David Barber and A. Taylan Cemgil

Graphical Models for Time Series

Gaining insight into their computational implementation

Time-series analysis is central to many problems in signal processing, including acoustics, image processing, vision, tracking, information retrieval, and finance, to name a few [1], [2]. Because of the wide base of application areas, having a common description of the models is useful in transferring ideas between the various communities. Graphical models provide a compact way to represent such models and thereby rapidly transfer ideas. We will discuss briefly how classical time-series models such as Kalman filters and hidden Markov models (HMMs) can be represented as graphical models and critically how this representation differs from other common graphical representations such as state-transition and block diagrams. We will use this framework to show how one may easily envisage novel models and gain insight into their computational implementation.

TIME-SERIES AND GRAPHICAL MODELS

Classically, time-series analysis falls into two camps in which the central assumption is that the process generating the time series is continuous or alternatively discrete. In the continuous case, classical textbook methods such as Kalman filters depend heavily on linear dynamical systems (LDSs) for which the underlying theory is well understood [3]. In the discrete case, the well-known HMM has enjoyed considerable success [4]. However, recent developments in engineering, statistics, and machine learning consider underlying processes that can be both discrete and continuous. Such models are natural in many applications in control, tracking, and signal processing where one may wish to discover step-changes in an underlying continuous dynamical process, such as might occur for example with a fault. Working with these increasingly sophisticated models requires specialized treatments and often approximations [5]. There are, however, important special cases, such as the reset models, where the computational complexity of inference is relatively modest [6]. Here we take advantage of the graphical models framework to describe some of the basic time-series models, their extensions, and applications in signal processing.

DEVELOPING A GRAPHICAL REPRESENTATION

A probabilistic model of a time series $y_{1:T} = \{y_1, \ldots, y_T\}$ is a specification of a joint distribution $p(y_{1:T})$. In time series, it is natural to consider models consistent with the causal nature of time. To achieve this, we may use Bayes' rule of the probability of A conditioned on knowing B, $p(A|B) = p(A, B)/p(B)$, and write

Digital Object Identifier 10.1109/MSP.2010.938028

1053-5888/10/$26.00©2010IEEE

$$p(y_{1:T}) = p(y_T|y_{1:T-1})p(y_{1:T-1}). \quad (1)$$

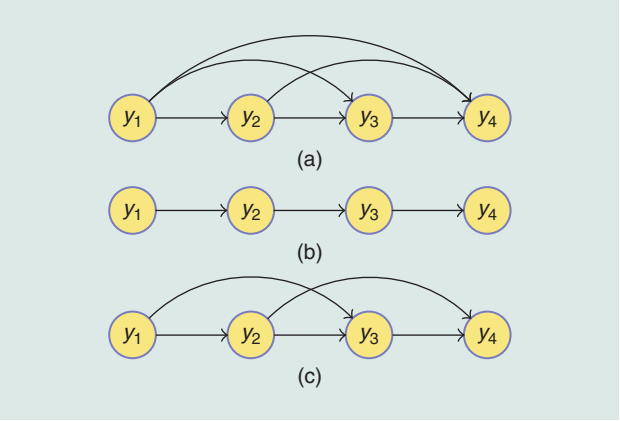
By recursively applying Bayes' rule to the last factor, any distribution can be written in a causal form

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t|y_{1:t-1}) \quad (2)$$

with the convention that $y_{a:b} = y_a$ if $b \leq a$. This has a natural causal interpretation in which for each factor the present depends only on the past. The distribution can be represented by a belief network in a distribution over N variables

$$p(y_1, \ldots, y_N) = \prod_{i=1}^{N} p(y_i|\mathrm{pa}(y_i)), \quad (3)$$

where $\mathrm{pa}(y_i)$ denotes the set of parental variables for variable y_i. We depict a belief network using a graph in which a node represents a variable y_i and the variables that point to y_i are the parents of this variable. Each node in the belief network then corresponds to a factor in the joint distribution over all variables; see Figure 1. By Bayes' recursive construction, the graph must be acyclic. The most general form of belief network is therefore the cascade graph in which the parents of a variable are all the previous variables in the ordering. Any valid belief network can be obtained by removing edges in the cascade graph, with each removal corresponding to a conditional independence assumption. The first-order Markov model can be represented in this form in which i indexes time and $\mathrm{pa}(y_i) = y_{i-1}$. A second-order Markov model has $\mathrm{pa}(y_i) = \{y_{i-1}, y_{i-2}\}$. As an example, the classical Lth-order



[FIG1] Belief network representations of time-series models: (a) Cascade graph. (b) First-order Markov model $p(y_4|y_3)p(y_3|y_2)p(y_2|y_1)p(y_1)$. (c) Second-order Markov model $p(y_4|y_3, y_2)p(y_3|y_2, y_1)p(y_2|y_1)p(y_1)$.

scalar auto-regressive (AR) model $y_t = \sum_{l=1}^{L} a_l y_{t-l} + \epsilon_t$ for coefficients a_l, $l = 1, \ldots, L$ and Gaussian noise $\epsilon_t \sim \mathcal{N}(\epsilon_t|0, \sigma^2)$ corresponds to the transition

$$p(y_t|y_{t-L:t-1}) = \mathcal{N}\left(y_t \,\middle|\, \sum_{l=1}^{L} a_l y_{t-l}, \sigma^2\right)$$

with a belief network representation in which the parent set of each variable contains the previous L observations. When the parameters of the model are also unknown, they can be incorporated into the graphical description as well; see "Parameter Learning." Graphs have a long history in the description of and concentrate here on inference, assuming that parameters are known.

PARAMETER LEARNING

So far we've assumed that any parameters θ of a model $p(y_{1:T}|\theta)$ are known. From a Bayesian perspective, a parameter is treated as another random variable,

$$p(y_{1:T}, \theta) = p(y_{1:T}|\theta)p(\theta).$$

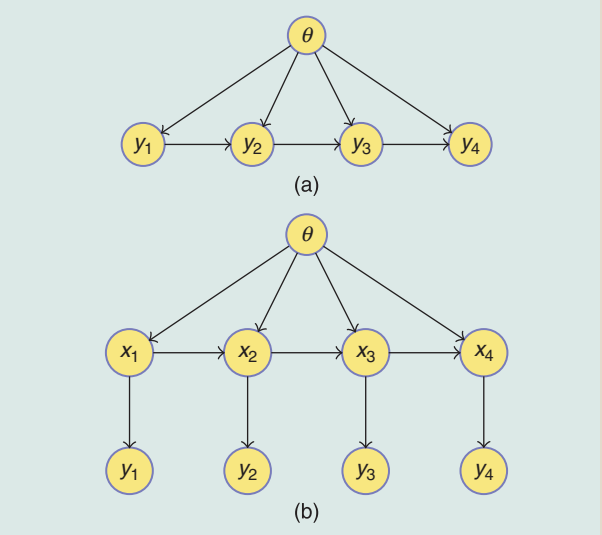
All questions relating to parameter estimation are computed from the parameter posterior density

$$p(\theta|y_{1:T}) = \frac{p(y_{1:T}|\theta)p(\theta)}{p(y_{1:T})}.$$

For example, for a first-order Markov model we have

$$p(y_{1:T}, \theta) = p(\theta)\prod_t p(y_t|y_{t-1}, \theta),$$

whose belief network is given in Figure S1(a). Calculating the posterior probability of a parameter $p(\theta|y_{1:T})$ then becomes a problem in marginal inference in a graphical model. For a continuum of parameters this can present difficulties. Finding, for example, the most likely single set of parameters $\arg\max_\theta p(\theta|y_{1:T})$ may not be a computationally straightforward problem, for which one may resort to numerical approximations such as Monte Carlo or deterministic techniques. While parameter estimation therefore fits neatly within the graphical models framework, we leave the numerical details of how this can be achieved to one side



[FIGS1] Dealing with parameters. (a) First-order Markov model with a parameter θ tied across the separate transition. (b) State-space model with a tied parameter on the latent transitions.

