Deep Decoding of Strategies

Technical Note, June 6, 2022

Jean-Jacques Ohana [1] **Eric Benhamou** [1 2] **David Saltiel** [1 2 3] **Beatrice Guez** [1]

Abstract

To the best of our knowledge, the application of machine learning and in particular graphical models in the field of quantitative risk management is still a relatively recent and new phenomenon. This paper presents a new and effective methodology for decoding strategies. Given an investment universe, we calculate dynamic weights for a sparse portfolio whose aim is to replicate the strategy with the most stable allocation rules. Naturally, this can be formulated as a reinforcement learning problem whose reward is a weighted sum of tracking error and turnover. We show on stylized examples that we can accurately decode strategies or funds with meaningful factors and allocations.

1. Introduction

Decoding or deciphering a financial strategy is a traditional problem often framed as a replication or index tracking problem. However, index tracking is not that trivial even if we have the full knowledge of the index's composition, due to very frequent rebalancing, churn in index members, and no liquid assets making it a difficult task (Strub & Baumann, 2018), (Benidis et al., 2018). Indeed, it is highly recommended to take into account not only the capacity to replicate the financial strategy but also the overall turnover of the replicated strategy and the number of selected assets (Canakgoz & Beasley, 2009), (Takeda et al., 2013), (Fastrich et al., 2014).

In this note, we show that we can write this as a graphical inference problem and hence use efficient machine learning methods to decipher the strategy. Graphical models are a powerful tool that allows learning latent variables efficiently and within a reinforcement learning setting (Levine, 2018), or with complex optimization based on natural gradient descents (Benhamou et al., 2019). Moreover, Deep reinforcement learning has turned out to be effective for a variety of traditional financial problems (Benhamou et al., 2020c), (Benhamou et al., 2021b), (Benhamou et al., 2021c), (Benhamou et al., 2020a), (Benhamou et al., 2021a), (Benhamou et al., 2020b)

Let us write $(S_t)_{t=1..N}$ a time series of observation prices for a fund or a strategy that we want to decode. Let us define an investment universe of m assets or factors $i = 1...m$ whose returns for the same observation times as the strategy to decode are denoted by by r_t^i for $t \in [1, N]$ and for $i \in [1, m]$. The decoding consists in finding the dynamic weights w_t^i such that we replicate the initial strategy $(S_t)_{t=1..N}$. By replication, we mean that the portfolio invested in factors according to our weights and with transaction cost is close to the strategy to decode. If the value of this portfolio starts at time t_0 with an initial value equal to the strategy to decode S_{t_0} and is invested according to the dynamic weights, its value at time u is given by the corresponding compounded returns from $t = t_0$ to $t = t_u$. This writes as

$$\hat{S}_{t_u} = S_{t_0} \prod_{j=1}^{u} \frac{S_{t_j}}{S_{t_{j-1}}} \tag{1.1}$$

$$= S_{t_0} \prod_{j=1}^{u} (1 + \sum_{i=1}^{m} w_{t_{j-1}}^i r_{t_j}^i - cost_{t_j}^i) \tag{1.2}$$

Cost at time t_j will be specified later on in this note. Index difference between weights and returns comes from the fact that weights are decided at previous day's close.

*Equal contribution [1]Ai for Alpha, France [2]Paris Dauphine University PSL, France [3]University of the Littoral Opal Coast, France. Correspondence to: Beatrice Guez <beatrice.guez@aiforalpha.com>.

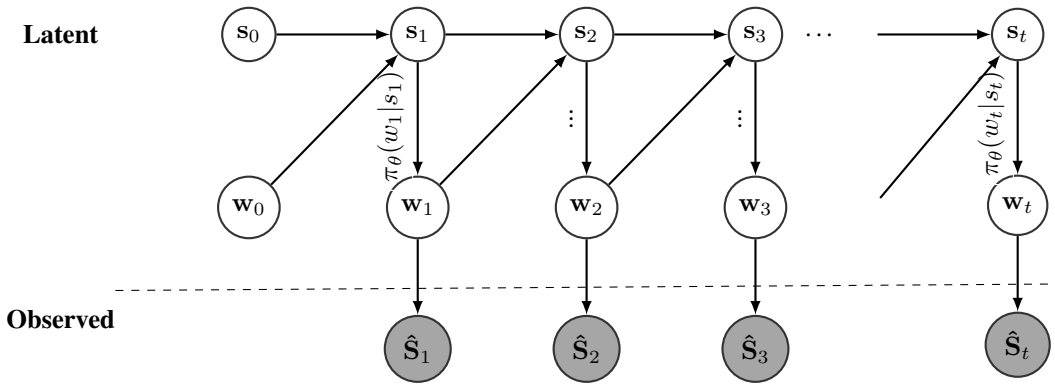


Figure 1. Graphical model representation of decoding RL model

Hence weights time index is one period before: $t \in [0, u-1]$. To infer these weights, we use two conditions:

- best match between our decoded strategy and the initial strategy
- most stable rules for our dynamic weights

2. Financial formulation

The two conditions stated above can be translated in financial terms.

Our decoded strategy $\hat{S}_{t=1..N}$ results from taking a position w^i_{t-1} into the asset or factor i at time $t-1$. Hence we receive the corresponding next day return r^i_t reduced by the transaction costs whose cost for asset i is estimated to be proportional by a coefficient b_i to the absolute difference of positions between today and yesterday. This translates into the following evolution for our decoded strategy for any observation time t_u with $u \in [1, N]$

$$\hat{S}_{t_u} = S_{t_0} \prod_{t=1..u} (1 + \sum_{i=1}^{m} w^i_{t-1} r^i_t - b_i |w_{t-1} - w_t|) \quad (2.1)$$

This equation is the straight mathematical translation of equation (1.1). We can translate the first condition of similarities between our two strategies into a condition of minimum tracking error.

We are interested in the decoded strategy whose tracking error is minimal. This tracking error is traditionally computed as the annual standard deviation of the difference between the strategy to decode 's and the decoded strategy's return. Denoting by Std the standard deviation operator and assuming 252 days per year, the tracking error (TE) writes as follows:

$$\text{TE} = Std(S_t/S_{t-1} - \hat{S}_t/\hat{S}_{t-1}) \times \sqrt{252} \quad (2.2)$$

Similarly, we can translate the second condition into a condition of minimum turnover (To) whose definition is simply given by

$$\text{To} = \frac{252}{N} \sum_{t=1..N} |w_{t-1} - w_t| \quad (2.3)$$

We are then ready to formulate our problem as a reinforcement learning problem. Although we do not know the solution, we can use the two conditions and create a mix, whose balance between tracking error and turnover is Ai for Alpha' secret, that acts as a reward mechanism to learn the decoded strategy. In other terms, we create a graphical model that represents all the connections between the dynamic weights and use reinforcement learning to learn these parameters given this reward.

3. Reinforcement learning formulation

At time t, we create a state that is given by the previous weights $(w^i_{t-1})_{i=1..m}$ and by a window of previous universe asset returns $\left(r^i_u\right)_{u=t-k..t}$ and their standard deviations denoted by $(\sigma^i_u)_{u=t-k..t}$.

The goal of our agent is to find the next weights $(w^i_t)_{i=1..m}$ that achieve the highest final reward. From a graphical model point of view, this creates a graphical model that models the interaction between states s_t and actions w_t at each time, where weights are the stacked vector of all individual weights. What we observed is the resulting decoded portfolio

Our machine learning algorithm learns the mapping from states to actions. This mapping is traditionally referred to as the agent policy and denoted by $\pi_\theta(w_t|s_t)$ where θ are our

model parameters. It provides in other words the conditional probability of taking an allocation w_t at time t given that we are in the state t_t. Inference of model parameters and the corresponding optimal policy can be done by an actor critic method whose property is to be an effective variance reduction as explained in (Benhamou, 2019).

4. Connections with statistical methods

In this section, we prove that graphical models generalizes Kalman filters and HMM models. Historically, Hidden Markov Model (HMM) and Kalman filter were developed in distinct and unconnected research communities. Hence, their close relationship has not always been widely emphasized and appreciated. Part of the explanation lies also to the fact that the general framework for unifying these two approaches, namely graphical models came much later than HMM and Kalman filter. Without Bayesian graphical framework, the two algorithms underlying the inference calculation look rather different and unrelated. However, their difference is simply a consequence of the differences between discrete and continuous hidden variables and more specifically between multinomial and normal distribution. These details, important as they may be in practice, should not obscure us from the fundamental similarity between these two models. As we shall see in this section, the inference procedure for the state space model (SSM) shall prove us shortly that HMM and Kalman filter's model are cousin and share the same underlying graphical model structure, namely a hidden state space variable and an observable variable. The interest of using Bayesian Probabilistic Graphical model is multiple. First, it emphasizes the general graphical model architecture underpinning both HMM and Kalman filter. Second, it provides modern computational tools used commonly in machine learning to do the inference calculation. It shows how to generalize Kalman filter in the case of non Gaussian assumptions. It is interesting to realize that graphical models have been the marriage between probability theory and graph theory.

They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering – uncertainty and complexity – and in particular they are playing an increasingly important role in the design and analysis of machine learning algorithms.

From a literature point of view, Hidden Markov models were discussed as early as (Rabiner & Juang, 1986), and expanded on in (Rabiner, 1989). The first temporal extension of probabilistic graphical models is due to (Dean & Kanazawa, 1989), who also coined the term dynamic Bayesian network. Much work has been done on defining various representations that are based on hidden Markov models or on dynamic Bayesian networks; these include generalizations of the basic framework, or special cases that allow more tractable inference. Examples include mixed memory Markov models (see (Saul & Jordan, 1999)); variable-duration HMMs ((Rabiner, 1989)) and their extension segment models ((Ostendorf et al., 1996)); factorial HMMs ((Ghahramani & Jordan, 1994)); and hierarchical HMMs ((Fine et al., 1998) and (Bui et al., 2002)). (Smyth et al., 1997) is a review paper that was influential in providing a clear exposition of the connections between HMMs and DBNs. (Murphy & Paskin, 2001) show how hierarchical HMMs can be reduced to DBNs, a connection that provided a much faster inference algorithm than previously proposed for this representation. (Murphy, 2002) (and lately the book (Murphy, 2013)) provides an excellent tutorial on the topics of dynamic Bayesian networks and related representations as well as the non published book of (Jordan, 2016)

4.1. State space model

The state space model as emphasized for instance in (Murphy, 2013) is described as follows:

- there is a continuous chain of states denoted by $(\mathbf{x}_t)_{t=1,\dots,n}$ that are non observable and influenced by past states only through the last realization. In other words, $\mathbf{x}_t$ is a Markov process, meaning $\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1}) = \mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ Using graphical model, this can also be stated as given a state at one point in time, the states in the future are conditionally independent of those in the past.

- for each state, we can observe a space variable denoted by $\mathbf{z}_t$ that depends on the non observable space $\mathbf{x}_t$

Compared to the original presentation of the Kalman filter model, this is quite different. We now assume that there is an hidden variable (our state) and we can only measure a space variable. Since at each step, the space variable only depends on the non observable, there is only one arrow or edge between two latent variables horizontal nodes. This model is represented as a graphical model in figure 2.

Obviously, the graphical model provided in figure 2 can host both HMM and Kalman filter model. To make our model tractable, we will need to make additional assumptions.

