mathematics

MDPI

Article

Hierarchical Vector Mixtures for Electricity Day-Ahead Market Prices Scenario Generation

Carlo Mari [1,*,†] **and Carlo Lucheroni** [2,†]

[1] Department of Economics, Engineering, Society, Business Organizaton, University of Tuscia, 01100 Viterbo, Italy
[2] School of Sciences and Technology, University of Camerino, 62032 Camerino, Italy; carlo.lucheroni@unicam.it
* Correspondence: carlo.mari@unitus.it
† These authors contributed equally to this work.

Abstract

In this paper, a class of fully probabilistic time series models based on Gaussian Vector Mixtures (VMs), i.e., on linear combinations of multivariate Gaussian distributions, is proposed to model electricity Day Ahead Market (DAM) hourly prices and to generate consistent related DAM prices dynamic scenarios. These models, based on latent variables, intrinsically allow for organizing DAM data in hierarchically organized clusters, and for recreating the delicate balance of price spikes and baseline price dynamics present in the DAM data. The latent variables and the parameters of these models have a simple and clear interpretation in terms of market phenomenology, like market conditions, spikes and night/day seasonality. In the machine learning community, different to current deep learning models, VMs and the other members of the class discussed in the paper could be seen as just 'oldish' probabilistic models. In this paper it is shown, on the contrary, that they are still worthy models, excellent at extracting relevant features from data, and directly interpretable as a subset of the regime switching autoregressions still currently largely used in the econometric community. In addition, it is shown how they can include mixtures of mixtures, thus allowing for the unsupervised detection of hierarchical structures in the data. It is also pointed out that, as such, VMs cannot fully accommodate the autocorrelation information intrinsic to DAM data time series, hence extensions of VMs are needed. The paper is thus divided into two parts. In the first part, VMs are estimated and used to model daily vector sequences of 24 prices, thus assessing their scenario generation capability. In this part, it is shown that VMs can very well preserve and encode infra-day dynamic structure like autocorrelation up to 24 lags, but also that they cannot handle inter-day structure. In the second part, these mixtures are dynamically extended to incorporate dynamic features typical of hidden Markov models, thus becoming Vector Hidden Markov Mixtures (VHMMs) of Gaussian distributions, endowed with daily latent dynamics. VHMMs are thus shown to be very much able to model both infra-day and inter-day phenomenology, hence able to include autocorrelation beyond 24 lags. Building on the VM discussion on latent variables and mixtures of mixtures, these models are also shown to possess enough internal structure to exploit and carry forward hierarchical clustering also in their dynamics, their small number of parameters still preserving a simple and clear interpretation in terms of market phenomenology and in terms of standard econometrics. All these properties are thus also available to their regime switching counterparts from econometrics. In practice, these very simple models, bridging machine learning and econometrics, are able to learn latent price regimes from historical data in an unsupervised fashion, enabling the generation of realistic market scenarios while maintaining straightforward econometrics-like explainability.

Academic Editor: Pedro Carpena

Received: 12 July 2025
Revised: 22 August 2025
Accepted: 1 September 2025
Published: 4 September 2025

Citation: Mari, C.; Lucheroni, C. Hierarchical Vector Mixtures for Electricity Day-Ahead Market Prices Scenario Generation. *Mathematics* **2025**, *13*, 2852. https://doi.org/10.3390/math13172852

Keywords: machine learning; stochastic processes; nonlinear time series analysis; electricity markets

MSC: 37M10

1. Introduction

Day-Ahead Markets (DAM) of electricity are markets in which, on each day, prices for the 24 h of the next day form at once in an auction, usually held at midday. The data obtained from these markets are organized and presented in a discrete hourly time sequence, but actually the 24 prices of each day share the same information. Accurate encoding and synthetic generation of these time series is important not only as a response to a theoretical challenge, but also for the practical purposes of short term price risk management and derivative pricing. The most important features of the hourly price sequences are night/day seasonality, casual sudden upward spikes appearing only at daytime, downward spikes appearing at night time, spike clustering, and long memory with respect to previous days. From a modeling point of view, all of this is very hard to model accurately, and can mean nonlinearity and multi-scale analysis. Modeling is not forecasting, but since good quality forecasting is based on good modeling, developing good models can help quality forecasting, and this topic is worth research for this reason too.

Different research communities have developed different DAM price modeling methods, discussed and neatly classified some time ago in Ref. [1], especially dedicated to forecasting. Interestingly, in Ref. [1] it is also noted that a large part of papers and models in this research area can be mainly attributed to just two cultures, that of econometricians/statisticians and that of engineers/computational intelligence (CI) people. This bipartition seems to be still valid nowadays. DAM econometricians tend to use discrete time stochastic autoregressions (after due model identification) for point forecasting, and quantile regressions for probabilistic forecasting [2]. DAM engineering/CI people prefer to work with machine learning methods, in some cases of the probabilistic type [3,4].

Noticeably, writing models directly as distributions [5,6] and not as stochastic equations (like it is done in econometrics), allows for direct and stable sampling of scenario trajectories for a very large set of situations. Hence, probabilistic modeling as a background for probabilistic DAM forecasting has been appearing more and more frequently in the CI community in recent years, like for example the NBEATSx model [7,8] applied to DAM data, or the normalizing flows approach [9–11]. In addition, there exist machine learning probabilistic models that were never tested on DAM price forecasting. For example, the Temporal Fusion Transformer [12] can do probabilistic forecasting characterized by learning temporal relationships at different scales, emphasizing the importance of combining probabilistic accuracy with interpretability. DeepAR [13] is able to predict multivariate Gaussian distributions (in its DeepVAR form). Noticeably, these deep models often use the depth dimension for trying to capture multi-scale details (along the day time coordinate and along hours of the same day), at the expense of requiring a very large number of parameters. All these models allow for both forecasting and scenario generation. In any case, this 'two-cultures paradigm' finding, earlier discussed in Ref. [14] for a broader context, is stimulating in itself because it can orientate further research.

As said, the main advantage of modeling directly by probability distributions is the possibility of scenario generation. For this reason, at their inception, machine learning probabilistic models were called 'generative' (in contrast to the 'discriminative' standard econometrics models based on stochastic equations). Nowadays, the word generative has entered everyday use in relation to large language models (LLMs) and chatbots. That word

sense is the same as that used in this paper: LLMs do model corpora of documents in a probabilistic way, and, when they are triggered by a prompt, they 'emit' a stochastic sequence of words coherent with the properties of the modeled corpus. This scenario generation feature of LLMs is so important that it is changing our language. Thus, one can actually think of this linguistic property as a direct analog of DAM scenario generation. Indeed, among the first language models developed, there were the hidden Markov models (HMMs) [15], which will be discussed in the rest of the paper in relation to generation of coherent DAM price dynamics. Notice that in the case of language modeling, the relationship between modeling and forecasting becomes very subtle and would deserve an accurate discussion (think of the suggestive definition of LLMs as 'stochastic parrots' proposed in Ref. [16]). In analogy, in this paper, probabilistic forecasting will be discussed only in passing, just as a possible interesting task for the proposed models, but, for reasons of space, it will not be considered central to our discussion, which is mainly aimed at showing the DAM generation aspect of the proposed mathematical method, and at elucidating the double econometric/CI nature of the proposed models. The starting idea of this paper is that of picking up a time series that is very difficult to model (a DAM series), and showing that one can build up a satisfactory generative model, very specific to this series, but very effective to the task. One important feature of this kind of probabilistic modeling will be that it is based on latent variables, which can be directly interpreted as peeping into internal states of the electricity market, thus extracting very interesting hidden but useful information from it.

Building on these grounds, this paper, taking the stance of econometrics but still making use of 'old style', shallow probabilistic machine learning, will discuss the use of the Vector Mixture (VM) and Vector Hidden Markov Mixture (VHMM) family of models in the context of DAM prices modeling and scenario generation, with just some hints at how these models can be used for forecasting. It will study the behavior of these models on a freely downloadable specific DAM prices data set from the DAM of Alberta in Canada. All market data used in this paper are available from the AESO web site http://www.aeso.ca/ (accessed on 11 January 2024) via the link http://ets.aeso.ca/ (accessed on on 11 January 2024). In the basic Gaussian form in which it will be presented in this paper, the VM/VHMM approach works with stochastic variables with support on the full real line, so that this approach can be directly applied to series which allow for both negative and positive values. Yet, since in the paper the approach will be tested on one year of prices from the Alberta DAM ($24 \times 365 = 8760$ values), which are positive and capped, logprices will be used everywhere instead of prices. Estimation of VM/VHMM models relies on sophisticated and efficient algorithms like the Expectation–Maximization, Baum–Welch and Viterbi algorithms. Some software packages exist to facilitate estimation. The two very good open source free Matlab packages `BNT` and `pmtk3` were used for many of the computations of this paper. As an alternative, one can also use `pomegranate` [17], a package written in Python.

The presented VM/VHMM approach will be discussed from four perspectives. First, from the two-culture paradigm. Second, from the generative vs. discriminative view on models (which implies an extensive discussion of generative/probabilistic modeling). Third, from the practical ability of these models of extracting relevant features from the DAM specific data, and of generate synthetic time series very similar to the original data. Fourth, from the impact of time on generative modeling. In brief, these very simple models will be shown to be able to learn latent price regimes from historical data in a highly structured and unsupervised fashion, enabling the generation of realistic market scenarios, while maintaining straightforward econometrics-like explainability.

The idea of using HMM on electricity prices data is not new in itself. This approach is, however, usually aimed at price forecasting (which we do not do; we simply propose scenario generation, and in the text we will explain why we do not think that forecasting DAM prices with HMMs is a good idea). Available papers from the literature that apply HMMs to DAM electricity prices will be listed below. In addition, once this approach is selected, attaching different distributions with a continuous support to different states is natural. This can also lead to the idea of directly attaching mixtures to the states, like we do, although we propose an advanced type of mixture, hierarchical mixtures, never used before on DAM data. One can go even beyond that, and attach vector mixtures to the states. Even the dataset we chose has been already explored with HMM-based models. However, all HMM papers we know miss one main point in using DAM data, from Alberta or from other DAMs. As said, the 24 DAM prices are set daily, with all 24 at once, hence same-day prices share the same information. Trying to apply an HMM model to the scalar hourly sequence of prices staggered in a row of hourly values introduces spurious causality, breaks down infra-day coherency, and cannot be used in a real trading situation, since for forecasting tomorrow's hour 6, you will need the price of tomorrow's hour 5, which you do not have today. This leads us to propose a vector hierarchical mixtures HMM approach to model in a probabilistic way DAM data. This approach should be able to 'speak DAM language', that is, doing scenario generation. We also noticed that the papers we collected don't take into account the questions about interesting mathematical problems, which the use of this technique poses. We thus tried to formalize these questions in terms of dynamical modeling mathematics in relation to current mathematical research. We also found it interesting (but also maybe obvious) to notice that interest in HMM-based models applied to DAM data began at the beginning of the year 2000 with the advent of DAMs, and faded in just a few years. Actually, the interest in DAM prices grew very strongly, but newer machine learning techniques, especially deep learning, took over in the meantime, making HMMs sort of obsolete. However, considering HMM-related techniques too old or obsolete is, in our opinion, wrong because this approach still has many ideas to suggest even deep learning methodologies. The reason for that will be part of the discussions of this paper. Finally, why in the end is it useful to 'learn speaking DAM'? Probabilistic modeling means learning the structure of uncertainty present in the data. Energy Finance is very much about uncertainty management. This is made by means of scenario analysis, derivative pricing, correlation quantification, anticipative control for optimal portfolio composition, and probabilistic forecasting, which are all techniques that require joint probabilistic information about data. The more accurate is the representation of uncertainty, the better comes out energy risk management performance. If this representation includes also some analytic aspects, that is even better.

In the literature, the related oldest paper we found is an IEEE proceedings by Zhang et al. from 2010 [18]. A 4-states HMM is used to forecast hourly prices by the use of four distributions, none of them a mixture. No mathematical discussion is carried out. It is not clear which dataset was used. A preprint from 2015 by Wu et al. [19] works on forecasting prices of the Alberta DAM. It is interesting to notice that the paper opens with a short discussion about previous experiences of other groups on switching Markov models and clustering on European prices. In this HMM paper they work with hand-preprocessed discrete distributions, hence with symbolic HMMs. There is more attention to the mathematics of the problem, but no advancement of the techniques. We found a proceedings from 2016 by Dimoulkas et al. [20], working with HMM and Nordpool balancing market data on forecasting. Noticeably, the balancing market has statistical features very different from the wild price dynamics of the Alberta DAM. Interestingly for us, they work with a mixture of Gaussians attached to each state, like those we start with. It is an application

paper, without theory. We found a full paper by Apergis et al. from 2019 in the Energy Economics journal [21], working on a parent research line, that of semi-Markov models, applied to Australian electricity prices. They work with scalar sequences of weekly volume weighted average spot prices, which is a choice more sensible than working on hourly prices. Their aim, as much as is sensible and similar to one of our aims, is that of studying the inner working of the market by means of the dynamics of the hidden coordinates. We also found a very recent conference paper by Duttilo et al. from 2025 [22] applied to forecasting Italian DAM prices, based on a standard HMM, but using non-standard generalized normal distribution components (but not mixtures), on daily (not hourly) returns, a quite sensible target. Based on that, we think that we can say that our proposal adds some new possibly interesting issues to the study of DAM prices with HMM-based models, to the specific literature on Alberta DAM prices dynamics, and to the literature on HMM-based models itself.

The plan of the paper is as follows:

After this Introduction, Section 2, will define suitable notation and review usual vector autoregression modeling to prepare the discussion on VMs and VHMMs for DAM prices. The difference between the discriminative approach (stochastic equations) and the purely probabilistic generative approach to time series modeling will be highlighted in this discussion. Section 3, starting from a simplified time structure for the models, will introduce VMs as machine learning dynamical mixture systems uncorrelated in inter-day dynamics but correlated in infra-day dynamics, and will discuss their inherent capacity of doing clustering in a hierarchical way. In passing, this section will discuss how their structure can be thought as similar to that of deep learning. Stochastic equations formalism will be contrasted with purely probabilistic formalism. This is a core theoretical section of the paper. Section 4, after discussing the very complex features of the chosen time series, will show how the proposed VM models perform on DAM data, especially in terms of the features that they can automatically extract, and in terms of the synthetic series that they can generate. Section 5, based on the uncorrelated structures illustrated in Section 3, will discuss VHMMs as models fully correlated in both infra-day and inter-day dynamics, and their relationship to their HMM backbone. Section 6 will show the behavior of VHMMs when applied to Alberta DAM data, their remarkable scenario generation ability and their special ability of accurately modeling and mimicking price spike clustering, due to time inclusion in their structure. This is a core practical section of the paper. Section 7 will conclude.

2. Notation and Discriminative vs. Generative Autoregression Modeling

This section will define suitable notation and review usual vector modeling. This will help put into context the related literature and to prepare the discussion on vector mixtures and Markov models proposed in the following sections.

Consider the DAM equispaced hourly sample time series $\{\hat{x}_t\}$ of values $\hat{x}_t \in [-\infty, \infty]$, where t indicates a given hour starting from the beginning of the sample.

2.1. Why Vector Modeling for DAM Data

In the specific DAM literature, it is univariate (i.e., scalar) autoregressions directly on the hourly series $\{\hat{x}_t\}$ [23,24], more or less nonlinear and complicated [25,26], which are applied most often. Noticeably, the information structure implicitly assumed by univariate models is the typical causal chain in which prices at hour h depend only on prices of previous hours. On one hand researchers are aware that this structure does not correspond to typical DAM prices information structure [27], but on the other hand vector models are often considered impractical and heavier to estimate and discuss, especially before

the arrival of large neural network models. This issue led first to estimate the data as 24 concurrent but independent series like in Equation (2), then to the introduction of vector linear autoregressions [28], maybe using non-Gaussian disturbances distributions like in [29] (in this case multivariate Student's distributions). In the case of models like the VAR(1), for each of the K_D lags a matrix of 24×24 parameters (besides vector means and matrix covariances of the innovations) has to be estimated and interpreted. Adding lags is thus computationally expensive and makes the interpretation of the model more complicated, and weakens stability, but in contrast few lags imply short term memory in terms of days, i.e., quickly decaying autocorrelation. Seasonality and nonlinearity (like fractional integration) can be further added to these vector regressions [30], making them able to sustain a longer term memory. A parallel line of research that uses a regime switching type of nonlinearity in the scalar autoregression setting (discussed at length in [31]) does not concentrate on long term memory but can result in an even more satisfactorily modeling of many DAMs' key phenomena like concurrent day/night seasonality and price spiking.