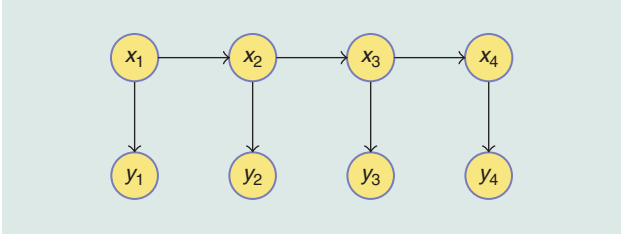
time-series models and it's important to stress the difference between a probabilistic graphical model and alternative graph representations such as state transition diagrams or block diagrams that use an entirely different set of semantic rules.

PROBABILISTIC GRAPHICAL MODELS ARE COMPACT DEPICTIONS OF INDEPENDENCE AND FACTORIZATION ASSUMPTIONS OF A PROBABILITY DENSITY.

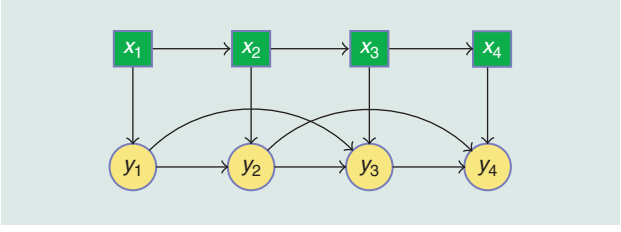
More generally, probabilistic graphical models are compact depictions of independence and factorization assumptions of a probability density. Besides the directed acyclic graphs, there exist also other formalisms. Two well-known formalisms are undirected models [7] and factor graphs [8]. To keep this survey self-contained, we focus only on the case of directed graphical models.

LATENT MARKOV MODELS

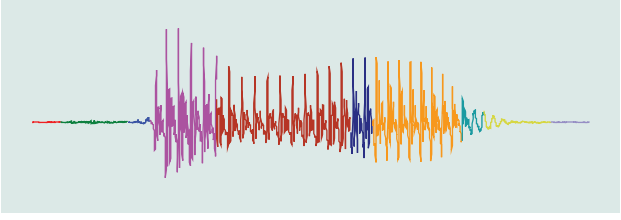
A more general framework for modeling time-series data uses a latent, unobserved variable x_t, from which the observations y_t are generated [9]. For example, in tracking, x_t might represent the position of an object that is assumed to move according a transition dynamics $p(x_t|x_{t-1})$. However, we cannot directly observe x_t, but some noisy function of it $p(y_t|x_t)$—for example a noisy radar reading y_t of the approximate distance to the object. We would like to use the observations $y_1, \ldots, y_t$ to track the likely position x_t of the object. Due to their development in different research communities, latent Markov models are variously called state-space models or HMMs. We use the term HMM to refer to a latent Markov model with discrete latent states. Both the classical Kalman filter and the HMMs have the same belief network representation, (see Figure 2), differing only in the domain of the variables and the specifics of the transition and observation model.



[FIG2] A first-order state-space model with "hidden" variables. For discrete hidden variables $x_t \in \{1, \ldots, H\}$, $t = 1{:}T$ the model is termed an HMM.



[FIG3] A switching (second order) AR model. Here the x_t indicates which of a set of available AR models is active at time t. In terms of inference, conditioned on $y_{1:T}$ (this is an HMM).



[FIG4] A spoken digit of the word "four" modeled by a SAR model. The SAR model was trained on many example sequences using $S = 10$ states with a left-to-right transition matrix. Given the particular audio sequence shown, the most likely set of states $x_{1:T}$ are computed. The colors indicate the states used at each time. The states found correspond to basic sound component models that when used in sequence generate realistic sounding waveforms.

DISCRETE LATENT STATE MARKOV MODELS

HMMs are models in which the latent variables x_t are discrete [4]. The observations y_t can be discrete or continuous. Since the latent x_t are discrete, HMMs are able to model discrete changes in the underlying state. To emphasize that the x_t are discrete, graphically, we use a square node. For example, in a switching AR (SAR) model, a set of S different AR models is available, and $x_t \in \{1, \ldots, S\}$ may be used to indicate which of the AR models is to be used at time t; see Figure 3. In Figure 4, a segment of a speech signal is shown; each of the ten available AR models is responsible for modeling the dynamics of a basic subunit of speech [10], [11]. The interest is to determine when each subunit is most likely to be active. This corresponds to the computation of the most-likely path $x_{1:T}$ given the observed signal $p(x_{1:T}|y_{1:T})$. Typically we use a lower-case version of a variable to denote instantiation. While models such as the SAR model contain both discrete and continuous variables, fundamentally, the underlying latent process is discrete.

CONTINUOUS STATE LATENT MARKOV MODELS

Dealing with continuous variable distributions is generally awkward and the set of models that are analytically tractable is limited. Within this tractable class, the LDS plays a special role, being essentially the continuous analog of the discrete-state HMM. An LDS has the following form [2], [3]:

$$x_t = Ax_{t-1} + \epsilon_t, \qquad y_t = Cx_t + \nu_t,$$

where the noise terms ϵ_t and ν_t are Gaussian distributed. This is more commonly referred to as a Kalman filter in the signal processing literature. The traditional focus is the use of this linear system to compute quantities of interest, in particular the expected mean of x_t given past observations. This terminology unfortunately confuses the distinction between a model

and the use of it to infer a quantity of interest. As such most students are familiar with the Kalman filter from an algorithmic perspective, unaware that the algorithm is an instance of the generic filtering algorithm for all graphs consistent with the belief network Figure 2. As a probabilistic model, the LDS corresponds to

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t|Ax_{t-1}, Q), \quad p(y_t|x_t) = \mathcal{N}(y_t|Cx_t, R).$$

The model is completed by choosing a suitable prior: $p(x_1) = \mathcal{N}(x_1|\mu_1, P_1)$. Furthermore, as we describe below, inference in this model is computationally straightforward. The graphical models perspective emphasizes the application of the independence assumptions of the model to derive generic recursions; these recursions may be then implemented in specific numerical instances of distributions consistent with the graphical model representation.

INFERENCE IN LATENT MARKOV MODELS

Latent Markov models have widespread application in a variety of tracking domains, for which one often wishes to infer the distribution of the latent state x_t based on noisy observations. These will be derived for general latent variables x_t using the notation $\int dX$ that either integrates or sums over the domain of X. The important conclusion we shall reach is that the same procedure applies in all models consistent with the belief network Figure 2, irrespective of the numerical form of the transition and observation distributions. As we also discuss, while the general procedure produces exact results, it can only be numerically implemented in a restricted class of transition and observation distributions, the two most common being i) discrete latent variables (HMM) and ii) linear Gaussian transition and observations (LDS), giving rise to the classical Kalman filter. We sketch below the generic filtering and smoothing recursions for the class of all latent Markov models.