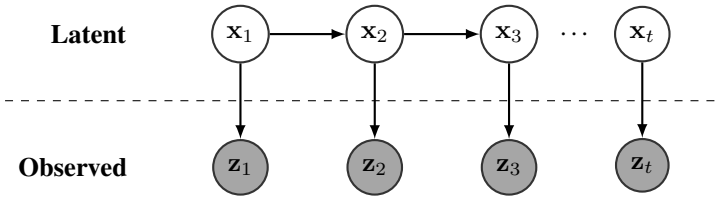


Figure 2. State Space model as a Bayesian Probabilistic Graphical model. Each vertical slice represents a time step. Nodes in white represent non observable or latent variables called the states and denoted by $\mathbf{x}_t$ while nodes in gray observable ones and are called the spaces and denoted by $\mathbf{z}_t$. Each arrow indicates that there is a relationship between the arrow originating node and the arrow targeting node. Dots indicate that there is many time steps. The central dot line is to emphasize the fundamental difference between latent and observed variables

We will emphasize the ones that are common to HMM and Kalman filter models and the ones that differ. We impose the following conditions:

- The relationship between the state and the space is linear (this is our measurement equation in the Kalman filter and this is common to Kalman filter and HMM models):
$$\mathbf{z}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t \tag{4.1}$$
where the noise term $\mathbf{v}_t$ is assumed to follow a multi dimensional normal distribution with zero mean and covariance matrix given by $\mathbf{R}_t$.
- We will make the simplest choice of dependency between the state at time $t-1$ and t and assume that this is linear (this is our state equation and this is common to Kalman filter and HMM models)
$$\mathbf{x}_t = \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_t + \mathbf{w}_t \tag{4.2}$$
where the noise term $\mathbf{w}_t$ is assumed to follow a multi dimensional normal distribution with zero mean and covariance matrix given by $\mathbf{Q}_t$ and where $\mathbf{B}_t\mathbf{u}_t$ is an additional trend term (that represents our control in Kalman filter). This control term is not common in HMM model but can be added without any difficulty. This results in slightly extended formula that we will signal. These formula are slight improvement of the common one found in the literature. Although, from a theoretical point of view, this control term may seem a futility, it is very important in practice and makes a big difference in numerical applications.
- We will assume as in the Kalman filter section that the initial state, and noise vectors at each step $\mathbf{x}_0, \mathbf{w}_1, \ldots, \mathbf{w}_t, \mathbf{v}_1, \ldots, \mathbf{v}_t$ are all mutually independent (this is common to HMM and Kalman filter models).
- Last but not least, we assume that the distribution of the state variable $\mathbf{x}_t$ follows a multi dimensional normal distribution. This is **Kalman filter specific**. For HMM, the state is assumed to follow a multinomial distribution.

The above assumptions restrict our initial state space model to a narrower class called the Linear-Gaussian SSM (LG-SSM). This model has been extensively studied and more can be found in (Durbin & Koopman, 2012) for instance.

Before embarking into the inference problem for the SSM, it is interesting to examine the unconditional distribution of the states $\mathbf{x}_t$. Using equation (4.2)

The unconditional mean of $\mathbf{x}_t$ is computed recursively
$$\prod_{k=2}^{t}\mathbf{F}_k\mathbf{x}_1 + \sum_{k=2}^{t}\prod_{l=k+1}^{t}\mathbf{F}_l\mathbf{B}_k \tag{4.3}$$
with the implicit assumption that empty product equals 1:
$$\prod_{l=t+1}^{t}\mathbf{F}_l = 1.$$
In the specific case of a null control term ($\mathbf{B}_t = 0$), the latter equation simplifies into
$$\prod_{k=2}^{t}\mathbf{F}_k\mathbf{x}_1 \tag{4.4}$$
The unconditional co-variance is computed as follows $\mathbf{P}_t = \mathbb{E}[\mathbf{x}_t\mathbf{x}_t^{\mathrm{T}}]$. Using our assumptions on independence as well as the state equation (4.2), we can compute it easily as:
$$\mathbf{P}_t \quad = \quad \mathbf{F}_t\mathbf{P}_{t-1}\mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t \tag{4.5}$$
This last equation remains unchanged in case of a non zero control term as the control term is deterministic. This last equation provides a dynamic equation for the unconditional variance and is referred to as the *Lyapunov equation*. It is also easy to checked that the unconditional covariance between neighboring states $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ is given by $\mathbf{F}_{t+1}\mathbf{P}_t\mathbf{F}_{t+1}^{\mathrm{T}}$.

4.2. Inference

The inference problem consists in calculating the posterior probability of the states given an output sequence. This calculation can be done both forward and backward. By forward, we mean that the inference information (called the evidence) at time t consists of the partial sequence of outputs up to time t. The backward problem is similar except that the evidence consists of the partial sequence of outputs after time t. Using standard graphical model terminology, we distinguish between filtering and smoothing problem.

In filtering, the problem is to calculate an estimate of the state $\mathbf{x}_t$ based on a partial output sequence $\mathbf{z}_0, \dots, \mathbf{z}_t$. That is, we want to calculate $\mathbb{P}(\mathbf{x}_t \mid \mathbf{z}_0, \dots, \mathbf{z}_t)$. This is often referred to as the alpha recursion in HMM models (see for instance (Rabiner & Juang, 1986) and (Rabiner, 1989)).

Using standard Kalman filter notations, we shall denote by

$$\hat{\mathbf{x}}_{t|t} \triangleq \mathbb{E}[\mathbf{x}_t \mid \mathbf{z}_0, \dots, \mathbf{z}_t] \tag{4.6}$$

$$\mathbf{P}_{t|t} \triangleq \mathbb{E}[(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^{\mathrm{T}} \mid \mathbf{z}_0, \dots, \mathbf{z}_t] \tag{4.7}$$

Under this settings, it is fairly easy to derive the following property that provides the conditional posterior distribution in the forward recursion. And to recover traditional results of Kalman filter, we shall decompose our time propagation into two steps:

- time update:
  $\mathbb{P}[\mathbf{x}_t \mid \mathbf{z}_0, \dots, \mathbf{z}_t] \quad \to \mathbb{P}[\mathbf{x}_{t+1} \mid \mathbf{z}_0, \dots, \mathbf{z}_t]$
- measurement update:
  $\mathbb{P}[\mathbf{x}_{t+1} \mid \mathbf{z}_0, \dots, \mathbf{z}_t] \to \mathbb{P}[\mathbf{x}_{t+1} \mid \mathbf{z}_0, \dots, \mathbf{z}_{t+1}]$

This can be represented nicely in terms of graphical models by the figure 3 below.

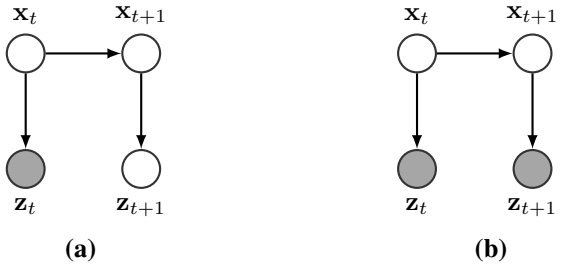


Figure 3. (a) A portion of the State Space Model before a measurement and (b) after a measurement update. White nodes are non observable variables while gray nodes are observed nodes.

Proposition 1. Conditioned on past outputs $\mathbf{z}_0, \dots, \mathbf{z}_t$, the variables $\mathbf{x}_{t+1}$ and $\mathbf{z}_{t+1}$ have a joint Gaussian distribution with mean and covariance matrix given by:

$$\begin{bmatrix} \hat{\mathbf{x}}_{t+1|t} \\ \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t} \end{bmatrix} \quad \text{and} \tag{4.8}$$

$$\begin{bmatrix} \mathbf{P}_{t+1|t} & \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \\ \mathbf{H}_{t+1}\mathbf{P}_{t+1|t} & \mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1} \end{bmatrix} \tag{4.9}$$

Proof. This is trivial as the considered state space model is the Linear Gauss State Space Model (LGSSM). □

As simple as it may seem, the previous proposition makes our graphical model a full powerhouse as it provides the building block to start the inference. Indeed, knowing the conditional posterior distribution at step (a) of figure 3 gives us the first bullet to conclude for the step (b). Moreover, using results from simpler graphical models like the factor analysis graphical model, we can immediately conclude that the second step is given as follows

Proposition 2. Conditioned on past outputs $\mathbf{z}_0, \dots, \mathbf{z}_{t+1}$, we have the following relationship

$$\begin{aligned} \hat{\mathbf{x}}_{t+1|t+1} &= \hat{\mathbf{x}}_{t+1|t} + \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \\ &\quad \left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \\ &\quad (\mathbf{z}_{t+1} - \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t}) \end{aligned} \tag{4.10}$$

$$\begin{aligned} \mathbf{P}_{t+1|t+1} &= \mathbf{P}_{t+1|t} - \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \\ &\quad \left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \\ &\quad \mathbf{H}_{t+1}\mathbf{P}_{t+1|t} \end{aligned} \tag{4.11}$$

Proof. This is a direct consequence of the spatial model and can be found for instance in (Murphy, 2013) or (Jordan, 2016). We provide a self contained proof in appendix B □

Summarizing all these results leads to the seminal recursions of the Kalman filter and given by the following proposition.

Proposition 3. Kalman filter consists in the following recursive equations:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t \tag{4.12}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t \tag{4.13}$$

$$\begin{aligned} \hat{\mathbf{x}}_{t+1|t+1} &= \hat{\mathbf{x}}_{t+1|t} + \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \\ &\quad \left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \\ &\quad (\mathbf{z}_{t+1} - \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t}) \end{aligned} \tag{4.14}$$

$$\begin{aligned} \mathbf{P}_{t+1|t+1} &= \mathbf{P}_{t+1|t} - \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \\ &\quad \left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \\ &\quad \mathbf{H}_{t+1}\mathbf{P}_{t+1|t} \end{aligned} \tag{4.15}$$

Proof. This is a trivial consequence of proposition 2. □

Remark 4.1. If we introduce the following intermediate variables:

$$\tilde{\mathbf{y}}_{t+1} = \mathbf{z}_{t+1} - \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t} \tag{4.16}$$
$$\mathbf{S}_{t+1} = \mathbf{R}_{t+1} + \mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \tag{4.17}$$
$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{S}_{t+1}^{-1} \tag{4.18}$$

The equations (4.14) and (4.15) transform into equations

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \mathbf{K}_{t+1}\tilde{\mathbf{y}}_{t+1} \tag{4.19}$$
$$\mathbf{P}_{t+1|t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})\,\mathbf{P}_{t+1|t} \tag{4.20}$$

which are useful equations often presented in control theory when doing Kalman filtering.

Remark 4.2. There is nothing new to fancy at this stage except that we have shown with graphical models the Kalman filter recursion. And we can check this is way faster, easier and more intuitive. It is worth noticing that we have also multiple ways to write the gain matrix. Using the Sherman–Morrison–Woodbury formula, we can also derive various forms for the gain matrix as follows:

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}\left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \tag{4.21}$$
$$= \left(\mathbf{P}_{t+1|t}^{-1} + \mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}\mathbf{H}_{t+1}\right)^{-1}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}^{-1} \tag{4.22}$$
$$= \left(\mathbf{P}_{t+1|t} - \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}^{-1} \tag{4.23}$$
$$= \mathbf{P}_{t+1|t+1}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}^{-1} \tag{4.24}$$

These forms may be useful whenever the reduced form (which is the last equation) is numerically unstable. Equation (4.24) is useful as it relates $\mathbf{K}_{t+1}$ to $\mathbf{P}_{t+1|t+1}$. It is interesting to notice the two form of the Kalman gain

$$\mathbf{K}_{t+1} \quad = \quad \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{S}_{t+1}^{-1} = \mathbf{P}_{t+1|t+1}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}^{-1} \tag{4.25}$$

Remark 4.3. The Kalman filter appeals some remarks. We can first note that Kalman filtering equations can be interpreted differently. Combining equations (4.19) and (4.12), we retrieve an error correcting algorithm as follows:

$$\hat{\mathbf{x}}_{t|t} \quad = \quad \mathbf{F}_t\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t + \mathbf{K}_t \left(z_t - \mathbf{H}_t(\mathbf{F}_t\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t\right) \tag{4.26}$$

or equivalently, regrouping the term $\hat{\mathbf{x}}_{t-1|t-1}$

$$\hat{\mathbf{x}}_{t|t} \quad = \quad (\mathbf{F}_t - \mathbf{K}_t\mathbf{H}_t\mathbf{F}_t)\,\hat{\mathbf{x}}_{t-1|t-1} + (\mathbf{B}_t\mathbf{u}_t + \mathbf{K}_t\,(z_t - \mathbf{H}_t\mathbf{B}_t\mathbf{u}_t)) \tag{4.27}$$

This shows us two things:

- Kalman filter can be seen as the discrete version of an Ornstein Uhlenbeck process
- Kalman filter can be seen as Auto Regressive process in discrete times

Since, recursive equation do appear similarly in Recursive Least Square (RLS) estimates, we also see here a connection with RLS. It is striking that these connections are not often made, mostly because Kalman filter was originally a control problem and not a statistician one.

