Mortality forecasting using variational inference

Patrik Andersson

Uppsala University

Mathias Lindholm

Stockholm University

Abstract

This template helps you to create a properly formatted LaTeX manuscript.

Keywords: Non-linear state-space-models , Mortality forecasting, Hidden Markov model, variational inference

1. Introduction

Attempts to forecast mortality goes back perhaps as far as [**?**]. A more recent example is the Lee-Carter model [**?**] and its extensions, see [**? ? ?**] for a survey. Applications of mortality forecasting can be found in for example demographic predictions and in the insurance industry.

The Lee-Carter model is a log-linear multivariate Gaussian model of mortality rates. A major criticism of Lee-Carter-type models is that the model training is done as a two-step process. In a first step, point estimates of the mortality rates are obtained, for example as the maximum likelihood estimate of a Poisson distribution, and in a second step, a latent process is fitted to these estimates. This method has the advantage that it is simple and fast to implement, but it is inefficient when compared to a simultaneous estimation of all unknown parameters. Also, it is not possible to distinguish between the finite population noise of the mortality estimates and the noise from the latent process. Both of these issues can potentially affect the quality of the forecasts.

Simultaneous estimation of parameters have been considered in [**?**] where particle filtering methods are used for a Poisson Lee-Carter model similar to [**?**

]. However, this method has its drawbacks. It could be considered cumbersome for practitioners as it requires custom implementation and tuning, and since the particle filter methods are computationally expensive, the number of parameters must not be too large. The complexity of the modelling, is of course, reduced when removing the Poisson assumption, see e.g. [**?**] for a simpler state-space model treatment of a standard Gaussian Lee-Carter model.

Recently it has been suggested to use models from deep learning, sometimes called deep factor models, to forecast high-dimensional multivariate time series. Some examples of this can be found in [**? ? ? ?**]. The applications presented in those articles differ from mortality forecasting in the scale of the problem. In mortality forecasting the dimension of the time series is about 100 (the lifetime in years of a human) and the number of observations of time series is also about 100 (although some countries do have reliable data for longer than that). As a consequence, to avoid overfitting, we need to consider simpler models. This includes shallower networks for mapping latent variables to the observed time series, linear Gaussian models instead of RNNs for propagating the latent variables forward in time and fewer latent factors.

Compared to previous mortality forecasting models, the novelty of this paper is therefore to use black-box variational inference (?) (VI) [**?**] to solve the inference problem. This means that after specifying how to sample from the model and the approximate posterior, the inference is done automatically without any model specific customisation. The family of models that can be handled is also expanded. For example one can consider neural networks for mapping the latent process to mortality rates. Also, in this case all the parameters can be estimated simultaneously. This latter point is problematic when using particle filter techniques, where it is necessary to estimate the linear mapping from the latent process to the mortality rates in isolation before continuing with the estimation of the other parameters.

Other approaches that use machine learning techniques for forecasting mortality rates can be found in e.g. [Wüthrich & co] that consider various types of Gaussian recurrent neural network structures, [Italienarna, Lindholm & Palmborg] that consider univariate LSTM NNs, both with and without a Poisson population assumption, and see [xx] that consider tree-based techniques.

The model is implemented using the probabilistic programming language Pyro [**?**] and the code is available on Github, see [reference].

The rest of the paper is organised as follows: In Section 2 we describe the probabilistic model that will be used for forecasting. In Section 3 we give a brief introduction to variational inference and in Section 4 we describe how to forecast the mortality once the model has been trained and how we validate the forecast. In Section 5 we demonstrate our method on a few examples. Section 6 concludes the paper.

2. Model

In this section we describe the probabilistic model that define the mortality dynamics. Uppercase letters will denote random variables and lowercase letters the corresponding observed value. Greek letters will denote unknown parameters that are to be estimated.

Mortality data can be aggregated in different forms and clearly the choice of model will have to be adjusted accordingly. For example, the data could contain the population size of each age group at the beginning of the year and the number of deaths during the year. In this case a binomial model seems natural. We however consider data on the yearly number of deaths and the *exposure to risk* in each age group. The exposure to risk in this setting is the total time that the individuals in the population were a certain age in a certain year. The number of deaths in age $a \in \{0, 1, \dots, \bar{a}\}$, year $t \in \{0, 1, \dots, \bar{t}\}$ is denoted by $D_{a,t}$ and the exposure by $E_{a,t}$.