2.2. Discriminative Modeling

In order to link the following discussion with the Alberta data, in what follows $\hat{x}_t$ will be assumed to be logprices. In DAMs, the next day 24 logprices form at once on each day and share the same information, so that it makes microeconomic sense to regroup the hourly scalar series into a daily vector sample series $\{\hat{\boldsymbol{\xi}}_{\boldsymbol{d}}\}_N$ of N days, where the vectors $\hat{\boldsymbol{\xi}}_{\boldsymbol{d}}$ have 24 coordinates $\hat{\xi}_{d,h}$ each obtained from $\{\hat{x}_t\}$ by the mapping $t \to \{h, d\}$, and are labeled by the day index d and the day's hour h. The series $\{\hat{\boldsymbol{\xi}}_{\boldsymbol{d}}\}_N$ can be thought as having been sampled from a stochastic chain $\{\boldsymbol{\xi}_d\}$ of vector stochastic variables $\boldsymbol{\xi}_d$ modeled as the vector autoregression

$$\boldsymbol{\xi}_{d+1} = \boldsymbol{D}(\{\boldsymbol{\xi}\}_D|\{\theta_D\}) + \boldsymbol{\eta}_{d+1}. \tag{1}$$

In Equation (1) d labels the current day (at which the future at day $d+1$ is uncertain), $\boldsymbol{D}$ is a day-independent vector function with hourly components D_h that characterizes the structure of the autoregression model, $\{\boldsymbol{\xi}\}_D = \{\boldsymbol{\xi}_d, \boldsymbol{\xi}_{d-1}, \ldots, \boldsymbol{\xi}_{d-K_D+1}\}$ labels a set of lagged regressors of $\boldsymbol{\xi}_{d+1}$ where

$$K_D \geq 0$$

is the number of present plus past days on which day $d+1$ is regressed, with the convention that for

$$K_D = 0$$

no $\boldsymbol{\xi}_d$ variables appear in $\boldsymbol{D}$ (see Figure 1).

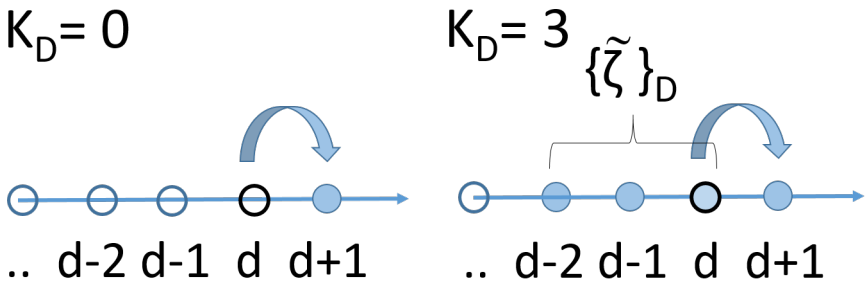

Figure 1. Sets of regressor variables: d is the current day. Empty circles represent lack of variable, the curved arrow represents the dynamic transition. Left: $K_D = 0$ (empty set), representing the 0 - lag autoregression $\boldsymbol{\xi}_{d+1} = \boldsymbol{\eta}_{d+1}$ where the model sequence consists of independent Gaussians, case of VMs. Right: $K_D = 3$, case which includes VHMMs.

The earliest day in $\{\boldsymbol{\xi}\}_D$ is hence $d - K_D + 1$, so that for example for $K_D = 1$ only $\boldsymbol{\xi}_d$ is used as a regressor (i.e., 24 h) and, for $K_D = 2$, both $\boldsymbol{\xi}_d$ and $\boldsymbol{\xi}_{d-1}$ (i.e., 48 h) are used. Hence the autoregression in Equation (1) uses

$$N_D = K_D + 1$$

vector stochastic variables in all, which include the $d + 1$ variable. For $K_D = 0$, i.e., $N_D = 1$, Equation (1) thus represents a 0-lag autoregression. In Equation (1) $\{\theta_D\}$ represents the set of model parameters. Moreover, in Equation (1) $\boldsymbol{\eta}_{d+1}$ represents the member at time $d + 1$ of a daily vector stochastic chain $\{\boldsymbol{\eta}_d\}$ of i.i.d. vector innovations $\boldsymbol{\eta}_d = (\eta_{1,d}, \eta_{2,d}, \ldots, \eta_{24,d})$ with coordinates $\eta_{h,d}$ related to days and hours. By definition, the $\boldsymbol{\eta}_d$ vector stochastic variables have their marginal (w.r.t. daily coordinates d) joint (w.r.t. intraday coordinates h) density distributions $p(\boldsymbol{\eta}_d)$ all equal to a fixed $p(\boldsymbol{\eta})$. This $p(\boldsymbol{\eta})$ will be chosen as a 24-variate Gaussian distribution $\mathcal{N}(\boldsymbol{\eta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are (vector) mean and (matrix) covariance of the distribution. Thus in this vector notation, $\{\hat{\boldsymbol{\xi}}_{\boldsymbol{d}}\}_N$ can represent also one N-variate (in the daily sense) draw from the chain $\{\boldsymbol{\xi}_{\boldsymbol{d}}\}$, driven by a i.i.d. Gaussian noise $\{\boldsymbol{\eta}_{\boldsymbol{d}}\}$, which yet includes intraday structure.

Incidentally, it can be noticed that the model in Equation (1) includes for a suitable form of $\boldsymbol{D}$ the possibility of regressing one day's hour λ on the same hour λ (only) of K_D past days, i.e., independently from other hours as

$$\xi_{\lambda,d+1} = D_\lambda(\{\xi\}_{\lambda,D}|\{\theta_D\}) + \eta_{\lambda,d+1},\ \lambda = 1, \ldots, 24, \tag{2}$$

where $\xi_{\lambda,d}$ are now scalar stochastic variables and $\{\xi\}_{\lambda,D} = \{\xi_{\lambda,d}, \xi_{\lambda,d-1}, \ldots, \xi_{\lambda,d-K_D+1}\}$, an approach occasionally used in the literature. Seen as a restriction of Equation (1), this 'independent hours' scalar model is easier to handle than the full vector model of Equation (1), but it is of course less accurate.

In contrast, the full vector model can take into account both intraday and interday dependencies, and allows for a complete panel analysis of data, cross-sectional and longitudinal, coherent with the microeconomic mechanism that generates the data. Commonly used forms of $\boldsymbol{D}$ are those linear in most of the parameters, like $\boldsymbol{D} = \boldsymbol{C_0} + \sum_{d'=d-K_D+1}^{d'} \boldsymbol{C_{d'}} \boldsymbol{h}(\boldsymbol{\xi}_{\boldsymbol{d'}})$ where $\boldsymbol{C_{d'}}$ are the matrices of coefficients and $\boldsymbol{h}$ is a vector nonlinear function. If $\boldsymbol{D}$ is linear in the $\boldsymbol{\xi}_{\boldsymbol{d'}}$ as well, a vector AR(K_D) model (VAR(K_D)) is obtained, like for example the $K_D = 1$ Gaussian VAR(1) model

$$\boldsymbol{\xi}_{d+1} = \mathbf{C_0} + \mathbf{C}\,\boldsymbol{\xi}_d + \boldsymbol{\eta}_{d+1}. \tag{3}$$

For VAR(K_D) models the Box–Jenkins model identification (i.e., selection) procedure can be used to select optimal K_D and to estimate the coefficients. When $E[\boldsymbol{\eta}_{d+1}] = 0$, taking the expectation of both sides of equations like Equation (3) is the usual way of transforming an identified linear autoregression model into a 'point forecaster', to give

$$\boldsymbol{\xi}^f = \mathbf{C_0} + \mathbf{C}\,\boldsymbol{\xi}_d. \tag{4}$$

a VAR(1) forecasting model in this case. For this reason, in practice, when using linear stochastic equations, modeling and forecasting can be thought as the same thing. More in general, point forecasting for Equation (1) can be indicated as

$$\boldsymbol{\xi}^f(\{\hat{\boldsymbol{\xi}}\}_D) = \boldsymbol{D}(\{\hat{\boldsymbol{\xi}}\}_D|\{\theta_D\}) \tag{5}$$

where θ_D represents the parameter set. In turn, Equation (5) allows for defining (conditional) forecast vector errors as

$$\boldsymbol{e}^{\boldsymbol{f}}_{\boldsymbol{d+1}}(\{\hat{\boldsymbol{\xi}}\}_D) = \hat{\boldsymbol{\xi}}_{\boldsymbol{d+1}} - \boldsymbol{\xi}^{\boldsymbol{f}}(\{\hat{\boldsymbol{\xi}}\}_D), \tag{6}$$

on which scalar error functions $E_{\boldsymbol{D}}(\boldsymbol{e}^{\boldsymbol{f}}_{\boldsymbol{d+1}})$ can be defined. It is by means of these modeling errors that the model is identified in the Box–Jenkins frame.

One can finally notice that at day d the only stochastic variable in the right hand side of Equation (1) is $\boldsymbol{\eta}_{d+1}$, endowed with a chosen distribution (usually a Gaussian). This allows one to look at Equation (1) from a probabilistic modeling point of view. Actually, the function $\boldsymbol{D}$ in Equation (1) induces a relationship between $\boldsymbol{\xi}_{d+1}$ and $\{\boldsymbol{\xi}\}_D$ that can be written as a conditional distribution

$$p(\boldsymbol{\xi}_{d+1}|\{\boldsymbol{\xi}\}_D; \{\theta_D\}) \tag{7}$$

In addition to using stochastic equations like Equation (1), vector autoregression models can also be directly defined by assuming a basic conditional probability distribution like that in Equation (7), for example a Gaussian like in the standard VAR model. When based on a distribution conditional on some of their variables, probabilistic models are called *discriminative*. Hence, even linear autoregressive models like AR and VAR can be considered probabilistic models, although defined by stochastic equations, but of discriminative type. The relationship in Equation (7), often obtained only numerically, can be also used for probabilistic forecasting.

2.3. Generative Modeling

In contrast, when directly defined by means of a full joint distribution like

$$p(\boldsymbol{\xi}_{d+1}, \{\boldsymbol{\xi}\}_D|\{\theta_D\}). \tag{8}$$

data models are called probabilistic *generative* models [32,33]. Dynamic probabilistic generative modeling consists in practice in obtaining models which, once put on data, adapt to these data and become a distribution with the same statistical characteristics of the sequences (at least those long up to N_D) present in the training set. It is thus important to notice the difference in conditioning between Equations (7) and (8).

There is actually a direct bridge between probabilistic generative modeling and stochastic autoregression modeling (which in turn is probabilistic discriminative modeling). Notice that in the 0-lag case, i.e., for $K_D = 0$, the set $\{\boldsymbol{\xi}\}_D$ is empty, and Equation (1) becomes the Gaussian VAR(0) model

$$\boldsymbol{\xi}_{d+1} = \boldsymbol{\eta}_{d+1} \tag{9}$$

(where a possible constant is omitted). The corresponding probabilistic version of this model, i.e., Equation (7) specialized according to Equation (9), becomes the product of factors

$$p(\boldsymbol{\xi}_{d+1}|\{\boldsymbol{\xi}\}_D; \{\theta_D\}) = \mathcal{N}(\boldsymbol{\xi}_{\boldsymbol{d+1}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{10}$$

In this case the defining distribution is unconditional, and the probabilistic model is both of discriminative and generative type. Estimation becomes probability density estimation. The point forecast becomes equal to $\boldsymbol{\mu}$ at all days. The probabilistic forecast, which forecasts a distribution, is the distribution itself.

2.4. Generative Dynamics

Where are the dynamics? Notice that a model is defined for all available days d. This means that the full dynamic model is a chain of independent distributions. It also means that equations like Equation (8) have to be interpreted as distribution products like

$$\cdots p(\boldsymbol{\xi}_{d+2}, \{\boldsymbol{\xi}\}_D | \{\theta_D\}) \cdot p(\boldsymbol{\xi}_{d+1}, \{\boldsymbol{\xi}\}_D | \{\theta_D\}) \tag{11}$$

where all factors are the same. Generation means sampling at once from this product distribution, starting from day d.

2.5. How to Estimate a Density Distribution

Nomenclature in generative modeling often follows conventions used in Bayesian analysis and in the literature about the Bayes Theorem. As a quick recall, if A and B are two stochastic variables for which distributions $p(A)$ and $p(B)$ are known, and the conditional distribution $p(A|B)$ is available, the Bayes theorem (actually a mere identity) is written as

$$p(B|A) = p(A|B)\ p(B)\ \frac{1}{p(A)}. \tag{12}$$

The distribution on the l.h.s. is called posterior, the first factor in the r.h.s. is called likelihood, the second is called bias, and the third is called normalizing factor. In the Bayesian frame parameters can be considered stochastic variables. Consider a stochastic variable $B = \xi$ for which the distribution $p(\xi|\theta)$ depends on a parameter $A = \theta$ (think of Equation (8)), and some samples $\hat{\xi}$ are given. Choose the bias $p(\theta) = 1$, which tells nothing special about θ. The likelihood of seeing $\hat{\xi}$, as it happened, is maximum when Equation (12), or its logarithm, called loglikelihood (LL), is maximized with respect to θ, that is when

$$\ln p(\xi|\theta) = \ln p(\theta|\xi)\ + \ln \frac{1}{p(\hat{\xi})} \tag{13}$$

is maximum for an optimum θ^*, where the last term on the r.h.s. is just a constant when optimizing w.r.t. θ, not entering the calculation. This happens when the likelihood, a probability of the parameter θ, is maximum. For this reason often the posterior $p(\xi|\theta)$ is called likelihood too. For example, if the generative model studied is a univariate Gaussian, and some samples are given, in order to identify the model one can find mean and variance that maximize the posterior of the chosen distribution. This is also expressed by saying that a maximum likelihood estimate is carried out. Actually, a more complicated but more flexible version of this procedure can be given when $p(\theta) \neq 1$, that is, with more sophisticated biases which can represent some previously gathered information about θ. The term bias is indeed very often used in this sense in probabilistic machine learning.

This short discussion can highlight some important differences between generative and discriminative modeling. Whereas in generative modeling estimation errors are usually not even defined, in discriminative modeling they are central to any quality analysis (as in any Box–Jenkins model selection procedure). In the generative case a generated sample from the modeled chain includes $N_D = K_D + 1$ variables, neither more or less, differently from the discriminative case, which usually includes only one, the forecast. Indeed, generative models are clearly different from discriminative models, and this is probably why their are seldom considered by econometricians.

Finally, it is interesting to compare the generated dynamics for N days implied by probabilistic discriminative models like that in Equation (7) with the dynamics implied by the VAR(0) model in Equation (10). Whereas in the former, at each given day d, the next day $d + 1$ vector is obtained by sampling from a marginal distribution that changes each day, obtained by feeding the last sampled value back to the distribution, in the latter the

sampled distribution never changes. The VAR(0) model is uncorrelated from the point of view of inter-day dynamics, whereas it remains correlated in infra-day dynamics. In contrast, in the case of the generic generative model of Equation (8), both intraday structure and interday dynamics can be interesting. Based on this point of view, generative models that originate factorized dynamics (like the VAR(0) model and the mixtures which will be introduced next) will be henceforth called uncorrelated, whereas generative models for which inter-day dynamics is more interesting (like the VHMMs) will be called correlated. It will be shown that both types, uncorrelated and correlated, can be in any case very useful for DAM market generative modeling.

3. Uncorrelated Generative Vector Models: Mixtures, Clustering and Complex Patterns

In this section the core block of the proposed generative modeling approach for DAM data is discussed. Vector Gaussian mixtures are introduced and related to their ability of automatically clustering and self-organizing data. Then, these mixtures are themselves used to build mixtures of mixtures, showing how hierarchical structure emerges from this construction. This will prepare the discussion of the next section, where this setup will be used to explore DAM data, and generate complex price sequences that look very much similar to original data.

3.1. Mixtures and Clustering

By Equation (10), the generative/discriminative VAR(0) model is defined in distribution as $\mathcal{N}(\boldsymbol{\xi_{d+1}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Estimation of this model on the data set $\{\hat{\boldsymbol{\xi}}_{\boldsymbol{d}}\}_N$ means estimating a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ of coordinates μ_h and symmetric covariance matrix $\boldsymbol{\Sigma}$ of coordinates $\Sigma_{h,h'}$. Parameter estimation is made by maximizing the likelihood L obtained multiplying Equation (10) N times, or maximizing the associated loglikelihood LL (which thus becomes a sum). The parameters can be thus be obtained analytically from the data, and since the problem is convex this solution is unique. Once $\boldsymbol{\Sigma}$ has been estimated, estimated 'hourly' univariate marginals $p(\xi_{h,d})$ and 'bi-hourly' bivariate marginals $p(\xi_{h,d}, \xi_{h',d})$ can be analytically obtained by partial integration of the distribution. In addition, for a generic market day, i.e., a vector $\hat{\boldsymbol{\xi}}_{\boldsymbol{n}} \in \{\hat{\boldsymbol{\xi}}_{\boldsymbol{d}}\}_N$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can define a distance measure l^2 in a 24 dimensions space from $\hat{\boldsymbol{\xi}}_{\boldsymbol{n}}$ to the center $\boldsymbol{\mu}$ of the Gaussian of Equation (10) as

$$l^2(\hat{\boldsymbol{\xi}}_{\boldsymbol{n}}) = (\hat{\boldsymbol{\xi}}_{\boldsymbol{n}} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\hat{\boldsymbol{\xi}}_{\boldsymbol{n}} - \boldsymbol{\mu})', \tag{14}$$

where the superscript -1 indicates the inverse and the superscript $'$ indicates the transpose. l^2 is called Mahalanobis distance [34]. This measure will be used later.