As nice as the Kalman filter equation may look like, they have one major problem. It is the numerical stability of the filter. If the process noise covariance $\mathbf{Q}_t$ is small, round-off error would lead to obtain numerically a negative number for small positive eigenvalues of this matrix. As the scheme will propagate round off errors, the state covariance matrix $\mathbf{P}_t$ will progressively become only positive semi-definite (and hence indefinite) while it is theoretically a true positive definite matrix fully invertible.

In order to keep the positive definite property, we can slightly modify the recursive equation to find recursive equations that preserve the positive definite feature of the two covariance matrices. Since any positive definite matrix $\mathbf{S}^d$ can be represented and also reconstructed by its upper triangular square root matrix $\mathbf{R}$ with $\mathbf{S}^d = \mathbf{R}^t\mathbf{R}^{\mathrm{T}}$, with the strong property that representing in this form will guarantee that the resulting matrix will never get a negative eigen value, it is worth using a square root scheme with square root representation. Alternatively, we can also represent our covariance matrix using the so called unit diagonal (U-D) decomposition form, with $\mathbf{S}^d = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}$ where $\mathbf{U}$' is a unit triangular matrix (with unit diagonal), and $\mathbf{D}$' is a diagonal matrix. This form avoids in particular many of the square root operations required by the matrix square root representation. Moreover, when comparing the two approaches, the U-D form has the elegant property to need same amount of storage, and somewhat less computation. This explains while the U-D factorization is often preferred (Thornton, 1976). A slight variation is the *LDL* decomposition of the innovation covariance matrix. The LDL decomposition relies on decomposing the covariance matrix $\mathbf{S}^d$ with two matrices: a lower unit triangular (unitriangular) matrix $\mathbf{L}$, and $\mathbf{D}$ a diagonal matrix. The algorithm starts with the LU decomposition as implemented in the Linear Algebra PACKage (LAPACK) and further it factors into the LDL form. Any singular covariance matrix is pivoted so that the first diagonal partition is always non-singular and well-conditioned (for further details see (Bar-Shalom et al., 2002)).

4.3. Connection to information filter

It is worth showing the close connection with particle and information filter. This is a direct consequence of the fact

that a multivariate Gaussian belongs to the exponential family and as such admits canonical parameters. Hence, we can rewrite the filter in terms of the later instead of the initial usage of moment parameters. This is an interesting rewriting of the Kalman filter as it makes it numerically more stable. The canonical parameters of a multi variate Gaussian distribution, denoted by $\mathbf{\Lambda}$ and η, are obtained from the moment parameters Σ and μ as follows: $\mathbf{\Lambda} = \Sigma^{-1}$ and $\eta = \Sigma^{-1}\mu$. We are interested in deriving the canonical parameters of $\mathbf{x}_t$ first, at the prediction phase, conditioned on $\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}$.

In the Kalman filter settings, the covariance Σ is denoted by $\mathbf{P}$ while the moment is given by $\mathbf{x}$ with the relationship $\eta = \mathbf{P}^{-1}\mathbf{x}$. Hence, we will write the precision matrix as $\mathbf{\Lambda}$ with the relationship with the covariance matrix given by $\mathbf{\Lambda} = \mathbf{P}^{-1}$. We shall write our new variables to be consistent with previous development as $\mathbf{\Lambda}_{t|t-1}$ and $\eta_{t|t-1}$. At the correction or measurement phase, we are interested in the same parameters but now conditioned on $\mathbf{z}_1, \ldots, \mathbf{z}_t$. We shall write them $\mathbf{\Lambda}_{t|t}$ and $\eta_{t|t}$. We can easily derive the following recursive scheme that is given by the proposition below:

Proposition 4. The filter equations are given by the following recursive scheme:

$$\begin{aligned}
\hat{\eta}_{t|t-1} &= \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\hat{\eta}_{t-1|t-1} \\
&\quad +(\mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1} \\
&\quad \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1})\mathbf{B}_t\mathbf{u}_t \qquad (4.28) \\
\hat{\eta}_{t|t} &= \hat{\eta}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t \qquad (4.29) \\
\mathbf{\Lambda}_{t|t-1} &= \mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1} \\
&\quad \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} \qquad (4.30) \\
\mathbf{\Lambda}_{t|t} &= \mathbf{\Lambda}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t \qquad (4.31)
\end{aligned}$$

Proof. The derivation is easy and given in section B.3 □

Remark 4.4. The recursive equation for the information filter are very general. They include the control term $\mathbf{B}_t\mathbf{u}_t$ that is often neglected in literature introduced as early as 1979 in (Anderson & Moore, 1979). It is worth noticing that we can simplify computation by pre-computing a term $\mathbf{M}_t$ as follows:

$$\begin{aligned}
\mathbf{M}_t &= \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1} \qquad (4.32) \\
\mathbf{\Lambda}_{t|t-1} &= \mathbf{Q}_t^{-1} - \mathbf{M}_t\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} \qquad (4.33) \\
\hat{\eta}_{t|t-1} &= \mathbf{M}_t\hat{\eta}_{t-1|t-1} + \mathbf{\Lambda}_{t|t-1}\mathbf{B}_t\mathbf{u}_t \qquad (4.34) \\
\hat{\eta}_{t|t} &= \hat{\eta}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t \qquad (4.35) \\
\mathbf{\Lambda}_{t|t} &= \mathbf{\Lambda}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t \qquad (4.36)
\end{aligned}$$

These equations are more efficient than the ones provided in proposition 4. As for the initialization of this recursion, we define the initial value as follows $\hat{\eta}_{1|0} = \hat{\eta}_1$ and $\mathbf{\Lambda}_{1|0} = \mathbf{\Lambda}_1$.

It is interesting to note that the Kalman filter and the information filter are mathematically equivalent. They both share the same assumptions. However, they do not use the same parameters. Kalman filter (KF) uses moment parameters while particle or information filter (IF) relies on canonical parameters, which makes the later numerically more stable in case of poor conditioning of the covariance matrix. This is easy to understand as a small eigen value of the covariance translates into a large eigen value of the precision matrix as the precision matrix is the inverse of the covariance matrix. Reciprocally, a poor conditioning of the information filter should convert to a more stable scheme for the Kalman filter as the he condition number of a matrix is the reciprocal of the condition number of its inverse. Likewise, for initial condition as they are inverse, a small initial state in KF should translate to a large state in IF.

4.4. Smoothing

Another task we can do on dynamic Bayesian network is smoothing. It consists in obtaining estimates of the state at time t based on information from t on-wards (we are using future information in a sense). Like for HMM, the computation of this state estimate requires combining forward and backward recursion, either by starting by a backward-filtered estimates and then a forward-filtered estimates (an 'alpha-beta algorithm"), or by doing an algorithm that iterates directly on the filtered-and-smoothed estimates (an "alpha-gamma algorithm"). Both kinds of algorithm are available in the literature on state-space models (see for instance (Koller & Friedman, 2009) and (Cappe et al., 2010) for more details), but the latter approach appears to dominate (as opposed to the HMM literature, where the former approach dominates) . The "alpha-gamma" approach is referred to as the "Rauch-Tung-Striebel (RTS) smoothing algorithm" (although it was developed using very different tool, namely control theory as early as 1965: see (Rauch et al., 1965)) while the other approach is just the "alpha-beta" recursion.

4.4.1. Rauch-Tung-Striebel (RTS) smoother

The Rauch-Tung-Striebel (RTS) smoother developed in (Rauch et al., 1965) relies precisely on the idea of of doing first a backward estimate and then a forward filter. Smoothing should not be confused with smoothing in times series analysis that is more or less a convolution. Smoothing for a Bayesian network means infering the distribution of a node conditioned on future information. It is the reciprocal of filtering that has also two meanings. Filtering for time series means doing a convolution to filter some noise. But filtering for Bayesian network means infering the distribution of a node conditioned on past information.

We can work out the RTS smoother and find the recursive

equations provided by the following proposition

Proposition 5. The RTS smoothing algorithm works as follows:

$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + \mathbf{L}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{4.37}$$

$$\hat{\mathbf{P}}_{t|T} = \mathbf{P}_{t|t} + \mathbf{L}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{L}_t^{\mathrm{T}} \tag{4.38}$$

$$\text{where } \mathbf{L}_t = \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1} \tag{4.39}$$

with an initial condition given by

$$\hat{\mathbf{x}}_{T|T} = \hat{\mathbf{x}}_T \tag{4.40}$$

$$\hat{\mathbf{P}}_{T|T} = \hat{\mathbf{P}}_T \tag{4.41}$$

Proof. The proof consists in writing rigorously the various equations and is given in section B.4 □

4.4.2. Alternative to RTS filter

It is worth noting that we can develop an alternative approach to the RTS filter that relies on the alpha beta approach without any observation. This approach is quite standard for HMM models (see for instance (Rabiner & Juang, 1986) or (Russell & Norvig, 2009)). Following (Kitagawa, 1987), this approach has been called in the literature the *two-filter algorithm*. It consists in combining the *forward* conditional probability $\mathbb{P}(\mathbf{x}_t \mid \mathbf{z}_1, \dots, \mathbf{z}_t)$ with the *backward* conditional probability $\mathbb{P}(\mathbf{x}_t \mid \mathbf{z}_{t+1}, \dots, \mathbf{z}_T)$. The intuition is illustrated by the figure 4.