Our model is a dynamic factor model that can be written as:

$$D_{a,t} \mid X_t, E_{a,t} \sim \mathsf{Poisson}\left(E_{a,t} \exp\left(f_a^{\psi}(X_t)\right)\right), \tag{1}$$

$$X_{i,t} = X_{i,t-1} + K_{i,t-1} + U_{i,t}, \quad U_{i,t} \sim \mathsf{Normal}(0, \sigma^2_{X,i}), \tag{2}$$

$$K_{i,t} = \mu_i + \varphi_i(K_{i,t-1} - \mu_i) + V_{i,t}, \quad V_{i,t} \sim \mathsf{Normal}(0, \sigma^2_{K,i}). \tag{3}$$

Here $i = 1, 2, \dots, d$, where d is the dimension of the latent variables. We also require $0 \leq \varphi_i \leq 1$. The function f_a^ψ is the a:th component of $f^\psi : \mathbb{R}^d \to \mathbb{R}^{\bar{a}+1}$. In our examples in Section 5 f will be given by a shallow neural network.

We remark here that the exact specification of the above model is not critical for the continuation. For example, the exponential link-function in the Poisson-distribution could be changed to some other positive differentiable function without complication. We are assuming that the components of the latent process are independent, instead letting any dependence be captured by f. However this latent process could be replaced with some other Markov process.

When you say “shallow”, you don’t allow for interactions?

see previous comment about “shallow” networks

3. Variational inference

Here we explain the main ideas of variational inference (VI) in a general setting. At the end of the section, we connect this to our specific model. For more on VI in general we refer to [**?**] and for the application to state space models, see [**?**].

We are observing y, whose distribution depends on a latent variable x and an unknown parameter ψ. This is modelled by the joint distribution

$$p_\psi(y, x) = p_\psi(y \mid x) p_\psi(x). \tag{4}$$

The likelihood,

$$L(\psi) = p_\psi(y) = \int p_\psi(y, x) \mathrm{d}x, \tag{5}$$

is in general not tractable and therefore approximations are needed in order to be able to estimate ψ. Consider a parametrised distribution, the approximate posterior, $q_\theta(x)$. Then observe that, due to Jensen’s inequality, the log-likelihood is

$$l(\psi) := \log L(\psi) = \log \int \frac{p_\psi(y, x)}{q_\theta(x)} q_\theta(x) \mathrm{d}x \geq \int \left(\log p_\psi(y, x) - \log q_\theta(x)\right) q_\theta(x) \mathrm{d}x =: \mathcal{L}(\psi, \theta). \tag{6}$$

The right-hand side is known as the evidence lower bound (ELBO). The idea of VI is to instead of maximising the log-likelihood, maximise the ELBO. Towards this we calculate the gradients

$$\partial_\psi \mathcal{L}(\varphi,\theta) = \int \partial_\psi \log p_\psi(y,x) q_\theta(x)\mathrm{d}x, \tag{7}$$

$$\partial_\theta \mathcal{L}(\psi,\theta) = \int \log p_\psi(y,x) q_\theta(x) \partial_\theta \log q_\theta(x)\mathrm{d}x. \tag{8}$$

We can then proceed to obtain unbiased estimates of the gradients by sampling from q_θ and maximise $\mathcal{L}$ using stochastic optimisation algorithms. Once converged, $q_\theta(x)$ can be used as an approximation of the posterior distribution of the latent variables $p_\psi(x \mid y)$.

Further, to obtain faster convergence, various variance reduction techniques are often used. Here we only mention the so-called reparametrisation trick. Suppose that we can find functions x_θ such that

$$\int f(x) q_\theta(x)\mathrm{d}x = \int f(x_\theta(z)) q(z)\mathrm{d}z, \tag{9}$$

which makes the sampling distribution independent of θ. In particular, the gradient satisfies

$$\partial_\theta \int f(x) q_\theta(x)\mathrm{d}x = \partial_\theta \int f(x_\theta(z)) q(z)\mathrm{d}z = \int \partial_\theta f(x_\theta(z)) q(z)\mathrm{d}z, \tag{10}$$

which usually improves the sampling variance compared to differentiating the density directly. An important example of a distribution that allows for reparametrisation according to (9) is the Gaussian, since if $Z \sim \mathsf{N}(0,1)$ then $\mu + \sigma Z \sim \mathsf{N}(\mu,\sigma^2)$.

In the numerical illustrations in Section 5, the approximate posterior is modelled as a Gaussian distribution with an autoregressive covariance. That is, the distribution of the process is given by

$$\tilde{X}_{i,t} = \tilde{\mu}^X_{i,t} + \alpha_{i,t}\tilde{X}_{i,t-1} + \tilde{e}^X_{i,t}, \quad \tilde{e}^X_{i,t} \text{ iid } \mathsf{N}(0,\tilde{\sigma}^2_{X,i}), \tag{11}$$

$$\tilde{K}_{i,t} = \tilde{\mu}^K_{i,t} + \beta_{i,t}\tilde{K}_{i,t-1} + \rho_{i,t}\tilde{X}_{t-1} + \tilde{e}^K_{i,t}, \quad \tilde{e}^K_{i,t} \text{ iid } \mathsf{N}(0,\tilde{\sigma}^2_{K,i}). \tag{12}$$

4. Forecasting and validation

Here we discuss how to forecast the mortality after maximising the ELBO and thus obtaining estimates of φ, θ and, in particular, the joint distribution of $(\tilde{X}_{i,\bar{t}}, \tilde{K}_{i,\bar{t}})$.