The VAR(0) model can become a new and more complex stochastic chain after replacing its driving Gaussian distribution with a driving 'Gaussian mixture' of a given number of 'components', making it a Vector Mixture VM.

An S-component VM is thus a generative dynamic model defined in distribution as a linear combination of Gaussians,

$$p(\boldsymbol{\xi}_{d+1}, \{\boldsymbol{\xi}\}_D | \{\theta_D\}) = \sum_{s=1}^{S} \pi^s \mathcal{N}(\boldsymbol{\xi_{d+1}}; s), \tag{15}$$

where $\mathcal{N}(\boldsymbol{\xi_{d+1}}; s)$ is an abbreviated notation for $\mathcal{N}(\boldsymbol{\xi_{d+1}}; \boldsymbol{\mu^s}, \boldsymbol{\Sigma^s})$ [35]. The S extra parameters $\pi^s \geq 0$ subject to $\sum_s \pi^s = 1$ are called mixing parameters or weights. Continuous distributions must integrate to 1, and discrete distributions must sum to 1. Integrating Equation (15) along ξ gives $\sum_{s=1}^{S} \pi^s$, which by definition sum to 1. Hence the r.h.s. of

Equation (15) is a sound distribution. Figure 2 shows an example with $S = 2$, $\pi^1 = 0.3$ and $\pi^2 = 0.7$, a model of type

$$\pi^1 \mathcal{N}(\boldsymbol{\xi}_{d+1}; 1) + \pi^2 \mathcal{N}(\boldsymbol{\xi}_{d+1}; 2).$$

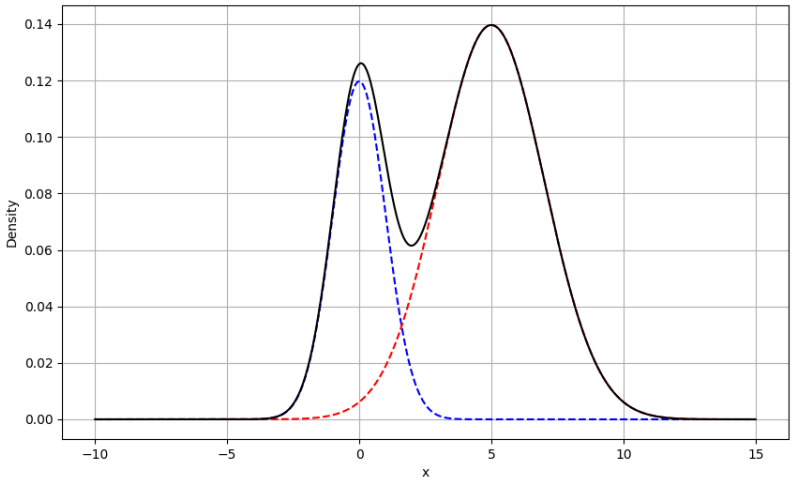

Figure 2. The distribution of an univariate mixture of two Gaussians, with $\mu_1 = 0, \sigma_1 = 1, \mu_2 = 5$, $\sigma_2 = 2$, and weights $\pi^1 = 0.3, \pi^2 = 0.7$.

Like the VAR(0), the VM is a probabilistic generative model, of 0-lag type, uncorrelated in the sense that it generates an interday independent dynamics, unlimited with respect to maximum attainable sequence length. In Equation (15) the $d + 1$ subscript is there to remark its dynamic nature, since mixture models are not commonly used for stochastic chain modeling, except possibly in GARCH modeling (see for example [36]). A VAR(0) model can thus be seen as a one-component $S = 1$ VM. For $S > 1$ the distribution in Equation (15) is multi-modal (in Figure 2 there are two distinct peaks, which are the mode of the Gaussians), and maximization of the LL can be made in a numeric way only, either by Montecarlo sampling or by the Expectation–Maximization (EM) algorithm [6,37]. When estimating a model, since EM can get stuck into local minima, multiple EM runs have to be run from varied initial conditions in order to be sure to have reached the global minimum.

Means and covariances of the S-component VM can be computed in an analytic way from means and covariances of the components. For example, in the $S = 2$ case in the vector notation the mean is

$$\boldsymbol{\mu} = \pi^1 \boldsymbol{\mu^1} + \pi^2 \boldsymbol{\mu^2} \tag{16}$$

and in coordinates the covariance is

$$\begin{aligned} \boldsymbol{\Sigma}_{h,h'} = &\left(\textstyle\sum_{i=1,2} \pi^i (\boldsymbol{\Sigma}^i_{h,h'} + \mu^i_h \, \mu^i_{h'}) \right) - \\ &\left(\textstyle\sum_{i=1,2} \pi^i \mu^i_h \right) \left(\textstyle\sum_{i=1,2} \pi^i \mu^i_{h'} \right). \end{aligned} \tag{17}$$

Hourly variance $(\sigma_h)^2$ is obtained for $h' = h$ in terms of component hourly variances $(\sigma^1_h)^2$ and $(\sigma^2_h)^2$ as

$$\begin{aligned} (\sigma_h)^2 = \pi^1 (\sigma^1_h)^2 + \pi^2 (\sigma^2_h)^2 + \\ (\pi^1 (\mu^1_h)^2 + \pi^2 (\mu^2_h)^2) - (\pi^1 \mu^1_h + \pi^2 \mu^2_h)^2. \end{aligned} \tag{18}$$

The mixture hourly variance is thus the weighted sum of component variances plus a correction term which is similar to covariance in the case of a weighted sum of two gaussian variables. Based on the component variances, in analogy with the distance defined in

Equation (14) a number S of Mahalanobis distances l_s^2 can be now associated to each data vector $\hat{\boldsymbol{\xi}}_d$. In Equation (15) the weights can be interpreted as probabilities of a discrete hidden variable s, i.e., as a distribution $p(s) = \pi^s$ over s.

All of this is important because an estimated VM implicitly clusters data, in as many Gaussians as appear in the model, in a probabilistic way.

In machine learning, clustering, i.e., unsupervised classification of vector data points called evidences into a predefined number K of classes, can be made in either a deterministic or a probabilistic way. Deterministic clustering is obtained using algorithms like K-means [38], based on the notion of distance between two vectors. K-means generates K optimal vectors called centroids and partitions the evidences into K groups such that each evidence is classified as belonging only to the group (cluster) corresponding to the nearest centroid in a yes/not way. Here 'near' is used in the sense of a chosen distance measure, often called linkage. In this paper, for the K-means analysis used in the Figures the Euclidean distance linkage is used.

Probabilistic clustering into $K = S$ clusters can in turn be obtained using S-component mixtures in the following way, which is called the EM algorithm. In a first step the estimation of a S-component mixture on all evidences $\hat{\boldsymbol{\xi}}_n$ finds the best placement of means and covariances of S component distributions, in addition to S weights π^s, seen as unconditional probabilities $p(s)$. Here best is intended in terms of maximum likelihood. The component means are vectors that can be interpreted as centroids, the covariances can be interpreted as parts of distance measures from the centroids, in the Mahalanobis linkage sense of Equation (14). In a second step, centroids are kept fixed. Conditional probabilities

$$\pi_n^s = p(s|\hat{\boldsymbol{\xi}}_n)$$

of individual data points (our daily vectors), technically called 'responsibilities', are then computed by means of a Bayesian-like inference approach which uses a single-point likelihood. In this way, to each evidence a probability π_n^s of belonging to one of the clusters is associated, a member relationship which is softer than the yes/not membership given by K-means. For this reason, K-means clustering is called hard clustering, and probabilistic clustering is called soft clustering.

It is very important to notice that s, which started as a component index, is now interpreted as a latent or hidden state, on which the model can find itself, which a probability that can be computed once the model is fitted on data.

Being a product of mixtures, an uncorrelated VM model can do soft clustering in a intrinsic way, partitioning the data without supervision and automatically incorporating this feature in the generative vector model. In addition, as time goes on (d grows), the system finds itself exploring the hidden state set available to the mixture. For example, when $S = 2$, a generated trajectory could correspond to the state sequence $1, 2, 2, \ldots$. Notice that these outcomes can never become explicit data, they cannot be observed, they can only be inferred. This is what makes of a VM a latent variable model.

3.2. Mixtures of Mixtures, and Hierarchical Regime Switching Models

Hence, in a dynamical setting, independent copies s_d of the mixture variable s can be associated with the days, thus forming a (scalar) stochastic chain $\{s_d\}$ ancillary to $\{\boldsymbol{\xi}_d\}$. In the example above, a realization of this chain could be $s_1 = 1, s_2 = 2, s_3 = 2, \ldots$.

This suggests a further and deeper interpretation of a VM, which will lead to a hierarchical extension of VMs themselves, with an attached econometric interpretation, and which will make them deep.

Consider as an example the $S = 2$ model. Each draw at day d from the two-component VM can be seen as a hierarchical two-step (i.e., doubly stochastic) process, which consists

of the following sequence. At first, flip a Bernoulli coin represented by the scalar stochastic variable ζ with support $(\zeta^1 = 1, \zeta^2 = 2)$ and probabilities $\pi = (\pi^1, \pi^2)$. Then, draw from (only) one of the two stochastic variables $\boldsymbol{\eta}^i$, $i = 1, 2$, specifically from the one which i corresponds to the outcome of the Bernoulli flip. The variables $\boldsymbol{\eta}^i$ are chosen independent from each other and distributed as $\mathcal{N}(\boldsymbol{\eta}; i)$. If $(s^1 = 1, s^2 = 2)$ is the support of s_{d+1}, the VM can thus be seen as the stochastic nonlinear time series

$$\begin{aligned} s_{d+1} &= \zeta \\ \boldsymbol{\xi_{d+1}} &= \begin{cases} \boldsymbol{\eta}^1 \text{ if } s_{d+1} = 1 \\ \boldsymbol{\eta}^2 \text{ if } s_{d+1} = 2, \end{cases} \end{aligned} \tag{19}$$

a vector Gaussian regime switching model in the time coordinate d, where the regimes 1 and 2 are formally autoregressions without lags, i.e., of VAR(0) form. A path of length N generated by this model consists of a sequence of N hierarchical sampling acts from Equation (19), which is called ancestor sampling. In Equation (19) the r.h.s. variables ζ, $\boldsymbol{\eta}^1$, $\boldsymbol{\eta}^2$ are daily innovations (i.e., noise generators) and the l.h.s. variables s_{d+1}, $\boldsymbol{\xi}_{d+1}$ are system dynamic variables. Even though the value of s_{d+1} is hidden and unobserved, it does have an effect on the outcome of observed $\boldsymbol{\xi}_{d+1}$ because of Equation (19). Notice that if the regimes had $K_D > 0$, for example $K_D = 1$ like the VAR(1) model of Equation (3), the model would have been discriminative and not generative. It would have had a probabilistic structure based on the conditional distribution $p(s_{d+1}, \boldsymbol{\xi_{d+1}} | s_d, \boldsymbol{\xi_d}; \{\theta_D\})$, like most regime switching models used in the literature [24]. Yet, it would not be capable of performing soft clustering.

Generative modeling requires at first to choose the joint distribution. The hierarchical structure interpretation of the basic VM (a simple mixture), Equation (19), implies that in this case the generative modeling joint distribution is chosen as the product of factors of the type

$$p(s_{d+1}, \boldsymbol{\xi_{d+1}}) = p(\boldsymbol{\xi_{d+1}} | s_{d+1})\, p(s_{d+1}), \tag{20}$$

From Equation (20) the marginal distribution of the observable variables $\boldsymbol{\xi_{d+1}}$ can be obtained by summation

$$p(\boldsymbol{\xi_{d+1}}) = \sum_{s_{d+1}} p(\boldsymbol{\xi_{d+1}} | s_{d+1})\, p(s_{d+1}). \tag{21}$$

A graphic representation of this interpretation of our VM model is shown in Figure 3, where model probabilistic dependency structure is shown in such a way that the dynamic stochastic variables, like those appearing on the l.h.s. of Equation (19), are encircled in shaded (observable type) and unshaded (unobservable type) circles, and linked by an arrow that goes from the conditioning to the conditioned variable. This structure is typical of Bayes Networks [6]. Specifically, in panel (a) of Figure 3 the model of Equation (19) is shown.

It will be now shown that models like that in Figure 3a can be made deeper, like those Figure 3b,c, and that this form is also necessary for a parsimonious form of modeling in VHMMs to be discussed in Section 5. Define the depth of a dynamic VM as the number J of hidden variable layers plus one, counting in the $\boldsymbol{\xi}_d$ layer as the first of the hierarchy. A deep VM can be defined as a VM with $J > 2$, i.e., with more than two layers in all, in contrast with the shallow (using machine learning terminology) VM of Equation (19). A deep VM is a hierarchical VM, that can be also used to detect cluster-within-clusters structures.

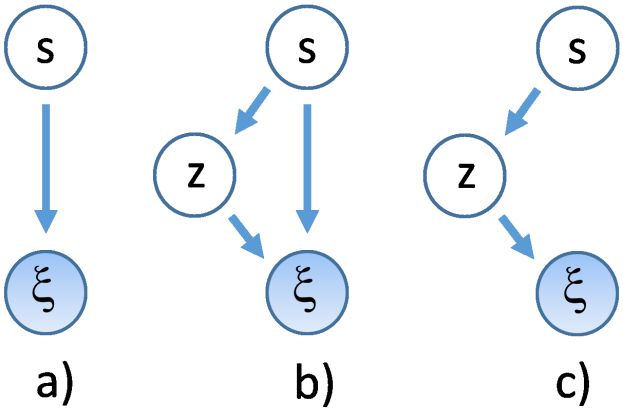

Figure 3. Graphic representation of the dynamic VM/regime switching models discussed in the text, with their probabilistic dependency structure. Dynamic stochastic variables are shown inside circles, shaded when variables are observable and unshaded when they are not observable. Arrows show the direction of dependencies among variables. (**a**) Model of Equation (19); (**b**) Model of Equation (22); (**c**) Model of Equation (27).

As an example, a $J = 3$ VM can be defined in the following way. For a given day d, let ζ_d and s_d to be again two scalar stochastic variables with S-valued integer support $s_d^i = i$, $i = 1, \ldots, S$ and same distribution $\pi^1, \ldots, \pi^S$. Let α^i to be S new scalar stochastic variables, all with same M-valued integer support $\alpha^{i,j} = j$, $j = 1, \ldots, M$, and with time-independent distributions $p(\alpha^i)$. Consider the following VM obtained for $S = 2$ and $M = 2$:

$$\begin{aligned} s_{d+1} &= f \\ z_{d+1} &= \begin{cases} \alpha^1 & \text{if } s_{d+1} = 1 \\ \alpha^2 & \text{if } s_{d+1} = 2 \end{cases} \\ \boldsymbol{\xi}_{d+1} &= \begin{cases} \boldsymbol{\eta}^1 & \text{if } s_{d+1} = 1 \text{ and } z_{d+1} = 1 \\ \boldsymbol{\eta}^2 & \text{if } s_{d+1} = 2 \text{ and } z_{d+1} = 1 \\ \boldsymbol{\eta}^3 & \text{if } s_{d+1} = 1 \text{ and } z_{d+1} = 2 \\ \boldsymbol{\eta}^4 & \text{if } s_{d+1} = 2 \text{ and } z_{d+1} = 2, \end{cases} \end{aligned} \tag{22}$$

where

$$f = \zeta. \tag{23}$$

In Equation (22) the second layer variables z_d are thus described by conditional distributions $p(z^j|s)$. This is a regime switching chain which switches among four different VAR(0) autoregressions. Seen the model as a cascade chain, at time $d + 1$, a Bernoulli coin ζ is flipped at first. This first flip assigns a value to s_{d+1}. After this flip, one out of a set of two other Bernoulli coins is flipped in turn, either α^1 or α^2, its choice being conditional on the first coin outcome. This second flip assigns a value to z_{d+1}. Finally, one Gaussian out of four is chosen conditional on the outcomes of both the first and the second flip. Momentarily dropping for clarity the subscript $d + 1$ from the dynamic variables and reversing to mixture notation, the observable sector of this partially hidden hierarchical system is expressed in distribution as a (marginal) distribution in $\boldsymbol{\xi}$, with a double sum on hidden component sub-distribution indexes

$$p(\boldsymbol{\xi}; \{\theta_D\}) = \sum_{j=1}^{M} \sum_{i=1}^{S} p(z^j|s^i) p(s^i) \mathcal{N}(\boldsymbol{\xi}; j, i). \tag{24}$$

The number of components used in Equation (24) is $M \times S \geq S$, and this can be immediately read out from the symbol for the components $\mathcal{N}(\boldsymbol{\xi}; j, i)$. The Gaussians are here conditioned on both second and third hierarchical layer i and j indices. See panel (b) of Figure 3 for a graphical description. Intermediate layer information about the discrete distributions $p(z^j|s^i)$ can be collected in a $M \times S$ matrix

$$W = \begin{pmatrix} w^{1,1} & \dots & w^{1,S} \\ \vdots & w^{m,s} & \vdots \\ w^{M,1} & \dots & w^{M,S} \end{pmatrix}, \tag{25}$$

which supplements the piece of information contained in $\pi = p(s)$. In Equation (25) the rows sum to 1 in order to preserve the probabilistic structure. In this notation Equation (24) becomes

$$p(\boldsymbol{\xi}; \{\theta_D\}) = \sum_{j=1}^{M} \sum_{i=1}^{S} w^{j,i} \pi^i \mathcal{N}(\vec{\xi}; j, i). \tag{26}$$

Notice also that this hierarchical structure is more flexible than writing the same system as a shallow flip of an $M \times S$ faces dice because it fully exposes all possible conditionalities present in the chain, better represents hierarchical clustering, and allows for asymmetries–like having one cluster with two subclusters inside and another cluster without internal subclusters. In addition to the structure of Equation (22), another more compact $J = 3$ hierarchical structure, a useful special case of Equation (22), is

$$\begin{aligned} s_{d+1} &= f \\ z_{d+1} &= \begin{cases} \alpha^1 & \text{if } s_{d+1} = 1 \\ \alpha^2 & \text{if } s_{d+1} = 2 \end{cases} \\ \boldsymbol{\xi}_{d+1} &= \begin{cases} \boldsymbol{\eta}^1 & \text{if } z_{d+1} = 1 \\ \boldsymbol{\eta}^2 & \text{if } z_{d+1} = 2 \end{cases} \end{aligned} \tag{27}$$

with the same f as in Equation (23). This system has distribution (in the simplified notation)

$$p(\boldsymbol{\xi}; \{\theta_D\}) = \sum_{j=1}^{M} \sum_{i=1}^{S} \mathcal{N}(\boldsymbol{\xi}; j) p(z^j|s^i) p(s^i), \tag{28}$$

or, for $h^{j,i} = p(z^j|s^i)$,

$$p(\boldsymbol{\xi}; \{\theta_D\}) = \sum_{j=1}^{M} \sum_{i=1}^{S} \mathcal{N}(\boldsymbol{\xi}; j) h^{j,i} p(s^i), \tag{29}$$

where $S = 2$ and $M = 2$, and where now the Gaussians are conditioned to the next layer only. See panel (c) of Figure 3 for a graphical description. This chain can also model a situation in which two different mixtures of the same pair of Gaussians are drawn. In this case, each state $s_d = i$ corresponds to a bimodal distribution $\sum_{j=1}^{M} \mathcal{N}(\boldsymbol{\xi}; j) h^{j,i}$, unlike the shallow model. This structure, which is a three-layer two-component regime switching dynamics from an econometric point of view, will be specifically used later when discussing hidden Markov mixtures.