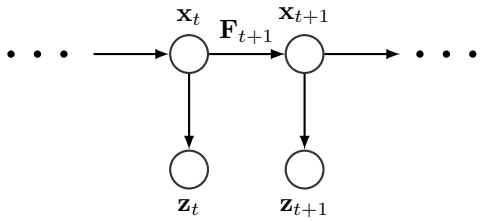


Figure 4. SSM with no observations

The graphical model provided by figure 4 can be easily characterized. The joint probability distribution of $(\mathbf{x}_t, \mathbf{x}_{t+1})$ is a multi variate normal whose Lyapunov equation (equation of the covariance matrix) is given by

$$\mathbf{P}_{t+1} = \mathbf{F}_{t+1}\mathbf{P}_t\mathbf{F}_{t+1}^{\mathrm{T}} + \mathbf{Q}_{t+1} \tag{4.42}$$

Hence the covariance matrix of $(\mathbf{x}_t, \mathbf{x}_{t+1})$ is given by

$$\begin{bmatrix} \mathbf{P}_t & \mathbf{P}_t\mathbf{F}_{t+1}^{\mathrm{T}} \\ \mathbf{F}_{t+1}\mathbf{P}_t & \mathbf{F}_{t+1}\mathbf{P}_t\mathbf{F}_{t+1}^{\mathrm{T}} + \mathbf{Q}_{t+1} \end{bmatrix} \tag{4.43}$$

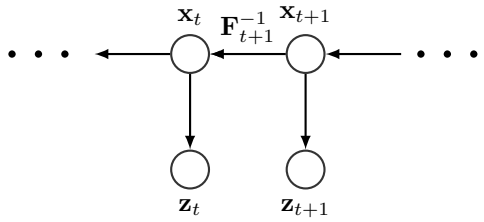


Figure 5. Inverted arrows for the SSM with no observations

The underlying idea in this approach is to invert the arrows in the graphical model leading to a new graphical model given by

Mathematically, this imply to change the relationship and now solve for $\mathbf{P}_t$ in terms of $\mathbf{P}_{t+1}$ using equation (4.42). We get:

$$\mathbf{P}_t = \mathbf{F}_{t+1}^{-1}(\mathbf{P}_{t+1} - \mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-\mathrm{T}} \tag{4.44}$$

where we have assumed that $\mathbf{F}_{t+1}$ is invertible. As a matter of fact, if it is not the case, we can rewrite everything with an approaching matrix to $\mathbf{F}_{t+1}$ (in the sense of the Frobenius norm) that is invertible, whose distance is less than ε, derive all the relationship below and then take the limit as ϵ tends to zero. We will therefore assume in the following that $\mathbf{F}_{t+1}$ is invertible as this is not a strict condition for our derivation. Hence we can rewrite the covariance matrix now as follows:

$$\begin{bmatrix} \mathbf{F}_{t+1}^{-1}(\mathbf{P}_{t+1} - \mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-\mathrm{T}} & \mathbf{F}_{t+1}^{-1}(\mathbf{P}_{t+1} - \mathbf{Q}_{t+1}) \\ (\mathbf{P}_{t+1} - \mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-T} & \mathbf{P}_{t+1} \end{bmatrix} \tag{4.45}$$

If we define the new matrix $\widetilde{\mathbf{F}}_{t+1}$ as:

$$\widetilde{\mathbf{F}}_{t+1} = \mathbf{F}_{t+1}^{-1}(\mathbf{I} - \mathbf{Q}_{t+1}\mathbf{P}_{t+1}^{-1}) \tag{4.46}$$

we can simplify the covariance matrix as follows:

$$\begin{bmatrix} \widetilde{\mathbf{F}}_{t+1}\mathbf{P}_{t+1}\mathbf{F}_{t+1}^{-T} & \widetilde{\mathbf{F}}_{t+1}\mathbf{P}_{t+1} \\ \mathbf{P}_{t+1}\widetilde{\mathbf{F}}_{t+1}^{\mathrm{T}} & \mathbf{P}_{t+1} \end{bmatrix} \tag{4.47}$$

As this looks like a forward covariance matrix. Recall that the forward dynamics is defined by:

$$\mathbf{x}_{t+1} = \mathbf{F}_{t+1}\mathbf{x}_t + \mathbf{B}_{t+1}\mathbf{u}_{t+1} + w_{t+1} \tag{4.48}$$

The following proposition gives the inverse dynamics:

Proposition 6. The inverse dynamics is given by:

$$\mathbf{x}_t = \widetilde{\mathbf{F}}_{t+1}\mathbf{x}_{t+1} + \widetilde{\mathbf{B}}_{t+1}\mathbf{u}_{t+1} + \widetilde{w}_{t+1} \tag{4.49}$$

with

$$\widetilde{w}_{t+1} = -\mathbf{F}_{t+1}^{-1}(w_{t+1} - \mathbf{Q}_{t+1}\mathbf{P}_{t+1}^{-1}\mathbf{x}_{t+1}) \quad (4.50)$$
$$\widetilde{\mathbf{B}}_{t+1} = -\mathbf{F}_{t+1}^{-1}\mathbf{B}_{t+1} \quad (4.51)$$

and with $\widetilde{w}_{t+1}$ independent of the *past* information $\mathbf{x}_{t+1}, \dots, \mathbf{x}_T$ and with the covariance matrix of $\widetilde{w}_{t+1}$ given by

$$\begin{aligned}\widetilde{\mathbf{Q}}_{t+1} &\triangleq \mathbb{E}[\widetilde{w}_{t+1}\widetilde{w}_{t+1}^{\mathrm{T}}] \\ &= \mathbf{F}_{t+1}^{-1}\mathbf{Q}_{t+1}(\mathbf{I} - \mathbf{P}_{t+1}^{-1}\mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-\mathrm{T}}, \quad (4.52)\end{aligned}$$

and with the forward Lyapunov equation given by:

$$\mathbf{P}_t = \widetilde{\mathbf{F}}_t\mathbf{P}_{t+1}\widetilde{\mathbf{F}}_t^{\mathrm{T}} + \widetilde{\mathbf{Q}}_{t+1} \quad (4.53)$$

Proof. The proof is straightforward and given for instance in (Jordan, 2016). □

The reasoning followed here gives us a new way to generate an information filter but now backward. We should note that the conversion between the state space and the observable variable is unchanged and given by

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$$

We can now apply the information filter developed previously to get the new filtering equations using canonical instead of moment parameterization. We get the following new information filter :

Proposition 7. The modified Bryson–Frazier as presented in (Bierman, 2006) is given by:

$$\widetilde{\mathbf{M}}_t = \mathbf{F}_t^T\mathbf{Q}_t^{-1}(\mathbf{\Lambda}_{t+1|t+1} + \mathbf{Q}_t^{-1} - \mathbf{P}_t^{-1})^{-1} \quad (4.54)$$
$$\mathbf{\Lambda}_{t|t+1} = \mathbf{F}_t^T(\mathbf{Q}_t - \mathbf{Q}_t\mathbf{P}_t^{-1}\mathbf{Q}_t)^{-1}\mathbf{F}_t - \widetilde{\mathbf{M}}_t\mathbf{Q}_t^{-1}\mathbf{F}_t \quad (4.55)$$
$$\hat{\eta}_{t|t+1} = \widetilde{\mathbf{M}}_t\hat{\eta}_{t+1|t+1} - \mathbf{\Lambda}_{t|t+1}\mathbf{F}_t\mathbf{B}_t\mathbf{u}_t \quad (4.56)$$
$$\hat{\eta}_{t|t} = \hat{\eta}_{t|t+1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t \quad (4.57)$$
$$\mathbf{\Lambda}_{t|t} = \mathbf{\Lambda}_{t|t+1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t \quad (4.58)$$

Proof. The proof consists in combining all previous results and is given in (Jordan, 2016). □

Remark 4.5. If we want to convert to to the moment representation, we can use the inverse transform given by $\hat{\mathbf{x}}_{t|t+1} = \mathbf{S}_{t|t+1}^{-1}\hat{\eta}_{t|t+1}$ and $\mathbf{P}_{t|t+1} = \mathbf{S}_{t|t+1}^{-1}$

4.5. Inferring final posterior distribution

This problem consists in inferring the final posterior distribution $\mathbb{P}(\mathbf{x}_t \mid \mathbf{z}_1, \dots, \mathbf{z}_T)$. We can easily derive

$$\hat{\mathbf{x}}_{t|T} = \mathbf{P}_{t|T}(\mathbf{P}_{t|t}^{-1}\hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t+1}^{-1}\hat{\mathbf{x}}_{t|t+1}) \quad (4.59)$$

and

$$\mathbf{P}_{t|T} = \left(\mathbf{P}_{t|t}^{-1} + \mathbf{P}_{t|t+1}^{-1} - \Sigma_t^{-1}\right)^{-1} \quad (4.60)$$

4.6. Parameter estimation

4.6.1. Intuition

The expectation–maximization (EM) algorithm is an iterative method to find the maximum likelihood or maximum a posteriori (MAP) estimates of the parameters of our graphical model.

This approach relies on the fact that the model depends on non observable (also called latent) variables. The EM algorithm alternates between performing an expectation (E) step, which provides the expectation of the conditional log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

The EM algorithm was explained and given its name in a classic 1977 paper by (Dempster et al., 1977). The intuition is to find a way to solve the maximum likelihood solution for a model with latent variables. Since variables are hidden, solving the maximum likelihood solution typically requires taking the derivatives of the likelihood function with respect to all the unknown values and then solving the resulting equations. Because of latent variables, this function is unknown and the problem can not be solved. Instead, we compute the expected maximum likelihood with respect to the latent variables. We can then solve this explicit maximum likelihood function with respect to the observable variables.

4.6.2. EM Algorithm

We assume that our statistical model has a set $\mathbf{X}$ of observed data and a set of unobserved latent $\mathbf{Z}$ that depend on unknown parameters $\boldsymbol{\theta}$ that we want to determine thanks to maximum likelihood. We obviously know the likelihood function $L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ given by

$$L(\boldsymbol{\theta}; \mathbf{X}) = p(\mathbf{X}|\boldsymbol{\theta}) = \int p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})d\mathbf{Z}$$

The later is often intractable as the number of values for $\mathbf{Z}$ is vary large making the computation of the expectation

(the integral) extremely difficult. In order to avoid the exact computation, we can work as follows:

First compute the expectation explicitly (called the *E step*) by taking the expected value of the log likelihood function of $\boldsymbol{\theta}$ with respect to latent variables $\mathbf{Z}$, denoted by $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}$:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}\left[\log L(\boldsymbol{\theta};\mathbf{X},\mathbf{Z})\right]$$

We then maximize the resulted function with respect to the parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

The EM method works both for discrete or continuous random latent variables. It takes advantage of algoirthm like the Viterbi algorithm for hidden Markov models or recursive equation fo Kalman filter. The various steps are the following:

Algorithm 1 EM algorithm:

init: Set $\boldsymbol{\theta}$ to some random values.
while Not Converged **do**
  Compute the probability of each possible value of $\mathbf{Z}$, given $\boldsymbol{\theta}$ (E-step)
  Use the computed values of $\mathbf{Z}$ to find the argmax values for $\boldsymbol{\theta}$ (M-Step).
end while

The convergence of this algorithm is given by the following proposition.

Proposition 8. The EM algorithm converges monotonically to a local minimum for the marginal log likelihood.

Proof. There are at multiple two ways of proving this results that are provided in (Jordan, 2016) □

5. L1 regularization and implied scarcity

We add in our model a $L1$ regularisation of our parameters to ensure scarcity of our parameters. Indeed L_1 regularization is more likely to create 0 weights in our model and is quite intuitive. To keep things simple, consider a model consisting of the weights $(w_1, w_2, .., w_m)$. L_1 penalize the model by a loss function $L_1(w) = \sum_i |w_i|$ while $L2$ regularizationuses a loss function $L_2(w) = \frac{1}{2}\sum_i w_i^2$.

When using gradient descent, you will iteratively make the weights change in the opposite direction of the gradient with a learning rate α that is multiplied to the gradient. As the gradient is not constant over time, we will incur different gradient step. In particular for large gradient, we will incur a larger step, while a more flattish gradient, we will incur a smaller step. Let us look at the sub-gradients to build some intuition in terms of the impact of the regularisation.