FILTERING: COMPUTING $p(x_t|y_{1:t})$

Filtering is the estimation of the current state given the observations so far. It is useful to first compute the joint marginal $p(x_t, y_{1:t})$ since the likelihood of the sequence can be obtained from this expression. A recursion for $p(x_t, y_{1:t})$ is obtained by considering the conditional independence assumptions of the model

$$p(x_t, y_{1:t}) = \int dx_{t-1} p(y_t|x_t) p(x_t|x_{t-1}) p(x_{t-1}, y_{1:t-1}). \quad (4)$$

Hence, if we define $\alpha(x_t) = p(x_t, y_{1:t})$ with $\alpha(x_1) = p(y_1|x_1)p(x_1)$ we have the so-called α-recursion

$$\alpha(x_t) = \underbrace{p(y_t|x_t)}_{\text{corrector}} \underbrace{\int dx_{t-1} p(x_t|x_{t-1})\alpha(x_{t-1})}_{\text{predictor}}, \; t > 1. \quad (5)$$

This recursion has the interpretation that the filtered distribution $\alpha(x_{t-1})$ is propagated forwards by the dynamics for one time step to reveal a new "prior" distribution at time t. Normalization gives the filtered posterior $p(x_t|y_{1:t}) \propto \alpha(x_t)$.

PARALLEL (FORWARD-BACKWARD) SMOOTHER $p(x_t|y_{1:t})$

In parallel smoothing, one separates the smoothed posterior into contributions from the past and future

$$p(x_t, y_{1:T}) = p(x_t, y_{1:t}, y_{t+1:T}) = \underbrace{p(x_t, y_{1:t})}_{\text{past}} \underbrace{p(y_{t+1:T}|x_t, y_{1:t})}_{\text{future}}$$
$$= \alpha(x_t)\beta(x_t). \quad (6)$$

The term $\alpha(x_t)$ is obtained from the "forward" α recursion (5). The term $\beta(x_t)$ may be obtained using a "backward" β recursion with $\beta(x_T) = 1$

$$\beta(x_{t-1}) = \int dx_t p(y_t|x_t) p(x_t|x_{t-1}) \beta(x_t), \qquad 2 \le t \le T.$$

SEQUENTIAL (CORRECTION) SMOOTHER $p(x_t|y_{1:T})$

The parallel smoothing method given above is perhaps best known in the HMM literature [4]. Particularly in the case of continuous variables, however, some care is required with its numerical implementation [12]. In practice, it is often more suitable to use a sequential method that is based on the fact that conditioning on the present makes the future redundant [13]

$$p(x_t|y_{1:T}) = \int dx_{t+1} p(x_t, x_{t+1}|y_{1:T})$$
$$= \int dx_{t+1} p(x_t|x_{t+1}, y_{1:t}, \cancel{y_{t+1:T}}) p(x_{t+1}|y_{1:T}). \quad (7)$$

This then gives a recursion for $\gamma(x_t) \equiv p(x_t|y_{1:T})$

$$\gamma(x_t) = \int dx_{t+1} p(x_t|x_{t+1}, y_{1:t}) \gamma(x_{t+1}) \quad (8)$$

with $\gamma(x_T) \propto \alpha(x_T)$. The term $p(x_t|x_{t+1}, y_{1:t})$ may be computed based on the filtered results $p(x_t|y_{1:t})$ using a dynamics reversal step

$$p(x_t|x_{t+1}, y_{1:t}) \propto p(x_{t+1}, x_t|y_{1:t}) = p(x_{t+1}|x_t) p(x_t|y_{1:t}), \quad (9)$$

where the proportionality constant is found by normalization. The procedure is sequential since we need to first complete the α recursions, after which the γ recursion may begin. This is also called a correction smoother, since it takes the filtered results and corrects them into smoothed results. A significant advantage of this sequential approach is that the recursion deals directly with densities, unlike the parallel approach that forms a recursion for a quantity that is itself not a density in x_t. This has important benefits for models (such as the SLDS described below) for which exact smoothing is not computationally feasible.

INFERENCE IN LINEAR DYNAMICAL SYSTEMS

Filtering and smoothing for the LDS follows the general approach, with the most common smoothing approach being the

sequential method. Since all updates for the LDS are linear-Gaussian, the filtered and smoothed distributions are Gaussians. The α and γ recursions can thus be represented by updates to the mean and covariance of the distributions. Working out these updates is a standard exercise in multivariate Gaussian integration resulting in the well-known Kalman filtering and smoothing recursions [2].

MOST STUDENTS ARE FAMILIAR WITH THE KALMAN FILTER, UNAWARE THAT THE ALGORITHM IS AN INSTANCE OF THE GENERIC FILTERING ALGORITHM.

THE SWITCHING LINEAR DYNAMIC SYSTEM

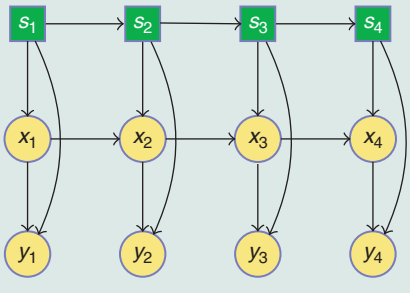
The HMM and LDS are two classical signal processing models. A more complex model is the switching LDS (SLDS) that marries the HMM and LDS by breaking the time series into segments, each modeled by a potentially different LDS; see Figure 5. Such models can handle situations in which the underlying linear model jumps from one parameter setting to another. Thus, the latent process contains both discrete and continuous variables. The SLDS is an attractive model and used in many disciplines, from econometrics to machine learning [14]–[17]. At each time t, a switch variable $s_t \in 1, \ldots, S$ selects a single LDS from the available set. The dynamics of s_t itself is Markovian, with transition $p(s_t \mid s_{t-1})$. The probabilistic model defines a joint distribution

$$p(y_{1:T}, x_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(y_t \mid x_t, s_t) p(x_t \mid x_{t-1}, s_t) p(s_t \mid s_{t-1})$$

with

$$p(y_t \mid x_t, s_t) = \mathcal{N}(y_t \mid C(s_t)x_t, R(s_t)),$$
$$p(x_t \mid x_{t-1}, s_t) = \mathcal{N}(x_t \mid A(s_t)x_{t-1}, Q(s_t)).$$

At time $t = 1$, $p(s_1 \mid x_1, s_1)$ denotes the prior $p(s_1)$, and $p(x_1 \mid x_1, s_1)$ denotes $p(x_1 \mid s_1)$. Due to its popularity in many different fields the SLDS has many different names; it is also called a jump Markov model/process, switching Kalman filter, switching linear Gaussian state-space model, conditional linear Gaussian model. Given its importance, we will spend some time considering the particular issues in dealing with the SLDS.



[FIG5] The independence structure of the SLDS. Square nodes s_t denote discrete switch variables; x_t are continuous latent/hidden variables and y_t continuous observed/visible variables. The discrete state s_t determines which LDS from a finite set of LDSs is operational at time t.