Finally, consider the two columns (a) and (c) of the further graphic representation of these structures shown in Figure 4. In Figure 4, column (a), a flat $S = 2$ VM is displayed on the top. Depending on the activated state, two different Gaussians can be activated, and emit in consequence. This structure corresponds to the simple regime switching model represented in Figure 3a, and to the stochastic equation Equation (19).

However, it is the third (rightmost) column c) of the Figure which is the best suited to show how these hierarchical systems are indeed mixtures of mixtures. Matrix Y^{tied} in the Figure gathers information about the composed weights of this structure. Consider the following symbolic example, based on Equation (26), of a $J = 3$ 'deep' system, where different rows correspond to three different Gaussians:

$$W\pi = Y^{\text{tied}}\pi = \begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix} \begin{pmatrix} \pi^1 \\ \pi^2 \end{pmatrix}. \tag{30}$$

The two columns of the matrix Y^{tied} represent the two states of a latent s, the rows represent available Gaussians G^i, and the rows of W (conditional probabilities) sum to 1. This system consists of a mixture of the two mixtures

$$\pi^1(aG^1 + bG^2 + cG^3) + \pi^2(bG^1 + cG^2 + dG^3) = \pi^1 M^1 + \pi^2 M^2,$$

and, however, when flattened, this becomes the linear combination of Gaussians

$$(\pi^1 a + \pi^2 d)G^1 + (\pi^1 b + \pi^2 e)G^2 + (\pi^1 c + \pi^2 f)G^3.$$

It is easy to check that the summed integral on this structure returns 1.

This combination of three Gaussians originating from $S = 2$ states can be used to study data where features could be nested within other structures, due to the underlaying $J = 3$ doubly-stochastic structure. It is the shape and the constraints of W which make it different from a 'shallow' $J = 2$ system with $S = 3$ states. This twin interpretation, in both econometric ($J = 3$ layers) and probabilitstic machine learning ($M = 2$ mixtures) terms of the same system, it is hoped to display how the two cultures can look from different perspectives at the same model.

This Figure will be commented further and in a deeper way in the next sections about correlated dynamics.

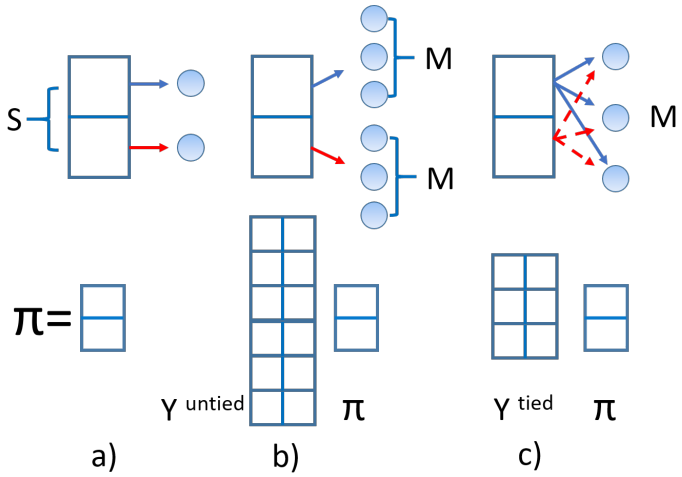

Figure 4. Types of S-state VMs and VHMMs, for $S = 2$, distribution π of s (2×1 vector), and intermediate layer weight matrices $W = Y^{\text{untied}}$ and $W = Y^{\text{tied}}$ (see Equation (26)). (**a**) $J = 2$ shallow mixture: one component per state, no intermediate layer; (**b**) $J = 3$ 'deep' untied model, $S = 2$ mixtures with $M = 3$ components per state that can differ across states, i.e., *six* components in all, associated with Y^{untied} (6×2) for the intermediate layer; (**c**) $J = 3$ 'deep' tied model, $S = 2$ mixtures, $M = 3$ components in all, associated to Y^{tied} (3×2) for the intermediate layer.

3.3. VMs and Forecasting

With VMs, a possible point forecast at day d can be obtained by the use of estimated means. For example, in the case of the model of Equation (22), using the indexed symbol $\boldsymbol{\mu}^{j,i}$ for the vector mean of the conditional Gaussian $\mathcal{N}(\boldsymbol{\xi}; j, i)$, $\hat{\boldsymbol{\mu}}^{j,i}$ for its estimate, and $\boldsymbol{\mu}^f$ for the forecast value, from Equation (26) one obtains

$$\boldsymbol{\mu}^f = \sum_{j=1}^{M} \sum_{i=1}^{S} w^{j,i} \pi^i \hat{\boldsymbol{\mu}}^{j,i}. \tag{31}$$

Once estimated, this 24 h-ahead point forecast cannot go beyond forecasting the same hourly profile $\boldsymbol{\mu}^f$ at any day. Hence, point forecasting of DAM (logarithmic) data with these uncorrelated generative model, independently on how complicated they are, is very bad. For example, for S = 2.

In passing, notice that Equation (31) can also be used to form makeshift daily errors of the type

$$\hat{\boldsymbol{\xi}}_d - \boldsymbol{\mu}^f. \tag{32}$$

As for forecasting, there is a hidden advantage, however. In addition to point forecasting, deep VMs can of course do deep volatility forecasting, through combinations of conditional covariances $\boldsymbol{\Sigma}^{j,i}$, and more complete probabilistic forecasting like quantile or other risk measure forecasting. In the case of a shallow two-component model, the volatility forecast has the form of Equation (17).

4. Uncorrelated Models on Data

Before extending the uncorrelated VM models to correlated models of VHMM type, this section will be discuss some examples of how VMs can extract features from data, and some examples of the scenarios can be generated by them.

4.1. Alberta DAM Data

The theory discussed in Sections 3 and 3.2 will be now applied to the Alberta DAM data. Figure 5 shows one year of Alberta DAM price $\hat{y}_t$ (in Canadian Dollars, upper panel) and logprice $\hat{x}_t = \ln \hat{y}_t$ (lower panel) data, i.e., 8670 hourly values reffering to the one-year period from 7 April 2006 to 6 April 2007 organized in one sequence. This period was chosen due to the variety of price structures observed, with large jumps and spikes, and the presence of a marked mean-reversion component.

In this year of the Alberta market, only positive prices are allowed, and prices are capped. In the upper panel of Figure 5, notice the heavy spiking behavior, which mathematically means strong mean reversion after each spike, and spike clustering. In the lower panel, in which logprices are shown, notice that, besides upward spikes, downward spikes (antispikes) are significantly present (they are magnified by the logarithm). Figure 6 shows a blowup of the full series, two weeks of hourly values, i.e., fourteen individual market data vectors presented in sequence.

Some days are of the spiky type, others are quieter. Notice how spikes appear only during daylight and antispikes appear only during night time, all this due to a sinusoidally time varying demand (see related mathematical discussions in [39,40]). In weekends spikes are less frequent, so that there is a weekly periodicity. Figure 7 shows the same Alberta data, organized in a different way.

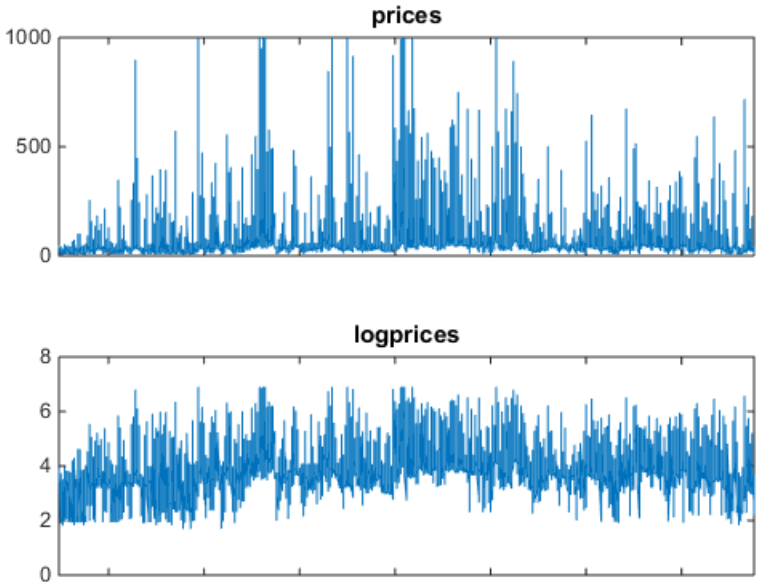

Figure 5. Alberta power market data: one year from 7 April 2006 to 6 April 2007 of hourly data (8760 values). Prices in Canadian Dollars (CAD) in the **upper** panel, and logprices in the **lower** panel. In this market, during this period, prices are capped at CAD 1000. Notice the frequent presence antispikes, better visible in the logprice panel.

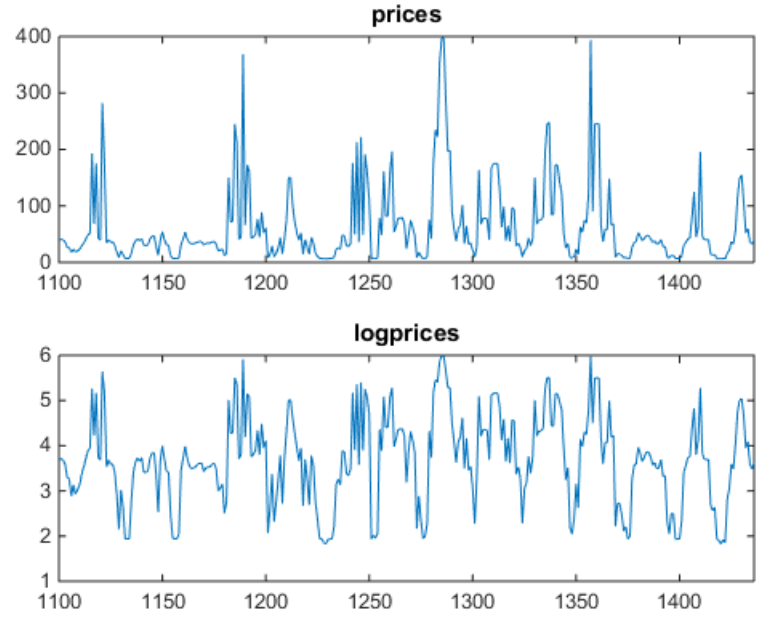

Figure 6. Alberta power market data blowup: two weeks (336 values) of hourly price and logprice profiles. Prices in the **upper** panel, and log-prices in the **lower** panel. Spikes concentrate in central hours, and they never appear during night time. Antispikes never appear during daylight. On the x axis, progressive hour indexes relative to the data set.

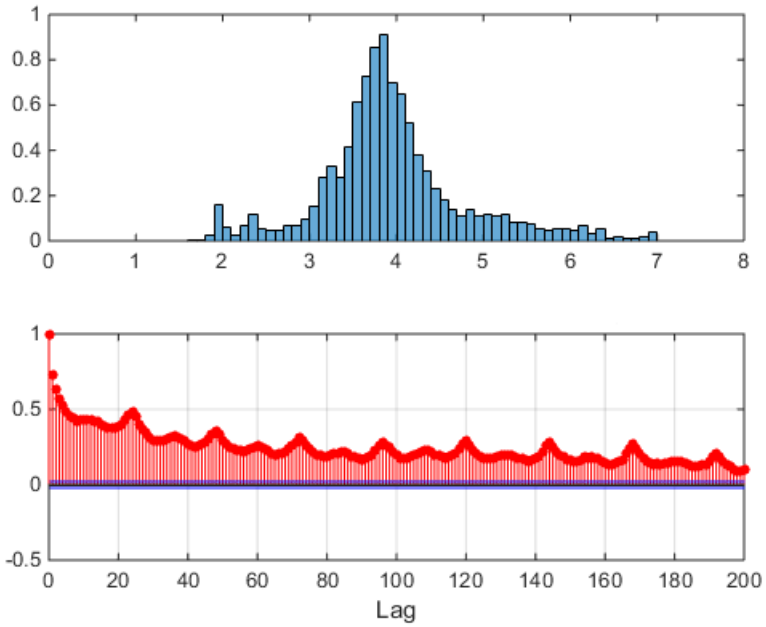

Figure 7. Alberta power market data: one year from 7 April 2006 to 6 April 2007. **Upper** panel: empirical unconditional distribution of logprices, bar areas sum to 1. **Lower** panel: autocorrelation of preprocessed timeseries. For each day, at each hour the full data set average of that hour's logprice was subtracted as $\hat{x}_{h,d} - \bar{\mu}_h$.

The upper panel shows the empirical, unconditional distribution of all logprices in the sample set. Notice that this distribution is not Gaussian, it has thick tails due to spikes and antispikes, and it is skewed to the right since there are more spikes than antispikes (compare with Figure 5). The lower panel shows the sample autocorrelation of logprices, where logprices were preprocessed subtracting at each $\hat{x}_{h,d}$ the corresponding full sample hourly mean

$$\bar{\mu}_h = \frac{1}{N} \sum_{d=1}^{N} \hat{x}_{h,d} \tag{33}$$

where $N = 365$ is the number of samples used. In the autocorrelation, notice the peaks at multiples of 24 h, due to the daily nature of the auctions, and the little bump at hour 12 and its multiples, due to night/day demand alternation. Notice also the very long slow decaying tail, indicating slow decorrelation among days. Such a long tail implies that autoregressions must include many lags. Figure 8 shows sample histograms of logprice distributions at four fixed hours.

Notice evident multimodality and skewness. Morning hour 1 distribution is rather Gaussian in shape, but other hours, like for example peak hours 12 and 18, are not Gaussian at all and at least bimodal. Superimposed to sample distributions, black curves (to be further commented later on), show how a two-component VM can fit these data, and hint at the possibility that mixtures with more components could model them even better. Incidentally notice that in the following analysis, in general, logprice data will be purposely neither detrended or deseasonalized before being used, and Equation (33) will be used only in relation to autocorrelations.

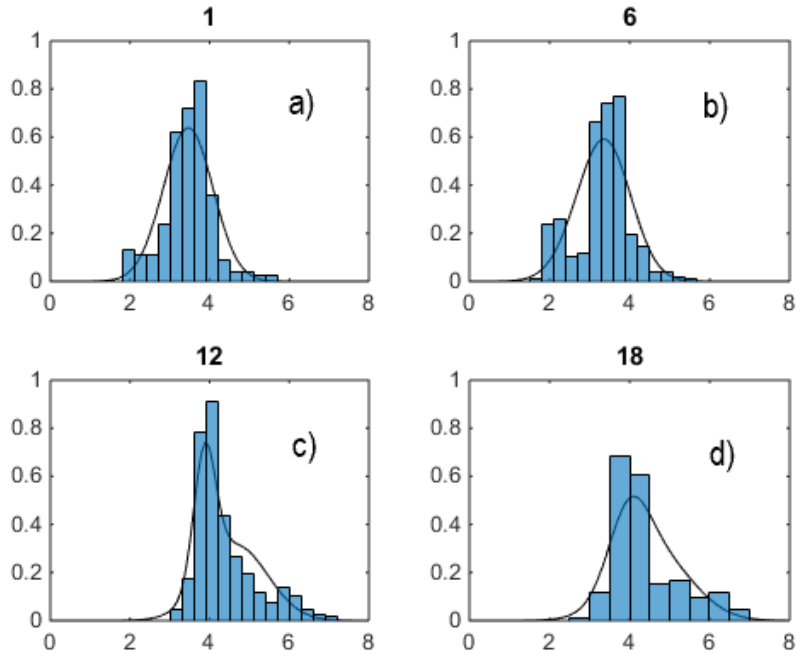

Figure 8. Four choices of logprice hourly sample distributions (histograms, bar areas sum to 1), and estimated marginals (continuous lines) of a two-component VM. (**a**) hour 1; (**b**) hour 6; (**c**) hour 12; (**d**) hour 18. Notice strong multimodality and skewness in panels (**b**–**d**).