For L_1 regularization, this writes as:

$$\frac{\partial L_1(w)}{\partial w} = sign(w), \tag{5.1}$$

where $sign(w) = (\frac{w_1}{|w_1|}, \frac{w_2}{|w_2|}, ..., \frac{w_m}{|w_m|})$ while for L_2 regularization, this is simply

$$\frac{\partial L_2(w)}{\partial w} = w \tag{5.2}$$

If we plot the loss function and it's derivative (or (sub-gradient in case of L_1) for a model consisting of just a single parameter, it shows the profound difference between these two types of regularisation:

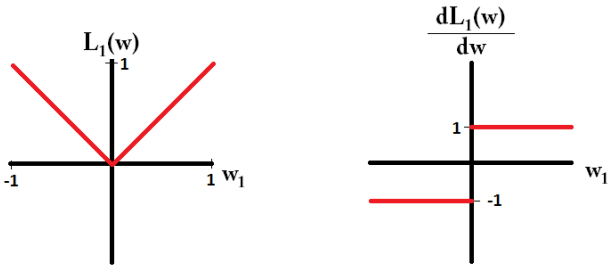


Figure 6. L_1 regularization for a single parameter

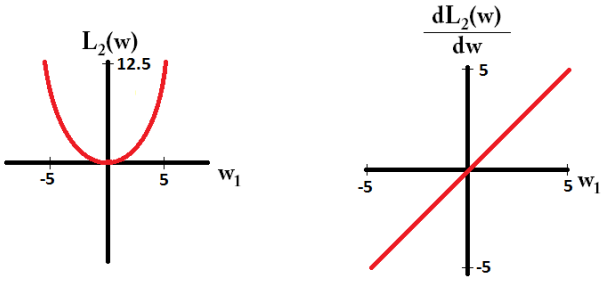


Figure 7. L_2 regularization for a single parameter

We can remark that for L_1 regularization (figure 6), the gradient is either 1 or -1, except for when the weight is null $w_1 = 0$. That means that L_1 regularization will shift any weight towards 0 with the same step size, regardless the initial weight's value. In contrast, the L_2 gradient (as presented in figure 7) is linearly decreasing towards 0 as the weight goes towards 0. Therefore, L_2 regularization will also move any weight towards 0, but at a path smaller and smaller as the weight approaches to 0.

This intuitively explains why L_1 regularization tends to create scarcity in models. This has two important implication for us:

- L_1 regularization favors a very limited number of factors

- in contrast to traditional methods like PCA, we can use factors that are not orthogonal as the scarcity generated by L_1 regularization does not imply to use orthogonal factors or to transform factors. This is very powerful as it does not modify factors and makes the approach very understandable in contrast to factor analysis methods based on PCA that transform factors and make the interpretation much harder

6. Decoding a fund

We apply our method to decode an existing global sustainable equity fund whose name is voluntarily anonymous. As stated in the technical section, decoding implies to only take historical reported prices for both the corresponding fund and the factors. We use this bare information to back out the most probable and stable allocations within the investment universe. We apply our reinforcement learning based graphical model method to find these allocations. We are able to replicate the fund with a very high correlation as illustrated in figure 9.

Quality of Ai For Alpha Decoding

Correlation	97%
Tracking Error	3.8%
Annual Turnover	139%

Figure 8. Decoding efficiency

Figure 8 indicates a correlation between the fund and the replicated strategy of 97%. This is a very high correlation indicating that the method is able to clearly identify most predictive weights. This comes with a turnover of 1.39, which is low and confirms that the allocation rules are stable over time. It validates that our machine learning model is able to identify stable and accurate weights. The corresponding tracking error is logically quite low: 3.8%.

As we are able to decode efficiently this fund, it comes at no surprise that we have similar annual returns, volatility and Sharpe ratio, between the fund (reported as the strategy) and the decoded strategy reported as Ai for Alpha. This is reported in figure 9. Hence we get 11.7% versus 11.2% for annual return, 16.1% versus 16.1% for annual volatility, and very similar numbers for Sharpe ratio, maximum draw down and return over maximum draw-down. In this specific case, it is impressive that we are able to recover the maximum draw-down indicating that the decoded strategy is really in par during challenging times with the funds. In our case, maximum draw-down, denoted by Max DD, is 64.4% versus 63.1% for the decoded strategy.

XXXXx Global Sustainable Equity Fund

Performance Statistics

	Strategy	Ai For Alpha
Annual Return	11.7%	11.2%
Annual Volatility	16.1%	16.1%
Sharpe Ratio	0.73	0.70
Max DD	64.4%	63.1%
Return/Max DD	18.2%	17.7%

Figure 9. Decoding statistics

As reported in figure 10, decoding gives us a synthetic view of the risks and factors related to our funds. Although it is a sustainable equity fund, decoding emphasizes that this fund has some substantial exposure to domestic US and European equity factors, namely with 55 and 42 percents at the date of the decoding (which is January 20 2022). These two exposures have been quite stable over time as reported by the different columns named Today, 3 month ago and 1 year ago. Likewise, we have a reported Dollar exposure that oscillates between 46 and 48 percents. This indicates that the fund does not hedge the currency risk between the US And European equity geographical sectors, hence a positive exposure to the dollar.

At the factor level, there are major exposures to Quality, Growth (short Value) and SRI, which are consistent with the fund objective. The quality factor dominates and has risen over the last year from 32% to 44%, while the short position on value has also increased from -11% to -24% . It is interesting to see that although this fund is labelled as an SRI fund, it has a substantial quality factor exposure which is not obvious when reading the prospectus. One would have anticipated the dominating factor to be the SRI, which is not the case and is currently (as reported in the today column that corresponds to Jan 20 2022) the third factor after Quality and Value (44.2% for quality, -24.2% for Value and 24.1% for SRI). Exposures to the 4 factors Quality, Momentum, Value and SRI are quite stable over time as reported by the graphic below the table in figure 10.

Replication main exposures

Asset		Today	3M ago	1Y ago
Equity	US	55.0%	54.6%	59.4%
Equity	EU	42.0%	38.1%	42.6%
Dollar		46.4%	48.1%	48.7%

Equity Factor	Today	3M ago	1Y ago
L/S Quality	44.2%	42.5%	31.9%
L/S Momentum	7.0%	2.6%	10.3%
L/S Value	-24.2%	-16.9%	-11.2%
L/S SRI	24.1%	22.8%	27.5%

*Each specific Equity Factor (Value, Quality, Momentum, SRI,) is defined by a long position on the MSCI Factor and a short position on the global MSCI Index.

Historical allocation to Equity Factors

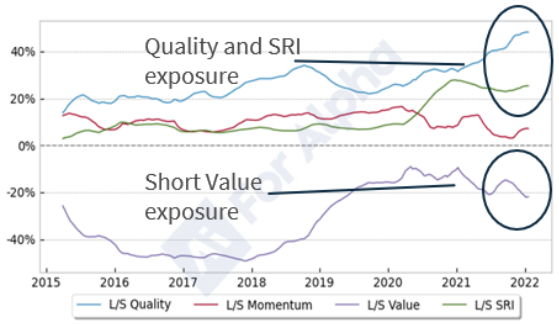


Figure 10. Replication main exposures

Figure 11 gives us the dependence to individual factors. As anticipated the highest exposure is on Equity, 83% and leads to the highest correlation, 89%. We can also compute the exposure to Commodity if we force our model to include the BCOM index (which is the general Bloomberg commodity index) and see that there is a small correlation. Dollar exposure is coherent with previous results.

Dependence to individual Factors

Asset		Correlation	Exposure
Equity	US S&P 500	89%	83%
Commodity	BCOM Industrial Metals	23%	19%
Dollar	Dollar Index	19%	44%

Figure 11. Dependence to individual factors

Figure 12 gives us some additional information in terms of performance attribution. Decoding indicates an evenly distributed performance contribution between the decline in equities and the decline in the exposure factors for the current month as shown by the two blue circles in figure 11. It also highlights that the fund is manly driven by Equities and specifically equities from the S&P 500, with a style tilted towards growth. This is indicated in the 1 year column with a total performance of 19.2% mainly explained by an equity performance of 19.7%. Overall, we can draw at least various conclusion from this show case example. First of all, decoding is effective in recovering implicit positions of a fund on a selected universe. Thanks to the stability and some modeling factor scarcity conditions, we are able to concentrate on effective factors and avoid spurious correlations. Using this top down decoding method, we can not only see factor exposures driving the fund but also source of performance and identify style change or drift.

Performance Attribution

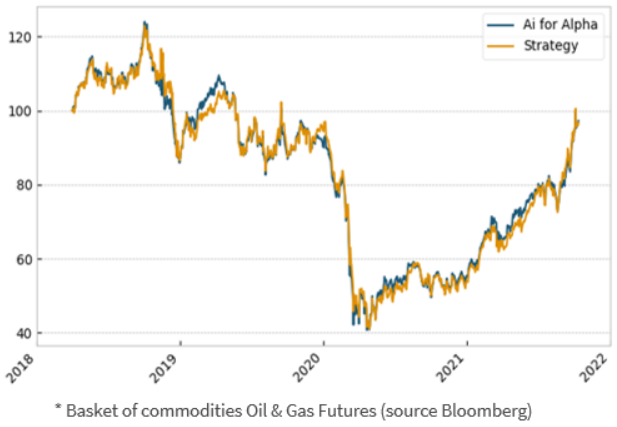
Asset	Daily	MTD	YTD	3M	1Y
Equity	-0.4%	-3.4%	-3.4%	1.2%	19.7%
Dollar	-0.1%	-0.1%	-0.1%	0.9%	3.1%
L/S Quality	0.0%	-1.5%	-1.5%	-1.0%	-0.9%
L/S Momentum	0.0%	-0.3%	-0.3%	-0.7%	-1.8%
L/S Value	0.1%	-1.0%	-1.0%	-1.0%	-0.6%
L/S SRI	0.0%	-0.8%	-0.8%	-0.5%	-0.4%
Total	-0.4%	-6.9%	-6.9%	-1.1%	19.2%

Figure 12. Performance attribution

7. Decoding a strategy

What is even more interesting lies in the generality of the method. Logically, decoding is not limited to decoding funds and can apply to an investment theme. Let us say we are interested in decoding of a macro theme, an energy theme thanks to the replication of the Bloomberg commodity index (BCOM Energy index) applied to European stocks as displayed in figure 13.

BCOM Energy *



* Basket of commodities Oil & Gas Futures (source Bloomberg)

Figure 13. Decoding of BCOM Energy index

The corresponding European stocks are the constituents of the Eurostoxx 600 index. This decoding exercise is inter-

esting for multiple reasons. It applies to a large universe namely 600 stocks. It tries to replicate an energy investment theme which is challenging to include in UCITs funds. Hence mapping this complicated underlying into stocks makes it trivial to be used in UCITs funds. It captures implicit correlation and relationship between stocks and energy that may seem quite non correlated at firt sight but are indeed related with time-varying correlation. Compared to traditional replication methods, decoding goes beyond. Indeed, traditional methods would can be summarized into two approaches that are very limited compared to decoding. Either one use constant weights in her/his replication and hence perform poorly given the dynamic correlation between stocks and the BCOM Energy index and its continuous modification over time. Or one tries to use time varying weights through regression and will incur very high turnover and unstable rules.

Quality of decoding

- **Correlation**: 97%
- **Turnover**: 7.1
- **Residual Error**: 8% vs Strategy volatility: 33%

Figure 14. Decoding efficiency

As reported in figure 14, we obtained like for the previous example a very high correlation: 97% with the initial strategy and a moderate turnover (7.1) given that it is a long short strategy and hence generates much higher turnover than long only strategies. Residual error is tiny about 8% compared to the strategy volatility of 33%. The match between the two curves is really impressive and translates graphically the efficiency of the decoding method.