EXACT INFERENCE IS COMPUTATIONALLY INTRACTABLE

In terms of the cluster variables $z_{1:T}$, with $z_t \equiv (s_t, x_t)$ and visible variables $y_{1:T}$, the belief network of the SLDS is a latent Markov model, for which the exact filtering and smoothing recursions are given in the section "Inference in Latent Markov Models." One might therefore envisage no difficulty in carrying out inference. However, both exact filtered and smoothed inference in the SLDS is intractable, scaling exponentially with time. As an informal explanation, consider filtered posterior inference, for which the forward pass is, by analogy with (5),

$$\alpha(s_t, x_t) = p(y_t \mid x_t, s_t) \times \sum_{s_{t-1}} \int_{x_{t-1}} p(s_t, x_t \mid s_{t-1}, x_{t-1}, y_t) \alpha(s_{t-1}, x_{t-1}). \quad (10)$$

At time step 1, $\alpha(s_1, x_1) \propto p(x_1 \mid s_1, y_1) p(s_1 \mid y_1)$ is an indexed set of Gaussians. At time step 2, due to the summation over the states s_1, $\alpha(s_2, x_2)$ will be an indexed set of S Gaussians; similarly at time step 3, it will be S^2 and, in general, gives rise to exponentially many Gaussians, S^{t-1}, at time t. The origin of the intractability of the SLDS therefore differs from structural intractability, resulting from the inability to form a singly connected structure by the clustering of a small number of variables [18]. Since filtering and smoothing in the SLDS require some form of approximation, we therefore have to choose which approximation strategy to follow. Approximate inference in the SLDS has a large associated literature describing available techniques that range from Monte Carlo methods to deterministic variational techniques [19], [20], [15]. One of the most robust techniques is the Gaussian sum approximation and, rather than giving a survey on the available techniques, we outline the rationale for this method below.

GAUSSIAN SUM FILTERING

A popular approximate SLDS filtering scheme is to keep in check the exponential explosion in the number of Gaussian components by projecting each filtered update to a limited number of components. A graphical depiction is given in Figure 6. At each stage, a single Gaussian component is propagated forwards by the S separate LDS dynamics, each giving rise to a separate filtered distribution according to LDS filtering. Subsequently, this S^2 Gaussian mixture is collapsed back to an S component Gaussian mixture, preventing the exponential explosion in mixture components. Such Gaussian sum

FILTERING IN THE RESET MODEL

Consider the filtering recursion for the two cases

$$\alpha(x_t, c_t = 0) = \int_{x_{t-1}} \sum_{c_{t-1}} p^0(y_t|x_t) p^0(x_t|x_{t-1}) p(c_t = 0|c_{t-1}) \times \alpha(x_{t-1}, c_{t-1}) \quad (11)$$

$$\alpha(x_t, c_t = 1) = \int_{x_{t-1}} \sum_{c_{t-1}} p^1(y_t|x_t) p^1(x_t) p(c_t = 1|c_{t-1}) \times \alpha(x_{t-1}, c_{t-1}) = p^1(y_t|x_t) p^1(x_t) \sum_{c_{t-1}} p(c_t = 1|c_{t-1}) \alpha(c_{t-1}). \quad (12)$$

Equation (12) shows that $\alpha(x_t, c_t = 1)$ contains only a single component proportional to $p^1(y_t|x_t)p^1(x_t)$. If we use this information in (11) we have

$$\alpha(x_t, c_t = 0) = \int_{x_{t-1}} p^0(y_t|x_t) p^0(x_t|x_{t-1}) p(c_t = 0|c_{t-1} = 0) \times \alpha(x_{t-1}, c_{t-1} = 0) + \int_{x_{t-1}} p^0(y_t|x_t) p^0(x_t|x_{t-1}) p(c_t = 0|c_{t-1} = 1) \times \alpha(x_{t-1}, c_{t-1} = 1). \quad (13)$$

If we assume that $\alpha(x_{t-1}, c_{t-1} = 0)$ is a mixture distribution with K components, then $\alpha(x_t, c_t = 0)$ will contain $K + 1$ components. In general, therefore, $\alpha(x_t, c_t = 0)$ will contain t components and $\alpha(x_t, c_t = 1)$ a single component. Since the number of components grows only linearly with time, the computational effort to perform exact filtering scales as $O(LT^2)$, compared with $O(LT2^T)$ in the general two-state switching case (L is the complexity of a filtered LDS update).

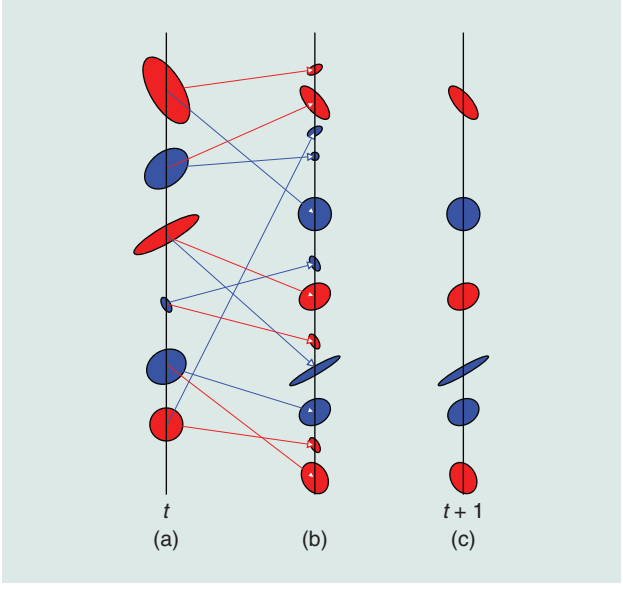
filtering approximations were developed early in the literature, in particular in the work by Alspach and Sorenson [21]. The method is a form of the general approximation class called assumed density filtering in which an approximate mixture density is projected back to a chosen approximation family at each update [22]. The complexity of the resulting approximate forward pass is $O(ISTL)$, where I is the number of mixture components of the collapsed distribution and L is the cost of performing a filtered update for the LDS. The recursion is initialized with $p(x_1, s_1 | y_1) \propto p(y_1 | x_1, s_1)\, p(x_1 | s_1)\, p(s_1)$, where $p(x_1|s_1)$ and $p(s_1)$ are given prior distributions.

GAUSSIAN SUM SMOOTHING

The γ recursion (8) suggests a convenient Gaussian sum smoothing approximation. Since the γ recursion can be interpreted as a backwards dynamics, one may propagate each component in a Gaussian sum smoothed approximation backwards according to each of the S dynamical systems. This results in an S^2 component Gaussian mixture distribution which, analogously to filtering, may be collapsed back to a smaller number of components to prevent the exponential explosion in components. A popular standard method to achieve this is called generalized pseudo-Bayes [15]. An alternative approach, which makes less severe approximation assumptions, is expectation-correction [23], which we use throughout our examples.

NOISY SIGNAL RECONSTRUCTION

Continuous observation models such as the SAR model have been successfully applied in many areas of signal processing, including audio signal processing [10]. Since such models are essentially HMMs, however, they are not well suited to signal reconstruction in which we aim to infer a clean continuous signal from a noisy observation. A natural extension is to include additional graphical links from the AR output y_t to form a noisy observation $\tilde{y}_t$. For example, additive zero mean Gaussian noise with variance σ_y^2 can be expressed as $p(\tilde{y}_t | y_t) = \mathcal{N}(\tilde{y}_t | y_t, \sigma_y^2)$. Given the noisy observation sequence $\tilde{y}_{1:T}$, our interest is then to reconstruct a clean signal $y_{1:T}$, based on the assumption that the clean signal is itself expressed as a SAR model; see Figure 7. This model is a form of SLDS and may be used to form noise-robust speech recognition systems [11]; see Figure 8.



[FIG6] Gaussian sum filtering. (a) Depicts the previous Gaussian mixture approximation $\alpha(x_t, s_t)$ for two states $S = 2$ (red and blue) and $I = 3$ mixture components. The area of each ellipse corresponds here to the relative weight of each component rather than the variance. There are $S = 2$ different linear systems that take each of the components of the mixture into a new filtered state, the color of the arrow indicating which dynamic system is used. After one time step, each mixture component branches into a further S components so that the joint approximation $\alpha(x_{t+1}, s_{t+1})$ contains (b) S^2I components. To keep the representation computationally tractable, the mixture of Gaussians for each state s_{t+1} is collapsed back to I components. This means that each colored state needs to be approximated by a smaller I component mixture of Gaussians. There are many ways to achieve this. A naïve but computationally efficient approach is to simply ignore the lowest weight components, as depicted in (c).

