ST451 - Lent term
Bayesian Machine Learning

Kostas Kalogeropoulos

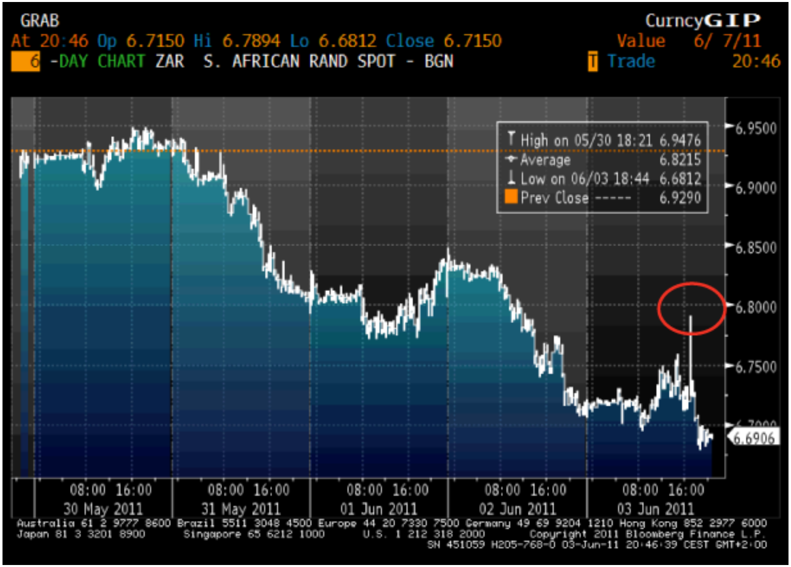
Sequential Data

Outline

Outline

Example 1: Self driving cars

Example 2: High frequency finance



Sequential Data

We have a sequence of observations $x_1, x_2, x_3, ..., x_t$. For example:

- Sequence of images
- Kinematic variables in a robot
- Speech signals
- Sensor readings from an industrial process
- Stock prices
- ...

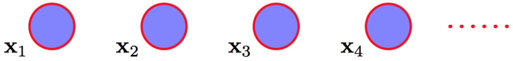
Goal: To build probabilistic model that can predict $\pi(x_t|x_{t-1}, x_{t-2}, \dots)$

Often the observations appear within short periods of time and we want to incorporate immediately in our prediction model.

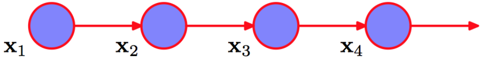
Directly observed models

A simple approach is to assume that the data are observed directly without error. We can then consider various models based on the dependence of the $x's$:

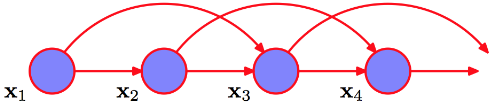
Model assuming independence:



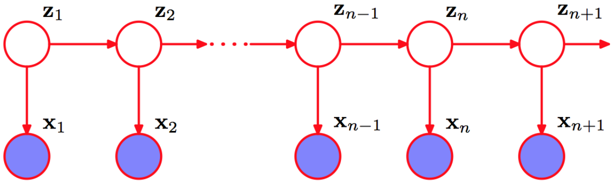
Markov model:



2nd order Markov model:



Causal structure and ‘Hidden variables’



Speech recognition:

- z - underlying phonemes or words
- x - acoustic waveform

Vision:

- z - object identities, poses, illumination
- x - image pixel values

Finance:

- z - underlying value of daily volatility
- x - observation index by intraday data or option prices

Outline

Linear Gaussian State Space Models

We can generally write

$$\pi_\theta(z_{1:T}, x_{1:T}) = \pi(z_1)\pi(x_1|z_1)\prod_{t=2}^{T}\pi(z_t|z_{t-1})\pi(x_t|z_t),$$

where z_t, x_t are both real vectors and $x_{1:T} = (x_1, \ldots, x_T)$

Linear Gaussian State Space Models are defined as

$$\begin{aligned} z_t &= A_t\, z_{t-1} + B_t\, u_t + \epsilon_t, \quad \epsilon_t \sim N(0, Q_t) \\ x_t &= C_t\, z_t + D_t\, u_t + \delta_t, \quad \delta_t \sim N(0, R_t). \end{aligned}$$

In the model above x_t and u_t (inputs) are observable.