4.2. Cluster Analysis, Without and with VMs (i.e., Hard and Soft)

Figure 9 shows a choice of four scatterplots of market logprices, respectively at hour pairs $(1,2)$, $(1,7)$, $(1,13)$, $(1,19)$ (i.e., with the first hour fixed at $h = 1$), where in each panel all pairs in the data set are plot. The dots on these four scatterplots are 2-dimensional projections of points (that is market days) living in the 24 coordinates space. The points tend to lay on the diagonal, i.e., with a positive correlation. A K-means cluster analysis for $S = 2$ finds the centroid components of the two clusters $c^i_{h,h'}$ (where $i = 1,2$, indicated on each panel with a cross), for each pair of hours $h = 1$ and $h' = 2,7,13,19$. Cluster membership of each point is indicated by the light red or dark blue color. Data are rather sharply clustered by the K-means algorithm. One cluster (in the panel south-west sector) contains logprices which are low at hour 1, deep in the night, and stay low at following hours. The other cluster (north-east) contains logprices which are higher and stay higher during the 24 h. A Variance Ratio analysis [41] not reported here graphically, which computes VR_S for a sequence of S values for K-means and soft clustering (i.e., using VMs), shows that VR_S is maximized by $\hat{S} = 2$ for all three methods. This result quantitatively confirms that the market days can be classified in a clear way into two different types. A Bayes Information Criterion (BIC) analysis gives the same result.

4.3. Estimate of a VM and Its Interpretation

Already at this uncorrelated stage, generative modeling based on the latent variables flat VM model with only two components shows a remarkable capability of structuring data automatically, gather both global and local information about market days, and deliver easily interpretable information.

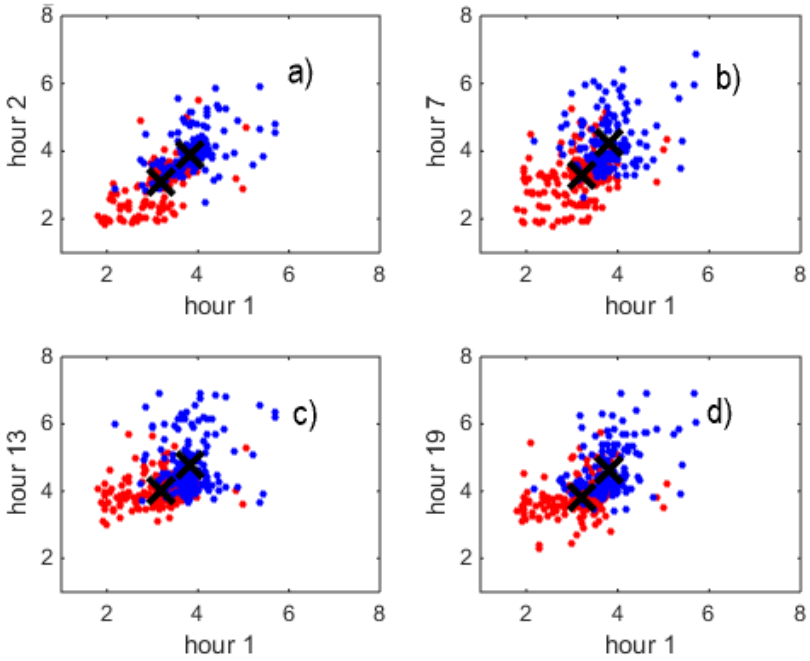

Figure 9. Hour/hour scatterplots, $S = 2$. 2-dimensional projections of clusters of logprices, hard-clustered with k-means in two clusters, a different model for each panel. Light red (south-west) and dark blue (north-east) distinguish the two populations found. Superimposed black crosses represent cluster centroids. First hour fixed at hour 1, second hour fixed at (**a**) 2; (**b**) 7; (**c**) 13; (**d**) 19. Clusters are not so well separated as with local k-means, but here the algorithm behaves in the same way for all cases.

Figure 10 shows a shallow two-component VM fit of the data vs. market data statistics. Each state in this model corresponds to a unimodal distribution. The upper panel shows the hourly sample means $\bar{\mu}_h$ of logprices (solid line), the hours being indicated on the x-axis, each mean bracketed by the related one sample standard deviation interval (the two dotted lines). In practice, the panel shows the average market day, with low logprices during night time and higher logprices during daylight. The lower panel shows the estimated hourly means $\hat{\mu}_h$ (solid line) and their one-standard-deviation boundaries (dotted lines), obtained from the vector $\hat{\boldsymbol{\mu}}$ of Equation (16) and from the square root of the diagonal elements of the matrix $\hat{\boldsymbol{\Sigma}}$ of Equation (17). Two plots are here used to show market data and modeling results because the two curves are indistinguishable.

Figure 11 shows estimated hourly component means $\boldsymbol{\mu}^i$ and component standard deviation from $\boldsymbol{\Sigma}^i$, and respective π^i values ($i = 1, 2$). The component hourly means should be compared to the $S = 2$ hard clustering information of Figure 9. For example, x-axis value $c^1_{1,h'} \approx 3.8$ of first cluster's hour 1 in that Figure has clearly the same value $\hat{\mu}^1_1 = 3.8$ of first component (upper panel of Figure 11), and x-axis value $c^2_{1,h'} \approx 3.8$ of second cluster's hour 1 has clearly the same value $\hat{\mu}^2_1 = 3.8$ of second component (lower panel). This means that the two 24-coordinate Gaussians found with the VM soft clustering analysis match well the two 24-coordinate clusters found with the hard clustering exploration, the main difference between the two methods being the type of distance used by VM clustering, which is the Mahalanobis metric of Equation (14).

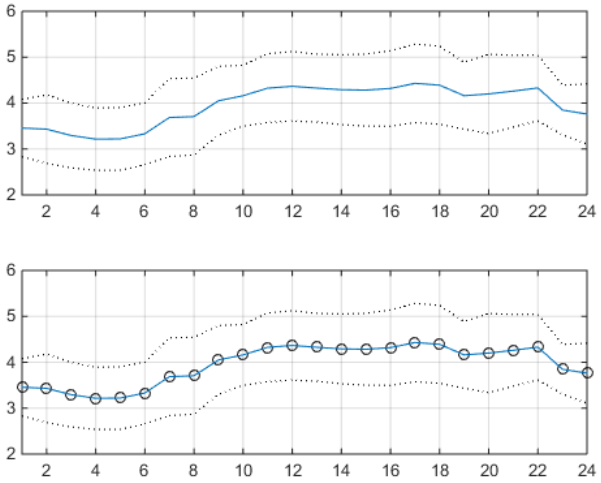

Figure 10. Two-component VM fit of market data statistics. Sample hourly means $\bar{\boldsymbol{\mu}}$ (**upper** panel) and estimated hourly means $\hat{\boldsymbol{\mu}}$ (lower panel) (solid lines) plus/minus corresponding one standard deviation curves (dotted lines). In the **lower** panel, $\bar{\boldsymbol{\mu}}$ is superimposed as a sequence of circles to $\hat{\boldsymbol{\mu}}$, to help comparison–they practically coincide. Here BIC = 13,162.1098. The number of reruns used to obtain this estimate is 300.

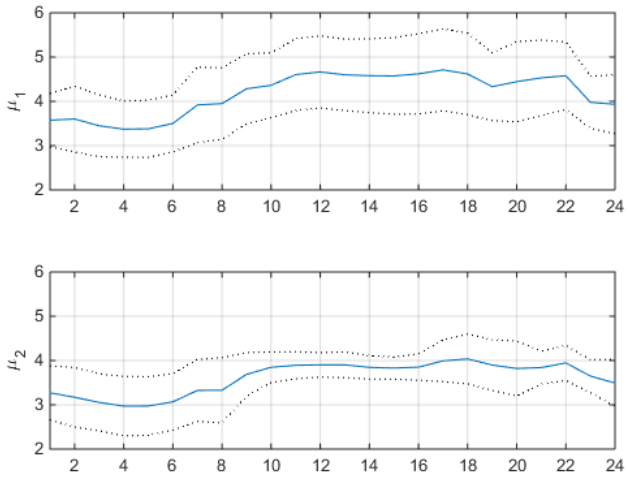

Figure 11. Two-component VM fit modeling the two components. The **upper** panel shows the estimated hourly mean $\hat{\boldsymbol{\mu}}^{\mathbf{1}}$ of the first component (solid line) plus/minus one standard deviation of the component (dotted lines). The **lower** panel shows the estimated hourly mean $\hat{\boldsymbol{\mu}}^{\mathbf{2}}$ of the second component (solid line) plus/minus one standard deviation of the component (dotted lines). Their weights are $\pi^1 = 0.60869$ and $\pi^2 = 0.39131$, their weighted sum is $\hat{\boldsymbol{\mu}}$ in the lower panel of Figure 10.

Notice that looking at the individual profiles of the two components, which have comparable weights π^i, the difference in their hourly standard deviations during daily hours is substantial. Recalling the two-levels interpretation of the shallow two-component VM of Equation (19), having the two components a similar hourly mean profile but having the second component a much lower variance during daylight, this second component can be interpreted as modeling a sort of nightly logprice pattern, to which the more spiky behavior of the first daylight component very often alternates (because $\pi^1 > \pi^2$). A fit of total covariance shows that the sample covariance matrix and the matrix $\hat{\boldsymbol{\Sigma}}$ estimated by Equation (17) practically coincide, as happened for the means.

More interestingly, Figure 12 shows estimated component covariances $\hat{\boldsymbol{\Sigma}}^i$. Here the interpretation of the two components in terms of 'day vs. night' becomes even clearer. The first component has a lot of variance along the daylight diagonal portion whereas the second component has none, showing some variance (the greenish square in the upper left corner of the panel) only during night time, that is when antispikes appear. The possibility of this interpretation in terms of abstraction and of representation learning is strictly connected to the use of a generative model like the VM, which allows for both classification and stochastic time series interpretation. In order to clarify what is the criterion with which the VM model classifies vector data, it is better to consider it as a regime switching zero-lag model.

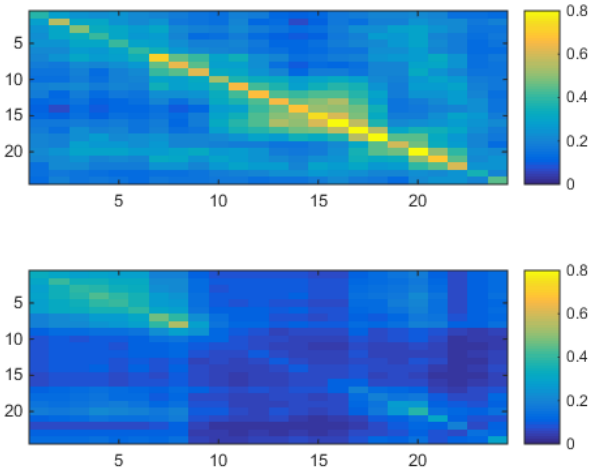

Figure 12. Two-component VM fit. Estimated hourly component covariances. **Upper** panel: first component $\hat{\Sigma}^1_{h,h'}$. **Lower** panel: second component $\hat{\Sigma}^2_{h,h'}$. h on the y- and x-axis. Covariance value scale on the r.h.s.

To illustrate this criterion even more pictorially, Figure 13 shows Mahalanobis distances of individual market days from component 1 and component 2 cluster centers (i.e., the two most typical market days selected by the model) in upper and middle panels, and state (i.e., cluster) membership inferred from posterior probability (i.e., responsibilities π^s_n) in the lower panel. Large distance from cluster 2 corresponds to large posterior probability of belonging to cluster 1. Market days can be classified ex post in terms of their (hidden) type. Mahalanobis distance, as Equation (14) shows, exploits all information available from $\boldsymbol{\mu}^i$ and $\boldsymbol{\Sigma}^i$ (i.e., from the cross-sectional structure shown in Figures 11 and 12). Thus, endowing the 24 dimensions market space with two of these distances (which for $\boldsymbol{\Sigma} = \mathbf{1}$ become the Euclidean distance), and similarity among daily patterns is exploited in a very simple, transparent, and human-readable way. This richness in information of a two-component model can be directly seen in visualizations that explore daily cross-sectional structure, like Figure 14, where the estimated two-component bivariate marginal distribution is shown (from above) for hours 1 and 12. The probability density function for which the isolevels are plot was obtained analytically, just by evaluating Equation (15), which consists of the weighted sum of two bivariate Gaussian functions of which all parameters are known. The small circles represent the selected coordinate pairs. Their color shade is obtained first analytically evaluating on each data point the two components π^i_n with the posterior formula, then associating the daily membership i^{memb}_n to cluster 1 or cluster 2 by computing

$$i^{\text{memb}}_n = \underset{i}{\text{argmax}}\ \pi^i_n. \tag{34}$$

A hard clustering very similar to hard K-means clustering is thus obtained, confirming the day/night interpretation also in terms of what happens at hour 12 to hour 1 logprices. In Figure 14 the two Gaussians have close centers (recall hard cluster positions from Figure 9 and component $\hat{\mu}_h^i$ values from Figure 11), and different peak heights. All of this has obvious potential for accurate scenario generation and for probabilistic forecasting.

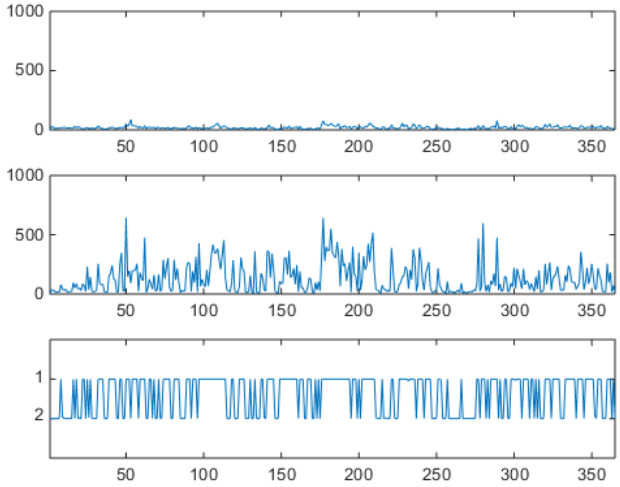

Figure 13. Two-component VM fit. Mahalanobis distances of individual market days from cluster 1 (**upper** panel) and cluster 2 (**middle** panel) centroids. In the **lower** panel the inferred membership at each day, $s_d = 1$ or $s_d = 2$, to one of the two clusters. Time scale in days. Large distance from cluster 2 corresponds to large posterior probability of belonging to cluster 1. Market days can be classified ex post in terms of their (hidden) type.

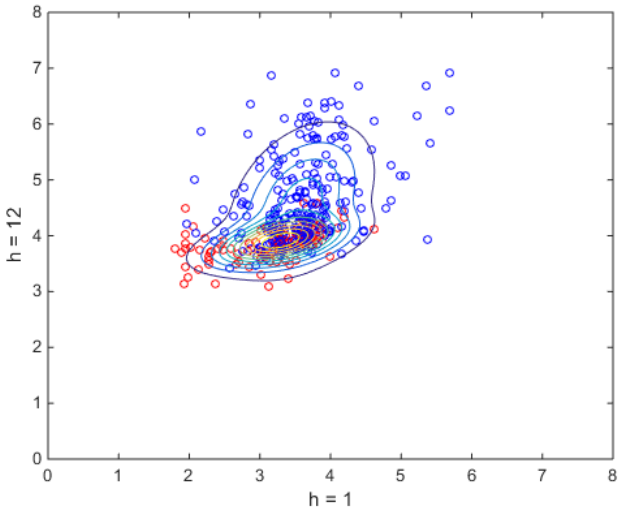

Figure 14. Fully probabilistic two-component VM fit of a choice of a two-hours marginal (bi-marginal), for the hours 1 and 12. Estimated bi-marginal probability density function, represented in terms of level curves, superimposed to the scatterplot of the market data–i.e., the distribution is seen from above. Data points (i.e., market days 2-d projections) of the scatterplot are marked as light red (south-west) or dark blue (north-east) according to membership inferred from soft clustering. The two peaks of the bimodal 2-d distribution are clearly visible. Compare this soft-clustered scatterplot with k-means hard clustered scatterplots, and their centroid positions. Notice that it is not possible to extract a distribution from k-means.