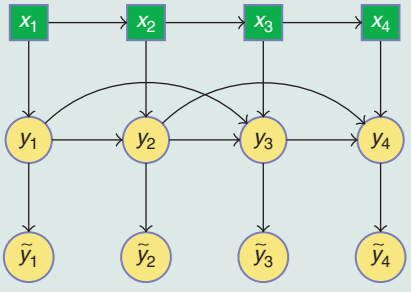
TRAFFIC FLOW

As an illustration of modeling and inference with a SLDS consider a traffic network; see Figure 9. There are four junctions a, b, c, d and traffic flows along the roads in the direction indicated. Traffic flows into junction a and then goes via different routes to d. Flow out of a junction must match the flow in to a junction (up to noise). There are traffic light switches at junctions a and b that, depending on their state, route traffic differently along the roads. Then using ϕ to denote flow, we model the flows using the switching linear system

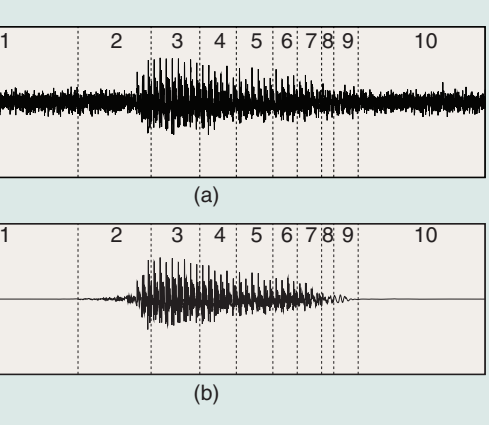
$$\left.\begin{matrix}\phi_a(t)\\ \phi_{a\to d}(t)\\ \phi_{a\to b}(t)\\ \phi_{b\to d}(t)\\ \phi_{b\to c}(t)\\ \phi_{c\to d}(t)\end{matrix}\right\} = \begin{cases}\phi_a(t-1)\\ \phi_a(t-1)(0.75\times[s_a(t)=1]+1\times[s_a(t)=2])\\ \phi_a(t-1)(0.25\times[s_a(t)=1]+1\times[s_a(t)=3])\\ \phi_{a\to b}(t-1)(0.5\times[s_b(t)=1])\\ \phi_{a\to b}(t-1)(0.5\times[s_b(t)=1]+1\times[s_b(t)=2])\\ \phi_{b\to c}(t-1)\end{cases}.$$

RESET MODELS ARE SPECIAL SWITCHING MODELS IN WHICH THE SWITCH CAN RESET THE LATENT STATE, ISOLATING THE PRESENT FROM THE PAST.

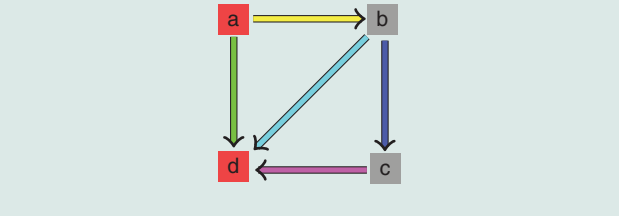
In the above, $[A]=1$ if A is true and is zero otherwise. By identifying the flows at time t with a six-dimensional vector hidden variable x_t, we can write the above flow equations as an SLDS in x_t for a set of suitably defined matrices $A(s)$ where the switch variable $s=s_a\otimes s_b$, takes $3\times 2=6$ states. The switch variables follow a simple Markov transition $p(s_t|s_{t-1})$ that biases the switches to remain in the same state in preference to jumping to another state. We additionally include small noise terms to model cars parking or deparking during a single time frame. The noise is larger at the inflow point a to model that the total volume of traffic entering the system can vary. Noisy measurements of the flow into the network are taken at a and d. Given an observed sequence (a_t, d_t), $t=1,\ldots,100$ (see Figure 10), the task is to infer the filtered and smoothed traffic flows throughout the network. A naïve approximation based on discretizing each continuous flow into 20 bins would contain $2\times 3\times 20^6$ or



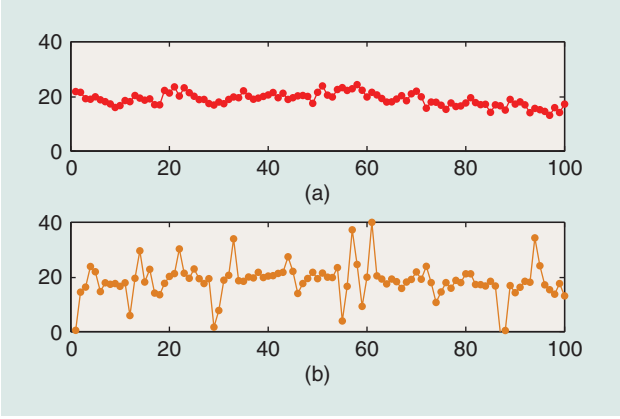
[FIG7] A latent switching (second order) AR model. Here the x_t indicates which of a set of available AR models is active at time t. The "clean" AR signal y_t, which is not observed, is corrupted by additive noise to form the noisy observations $\tilde{y}_t$. In terms of inference, conditioned on $\tilde{y}_{1:T}$, this can be expressed as a SLDS.



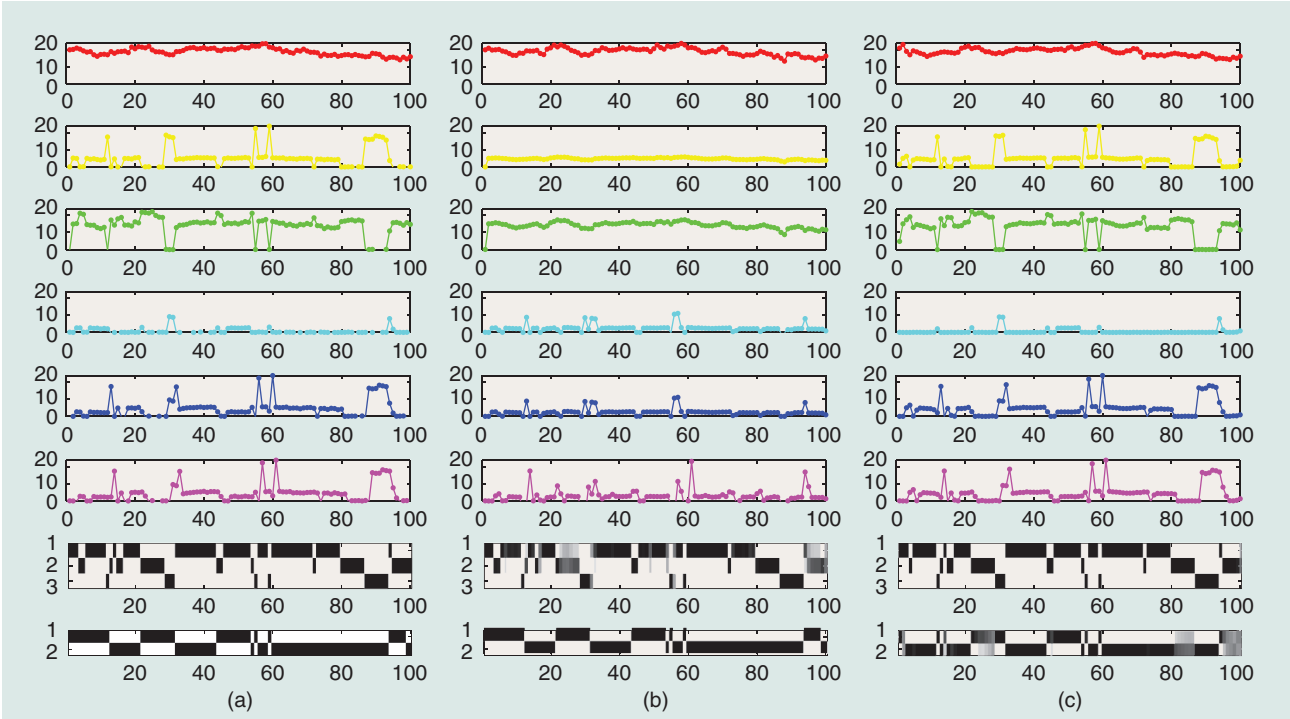
[FIG8] Signal reconstruction using a latent left-right SAR model; see Figure 7. (a) Noisy signal $\tilde{y}_{1:T}$. (b) Reconstructed clean signal $y_{1:T}$. The dashed lines and the numbers show the most-likely state segmentation $s^*_{1:T}$.