If the parameters $\theta_t = (A_t, B_t, C_t, D_t, Q_t, R_t)$ are independent of time, the model is called stationary.

Linear Gaussian State Space Models

There are three separate problems regarding inference regarding z_t:

- Filtering: What is the distribution of z_t given all information up to time t?

$$\pi(z_t | x_1, \ldots, x_t)$$

- Prediction: What is the distribution of z_{t+d} given all information up to time t?

$$\pi(z_{t+d} | x_1, \ldots, x_t)$$

- Smoothing: What is the distribution of z_t given all information up to time T?

$$\pi(z_t | x_1, \ldots, x_T)$$

A simple idea: running averages

Consider the following simple estimator:

For filtering take

$$\hat{z}_t = \tfrac{1}{t}\sum_{i=1}^{t} x_i$$

For prediction of z_t given $x_{1:t-1}$ take

$$\hat{z}_{t-1} = \tfrac{1}{t-1}\sum_{i=1}^{t-1} x_i$$

Note that we can write

$$\begin{aligned}\hat{z}_t &= \left(\tfrac{t-1}{t}\right)\hat{z}_{t-1} + \tfrac{1}{t}x_t \\ \hat{z}_t &= \hat{z}_{t-1} + \tfrac{1}{t}(x_t - \hat{z}_{t-1})\end{aligned}$$

We call $\frac{1}{t}$ as the Kalman gain a key component of the Kalman filter.

The Kalman filter main idea

As mentioned last week the Kalman filter was voted recently among the top 10 algorithms in sciences and engineering.

It is essentially the Bayes theorem under the Markov chain framework in order to obtain the filtering distribution (posterior) based on a sequentially updated prior

$$\begin{aligned}
\pi(z_t|x_{1:t}) &= \int \pi(z_t, z_{t-1}|x_t, x_{1:t-1})dz_{t-1} \\
&= \int \frac{\pi(z_t, z_{t-1}, x_t, |x_{1:t-1})}{\pi(x_t|x_{1:t-1})}dz_{t-1} \\
&\propto \int \pi(x_t|z_t, z_{t-1}, x_{1:t-1})\pi(z_t|z_{t-1}, x_{1:t-1})\pi(z_{t-1}|x_{1:t-1})dz_{t-1} \\
&\overset{\text{Markov}}{=} \int \pi(x_t|z_t)\pi(z_t|z_{t-1})\pi(z_{t-1}|x_{1:t-1})dz_{t-1}
\end{aligned}$$

The Kalman filter recursions

Notation: $\mu_t^i = \mathrm{E}[z_t | x_{1:i}]$, $\Sigma_t^i = \mathrm{Var}[z_t | x_{1:i}]$

Prediction

$$\begin{aligned} \mu_t^{t-1} &= A\,\mu_{t-1}^{t-1} + B\,u_t \\ \Sigma_t^{t-1} &= A\,\Sigma_{t-1}^{t-1}\,A^T + Q \end{aligned}$$

Kalman gain

$$K_t = \Sigma_t^{t-1} C^T \left(C \Sigma_t^{t-1} C^T + R) \right)^{-1}$$

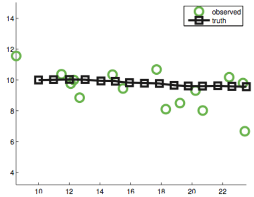
Correction

$$\begin{aligned} \mu_t^t &= \mu_t^{t-1} + K_t\,(x_t - C\,\mu_t^{t-1} - D\,u_t) \\ \Sigma_t^t &= \Sigma_t^{t-1} - K_t\,C\,\Sigma_t^{t-1} \end{aligned}$$

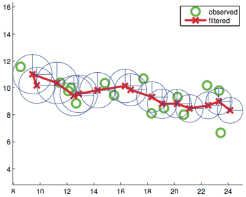
Derivations follow standard but tedious multivariate Normal properties.

The Kalman smoother

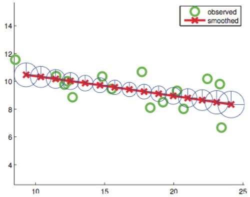
For $\pi(x_t | y_{1:T})$ we run the filter up to time T and then go backwards.