Figure 15 illustrates that decoding the BCOM Energy index requires long and short positions. The most obvious long positions are quite natural and composed of energy related stocks like the Oil Major Shell PLC. This is not surprising as BCOM Energy and Shell PLC are highly correlated. The second most important stock is Eiffage which is a construction company emphasizing connections between construction and energy resources. There are other obvious stocks selected like Galp Energia, Anglo American PLC or OMV on the long side. However, we can see through figure 16 that the companies are quite diverse and are spread across multiple sectors. On the short side, selected stocks are less obvious, with companies like Tenaris, Yara International, Gerresheirmer, Rotork Controls, Ackermans & van Haaren, Next plc, Adidas, Powszechny Zakład, Franz Colryut, Interk or Barry Callebaut.

Figure 16 emphasizes that decoding finds very diversified sectors that are not only related to the energy or more generally commodity sector indicating that the BCOM Energy index has some positive and negative correlation with European stocks.

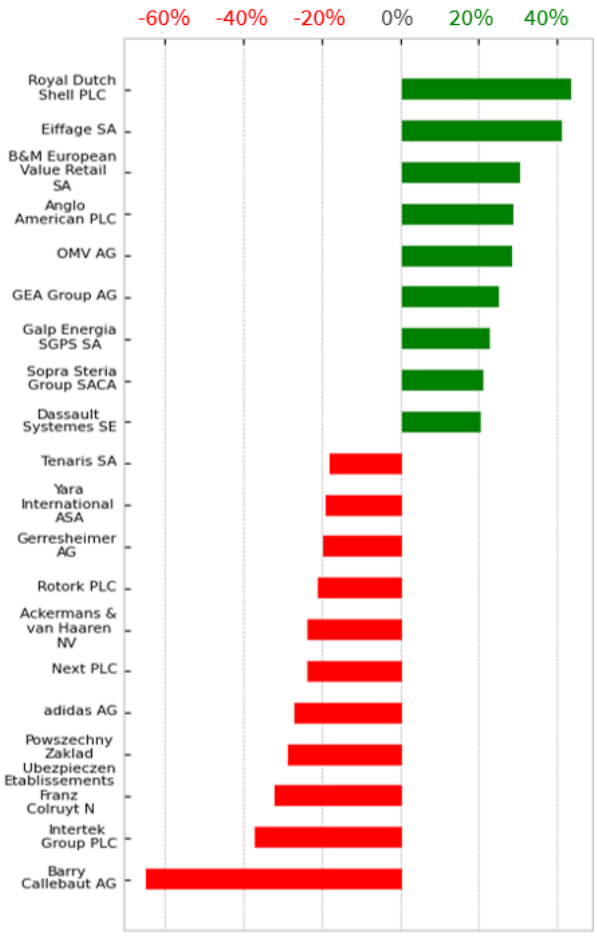


Figure 15. Stock exposure

8. Conclusion

Overall, machine learning enables us decoding efficiently strategies or funds. The underlying method relies on graphical models to capture dynamics among replication coefficients and scarcity of selected factors to reduce overall turnover and ensure stability. It comes at no surprise that reinforcement learning is efficient to find hyper-parameters thanks to a reward combining tracking error and turnover. We experiment high accuracy on practical cases illustrating the interest of this approach, with correlations between the strategy to decode and the generated replicating strategy close to one as well as relatively low turnover.

Company	sector
Shell plc	oil and gas
Eiffage	construction
B&M European value	retail
Anglo American plc	mining
OMV	oil gas and petrochemical
GEA Group	food & beverages
Galp Energia	energy
Sopra Steria	consulting
Dassault Systèmes	software
Tenaris	pipes supplier
Yara International	chemical
Gerresheirmer	manufacturing
Rotork Controls	industrial
Ackermans & van Haaren	diversified group
Next plc	retailer
Adidas	sportswear
Powszechny Zakład ..	insurance
Franz Colryut	retailer
Interk	quality assurance
Barry Callebaut	cocoa and chocolate

Figure 16. Stocks and sectors

References

Anderson, B. and Moore, J. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2002. ISBN 471221279.

Benhamou, E. Variance reduction in actor critic methods (ACM). *CoRR*, abs/1907.09765, 2019. URL http://arxiv.org/abs/1907.09765.

Benhamou, E., Atif, J., Laraki, R., and Saltiel, D. NGO-GM: Natural gradient optimization for graphical models. *SSRN Electronic Journal*, 01 2019. doi: 10.2139/ssrn.3387874.

Benhamou, E., Saltiel, D., Ungari, S., and Mukhopadhyay, A. Bridging the gap between markowitz planning and deep reinforcement learning. In *Proceedings of the 30th International Conference on Automated Planning and Scheduling (ICAPS): PRL*. AAAI Press, 2020a.

Benhamou, E., Saltiel, D., Ungari, S., and Mukhopadhyay, A. Time your hedge with deep reinforcement learning. In *Proceedings of the 30th International Conference on Automated Planning and Scheduling (ICAPS): FinPlan*. AAAI Press, 2020b.

Benhamou, E., Saltiel, D., Ungari, S., and Mukhopadhyay, A. Aamdrl: Augmented asset management with deep reinforcement learning. *arXiv*, 2020c.

Benhamou, E., Saltiel, D., Ohana, J.-J., and Atif, J. Detecting and adapting to crisis pattern with context based deep reinforcement learning. In *International Conference on Pattern Recognition (ICPR)*. IEEE Computer Society, 2021a.

Benhamou, E., Saltiel, D., Ohana, J. J., Atif, J., and Laraki, R. Deep reinforcement learning (drl) for portfolio allocation. In Dong, Y., Ifrim, G., Mladenić, D., Saunders, C., and Van Hoecke, S. (eds.), *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track*, pp. 527–531, Cham, 2021b. Springer International Publishing.

Benhamou, E., Saltiel, D., Ungari, S., and Abhishek Mukhopadhyay, Jamal Atif, R. L. Knowledge discovery with deep rl for selecting financial hedges. In *AAAI: KDF*. AAAI Press, 2021c.

Benidis, K., Feng, Y., and P. Palomar, D. Sparse portfolios for high-dimensional financial index tracking. *IEEE Transactions on Signal Processing*, 66(1):155–170, 2018.

Bierman, G. *Factorization Methods for Discrete Sequential Estimation*. Dover Books on Mathematics Series. Dover Publications, 2006. ISBN 9780486449814.

Brown, R. G. and Hwang, P. Y. C. *Introduction to random signals and applied kalman filtering: with MATLAB exercises and solutions; 3rd ed.* Wiley, New York, NY, 1997.

Bui, H. H., Venkatesh, S., and West, G. Hidden markov models. In *Hidden Markov Models*, chapter Tracking and Surveillance in Wide-area Spatial Environments Using the Abstract Hidden Markov Model, pp. 177–196. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002. ISBN 981-02-4564-5. URL http://dl.acm.org/citation.cfm?id=505741.505750.

Canakgoz, N. A. and Beasley, J. E. Mixed-integer programming approaches for index tracking and enhanced indexation. *European Journal of Operational Research*, 196(1):384–399, 2009.

Cappe, O., Moulines, E., and Ryden, T. *Inference in Hidden Markov Models*. Springer Publishing Company, Incorporated, 2010. ISBN 1441923195, 9781441923196.

Dean, T. and Kanazawa, K. A model for reasoning about persistence and causation. *Comput. Intell.*, 5(3):142–150, December 1989. ISSN 0824-7935. doi: 10.1111/j.1467-8640.1989.tb00324.x. URL http://dx.doi.org/10.1111/j.1467-8640.1989.tb00324.x.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

Durbin, J. and Koopman, S. *Time Series Analysis by State Space Methods*. Oxford Statistical Science Series. OUP Oxford, 2012. ISBN 9780191627194. URL https://books.google.fr/books?id=lGyshsfkLrIC.

Fastrich, B., Paterlini, S., and Winker, P. Cardinality versusq-norm constraints for index tracking. *Quantitative Finance*, 14(11):2019–2032, 2014.

Fine, S., Singer, Y., and Tishby, N. The hierarchical hidden markov model: Analysis and applications. *Mach. Learn.*, 32(1):41–62, July 1998. ISSN 0885-6125. doi: 10.1023/A:1007469218079. URL http://dx.doi.org/10.1023/A:1007469218079.

Gelb, A. *Applied Optimal Estimation*. The MIT Press, 1974. ISBN 0262570483, 9780262570480.

Ghahramani, Z. and Jordan, M. I. Supervised learning from incomplete data via an em approach. In Cowan, J. D.,

Tesauro, G., and Alspector, J. (eds.), *Advances in Neural Information Processing Systems 6*, pp. 120–127. Morgan-Kaufmann, 1994.

Jordan, M. I. *An introduction to probabilistic graphical models*. Berkeley, 2016. URL http://people.eecs.berkeley.edu/~jordan/prelims/.

Kitagawa, G. Non-gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1063, 1987.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193, 9780262013192.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *CoRR*, abs/1805.00909, 2018. URL http://arxiv.org/abs/1805.00909.

Murphy, K. P. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002. AAI3082340.

Murphy, K. P. *Machine learning : a probabilistic perspective*. MIT Press, August 2013. ISBN 262018020. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262018020.

Murphy, K. P. and Paskin, M. A. Linear time inference in hierarchical hmms. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pp. 833–840, Cambridge, MA, USA, 2001. MIT Press. URL http://dl.acm.org/citation.cfm?id=2980539.2980647.

Ostendorf, M., Digalakis, V. V., and Kimball, O. A. From hmm's to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(5):360–378, September 1996. ISSN 1063-6676. doi: 10.1109/89.536930.

Petersen, K. B. and Pedersen, M. S. *The Matrix Cookbook*. Technical University of Denmark, November 2012. URL http://www2.imm.dtu.dk/pubdb/p.php?3274. Version 20121115.

Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *PROCEEDINGS OF THE IEEE*, pp. 257–286, 1989.

Rabiner, L. R. and Juang, B. H. An introduction to hidden markov models. *IEEE ASSp Magazine*, 1986.

Rauch, H. E., Tung, F., and Striebel, C. T. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965.

Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009. ISBN 0136042597, 9780136042594.

Saul, L. K. and Jordan, M. I. Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Machine Learning*, 37 (1), October 1999. ISSN 1573-0565. doi: 10.1023/A:1007649326333. URL https://doi.org/10.1023/A:1007649326333.

Smyth, P., Heckerman, D., and Jordan, M. I. Probabilistic independence networks for hidden markov probability models. *Neural Comput.*, 9(2):227–269, February 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.2.227. URL http://dx.doi.org/10.1162/neco.1997.9.2.227.

Strub, O. and Baumann, P. Optimal construction and rebalancing of index-tracking portfolios. *European Journal of Operational Research*, 264(1):370–387, 2018.

Takeda, A., Niranjan, M., ya Gotoh, J., and Kawahara, Y. Simultaneous pursuit of out-of-sample performance and sparsity in index tracking portfolios. *Computational Management Science*, 10(1):21–49, 2013.

Thornton, C. *Triangular Covariance Factorizations for Kalman Filtering*. NASA-CR-149 147. University of California, Los Angeles–Engineering, 1976.