[FIG9] A representation of the traffic flow between junctions at a, b, c, d, with traffic lights at a and b. If $s_a=1$, $a\to d$ and $a\to b$ carry 0.75 and 0.25 of the flow out of a, respectively. If $s_a=2$, all the flow from a goes through $a\to d$; for $s_a=3$, all the flow goes through $a\to b$. For $s_b=1$, the flow out of b is split equally between $b\to d$ and $b\to c$. For $s_b=2$, all flow out of b goes along $b\to c$.



[FIG10] Time evolution of the traffic flow measured at two points in the network. (a) Sensors measure the total flow into the network $\phi_a(t)$ and (b) the total flow out of the network, $\phi_d(t)=\phi_{a\to d}(t)+\phi_{b\to d}(t)+\phi_{c\to d}(t)$. The total inflow at a undergoes a random walk. Note that the flow measured at d can momentarily drop to zero if all traffic is routed through $a\to b\to c$ in two consecutive time steps.



[FIG11] Given the observations from Figure 10, we infer the flows and switch states of all the latent variables. (a) The correct latent flows through time along with the switch variable state used to generate the data. The colors correspond to the colored edges and nodes in Figure 9. (b) Filtered flows based on a $I = 2$ Gaussian sum forward pass approximation. The filtered traffic light states of s_a and s_b are plotted below. (c) Smoothed flows and corresponding smoothed traffic light states using a two-component Gaussian sum smoothing approximation.

384 million states. Even for such a modest-size problem, an approximation based on a simple discretization is therefore impractical. As a practical alternative, filtering and smoothing for this SLDS can be carried out using a Gaussian sum approximation; see Figure 11.

RESET MODELS

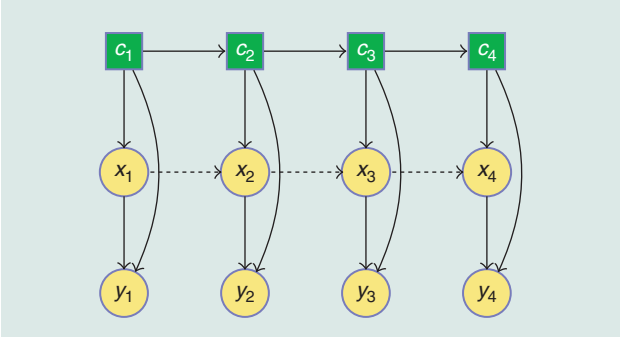
While switching models such as the SLDS are powerful, they are computationally difficult to implement. As such, it is interesting to consider special cases for which inference is computationally simpler. Reset models are special switching models in which the switch can reset the latent x_t, isolating the present from the past. These models are also known as change-point models [6], though the term is less precisely defined. We use the state $c_t = 1$, to denote a "change" that resets x_t, independent of the past, and $c_t = 0$ to denote that the standard dynamics continues. Then

$$p(x_t \,|\, x_{t-1}, c_t) = \begin{cases} p^0(x_t \,|\, x_{t-1}) & c_t = 0 \\ p^1(x_t) & c_t = 1 \end{cases}.$$

Similarly, we write

$$p(y_t \,|\, x_t, c_t) = \begin{cases} p^0(y_t \,|\, x_t) & c_t = 0 \\ p^1(y_t \,|\, x_t) & c_t = 1 \end{cases}.$$

The switch dynamics are first-order Markov with transition $p(c_t \,|\, c_{t-1})$. Under this model the dynamics follows a standard system $p^0(x_t \,|\, x_{t-1})$ until $c_t = 1$ when the continuous state is drawn from a "reset" distribution $p^1(x_t)$, independent of the past; see Figure 12. Such models are of interest when the time series is following a trend but suddenly changes and the past is forgotten. An SLDS with $S = 2$ states, one of which resets the continuous dynamics, is an example of such a reset model. Importantly, the complexity of filtered inference scales with $O(LT^2)$, compared to $O(LT2^T)$ in the general two-state switching case, as discussed in "Filtering in the Reset Model."



[FIG12] The independence structure of a reset model. Square nodes c_t denote the binary reset variables and s_t the state dynamics. The x_t are continuous variables, and y_t continuous observations. If the dynamics resets, the dependence of the continuous x_t on the past is cut.

POISSON RESET MODEL

Reset models are not limited to conditionally Gaussian cases. To illustrate this, we consider the following Poisson model. At each time t, we observe a count y_t that we assume is Poisson distributed with an unknown positive intensity x_t. The intensity is constant, but at certain unknown times t, it jumps to a new value. The indicator variable c_t denotes whether time t is such a change point or not. Mathematically, the model is

$$p(x_t|x_{t-1}, c_t) = [c_t = 0]\,\delta(x_t - x_{t-1}) + [c_t = 1]\,\mathcal{G}(x_t; \nu, b), \quad t \geq 2 \tag{14}$$

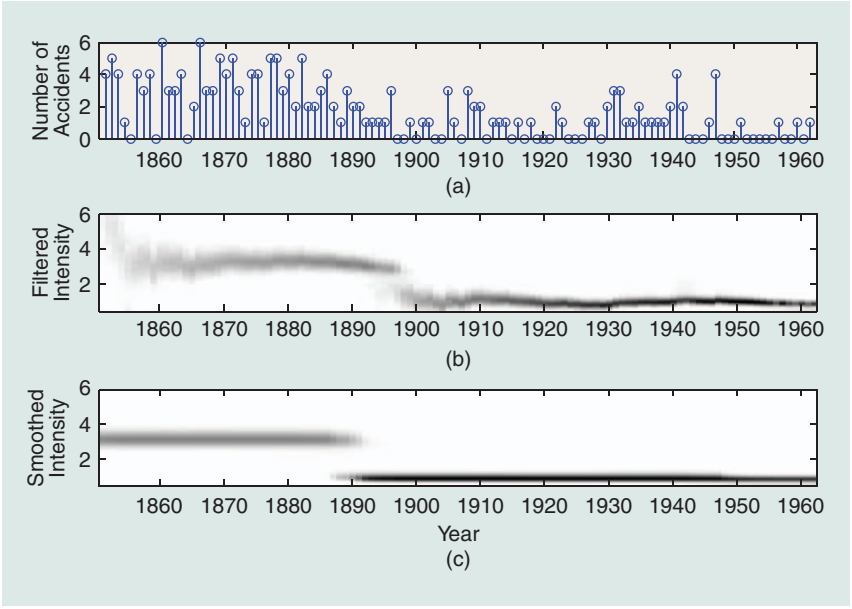
$$p(y_t|x_t) = \mathcal{PO}(y_t; x_t), \qquad p(c_t) = \mathcal{BE}(c_t; \pi) \tag{15}$$

