anava15.html

[2] H. Yu, N. Rao, and I. S. Dhillon, “Temporal regularized matrix factorization,” *CoRR*, vol. abs/1509.08333, 2015. [Online]. Available: http://arxiv.org/abs/1509.08333