Intuition from Figure 14 can be integrated with the results shown in Figure 8, where the solid curves of univariate marginals, analytically obtained, are superimposed to univariate

sample distribution histograms. In Figure 8, at hours 12 and 18 the two components' profiles can be clearly identified, showing that vector generative modeling can deliver accurate analytical information on DAM logprices' multivariate marginals already with a very small number of components. Three components could do better than two, but a simple shallow two-component modeling requires much less parameters, and can be considered satisfactory enough.

4.4. VM Dynamics and Scenario Generation

This is maybe the core section of this part of the paper, on uncorrelated modeling.

Generative models allow for generation of synthetic data by sampling the estimated distribution. In a sense, they take their name from this feature. In the case of DAM vector modeling, each draw represents one market day of 24 h.

Within the dynamic interpretation of an estimated VM, 365 independent draws reorganized in a scalar series of 8670 market hours simulate one market year. In Figure 15, such a 'synthetic Alberta market' year is shown, for prices (upper panel) and logprices (lower panel). Prices show spikes, logprices show spikes and antispikes. The height profile of synthetic spikes is varied, and should be compared with the profile shown in the upper panel of Figure 5. Spike and antispike variability is evident, whereas spike clustering is absent, since each draw (day) is independent from all other draws in the sequence. A blowup of Figure 15, as shown in Figure 16 where only two weeks are shown, is more striking, when compared with Figure 6 where two weeks of market data were shown. The intra-day structure of synthetic prices and logprices displays very clearly the night/day cycle, with some days more spiky than others, and spikes appearing only during daylight and antispikes appearing during night time, in a casual way. The mechanism behind this great accuracy in the modeling of details is radically different from mechanisms developed in current discrete- and continuous-time models of electricity markets. In multi-component VMs nonlinearity is mainly used for classification and determination of the covariances, and once the right covariances are found, when a component is drawn at the act of sampling, it is the orchestrated interplay among the values of its covariance that generates the daily logprice pattern, in a linear way. Since the components are Gaussian distributions, market configurations distant from cluster centers are exponentially suppressed, so that the model generates only 'credible' (from the model's point of view) market configurations. In contrast, usual models often use their nonlinearity for growing spikes and forcing mean reversion. In multi-component VMs nonlinearity enters again when fully probabilistic features are modeled, as was seen in scatterplots, because in that case the marginals are characterized by the full multi-component gaussian structure. For example, the dots on Figure 14 could be a pictorial representation of the 2d projection of 365 draws from the estimated joint distribution, their density being higher near the peaks of the two Gaussians. From a machine learning point of view, a VM learns many details of the daily patterns, then encodes them in a very abstract way in the covariance matrices. Each day thus becomes a lossy reconstruction of the encoded and compressed data set of the Alberta market. This reconstruction is concentrated in the few support values of the hidden dynamics.

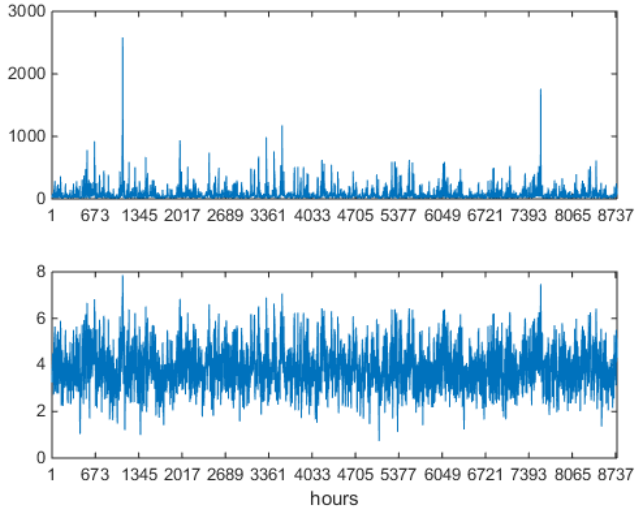

Figure 15. Synthetically generated series of 365 24-component logprice daily vectors, arranged in one hourly sequence 8760 h long, sampled from a two-component VM estimated on Alberta data. **Upper** panel: prices. **Lower** panel: logprices. On the x axis, progressive hour number. To be compared with the market data series of Figure 5.

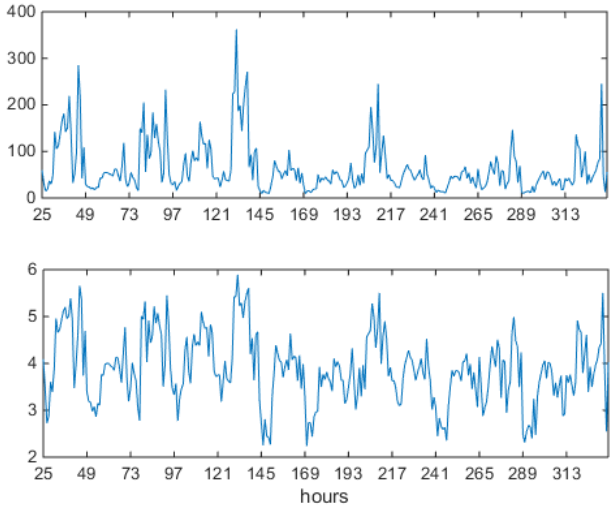

Figure 16. Blowup of the synthetically generated series of Figure 15, two weeks. Prices (**upper** panel) and logprices (**lower** panel). On the x axis, progressive hour number. To be compared with the market data blowup in Figure 6.

Moreover, VM classification into at least two components allows the model to interleave with finely tuned proportions (given by the two mixing weights) 'day' type market days and 'night' type market days (recall from discussion of Equation (19) that a mixture is not just a weighted sum of Gaussians, but also a sequence of individual components). Figure 17, upper panel, shows the unconditional empirical distribution of all synthetic logprices, and has to be compared with upper panel of Figure 7 where market logprices are shown. This panel highlights that the proportions of spikes (right hand tail) and antispikes (left hand tail) are not very well modeled, even though the distribution is correctly skewed to the right (like in the market data, where there are more spikes than antispikes). This distribution too can be found analitically, just summing all 24 gaussian estimated marginal distributions (like those shown in Figure 8) and dividing by 24. The lower panel of Figure 17 shows the autocorrelation function of preprocessed synthetic logprices. Unsurprisingly, there is nonnegligible autocorrelation, but only up to lag 24, that is to one day, even though

the VM model includes no lags for itself. Notice also the little night/day demand bump at hour 12, as in Figure 7. These features are due to the vector nature of the model, that never breaks the intraday logprice relationships, in accordance to the DAM nature of the data and the way information is injected in them. What the VM model does instead is break the interday relationships, not including any information about preceding days. This is reflected into the lack of autocorrelation beyond lag 24. This mechanism should be contrasted with the scalar autoregression approach which forces spurious autocorrelation inside the estimate, thus generating fake memory.

In sum, the econometric interpretation of VM parameters (means, covariances and weights) in the DAM case is straightforward. In constrast, the econometric interpretation of the estimated coefficients in VAR models is necessarily more vague and very difficult to connect to phenomenological features of series. In a sense, whereas a VM can abstract in a concentrated small number of states, a standard autoregression cannot be abstract. Finally, recall also being that simple, zero-lag switching autoregressions share with VMs the same modeling features.

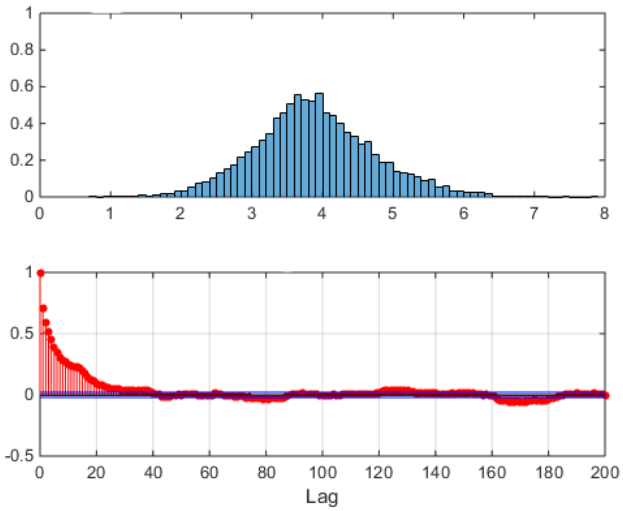

Figure 17. One year of synthetically generated series of logprices, two-component VM estimated on data. **Upper** panel: empirical unconditional distribution of logprices, bar areas sum to 1. **Lower** panel: preprocessed autocorrelation. To be compared with market data autocorrelation in Figure 7. Notice the tail arriving at lag 24 (1 day).

4.5. Forecasting

Equation (31) and the estimate result shown in Figure 10 imply that forecasting with shallow VMs is the same as forecasting by sample averages.

5. Autocorrelated Vector Models and VHMMs

The full power of the discussed generative models is deployed only when time is taken into account explicitly in the structures. In this section the uncorrelated VM models, flat or deep, are taken as the building blocks of a fully dynamic approach, based on a HHM backbone.

5.1. Mixtures of Mixtures and HMM Dynamics

Dynamic econometrics requires models, which have a causal temporal structure. This is usually automatically enforced by defining them by autoregressions like that in Equation (1), i.e., in a discriminative way.

In fully probabilistic (i.e., generative) modeling such a structure must be directly imposed on the joint distribution which defines the model. One possible approach to

enforce a causal structure in a generative model is that of leaving to the hidden dynamics the task of carrying the dynamics forward in time, whereas the observable dynamics is maintained conditionally independent on itself through time. This is what hidden Markov models are designed for [42]. Henceforth, in order to discuss this approach, the symbol for the set of N_D variables $\{\boldsymbol{\xi_{d+1}}, \{\boldsymbol{\xi}\}_D\}$ will be shortened to $\{\boldsymbol{\xi}_{N_D:1}\}$. This choice is intended for both keeping notation light and to remark the specific way in which the dynamic generative approach models the variables, that is all at once. In principle, if the data series has N variables, $N_D = N$.

A vector Gaussian hidden Markov mixture model is defined in distribution as a product chain of densities as

$$\begin{gathered} p(\boldsymbol{\xi}_{N_D:1}, s_{N_D:1}; \{\theta_D\}) = \\ p(s_{N_D:1})\, p(\boldsymbol{\xi}_{N_D:1}|s_{N_D:1}; \{\theta_D\}) = \\ \left(p(s_1) \prod_{d=2}^{N_D} p(s_d|s_{d-1})\right) \left(\prod_{d=1}^{N_D} p(\boldsymbol{\xi}_d|s_d)\right) \end{gathered} \tag{35}$$

for $N_D > 2$, and

$$p(\boldsymbol{\xi}_{N_D:1}, s_{N_D:1}; \{\theta_D\}) = p(s_1)p(\boldsymbol{\xi}_1|s_1) \tag{36}$$

for $N_D = 1$ (i.e., $K_D = 0$) [5]. In Equations (35) and (36) $p(s_1)$ is often called prior. In Equation (35) all $p(s_d|s_{d-1})$ 'dynamic engine' chain rings will be chosen equal, which makes the system time-homogeneous. The conditional distribution $p(\boldsymbol{\xi_d}|s_d)$ will be instead chosen as a convex combination of Gaussians, as explained in the following. For this system, as to the dynamics, stationarity is defined as that condition in which $p(s_{d+d'}) = p(s_d)\ \forall d, d'$. This means that the dynamics must stay the same in time. A time-homogeneous system which has reached stationarity has hence the property that

$$p(s_d) = \sum_{s_d} p(s_{d+1}|s_d)\, p(s_d) \tag{37}$$

for all d, a useful property. It should be yet also recalled out that homogeneity in time doesn't guarantee stationarity.

Notice the overall form $p(x|s)p(s)$ of this model, like in the VM case. The observable joint (w.r.t. intraday dynamics) marginal (w.r.t. observable variables) distribution $p(\boldsymbol{\xi}_{N_D:1})$ is obtained from Equation (35) by summing over the support of all s_n, which makes this model a generalized vector mixture. For example, by summing on s_1 Equation (36) one recovers the observable static S-component mixture

$$p(\boldsymbol{\xi}_1) = \sum_{s_1=1}^{S} p(s_1)p(\boldsymbol{\xi}_1|s_1),$$

used to define the shallow (i.e., not-hierarchical) VM. Since the s_n have a discrete support, it is possible to set up a vector/matrix notation for the daily hidden marginals

$$(\pi_d^1, \ldots, \pi_d^S)' \equiv \{p(s_d)\} \tag{38}$$

and for the constant transition matrix

$$A = \begin{pmatrix} a^{1,1} & \ldots & a^{1,S} \\ \vdots & & \vdots \\ a^{S,1} & \ldots & a^{S,S} \end{pmatrix} \equiv \{p(s_d|s_{d-1})\} \tag{39}$$

of entries $a^{i,j}$, where $\pi_d^i \in [0,1]$ and $a^{i,j} \in [0,1]$, with $\sum_{i=1}^{S} \pi_d^i = 1 \; \forall d$ and $\sum_{i=1}^{S} a^{i,j} = 1 \; \forall j$ (columns sum to one).

In this notation, for example for $S = 2$, the dynamics of the hidden distribution becomes a multiplicative rule of the form

$$\begin{pmatrix} \pi_d^1 \\ \pi_d^2 \end{pmatrix} = \begin{pmatrix} a^{1,1} & a^{1,2} \\ a^{2,1} & a^{2,2} \end{pmatrix} \begin{pmatrix} \pi_{d-1}^1 \\ \pi_{d-1}^2 \end{pmatrix} \tag{40}$$

which starts from an initial distribution $\pi_1 \equiv (\pi_1^1, \pi_1^2)' \equiv p(z_1)$. Under A, after n steps forward in time, π_1 becomes

$$\pi_n = A^n \pi_1, \tag{41}$$

where the superscript n on A indicates its n-th power. Time-homogeneous hidden Markov models can be hence be considered 'modular' because, for a given N_D, the hidden dynamics of $p(\boldsymbol{\xi}_{N_D:1})$ is completely described by the prior π_1 and a single matrix A which repeats itself as a module for $N_D - 1$ times. This means that, after an estimate on the N_D data, a global A will encode information from all $N_D - 1$ interday transitions, which are local in time. To highlight this feature, sometimes a subscript can be attached to A, like for example A_{N_D}.

The number of parameters required for the hidden dynamics of a hidden Markov model is thus $(S-1) + (S^2 - S)$ in all, usually a small number. To this number, a number of parameters for the observable sector should be added for each state. If Gaussians are used, this number can become very large. Hence, the total number of parameters depends in general mainly on the probabilistic structure of the observable sector of the model. The basic idea behind VHMMs is to take the distribution $p(\boldsymbol{\xi_d}|s_d)$ of the observable dynamics as a vector Gaussian mixture, i.e., to piggyback a VM module onto a hidden Markov dynamics backbone, taking also care not to end up with too many Gaussians. This will allow obtaining correlated dynamic vector generative models which are deep and rich in behavior but at the same time parsimonious in the number of parameters, and which can forecast in a more structured way than VMs.

Consider at first the shallow case in which $p(\boldsymbol{\xi_d}|s_d) = \mathcal{N}(\boldsymbol{\xi_d}; s_d)$. In this case, Equation (35) implies using S Gaussians in all, as many as the states. Ancestor sampling means picking up one of these Gaussians from this fixed set at each time d. This was pictorially represented in panel (a) of Figure 4 (already commented before), were one module of the model is shown with two hidden states emitting one Gaussian each. This model is the correlated equivalent of the (uncorrelated) shallow VM of Equation (19), which was represented in Figure 3a.

The shallow model can be extended by associating one M-component Gaussian mixture to each of the S hidden states, for S discrete distributions in all, each hidden state thus supporting a vector of M weights and a M-components mixture distribution. This configuration is represented in panel (b) of Figure 4, where the two hidden states emit three Gaussians each. The required weight vectors can be gathered into a matrix Y^{untied} with $M \times S$ non-negative entries, each row of which sums to 1. This is equivalent to using on each day a module of the $J = 3$ deep VM of Equation (22), which was represented in Figure 3b. This means that Y^{untied} corresponds to the matrix W of Equation (25) and that the Gaussians $\mathcal{N}(\boldsymbol{\xi_d}; j, i)$ of this model are conditioned both on the intermediate and on the top layer of the hierarchy. This configuration requires $M \times S$ Gaussians in all to be estimated, potentially a lot, each with $(U^2 + 3U)/2$ parameters where $U = 24$, in addition to A, π_1 and Y^{untied}.

A more parsimonious model can be obtained by choosing to use M Gaussians instead of $M \times S$. This is obtained by selecting M Gaussians and associating to each of the S hidden

states an s-dependent mixture from the fixed pool of these M Gaussians, in a tied way, in the same way as the $J = 3$ deep VM of Equation (27) represented in Figure 3c does. Information about the mixtures can be represented by a $M \times S$ matrix Y^{tied}, formally similar to Y^{untied}, but related to a different interpretation. The Gaussians now have form $\mathcal{N}(\xi_d; j)$ since they are conditioned on the next layer only. This setting is represented in panel (c) of Figure 4, where each of the two hidden states is linked to the same triplet of gaussians. Being able to tie a hierarchical mixture is therefore all-important for parsimony because in the tied case one can have a large number of hidden states but a very small number of Gaussians, maybe two or three, and an overall number of parameters comparable to that of a VAR(1), which has the same memory depth. This last model will be called *tied model* in contrast with the model with $M \times S$ Gaussians, which will be called *untied model*. The discussed tied model is the smallest model that contains all vector, generative, and hidden state features in a fully correlated dynamic way.