Appendix

A. Beyond Kalman filter

The proof is widely known and presented in various books (see for instance (Gelb, 1974) or (Brown & Hwang, 1997)). We present it here for the sake of completeness. It consists in three steps:

- Derive the posteriori estimate covariance matrix
- Compute the Kalman gain
- Simplify the posteriori error covariance formula

A.1. Step 1: A posteriori estimate covariance matrix

Proof. The error covariance $\mathbf{P}_{t|t}$ is defined as the covariance between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$: $\mathbf{P}_{t|t} = \mathrm{Cov}\left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}\right)$. We can substitute in this equation the definition of $\hat{\mathbf{x}}_{t|t}$ given by equation (4.19) to get:

$$\mathbf{P}_{t|t} = \mathrm{Cov}\left[\mathbf{x}_t - \left(\hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{y}}_t\right)\right] \tag{A.1}$$

Using the definition of $\tilde{\mathbf{y}}_t$ from equation (4.16), we get:

$$\mathbf{P}_{t|t} = \mathrm{Cov}\left(\mathbf{x}_t - \left[\hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}\right)\right]\right) \tag{A.2}$$

Using the measurement equation for $\mathbf{z}_t$, this transforms into:

$$\mathbf{P}_{t|t} = \mathrm{Cov}\left(\mathbf{x}_t - \left[\hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\left(\mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}\right)\right]\right) \tag{A.3}$$

By collecting the error vectors, this results in:

$$\mathbf{P}_{t|t} = \mathrm{Cov}\left[\left(\mathbf{I} - \mathbf{K}_t\mathbf{H}_t\right)\left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}\right) - \mathbf{K}_t\mathbf{v}_t\right] \tag{A.4}$$

Since the measurement error v_k is not correlated with the other terms, this simplifies into:

$$\mathbf{P}_{t|t} = \mathrm{Cov}\left[\left(\mathbf{I} - \mathbf{K}_t\mathbf{H}_k\right)\left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}\right)\right] + \mathrm{Cov}\left[\mathbf{K}_t\mathbf{v}_t\right] \tag{A.5}$$

By property of covariance for matrix: $\mathrm{Cov}(A\mathbf{x}_t) = A\,\mathrm{Cov}(\mathbf{x}_t)A^{\mathrm{T}}$ and using the definition for $\mathbf{P}_{t|t-1}$ and $\mathbf{R}_t = \mathrm{Cov}\left(\mathbf{v}_t\right)$, we get the final result:

$$\mathbf{P}_{t|t} = (\mathbf{I}-\mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1}(\mathbf{I}-\mathbf{K}_t\mathbf{H}_t)^{\mathrm{T}}+\mathbf{K}_t\mathbf{R}_t\mathbf{K}_t^{\mathrm{T}} \tag{A.6}$$

This last equation is referred to as the *Joseph form* of the covariance update equation. It is true for any value of Kalman gain $\mathbf{K}_t$ no matter if it is optimal or not. In the special case of optimal Kalman gain, this further reduces to equation (4.20) which we will prove in A.3 □

A.2. Step 2: Kalman gain

Proof. The goal of the Kalman filter is to find the minimum mean-square error estimator. It is obtained by minimizing the error of the *a posteriori* state given by $\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}$. This error is stochastic, hence we are left with minimizing the expected value of the square of the L_2 norm of this vector given by $\mathrm{E}\left[\left\|\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}\right\|^2\right]$. Since the error of the *a posteriori* state follows a normal distribution with zero mean and covariance given by $\mathbf{P}_{t|t}$, the optimization program is equivalent to minimizing the matrix trace of the a posteriori estimate covariance matrix $\mathbf{P}_{t|t}$.

Expanding out and collecting the terms in equation (A.6), we find:

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{H}_t\mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{K}_t^{\mathrm{T}} + \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^{\mathrm{T}} \tag{A.7}$$

This minimization program is very simple. It is quadratic in the Kalman gain, $\mathbf{K}_t$. The optimum is obtained when the derivative with respect to $\mathbf{K}_t$ is null. The equation for the critical point writes as:

$$\frac{\partial \mathrm{tr}(\mathbf{P}_{t|t})}{\partial\,\mathbf{K}_t} = -2(\mathbf{H}_t\mathbf{P}_{t|t-1})^{\mathrm{T}} + 2\mathbf{K}_t\mathbf{S}_t = 0. \tag{A.8}$$

We can solving for $\mathbf{K}_t$ to find the *optimal* Kalman gain as follows:

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{S}_t^{-1} \tag{A.9}$$

□

A.3. Step 3: Simplification of the posteriori error covariance formula

Proof. We can work on our formula further to simplify equation (A.6) as follows. A trick is to remark that in equation (A.9), multiplying on the right both sides by $\mathbf{S}_t\mathbf{K}_t^{\mathrm{T}}$, we find that:

$$\mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^{\mathrm{T}} = \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{K}_t^{\mathrm{T}} \tag{A.10}$$

Injecting this equality in equation (A.7) and noticing that the last two terms cancel out, we get:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1}. \tag{A.11}$$

This last formula appeals some remarks. It makes Kalman filter dammed simple. It is computationally cheap and much cheaper than the equation (A.7). Thus it is nearly always used in practice. However, this equation is only correct for

the optimal gain. If by back luck, arithmetic precision is unusually low due to numerical instability, or if the Kalman filter gain is non-optimal (intentionally or not), equation A.11 does not hold any more. Using it would lead to inaccurate results and can be troublesome. In this particular case, the complete formula given by equation (A.7) must be used. □

B. Factor Analysis model

B.1. Model Presentation and proof

We are interested in the following very simple graphical model called the Gaussian Factor analysis model. It is represented by a two nodes graphical models in figure 17

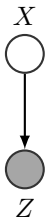


Figure 17. Factor analysis model as a graphical model

We assume that X follows a multi dimensional normal distribution $X \sim \mathcal{N}(0, \Sigma)$. We assume that X is a latent variable and that only Z is observed. We also assume that there is a direct linear relationship between X and Z given by

$$Z = \mu + \Lambda X + W$$

where W is also distributed as a a multi dimensional normal distribution $W \sim \mathcal{N}(0, \Psi)$ independent of X. This is obviously a simplified version of our Linear Gaussian State Space Model and can provide building blocks for deriving the Kalman filter. Trivially, we can compute the unconditional distribution of Z as a normal distribution whose mean is $\mathbb{E}[Z] = \mathbb{E}[\mu + \Lambda X + W] = \mu$. Likewise, it is immediate to compute the unconditional covariance matrix of Z as

$$\begin{aligned} \mathrm{Var}(Z) &= \mathbb{E}[(\mu + \Lambda X + W)(\mu + \Lambda X + W)^{\mathrm{T}}] && \text{(B.1)} \\ &= \Lambda\Sigma\Lambda^{\mathrm{T}} + \Psi && \text{(B.2)} \end{aligned}$$

The covariance between X and Z is also easy to compute and given by

$$\begin{aligned} \mathrm{Cov}(X, Z) &= \mathbb{E}[X(\mu + \Lambda X + W)^{\mathrm{T}}] && \text{(B.3)} \\ &= \Sigma\Lambda^{\mathrm{T}} && \text{(B.4)} \end{aligned}$$

Hence, collecting all our results, we have shown that the joint distribution of X and Z is a Gaussian given by

$$\mathcal{N}\left(\begin{bmatrix} 0 \\ \mu \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma\Lambda^{\mathrm{T}} \\ \Lambda\Sigma & \Lambda\Sigma\Lambda^{\mathrm{T}} + \Psi \end{bmatrix}\right).$$

We shall be interested in computing the conditional distribution of X given Z. Using lemma provided in B.2, we obtained that $X \mid Z$ is a multi dimensional Gaussian whose mean is:

$$\mathbb{E}(X \mid z) = \Sigma\Lambda^{\mathrm{T}}(\Lambda\Sigma\Lambda^{\mathrm{T}} + \Psi)^{-1}(z - \mu) \quad \text{(B.5)}$$

Using the Woodbury matrix inversion formula (see formula (158) in (Petersen & Pedersen, 2012)), this is also

$$\mathbb{E}(X \mid z) = (\Sigma^{-1} + \Lambda^{\mathrm{T}}\Psi^{-1}\Lambda^{-1})\Lambda^{\mathrm{T}}\Psi^{-1}(z - \mu) \quad \text{(B.6)}$$

The two formula are equivalent and one should use the one that is easier to compute depending on the dimension of X and Z. Our partitioned Gaussian lemma B.2 gives us also the conditional variance of X:

$$\begin{aligned} \mathrm{Var}(X \mid z) &= \Sigma - \Sigma\Lambda^{\mathrm{T}}(\Lambda\Sigma\Lambda^{\mathrm{T}} + \Psi)^{-1}\Lambda\Sigma && \text{(B.7)} \\ &= (\Sigma^{-1} + \Lambda^{\mathrm{T}}\Psi^{-1}\Lambda)^{-1} && \text{(B.8)} \end{aligned}$$

where in the last equation, we have used again the Woodbury matrix inversion formula (Woodbury variant formula (157) in (Petersen & Pedersen, 2012)).

B.2. A quick lemma about partitioned Gaussian

Lemma B.1. Let Y be a multivariate normal vector $\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Consider partitioning $\boldsymbol{Y}$, $\boldsymbol{\mu}$, Σ into

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

Then, $(\boldsymbol{y}_1 | \boldsymbol{y}_2 = \boldsymbol{a})$, the conditional distribution of the first partition given the second, is $\mathcal{N}(\overline{\boldsymbol{\mu}}, \overline{\Sigma})$, with mean

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \Sigma_{12}{\Sigma_{22}}^{-1}(\boldsymbol{a} - \boldsymbol{\mu}_2)$$

and covariance matrix

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}{\Sigma_{22}}^{-1}\Sigma_{21}$$

Proof. Let $\mathbf{y}_1$ be the first partition and $\mathbf{y}_2$ the second. Now define $\mathbf{z} = \mathbf{y}_1 + \mathbf{A}\mathbf{y}_2$ where $\mathbf{A} = -\Sigma_{12}\Sigma_{22}^{-1}$. We can easily compute the covariance between z and y as follows:

$$\begin{aligned}\operatorname{cov}(\mathbf{z},\mathbf{y}_2) &= \operatorname{cov}(\mathbf{y}_1,\mathbf{y}_2)+\operatorname{cov}(\mathbf{A}\mathbf{y}_2,\mathbf{y}_2)\\ &= \Sigma_{12}+\mathbf{A}\operatorname{Var}(\mathbf{y}_2)\\ &= 0\end{aligned}$$

Since $\mathbf{z}$ and $\mathbf{y}_2$ are jointly normal and uncorrelated, they are independent. The expectation of $\mathbf{z}$ is trivially computed as $E(\mathbf{z})=\boldsymbol{\mu}_1+\mathbf{A}\boldsymbol{\mu}_2$. Using this, we compute the conditional expectation of $\mathbf{y}_1$ given $\mathbf{y}_2$ as follows:

$$\begin{aligned}E(\mathbf{y}_1|\mathbf{y}_2) &= E(\mathbf{z}-\mathbf{A}\mathbf{y}_2|\mathbf{y}_2)\\ &= E(\mathbf{z}|\mathbf{y}_2)-E(\mathbf{A}\mathbf{y}_2|\mathbf{y}_2)\\ &= E(\mathbf{z})-\mathbf{A}\mathbf{y}_2\\ &= \boldsymbol{\mu}_1+\Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2-\boldsymbol{\mu}_2)\end{aligned}$$

which proves the first part. The second part is as simple. Note that

$$\begin{aligned}\operatorname{Var}(\mathbf{y}_1|\mathbf{y}_2) &= \operatorname{Var}(\mathbf{z}-\mathbf{A}\mathbf{y}_2|\mathbf{y}_2)\\ &= \operatorname{Var}(\mathbf{z}|\mathbf{y}_2)\\ &= \operatorname{Var}(\mathbf{z})\end{aligned}$$

since $\mathbf{z}$ and $\mathbf{y}_2$ are independent. We can trivially conclude using the following last computation:

$$\begin{aligned}\operatorname{Var}(\mathbf{y}_1|\mathbf{y}_2)=\operatorname{Var}(\mathbf{z}) &= \operatorname{Var}(\mathbf{y}_1+\mathbf{A}\mathbf{y}_2)\\ &= \operatorname{Var}(\mathbf{y}_1)\\ &\quad+\mathbf{A}\operatorname{Var}(\mathbf{y}_2)\mathbf{A}^{\mathrm{T}}+\mathbf{A}\operatorname{cov}(\mathbf{y}_1,\mathbf{y}_2)\\ &\quad+\operatorname{cov}(\mathbf{y}_2,\mathbf{y}_1)\mathbf{A}^{\mathrm{T}}\\ &= \Sigma_{11}+\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22}\Sigma_{22}^{-1}\Sigma_{21}\\ &\quad-2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\\ &= \Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\end{aligned}$$