Since both the approximate posterior and the latent process is Gaussian, the forecasting distribution of the latent process is also Gaussian. That is, for $t > \bar{t}$,

$$\begin{pmatrix} \hat{X}_{i,t} \\ \hat{K}_{i,t} \end{pmatrix} \sim \mathsf{N}\left(\begin{pmatrix} \hat{\mu}_{i,t}^{X} \\ \hat{\mu}_{i,t}^{K} \end{pmatrix}, \begin{pmatrix} \hat{\sigma}_{X,t}^{2} & \hat{\rho}_{i,t}\hat{\sigma}_{X,t}\hat{\sigma}_{K,t} \\ \hat{\rho}_{i,t}\hat{\sigma}_{X,t}\hat{\sigma}_{K,t} & \hat{\sigma}_{K,t}^{2} \end{pmatrix} \right), \tag{13}$$

where the parameters can be calculated iteratively from (2) and (3), by using the initial value

$$\begin{pmatrix} \hat{X}_{i,\bar{t}} \\ \hat{K}_{i,\bar{t}} \end{pmatrix} := \begin{pmatrix} \tilde{X}_{i,\bar{t}} \\ \tilde{K}_{i,\bar{t}} \end{pmatrix}, \tag{14}$$

and the forecast of ~~the~~ mortality rates is given by $\exp(f^{\hat{\psi}}(\hat{X}_t))$.

If one wants to forecast the actual number of deaths, a forecast of the number of living at the beginning of the year is also needed, together with some assumption on the distribution of when in the year people are born. For a longer discussion on how this can be done, we refer to [**?**].

The forecast is validated by calculating the logarithmic score of the forecast on the validation data set, see for example [**?**]. That is, the mortality rates are forecasted and multiplied by the observed exposure-to-risk in each age group, giving the intensity of the Poisson distributed number of deaths in each age group. The logarithmic score of each age group is summed to give the total logarithmic score for a given year.

5. Results

In this section we illustrate the method on two datasets, Swedish males between the years 1920 and 2000, and US males between 19xx and 2000. We then validate the forecasts on the years 2000 to 2019. Both datasets are collected from [**?**].

The model is as in (1) - (3) where f is either a 1 (i.e. affine) or a 2-layer neural network with ReLU activation and the middle layer is of dimension 100. We compare $d = 1, \dots 6$.

6. Conclusions

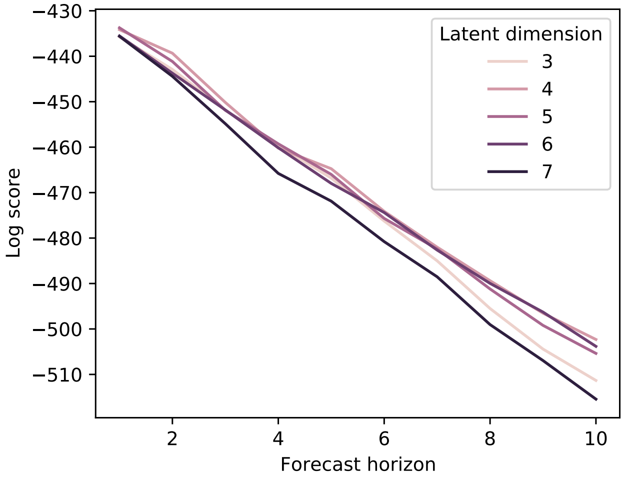

Figure 1

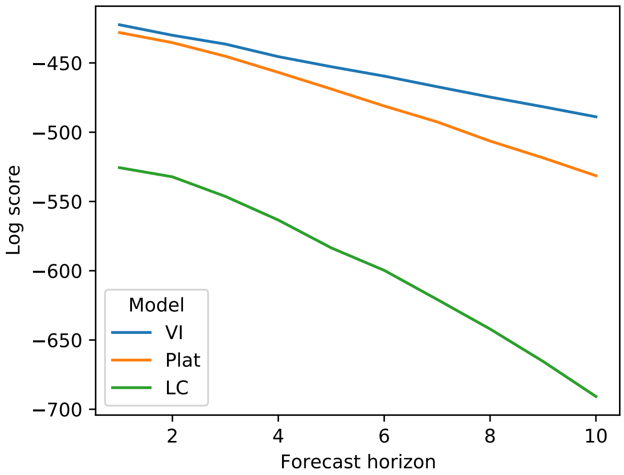

Figure 2