From an econometric point of view, the untied and tied VHMM models are deep regime switching VAR(0) autoregressions, with stochastic equations given by Equation (22) and Equation (27), where f of Equation (23) is replaced with

$$f = S_A \, s_d. \tag{42}$$

In Equation (42), S_A is the stochastic matrix associated to the transition matrix A of Equation (39), i.e., the stochastic generator of the hidden dynamics. Therefore in these models the hidden dynamics evolves according to a linear multiplicative law for the innovations, whereas the overall model results nonlinear. Since the underlying hidden Markov model is modular, it is possible to write in probability density the representative module of these models. For the untied model this module is

$$\begin{aligned} \pi_{d+1} &= A \, \pi_d \\ p(\xi_{d+1}) &= \sum_{j=1}^{M} \sum_{i=1}^{S} \mathcal{N}(\xi_{d+1}; j, i) w^{j,i} \pi^i_{d+1}, \end{aligned} \tag{43}$$

having used a mixed notation and having compressed in $w^{j,i}$ (i.e., in $W = Y^{\text{untied}}$) the 'static dynamics' of the α^i, i.e., the parametrization of the intermediate layer. Equation (43) corresponds to the one-lag regime switching chain of Equation (22). For the tied model,

$$\begin{aligned} \pi_{d+1} &= A \, \pi_d \\ p(\xi_{d+1}) &= \sum_{j=1}^{M} \sum_{i=1}^{S} \mathcal{N}(\xi_{d+1}; j) h^{j,i} \pi^i_{d+1}, \end{aligned} \tag{44}$$

where $h^{j,i}$ are the entries of $H = Y^{\text{tied}}$, i.e., the piece of information about the intermediate layer. This dynamic probabilistic equation corresponds to the stochastic chain of Equation (27). Hence, suitable regime switching VAR(0) autoregressions can have all the properties of the generative correlated VHMM models, clustering capabilities included. One could also say that Equations (43) and (44) are the machine learning, $K_D = 0$ vector generative correlatives of the vector discriminative dynamics of Equation (1), which in contrast has no hidden layers, it is based on additive innovations, and can have $K_D > 0$. Noticeably, the hidden stochastic variables α^i of the intermediate layer of the untied and tied models don't have an autonomous dynamics, like s_d has. But, if needed, they can be promoted to have it as well without changing the essential architecture of the models. More-

over, being Markovian, these dynamic models incorporate a one-lag memory, controlled by A. But, if needed, this memory can be in principle extended to more lags, by replacing the first order Markov chain of the hidden dynamics with an higher order Markov chain.

Finally, from a simulation point of view, each draw from these systems will consist of one joint draw of all variables $\{\vec{\xi}_{N_D:1}, s_{N_D:1}\}$ at once. This is typical of generative models. From a more econometric point of view, this one draw can be seen as a sequence of draws from s_d, local in time, each causally dependent on the preceding one only, and with a deep cascading component on their top at each time.

5.2. VHMMs and Estimation

Estimation of VHMMs on data follows the steps of standard HMM estimation. For HMM estimation there is no analytic solution, although a maximum loglikelihood can be derived efficiently using the Baum–Welch algorithm [15] developed in the early 1970s (as hinted to in the Introduction, one of its main original applications was to speech and language processing). However, like in the case of mixtures, the algorithm can search for local maxima only, and is not able to guarantee convergence to the local optimum, a problem in common with most kinds of contemporary neural network. The Baum–Welch algorithm, based on an iteration of forward and backward passes along the Markov chain, is an extension, or maybe a special case of the expectation–maximization algorithm already discussed in the uncorrelated part.

As described in the Introduction, all numerical results for this paper were obtained with `BNT` and `pmtk3`.

5.3. VHMMs and Forecasting

Forecasting with correlated models is not direct, and it relies on some assumptions. It is based on Equations (43) and (44). In the following it will be discussed only the case of the untied model, in order to compare it with the VM case of Section 4.5. A discussion of the tied model leads to the same conclusions.

Point forecast $\boldsymbol{\mu}^f$ at day d for the next 24 h can be made using the mean. Estimating an untied S-component model gives $\hat{\boldsymbol{\mu}}^{j,i}$, $\hat{\boldsymbol{\Sigma}}^{j,i}$, W, A and π_1. Inserting the first row of Equation (43) in the l.h.s. of the second equation, then taking the expected value leads to

$$\boldsymbol{\mu}^f = \sum_{j=1}^{M} \sum_{i=1}^{S} \sum_{k=1}^{S} \hat{\boldsymbol{\mu}}^{j,i} w^{j,i} a^{i,k} \pi_d^k. \tag{45}$$

In Equation (45), the distribution of the current hidden variable π_d is in principle unknown. There are a few different ways to estimate π_d, with different consequences on the forecast. The most interesting of them requires one to write Equation (43) at $d-1$, with A and W taken from the estimated model, and to assume $\pi_{d-1} = \pi$. In this way a maximum likelihood estimate $\hat{\pi}_d$ of π_d can be made as an inference, conditional on having seen on day d the recorded value of $\boldsymbol{\xi_d}$, and containing all of the estimated information.

A key limit of VHMM forecasting comes from the following. Since Equation (45) is a convex combination of $\hat{\boldsymbol{\mu}}^{j,i}$, all values (but only those) between maximum and minimum of these values can be obtained, in a continuous way. Unlike the VM case, these combinations can now change without always repeating themselves as time goes by. However, since the dynamics of the DAM dataset are really wild, such a limitation kills reactivity, eading to results very similar to those of the linear VAR approaches.

However, like in the VM case, the VHMM approach has the advantage that variance and actually covariance can be directly forecast as well, in the same way. Probabilistic forecasting, then quantile and VaR forecasting, can be made in the same way.

6. Correlated Models on Data

Correlated deep models like the untied and the tied VHMMs overcome the structural limit that prevents uncorrelated VMs to generate series with autocorrelation longer than 24 h. In order to discuss this feature in relation to Alberta data, two technical results [43] are first needed.

First, if the Markov chain under the VHMM is irreducible, aperiodic and positive recurrent, as usually estimated matrices A with small S ensure, then

$$\lim_{n\to\infty}(a^{j,i})^n = \pi^j_\infty \qquad \forall i. \tag{46}$$

Recalling Equation (41), Equation (46) means that after some time n the columns of the square matrix A^n, seen as vectors, become all equal to the same column vector $\pi_\infty = (\pi^1_\infty, \ldots, \pi^S_\infty)'$. Second, at the same conditions and at stationarity, if $v_n(j)$ counts the number of times a state $s = j$ has been visited up to time n, then

$$\lim_{n\to\infty}\frac{v_n(j)}{n} = \pi^j_\infty, \tag{47}$$

i.e., the components of the limit vector π_∞ give the percent of time the state j is occupied during the dynamics.

Figure 18 shows for a two-component $S = 2$ tied VHMM the two estimated covariance matrices of the model, to be compared with Figure 12 for which a two-component uncorrelated shallow VM model was used.

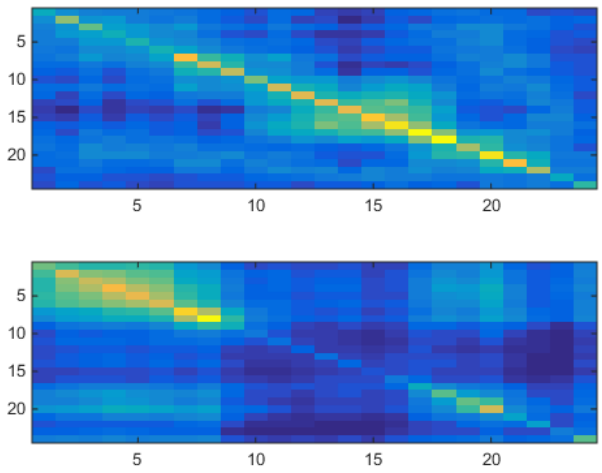

Figure 18. Two-component $S = 2$ tied VHMM fit. Estimated hourly component covariances. **Upper** panel: first component $\hat{\Sigma}^1_{h,h'}$. **Lower** panel: second component $\hat{\Sigma}^2_{h,h'}$. h on the y- and x-axis. Compare with the two components of Figure 12 obtained in the $S = 2$ shallow VM case.

The upper panel of Figure 18 has an analog in the upper panel of Figure 12. Both covariances have their highest values along their diagonal, very low values off-diagonal, and high values concentrated in the daily part. The lower panel of Figure 18 has an analog in the lower panel of Figure 12. Both covariances have their highest values along their diagonal, very low values off-diagonal, and high values concentrated in the night part. Namely, the correlated VHMM extracts the same structure as that extracted by the uncorrelated VM, i.e., a night/day structure.

Besides covariances and means, another estimated quantity is

$$\Upsilon_{yr}^{\text{tied}} = \begin{pmatrix} 0.8754 & 0.1246 \\ 0.2530 & 0.7470 \end{pmatrix}$$

(weights of each hidden state value are along rows). This means that, from the point of view of the tied model, each market day contains the possibility of being both night- or day-like, but in general each day is very biased towards being mainly day-like or mainly night-like. The last piece of information is contained in the estimated

$$A_{yr} = \begin{pmatrix} 0.8754 & 0.2530 \\ 0.1246 & 0.7470 \end{pmatrix}, \quad \pi_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

(columns sum to one). After about $n = 20$ days A_{yr}^n reaches stationarity becoming

$$A_{yr}^{n>20} = \begin{pmatrix} 0.6700 & 0.6700 \\ 0.3300 & 0.3300 \end{pmatrix}.$$

As Equation (46) indicates, this gives $\pi_\infty = (0.67, 0.33)'$, which means, in accordance with Equation (47), that the system tends to spend about two thirds of its time on the first of the two states, in an asymmetric way.

Once this information is encoded in the system, i.e., the parameters $\{\hat{\theta}_D\}$ are known by estimation, a yearly synthetic series ($N_D = 365$) can be generated, which will contain the extracted features. The series is obtained by ancestor sampling, i.e., first by generating a dynamics for the hidden variables s_d using the first line of Equation (27) with

$$f = S_A s_d \tag{48}$$

(to be compared with Equation (23)), then for each time d by cascading through the two-level hierarchy of the last lines of Equation (27) down to one of the two components. The obtained emissions (i.e., logprices and prices), organized in a hourly sequence, are shown in upper and middle panels of Figure 19, to be compared both with Figure 15 obtained with the shallow $S = 2$ VM and with the original series in Figure 5.

The hourly series shows spikes and antispikes, but now spikes and antispikes appear in clusters of a given width. This behavior was not possible for the uncorrelated VM. The VHMM mechanism for spike clustering can be evaluated by looking at the lower panel of Figure 19 where the daily sample dynamics $\hat{s}_{N_D:1}$ is shown in relation with the hourly logprice dynamics. Spiky, day-type market days are generated mostly when $s_d = 1$. Look for example at the three spike clusters respectively centered at day 200, beginning at day 250, and beginning at day 300. Between day 200 and 250, and between day 250 and 300 night-type market days are mostly generated with $s_d = 2$. Once in a spiky state, the system tends to remain in that state. Incidentally, notice that the lower panel of Figure 19 is not a reconstruction of the hidden dynamics because when the generative model is used for synthetic series generation the sequence s_d is known and it is actually not hidden. It should also be noticed that the VHMM logprice generation mechanism is slightly different from the VM case for a further reason too. In the VM case, the relative frequency of spiky and not spiky components is directly controlled by the ratio of the two component weights. In the VHMM case each state supports a mixture of both components. The estimation creates two oppositely balanced mixtures, one mainly day-typed, the other mainly night-typed. The expected permanence time on the $s_d = 1$ state, given by π_∞ (i.e., by A_{yr}), controls the width of the spike clusters. A blowup of the sequence of synthetically generated hours is

shown in Figure 20, to be compared with the VM results in Figure 15 and the original data in Figure 6.

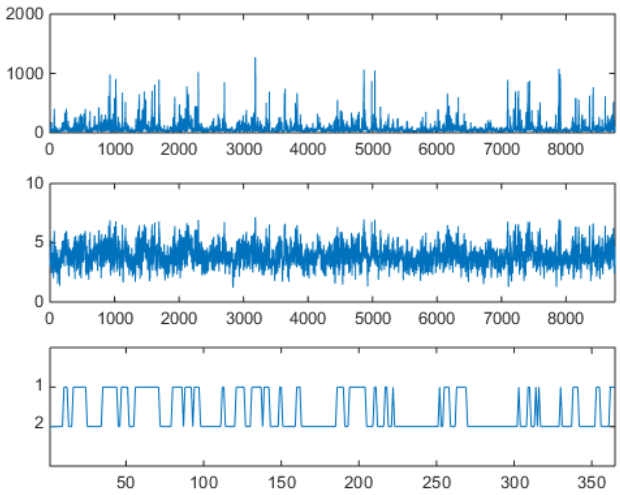

Figure 19. Two-component $S = 2$ tied VHMM estimated yearly on logprice data. Synthetically generated series for prices, logprices and hidden states. **Upper** panel: prices. **Middle** panel: logprices. Hours on x-axis for **upper** and **middle** panels. **Lower** panel: state trajectory s_d. Days on x-axis for the **lower** panel. State membership controls spikiness: for $s_d = 1$ spikiness is higher than for $s_d = 2$. Compare with component covariances of Figure 18.

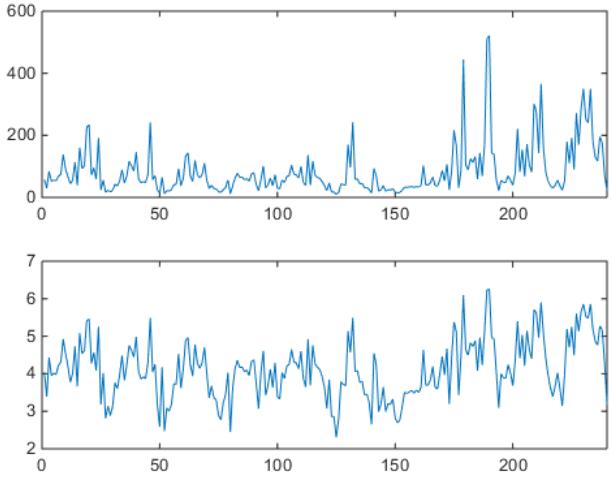

Figure 20. Two weeks of synthetically generated series of prices (**upper** panel) and logprices (**lower** panel), one full year estimate, $S = 2$ two-component tied VHMM estimated on logprice data. Hours on the x-axis for both panels. To be compared with market data series of Figure 6.

The aggregated, unconditional logprice sample distribution of all hours is shown in the upper panel of Figure 21, where left and right thick tails remind the asymmetrically contributing spikes and antispikes, whose balance depends now not only on static weight coefficients but also on the type of the dynamics that A_{yr} is able to generate.

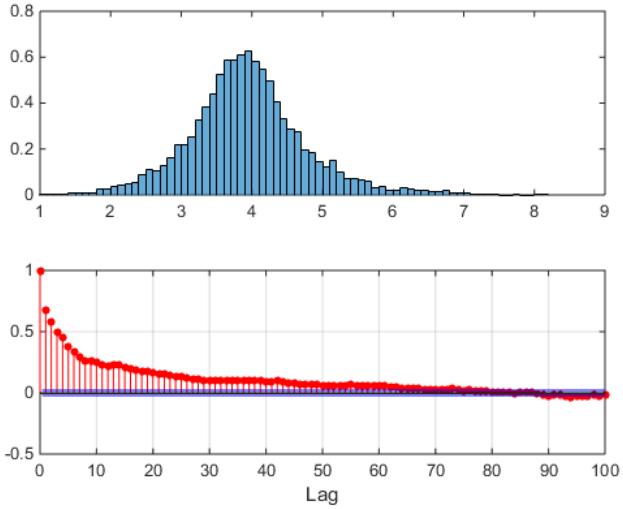

Figure 21. One year of synthetically generated series of logprices, $S = 2$ two-component tied VHMM estimated on one year of logprice data. **Upper** panel: empirical unconditional distribution of logprices, bar areas sum to 1, logprices on the x - axis. **Lower** panel: preprocessed autocorrelation. Notice the tail arriving at lag 48 (2 days). To be compared with the autocorrelation of original market data series of Figure 7 and to the VM autocorrelation of Figure 17, which arrives at lag 24.

In the lower panel of Figure 21, the sample preprocessed correlation function is shown, obtained subtracting Equation (33) from the hourly data, as in Figure 17 (VM) and Figure 7 (market data). Now the sample autocorrelation for the sample trajectory of Figure 19 extends itself to hour 48, i.e., to 2 days, due to the interday memory mechanism. Not all generated trajectories will of course have this property, each trajectory being just a sample from the joint distribution of the model, which has peaks (high probability regions) and tails (low probability regions). An example of this varied behavior is shown in Figure 22, where sample autocorrelation was computed on three different draws. The lag 1 effect is always possible, but it is not realized in all three samples, as can be seen in the lower panel of Figure 22.