with $p(x_1) = \mathcal{G}(x_1; a_1, b_1)$. The symbols $\mathcal{G}$, $\mathcal{BE}$, and $\mathcal{PO}$ denote the Gamma, Bernoulli, and the Poisson densities, respectively. Given observed counts $y_{1:T}$, the task is to find the posterior probability of a change and the associated intensity levels for each region between two consecutive change points. Plugging the above definitions in the generic updates (11) and (12), we see that $\alpha(x_t, c_t = 0)$ is a Gamma potential, and that $\alpha(x_t, c_t = 1)$ is a mixture of Gamma potentials, where a Gamma potential is defined as

$$\phi(x) = e^l \mathcal{G}(x; a, b) \tag{16}$$

via the triple (a, b, l). For the corrector update step, we need to calculate the product of a Poisson term with the observation model $p(y_t|x_t) = \mathcal{PO}(y_t; x_t)$. A useful property of the Poisson distribution is that, given the observation, the latent variable is Gamma distributed as $\mathcal{PO}(y; x) = \mathcal{G}(x; y + 1, 1)$. Hence, the update equation requires multiplication of two Gamma potentials. A nice property of the Gamma density is that the product of two Gamma densities is another Gamma potential. The α recursions for this reset model are therefore closed in the space of a mixture of Gamma potentials, with an additional Gamma potential in the mixture at each time step. A similar approach can be used to form the smoothing recursions.

We illustrate the algorithm on a coal mining disaster data set [24]. The data consists of the number of deadly coal-mining disasters in England per year over a time span of 112 years from 1851 to 1962. It is widely agreed in the statistical literature that a change in the intensity (the expected value of the number of disasters) occurs around the year 1890, after new health and safety regulations were introduced. In Figure 13, we show the marginals $p(x_t|y_{1:T})$ along with the filtering density. Note that we are not constraining the number of change points a priori and in principle allow any number. The smoothed densities indeed suggest a sharp decrease around $t = 1890$.



[FIG13] Estimation of change points on the coal mining disaster data set. (a) The number of deadly disasters each year. (b) Filtered estimate of the marginal intensity $p(x_t|y_{1:t})$. Here, darker color means higher probability. (c) Smoothed estimate $p(x_t|y_{1:T})$.

RESET HIDDEN MARKOV MODEL

The reset models described are useful in many applications but limited since only a single standard dynamics is available. An important extension is to consider a set of available dynamical models, indexed by $s_t \in \{1, \ldots, S\}$, with a reset that cuts dependency of the continuous variable on the past [17]

$$p(x_t|x_{t-1}, s_t, c_t) = \begin{cases} p^0(x_t|x_{t-1}, s_t) & c_t = 0 \\ p^1(x_t|s_t) & c_t = 1 \end{cases}. \tag{17}$$

The states s_t follow a Markovian dynamics $p(s_t|s_{t-1}, c_{t-1})$; see Figure 14. A reset occurs if the state s_t changes, otherwise, no reset occurs

$$p(c_t = 1|s_t, s_{t-1}) = [s_t \neq s_{t-1}]. \tag{18}$$

The computational complexity of filtering for this model is $O(LS^2T^2)$ that can be understood by analogy with the reset α recursions, (11), (12) on replacing x_t by (x_t, s_t). In the next section, we describe a signal processing application for this model.

DYNAMIC HARMONIC MODEL AND RESET MODELS

A key problem in music signal processing is music transcription; the identification of note events. The fundamental frequency, corresponding to the largest common divisor of mode frequencies, is perceived as the pitch or "note" in

music. For transcription, we estimate which note is played and when. The polyphonic case assumes that more than one note may be played at any time. Here we concentrate on the monophonic case of a single note $s_t \in \{1, \ldots, S\}$ being played at any time. For each note s_t, a musical instrument creates oscillations with modes at frequencies that are roughly related by ratios of integers. One can model this using an LDS that consists of a bank of "phasors"

$$A(s_t) = \mathrm{diag}(Z_1(s_t), \ldots, Z_\nu(s_t), \ldots, Z_W(s_t)), \tag{19}$$

where each phasor corresponds to a rotation matrix around a multiple ν of a fundamental frequency $\omega(s_t)$

$$Z_\nu(s_t) = e^{-\nu\gamma(s_t)} \begin{pmatrix} \cos(\nu\omega(s_t)) & \sin(\nu\omega(s_t)) \\ -\sin(\nu\omega(s_t)) & \cos(\nu\omega(s_t)) \end{pmatrix}. \tag{20}$$

Here, the index ν gives the number of the harmonic and sets both the damping parameter γ and the frequency ω. The dynamics of a single phasor is plotted in Figure 15. A note changes when $s_t \neq s_{t-1}$ at which point we assume the continuous dynamics is reset. Assuming a simple Markov model dynamics for the notes s_t, the model is

$$p(x_t \,|\, x_{t-1}, s_t, c_t) = [c_t = 0]\mathcal{N}(x_t | A(s_t)x_{t-1}, qI) + [c_t = 1]\,\mathcal{N}(x_t | 0, QI) \tag{21}$$

$$p(y_t | x_t) = \mathcal{N}(y_t | Cx_t, r), \tag{22}$$

where $C = [1\ 0\ 1\ \ldots 1\ 0]$ is a projection matrix that sums the first components of each phasor and the observation noise variance is r. The identity matrix is denoted by I and q and Q are transition variances with $Q \gg q$. This model is then a reset HMM model, for which the computations can be carried out efficiently.

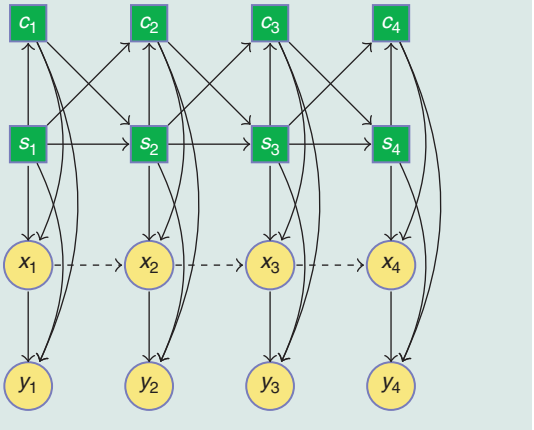
USING GRAPHICAL MODELS, IT IS EASY TO ENVISAGE NEW MODELS TAILORED FOR A PARTICULAR ENVIRONMENT.

An example from transcribing a real guitar recording is presented in Figure 16. An interesting, yet more complex problem is to deal with polyphony, or "chords." That is when more than one note can be simultaneously present. Using the graphical models perspective, this is straightforward to achieve by using a factorial construction in which each constituent of the chord is modeled with an independent reset HMM model. The elements of the chord are coupled via an observation model that combines all elements into a scalar observation at each time [25], [17].