□

B.3. Derivation of the filter equations

Proof. It is easy to start with the precision matrix (the inverse of the covariance matrix). We have

$$\begin{aligned}\boldsymbol{\Lambda}_{t|t-1} &= \mathbf{P}_{t|t-1}^{-1} && \text{(B.9)}\\ &= (\mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}}+\mathbf{Q}_t)^{-1} && \text{(B.10)}\\ &= \mathbf{Q}_t^{-1}-\mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{P}_{t-1|t-1}^{-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\\ &\quad\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} && \text{(B.11)}\\ &= \mathbf{Q}_t^{-1}-\mathbf{Q}_t^{-1}\mathbf{F}_t(\boldsymbol{\Lambda}_{t-1|t-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\\ &\quad\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} && \text{(B.12)}\end{aligned}$$

where in the previous equation, we have used extensively matrix inversion formula. Similarly, applying matrix inversion but for $\boldsymbol{\Lambda}_{t|t}$ (conditioned on t now)

$$\begin{aligned}\boldsymbol{\Lambda}_{t|t} &= \mathbf{P}_{t|t}^{-1} && \text{(B.13)}\\ &= (\mathbf{P}_{t|t-1}-\mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}(\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}+\mathbf{R}_t)^{-1}\\ &\quad\mathbf{H}_t\mathbf{P}_{t|t-1})^{-1} && \text{(B.14)}\\ &= \mathbf{P}_{t|t-1}^{-1}+\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t && \text{(B.15)}\\ &= \boldsymbol{\Lambda}_{t|t-1}+\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t && \text{(B.16)}\end{aligned}$$

We can now handle the parameter η. We have

$$\begin{aligned}\hat{\eta}_{t|t-1} &= \mathbf{P}_{t|t-1}^{-1}\hat{\mathbf{x}}_{t|t-1} && \text{(B.17)}\\ &= \mathbf{P}_{t|t-1}^{-1}\left(\mathbf{F}_t\hat{\mathbf{x}}_{t-1|t-1}+\mathbf{B}_t\mathbf{u}_t\right) && \text{(B.18)}\\ &= \mathbf{P}_{t|t-1}^{-1}\left(\mathbf{F}_t\mathbf{P}_{t-1|t-1}\hat{\eta}_{t-1|t-1}+\mathbf{B}_t\mathbf{u}_t\right) && \text{(B.19)}\\ &= (\mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}}+\mathbf{Q}_t)^{-1}\\ &\quad\left(\mathbf{F}_t\mathbf{P}_{t-1|t-1}\hat{\eta}_{t-1|t-1}+\mathbf{B}_t\mathbf{u}_t\right) && \text{(B.20)}\\ &= \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{P}_{t-1|t-1}^{-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\hat{\eta}_{t-1|t-1}\\ &\quad+(\mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}}+\mathbf{Q}_t)^{-1}\mathbf{B}_t\mathbf{u}_t && \text{(B.21)}\\ &= \mathbf{Q}_t^{-1}\mathbf{F}_t(\boldsymbol{\Lambda}_{t-1|t-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\hat{\eta}_{t-1|t-1}\\ &\quad+(\mathbf{Q}_t^{-1}-\mathbf{Q}_t^{-1}\mathbf{F}_t(\boldsymbol{\Lambda}_{t-1|t-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\\ &\quad\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1})\mathbf{B}_t\mathbf{u}_t && \text{(B.22)}\end{aligned}$$

Likewise, we derive the same type of equations for η but conditioned on t as follows:

$$\begin{aligned}\hat{\eta}_{t|t} &= \mathbf{P}_{t|t}^{-1}\hat{\mathbf{x}}_{t|t} && \text{(B.23)}\\ &= \mathbf{P}_{t|t}^{-1}(\hat{\mathbf{x}}_{t|t-1}+\mathbf{P}_{t|t}\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}(\mathbf{z}_t-\mathbf{H}_t\hat{\mathbf{x}}_{t|t-1})) && \text{(B.24)}\\ &= (\mathbf{P}_{t|t}^{-1}-\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t)\mathbf{P}_{t|t-1}\hat{\eta}_{t|t-1}+\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t && \text{(B.25)}\\ &= (\mathbf{P}_{t|t-1}^{-1}+\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t-\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t)\\ &\quad\mathbf{P}_{t|t-1}\hat{\eta}_{t|t-1}+\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t && \text{(B.26)}\\ &= \hat{\eta}_{t|t-1}+\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t && \text{(B.27)}\end{aligned}$$

Summarizing all these equation leads to the so called filter equations:

$$\begin{aligned}\hat{\eta}_{t|t-1} &= \mathbf{Q}_t^{-1}\mathbf{F}_t(\boldsymbol{\Lambda}_{t-1|t-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\hat{\eta}_{t-1|t-1}\\ &\quad+(\mathbf{Q}_t^{-1}-\mathbf{Q}_t^{-1}\mathbf{F}_t(\boldsymbol{\Lambda}_{t-1|t-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\\ &\quad\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1})\mathbf{B}_t\mathbf{u}_t && \text{(B.28)}\\ \hat{\eta}_{t|t} &= \hat{\eta}_{t|t-1}+\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t && \text{(B.29)}\\ \boldsymbol{\Lambda}_{t|t-1} &= \mathbf{Q}_t^{-1}-\mathbf{Q}_t^{-1}\mathbf{F}_t(\boldsymbol{\Lambda}_{t-1|t-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\\ &\quad\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} && \text{(B.30)}\\ \boldsymbol{\Lambda}_{t|t} &= \boldsymbol{\Lambda}_{t|t-1}+\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t && \text{(B.31)}\end{aligned}$$

which concludes the proof. □

B.4. Proof of RTS recursive equations

Proof. We start by writing down the distribution of $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ conditioned on $\mathbf{z}_1, \ldots, \mathbf{z}_t$. As $\hat{\mathbf{x}}_{t+1|t} = \mathbf{F}_{t+1}\mathbf{x}_t + \mathbf{B}_{t+1}\mathbf{u}_{t+1}$ and using the fact that the control term $\mathbf{B}_{t+1}\mathbf{u}_{t+1}$ is deterministic, we have

$$\begin{aligned}&\mathbb{E}[(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t})^{\mathrm{T}} \mid \mathbf{z}_1, \ldots, \mathbf{z}_t] \\ &= \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}} \end{aligned} \tag{B.32}$$

Thus the vector distribution of $\begin{bmatrix} \mathbf{x}_{t|t} \\ \mathbf{x}_{t+1|t} \end{bmatrix}$ has its mean given by $\begin{bmatrix} \hat{\mathbf{x}}_{t|t} \\ \hat{\mathbf{x}}_{t+1|t} \end{bmatrix}$ and its covariance matrix given by $\begin{bmatrix} \mathbf{P}_{t|t} & \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}} \\ \mathbf{F}_{t+1}\mathbf{P}_{t|t} & \mathbf{P}_{t+1|t} \end{bmatrix}$ Let us now work on the backward recursion. We are interested in computing the distribution of $\mathbf{x}_t$, conditioned on $\mathbf{x}_{t+1}$ and $\mathbf{z}_1, \ldots, \mathbf{z}_t$. Using the Factor analysis model, it is easy to see that

$$\begin{aligned}&\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_t] \\ &= \hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) \end{aligned} \tag{B.33}$$

Similarly, we have

$$\begin{aligned}&\mathrm{Var}\left[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_t\right] \\ &= \mathbf{P}_{t|t} - \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}\mathbf{F}_{t+1}\mathbf{P}_{t|t} \end{aligned} \tag{B.34}$$

But using the Markov property that states that

$$\begin{aligned}&\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \\ &= \mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_t] \end{aligned} \tag{B.35}$$

and

$$\begin{aligned}&\mathrm{Var}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_t] \\ &= \mathrm{Var}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \end{aligned} \tag{B.36}$$

We have

$$\begin{aligned}&\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \\ &= \hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) \end{aligned} \tag{B.37}$$

Similarly, we have

$$\mathrm{Var}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{B.38}$$

$$= \mathbf{P}_{t|t} - \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}\mathbf{F}_{t+1}\mathbf{P}_{t|t}^{-1} \tag{B.39}$$

Using conditional properties states in appendix section B.2, we can work the recursion as follows:

$$\hat{\mathbf{x}}_{t|T} \triangleq \mathbb{E}[\mathbf{x}_t \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{B.40}$$

$$= \mathbb{E}[\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{B.41}$$

$$= \mathbb{E}[\hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{B.42}$$

$$= \hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{B.43}$$

$$= \hat{\mathbf{x}}_{t|t} + \mathbf{L}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{B.44}$$

where

$$\mathbf{L}_t = \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1} \tag{B.45}$$

The latter equation is the basic update equation in the RTS smoothing algorithm. This equation provides an estimate of $\mathbf{x}_t$ based on the filtered estimate $\hat{\mathbf{x}}_{t|t}$ corrected by the convolution of $\mathbf{L}_t$ with the error term $\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}$ that represents the difference between the smoothed estimate of $\mathbf{x}_T$ and the filtered estimate $\hat{\mathbf{x}}_{t+1|t}$. The matrix $\mathbf{L}_t$ can be interpreted as a gain matrix that depends only on forward information. and can be computed in the forward pass.

As for the conditional variance, we have

$$\hat{\mathbf{P}}_{t|T} \triangleq \mathrm{Var}[\mathbf{x}_t \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{B.46}$$

$$= \mathrm{Var}[\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] + \mathbb{E}[\mathrm{Var}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{B.47}$$

$$= \mathrm{Var}[\mathbf{x}_{t|t} + \mathbf{L}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{B.48}$$

$$+ \mathbb{E}[\mathbf{P}_{t|t} - \mathbf{L}_t\mathbf{P}_{t+1|t}\mathbf{L}_t^{\mathrm{T}} \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] \tag{B.49}$$

$$= \mathbf{L}_t \mathrm{Var}[\mathbf{x}_{t|t} \mid \mathbf{z}_1, \ldots, \mathbf{z}_T]\mathbf{L}_t^{\mathrm{T}} + \mathbf{P}_{t|t} - \mathbf{L}_t\mathbf{P}_{t+1|t}\mathbf{L}_t^{\mathrm{T}} \tag{B.50}$$

$$= \mathbf{L}_t\mathbf{P}_{t+1|T}\mathbf{L}_t^{\mathrm{T}} + \mathbf{P}_{t|t} - \mathbf{L}_t\mathbf{P}_{t+1|t}\mathbf{L}_t^{\mathrm{T}} \tag{B.51}$$

$$= \mathbf{P}_{t|t} + \mathbf{L}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{L}_t^{\mathrm{T}} \tag{B.52}$$

We can summarize the RTS smoothing algorithm as follows:

$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + \mathbf{L}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{B.53}$$

$$\hat{\mathbf{P}}_{t|T} = \mathbf{P}_{t|t} + \mathbf{L}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{L}_t^{\mathrm{T}} \tag{B.54}$$

with an initial condition given by

$$\hat{\mathbf{x}}_{T|T} = \hat{\mathbf{x}}_T \tag{B.55}$$

$$\hat{\mathbf{P}}_{T|T} = \hat{\mathbf{P}}_T \tag{B.56}$$

□