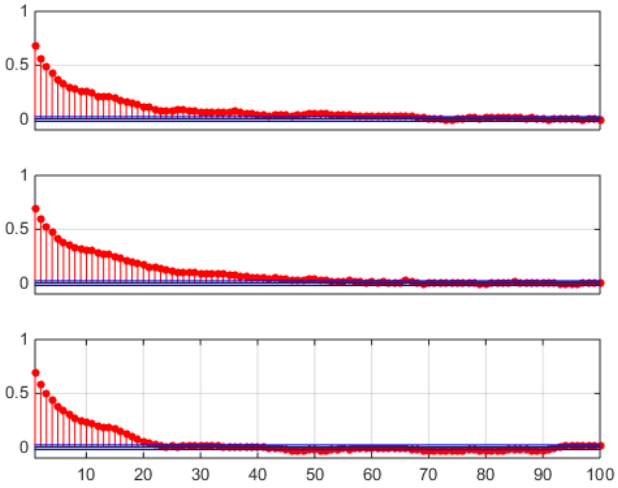

Figure 22. Sample autocorrelation functions of three synthetically generated samples from a two-component $S = 2$ tied VHMM estimated yearly on logprice data. **Upper**, **middle** and **lower** panels: sample autocorrelations. Lag scale on the x-axis extends to four days. These profiles are rather different. Notice the short autocorrelation in the **lower** panel.

These results were discussed using machine learning terminology, but they could have been discussed using switching $0-$lag autoregressions terminology as well.

As to probabilistic forecasting, these models must be improved. They are in a sense too much 'rigid'. They use most of their parameters to to excellent featurization and structure extraction (think of the covariance structure), but they relay on statistical means only to do point forecasting. An example of a tight VHMM model out-of-sample forecasting quality can be seen in Figure 23, where (makeshift) forecast errors aggregated over the 24 h, obtained from Equation (32) for the weekly two-component tied VHMM on logprices, are shown, computed on two distinct training and testing sets of equal length. In this plot, a week starting from Monday is used to forecast next Monday, for 50 weeks. In the figure, MAE stands for Mean Average Error, MAPE for Mean Absolute Percent Error, MSE for Mean Square Error, MSPE for Mean Square Percentage Error, all aggregated on the aggregated 24 h. Considering that the average hourly logprice is about 4 (this can be extracted by eye from Figure 5), and that from the MAE plot one can estimate by eye that this MAE is about 20 so that the hourly MAE is about 1 (20/24 h), the hourly error is about one fourth of the ground truth, which is very high. In practice, this is the forecast quality that one gets form standard VAR error on a such complicated DAM time series. Forecasting is not a strong point of this kind of generative modeling. Generation it is.

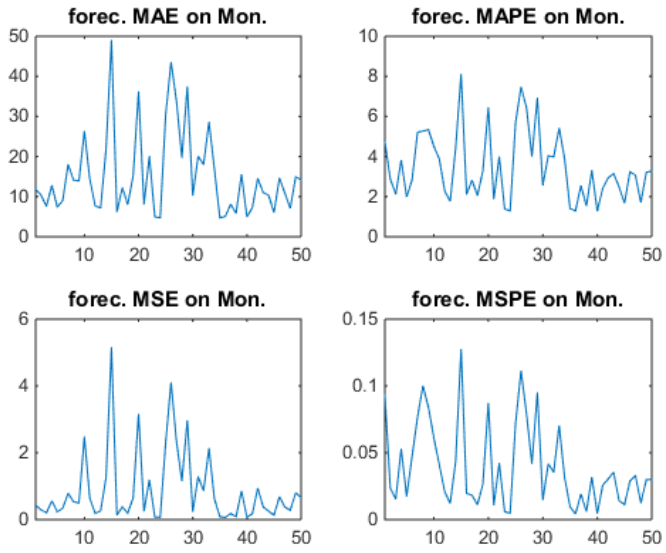

Figure 23. Forecast errors for the weekly two-component tied VHMM estimated on logprice data. A week starting from Monday is used to forecast next Monday. Week index on the x axis.

7. Conclusions

As discussed so far, mixture-based models are usually approached by the econometrics and CI communities with two different formalisms, stochastic equations and purely probabilistic approaches. Hence, our proposed VM/VHMM family could misleadingly appear from these two perspectives as two different mathematical entities to the two communities. From a mathematical point of view it is not correct, and this paper has tried to reconcile these seemingly different points of view at least as to the proposed specific modeling frame, strictly linked to DAM price modeling. The VM/VHMM family is based on Gaussian mixtures and HMMs, the HMM being a very well known approach used in the last 60 years for generative purposes, and the mixture approach has been very often used *per se* in practical situations [44]. In this paper, it was shown that the members of the VM probabilistic family correspond to the Gaussian regime switching 0-lag vector autoregressions of standard econometrics, studied there using a stochastic difference equations approach, and that the

other generative models presented can be written as doubly stochastic dynamic equations in terms of latent variables. Actually, already at a first inspection VMs/VHMMs and regime switching autoregressions should suggest that they share some features. They are both based on latent variables, and are both intrinsically nonlinear. Moreover, it was shown that VMs/VHMMs too do exploit a 'depth dimension' which can be very useful for modeling important details of the data. This can now be summarized in the sketch proposed in Figure 24, where it is depicted the 'spatio'-temporal structure of a VHMM seen as a deep machine learning model. In the figure, J is the model 'depth' indicated in the vertical axis. On the bottom axis it is shown the 24 h of the day related to the 24 components of each vector (a point) of the series dataset, on the horizontal axis the number of time lags used in a given model. As discussed, this means 0 lags for VMs, and 1 lag only for VHMMs, by contruction. This sketch thus also highlights the limits of the VHMM approach: the time memory is very limited.

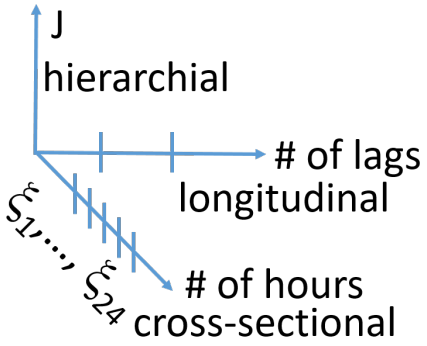

Figure 24. Spatio-temporal structure of a VHMM seen as a deep machine learning model. J is the model 'depth' z. On the bottom axis the 24 h of the day related to the 24 components of each vector (a point) of the series dataset, on the horizontal axis the number of time lags used in a given model (0 lag for VMs, 1 lag only for VHMMs, by contruction).

It is true that, being vector models, both VM and VHMM models can remove fake memory effects from hourly time series by directly modeling them as vector series. However, as seen in the autocorrelation plots, such HMM-based modeling cannot include more than some day in the generated series. Higher order (in time) HMMs do exist, but are very hard to implement. In addition, the discussions about their forecasting ability showed that they can indeed be used as forecasters, but their quality in this capacity is for sure not excellent. This is linked to a certain rigidity with which they forecast. These two weaknesses can be overcome with more research in these directions.

Yet, these model encapsulate common useful properties of mixtures that are usually ascribed to the machine learning side only, but not usually exploited in dynamical models. For example, it was shown that because VMs and VHMMs can do unsupervised deep learning (which however current deep learning models usually do not do), they can automatically organize time series information in a hierarchical and transparent way, which for DAMs lends itself to direct market interpretation. Being VM and VHMM models generative, the synthetic series which they produce are outstanding, and automatically include many of the features and fine details present in time series, in the specific of a very complex time series like that of Alberta DAM prices. These model features could be incorporated in more advanced, deep machine learning models.

Interestingly, VMs and VHMMs have also features shared with regime switching 0-lag vector autoregressions, which usually are not liked by econometricians. Being generative models, they are fit directly as distributions, and they are not based on errors

(i.e., innovations) unlike non-zero lag autoregressions and other discriminative models for time series [45,46], so that cross-validation, residual analysis and direct comparison with standard autoregressions is not straightforward. Moreover, being generative and based on hidden states, the way they forecast is different from usual non-zero lag autoregression forecasting. This dislike comes maybe from the culture split, and goes at detriment of research.

This paper thus shows that it can be interesting and useful to work out in detail these common properties hidden behind the two different formalisms adopted by the two communities. It can hence be also interesting to try to better understand how these common properties can be profitably used in DAM prices scenario generation and analysis when developing more sophisticated models.

Finally, one should consider that in DAM price forecasting, linear autoregressions have been always easily interpreted, whereas neural networks started out as very effective but a 'black box' type of tools. However, linear autoregressions are in general bad at reproducing fine details of DAM data (like big spikes or 'fountains' of spikes). The vector hierarchical hidden Markov mixtures discussed in this paper are on the contrary excellent at that. They are thus probably a seeding example of an intermediate class of models that are both accurate in dealing with data fine details, easy to interpret, not complex at all, sporting a very low number of parameters, and palatable for both communities. It is also hoped that this approach will lead to a more nuanced understanding of price formation dynamics through latent regime identification, while maintaining interpretability and tractability, which are two essential properties for deployment in real-world energy applications.

Author Contributions: Conceptualization, C.M. and C.L.; Methodology, C.M. and C.L.; Software, C.L.; Validation, C.M.; Formal analysis, C.M. and C.L.; Data curation, C.L.; Writing—original draft, C.M. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Weron, R. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *Int. J. Forecast.* **2014**, *30*, 1030–1081. [CrossRef]
2. Nowotarski, J.; Weron, R. Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renew. Sustain. Energy Rev.* **2018**, *81*, 1548–1568. [CrossRef]
3. Mari, C.; Baldassari, C. Ensemble methods for jump-diffusion models of power prices. *Energies* **2021**, *14*, 2084. [CrossRef]
4. Nitsch, F.; Schimeczek, C.; Bertsch, V. Applying machine learning to electricity price forecasting in simulated energy market scenarios. *Energy Rep.* **2024**, *12*, 5268–5279. [CrossRef]
5. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; MIT Press: Boston, MA, USA, 2012.
6. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2007.
7. Olivares, K.G.; Challu, C.; Marcjasz, G.; Weron, R.; Dubrawski, A. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *Int. J. Forecast.* **2023**, *39*, 884–900. [CrossRef]
8. Jiang, H.; Dong, Y.; Dong, Y.; Wang, J. Probabilistic electricity price forecasting by integrating interpretable model. *Technol. Forecast. Soc. Change* **2025**, *210*, 123846. [CrossRef]
9. Walter, V.; Wagner, A. Probabilistic simulation of electricity price scenarios using conditional generative adversarial networks. *Energy AI* **2024**, *18*, 100422. [CrossRef]
10. Dumas, J.; Wehenkel, A.; Lanaspeze, D.; Cornélusse, B.; Sutera, A. A deep generative model for probabilistic energy forecasting in power systems: Normalizing flows. *Appl. Energy* **2022**, *305*, 117871. [CrossRef]
11. Lu, X.; Qiu, J.; Lei, G.; Zhu, J. Scenarios modelling for forecasting day-ahead electricity prices: Case studies in Australia. *Appl. Energy* **2022**, *308*, 118296. [CrossRef]

12. Lim, B.; Ö. Arık, S.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
13. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowsk, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]
14. Breiman, L. Statistical modeling: The two cultures. *Stat. Sci.* **2001**, *16*, 199–215. [CrossRef]
15. Rabiner, L.M. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [CrossRef]
16. Bender, E.M.; Gebru, T.; McMillan-Major, A.; Mitchell, M. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual, 3–10 March 2021. [CrossRef]
17. Schreiber, J. Pomegranate. 2023. Available online: https://pomegranate.readthedocs.io/en/latest/index.html (accessed on 29 April 2025).
18. Zhang J.; Wang J.; Wang R.; Hou G. Forecasting next-day electricity prices with Hidden Markov Models. In Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications, Taichung, Taiwan, 15–17 June 2010. [CrossRef]
19. Wu, O.; Liu, T.; Huang, B.; Forbes, F. Predicting Electricity Pool Prices Using Hidden Markov Models. *IFAC-PapersOnLine* **2015**, *48*, 343–348. [CrossRef]
20. Dimoulkas, I.; Amelin, M.; Hesamzadeh, M.R. Forecasting Balancing Market Prices Using Hidden Markov Models. In Proceedings of the13th International Conference on the European Energy Markets, Porto, Portugal, 6–9 June 2016. [CrossRef]
21. Apergis, N.; Gozgor, G.; Lau, C.K.M.; Wang, S. Decoding the Australian electricity market: New evidence from three-regime hidden semi-Markov model. *Energy Econ.* **2019**, *78*, 129–142. [CrossRef]
22. Duttilo, P.; Bertolini, M.; Kume, A.; Gattone, S.A. High volatility, high emissions? a hidden-Markov model approach. In *Methodological and Applied Statistics and Demography III. SIS 2024*; Pollice, A., Mariani, P., Eds.; Italian Statistical Society Series on Advances in Statistics; Springer: Cham, Switzerland, 2019. [CrossRef]
23. Garcia-Martos, C.; Conejo, A.J. Price forecasting techniques in power systems. *Wiley Encycl. Electr. Electron. Eng.* **2013**, 1–23. [CrossRef]
24. Weron, R.; Misiorek, A. Short-term electricity price forecasting with time series models: A review and evaluation. In *Complex Electricity Markets*; Mielczarski, W., Ed.; IEPŁ and SEP: Łódź, Poland, 2006.
25. Lucheroni, C. A hybrid SETARX model for spikes in tight electricity markets. *Oper. Res. Decis.* **2012**, *1*, 13–49.
26. Mari, C.; De Sanctis, A. Modelling spikes in electricity markets using excitable dynamics. *Phys. A Stat. Mech. Its Appl.* **2007**, *384*, 457–467.
27. Huisman, R.; Huurman, C.; Mahieu, R. Hourly electricity prices in day-ahead markets. *Energy Econ.* **2007**, *29*, 240–248. [CrossRef]
28. Raviv, E.; Bouwman, K.E.; van Dijk, D. Forecasting day-ahead electricity prices: Utilizing hourly prices. *Energy Econ.* **2015**, *50*, 227–239. [CrossRef]
29. Panagiotelis, A.; Smith, M. Bayesian forecasting of intraday electricity prices using multivariate skew-elliptical distributions. *Int. J. Forecast.* **2008**, *24*, 710–727. [CrossRef]
30. Ergemen, Y.E.; Haldrup, N.; Rodríguez-Caballero, C.V. Common long-range dependence in a panel of hourly Nord Pool electricity prices and loads. *Energy Econ.* **2016**, *60*, 79–96. [CrossRef]
31. Janczura, J.; Weron, R. An empirical comparison of alternate regime-switching models for electricity spot prices. *Energy Econ.* **2010**, *32*, 1059–1073. [CrossRef]
32. Jordan, M.I. An Introduction to Probabilistic Graphical Models. Available online: https://eecs.berkeley.edu/book/phd/prelims/ (accessed on 5 August 2017).
33. Ng, A.Y.; Jordan, M.I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14 (NIPS 2001)*; Dietterich, T.G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Boston, MA, USA, 2021; pp. 841–848.
34. Mahalanobis, P.C. On the generalised distance in statistics. *Proc. Natl. Inst. Sci. India* **1936**, *2*, 49–55.
35. McLachlan, G.; Peel, D. *Finite Mixture Models*; Wiley: Hoboken, NJ, USA, 2000.
36. Alexander, C.; Lazar, E. Normal mixture GARCH(1,1): Applications to exchange rate modelling. *J. Appl. Econ.* **2006**, *21*, 307–336. [CrossRef]
37. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–38. [CrossRef]
38. Everitt, B.S.; Landau, S.; Leese, M. *Cluster Analysis*; Oxford University Press: Oxford, UK, 2001.
39. Lucheroni, C. Resonating models for the electric power market. *Phys. Rev. E* **2007**, *76*, 56116. [CrossRef]
40. Lucheroni, C. Spikes, antispikes and thresholds in electricity logprices. In Proceedings of the 10th International Conference on the European Energy Market (EEM), Stockholm, Sweden, 27–31 May 2013; pp. 1–5. [CrossRef]
41. Calinski, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat.* **1974**, *3*, 1–27.
42. Barber, D. *Bayesian Reasoning and Machine Learning*; Cambridge University Press: Cambridge, UK, 2012.

43. Levin, D.A.; Peres, Y.; Wilmer, E.M. *Markov Chains and Mixing Times*; American Mathematical Society: Providence, RI, USA, 2008.
44. Mari, C.; Baldassari, C. Unsupervised expectation-maximization algorithm initialization for mixture models: A complex network-driven approach for modeling financial time series. *Inf. Sci.* **2022**, *617*, 1–16. [CrossRef]
45. Jebara, T. *Machine Learning–Discriminative and Generative*; Springer: Berlin/Heidelberg, Germany, 2010.
46. Bishop, C.M.; Lasserre, J. Generative or discriminative? Getting the best of both worlds. In *Bayesian Statistics 8*; Bernardo, J.M., Bayarri, M.J., Berger, J.O., Dawid, A.P., Heckerman, D., Smith, A.F.M., West, M., Eds.; Oxford University Press: Oxford, UK, 2007.