DISCUSSION

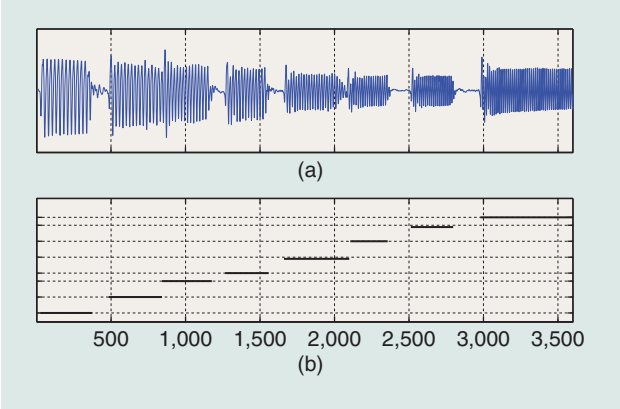
We presented an overview of the graphical models viewpoint of time-series modeling. Graphical models provide a compact description of the basic independence assumptions behind a model and, as such, are a useful way to communicate ideas. This also stresses general-purpose inference routines, for which classical algorithms such as the Kalman filter or forward-backward in the HMM are special cases.

Using graphical models, it is easy to envisage new models tailored for a particular environment. For example, we highlighted the switching state-space models and their



[FIG14] The independence structure of a reset HMM model. Square nodes c_t denote binary change point variables, x_t are continuous latent variables, and y_t continuous observations. The discrete state s_t determines which LDS from a finite set of LDSs is operational at time t.

[FIG15] A single phasor plotted as a damped two dimensional rotation. By taking a projection onto the y axis, the phasor generates a damped sinusoid.



[FIG16] Note transcription of a signal recorded from a bass guitar playing a major scale. (a) Raw acoustic signal. (b) The most probable joint note trajectory is shown with the vertical axis denoting the note index.

potential in signal processing applications. Such models are natural extensions of traditional signal processing techniques that are limited to the assumption that the underlying process generating the data is either discrete or continuous. In particular, the SLDS is a simple graphical marriage of a continuous and discrete latent Markov model. We showed how these models can be used to detect changes in the underlying dynamics of a system, and gave examples of their use in signal reconstruction and system monitoring.

Issues with computational tractability do not magically disappear within this framework, and the switching models are formally computationally intractable in general. Nevertheless, in some cases simple deterministic approximations based on mixture models can be effective, for which the graphical model helps guide intuition in the approximation. Alternatives to the deterministic approximation method we discussed are based on Monte Carlo sampling. Typical strategies use Markov chain Monte Carlo (MCMC) or sequential Monte Carlo, also known as sequential importance sampling or particle filtering [26], [27], with specialized algorithms designed for switching state-space models; for MCMC (see [28] and [29]).

The effective application of switching models in the real world is gaining pace, partly through the restricted reset models, but also via increased computational power that brings the more general models into consideration through carefully developed approximations. As such, developing new models and associated approximate inference schemes is likely to remain an active area of research, with graphical models playing an important role in facilitating communication and guiding intuition.

AUTHORS

David Barber (D.Barber@cs.ucl.ac.uk) received his B.A. degree in mathematics from Cambridge University and his Ph.D. degree in theoretical physics (statistical mechanics) from Edinburgh University. He is currently a reader in information processing in the Department of Computer Science, University College London (UCL), where he develops novel information processing schemes, mainly based on the application of probabilistic reasoning. Prior to joining UCL he was a lecturer at Aston and Edinburgh Universities.

A. Taylan Cemgil (taylan.cemgil@boun.edu.tr) received his B.Sc. and M.Sc. degrees in computer engineering from Bogazici University, Turkey. He received his Ph.D. degree from Radboud University, Nijmegen, The Netherlands, in 2004. He worked as a postdoctoral researcher at the University of Amsterdam and the University of Cambridge. Since 2008, he has been an assistant professor of computer engineering at Bogazici University. His research is focused on developing computational techniques for statistical information processing. He is a Member of the IEEE.

REFERENCES

[1] S. J. Godsill and P. J. W. Rayner, *Digital Audio Restoration—A Statistical Model-Based Approach*. New York: Springer-Verlag, 1998.

[2] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*, 2nd ed. New York: Springer-Verlag, 1997.

[3] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 2000.

[4] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[5] R. H. Shumway and D. S. Stoffer, "Dynamic linear models with switching," *J. Amer. Statist. Assoc.*, no. 86, no. 415, pp. 763–769, 1991.

[6] P. Fearnhead, "Exact and efficient Bayesian inference for multiple changepoint problems," *Statist. Compute.*, vol. 16, no. 2, pp. 203–213, 2006.

[7] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1–2, pp. 1–305, 2008.

[8] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[9] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models*. New York: Springer-Verlag, 2005.

[10] Y. Ephraim and W. J. J. Roberts, "Revisiting autoregressive hidden Markov modeling of speech signals," *IEEE Signal Processing Lett.*, vol. 12, no. 2, pp. 166–169, Feb. 2005.

[11] B. Mesot and D. Barber, "Switching linear dynamical systems for noise robust speech recognition," *IEEE Trans. Audio, Speech Lang. Processing*, vol. 15, no. 6, pp. 1850–1858, 2007.

[12] M. Verhaegen and P. van Dooren, "Numerical aspects of different Kalman filter implementations," *IEEE Trans. Automat. Contr.*, vol. 31, no. 10, pp. 907–917, 1986.

[13] H. E. Rauch, G. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *Amer. Inst. Aeronaut. Astronaut. J.*, vol. 3, no. 8, pp. 1445–1450, 1965.

[14] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques and Software*. Norwood, MA: Artech House, 1998.

[15] C.-J. Kim and C. R. Nelson, *State-Space Models with Regime Switching*. Cambridge, MA: MIT Press, 1999.

[16] S. Chib and M. Dueker, "Non-Markovian regime switching with endogenous states and time-varying state strengths," Working Paper 2004–030, Federal Reserve Bank of St. Louis, 2004.

[17] A. T. Cemgil, B. Kappen, and D. Barber, "A generative model for music transcription," *IEEE Trans. Audio, Speech Lang. Processing*, vol. 14, no. 2, pp. 679–694, 2006.

[18] M. I. Jordan, *Learning in Graphical Models*. Cambridge, MA: MIT Press, 1998.

[19] S. Frühwirth-Schnatter, *Finite Mixture and Markov Switching Models*. New York: Springer-Verlag, 2006.

[20] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural Comput.*, vol. 12, no. 4, pp. 963–996, 1998.

[21] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Automat. Contr.*, vol. 17, no. 4, pp. 439–448, 1972.

[22] T. Minka, "Expectation propagation for approximate Bayesian inference," Ph.D. dissertation, MIT, 2001.

[23] D. Barber, "Expectation correction for smoothing in switching linear Gaussian state space models," *J. Mach. Learn. Res.*, vol. 7, pp. 2515–2540, 2006. [Online]. Available: http://jmlr.csail.mit.edu/papers/volume7/barber06a/barber06a.pdf

[24] R. G. Jarrett, "A note on the intervals between coal-mining disasters," *Biometrika*, no. 66, no. 1, pp. 191–193, 1979.

[25] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, no. 2, pp. 245–273, 1997.

[26] A. Doucet, N. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[27] N. Whiteley, A. Doucet, and C. Andrieu, "Particle MCMC for multiple changepoint models," *Bristol Univ., Statist. Res.* Rep. 09:11, 2009.

[28] C. K. Carter and R. Kohn, "Markov chain Monte Carlo in conditionally Gaussian state space models," *Biometrika*, vol. 83, no. 3, pp. 589–601, 1996.

[29] R. Chen and J. S. Liu, "Mixture Kalman filters," *J. R. Statist. Soc. Series B*, vol. 62, no. 3, pp. 493–508, 2000. [SP]