(a)



(b)



(c)

Smoothing

$$
\begin{aligned}
J_t &= \Sigma_t^T A^T \left(\Sigma_t^{t+1}\right)^{-1} \\
\mu_t^T &= \mu_t^t + J_t \left(\mu_{t+1}^T - A\,\mu_t^{t-1} - B\,u_t\right) \\
\Sigma_t^T &= \Sigma_t^t - J_t \left(\Sigma_{t+1}^T - \Sigma_{t+1}^t\right) J_t^T
\end{aligned}
$$

Statistical Inference based on Kalman filter

To estimate parameters θ, we can write down the marginal likelihood

$$\pi(x_{1:T}|u_{1:T},\theta) = \prod_{t=1}^{T} \pi(x_t|x_{1:t-1},u_{1:t},\theta) = \prod_{t=1} N(x_t|M_t, S_t)$$

where $M_t = C\mu_t^{t-1} + D_t u_t$ and $S_t = C\Sigma_t^{t-1}C^T + R$

Note that the above is free of the z's, hence MLEs can be obtained by direct maximisation. The EM algorithm can also be applied; it requires the smoothing densities obtained by the Kalman smoother.

For Bayesian inference one needs to assign priors $\pi(\theta)$ and implement Metropolis-Hastings or Hamiltonian MCMC methods.

Extensions of the Kalman filter

Kalman filter covers linear Gaussian state space models only.

For non-linear cases similar approximations are available such as the Extended Kalman filter (EKF) or the Unscented Kalman filter (UKF).

For non-linear non-Gaussian cases other schemes can be used such as sequential Monte Carlo, e.g. the particle filter.

Outline

Hidden Markov Models

Discrete hidden states $z_t \in \{1..., K\}$, and outputs y_t (discrete or continuous).

Joint probability as before

$$\pi(z_{1:T}, x_{1:T}) = \pi(z_1)P(x_1|z_1)\prod_{t=2}^{T}\pi(z_t|z_{t?1})\pi(z_t|z_t)$$

Can be viewed as a Markov chain with stochastic measurements.

or a mixture model with states coupled across time.

Generative model

A first-order Markov chain generates the hidden state sequence (path):

$$\begin{aligned} \text{Initial State:} \quad \pi_j = &\quad P(z_1 = j) \\ \text{transition matrix:} \quad T_{ij} = &\quad P(z_{t+1} = j | z_t = i) \end{aligned}$$

A set of output distributions converts the state path into a sequence of observations x_t:

$$\begin{aligned} A_j(x) &= f(x_t | z_t = j) && \text{if x is continuous} \\ A_{jk} &= P(x_t = k | z_t = j) && \text{if x is discrete} \end{aligned}$$

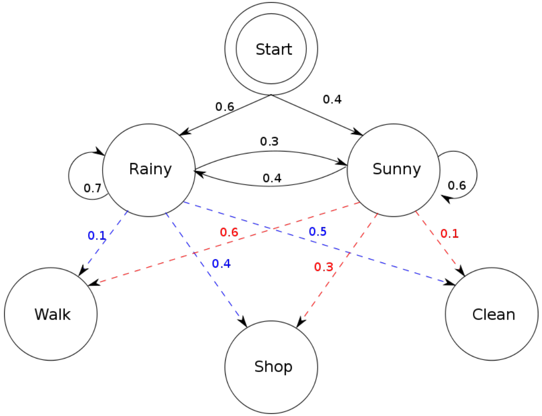
Even though hidden state sequence is first-order Markov, the output process may not be Markov.

Toy Example: Bob and Alice

- Two friends from different countries, Bob and Alice talk over the phone every day.
- Bob tells Alice whether he went walking, shopping or cleaned his apartment, activities determined by the weather (rainy or sunny)
- Alice wants to guess the weather in Bob's country based on what he says.
- She also believes that weather behaves like a Markov chain.

The above can be described by a Hidden Markov Model.

Toy Example: Bob and Alice



$$T = \begin{pmatrix} 0.7 & 0.3 \\ 0.6 & 0.4 \end{pmatrix}, A = \begin{pmatrix} 0.1 & 0.4 & 0.5 \\ 0.6 & 0.3 & 0.1 \end{pmatrix}, \pi = (0.6, 0.4).$$

Inference tasks

Assume that Bob says (x) 'walk', 'clean', 'shop', 'shop', 'clean', 'walk', 'shop', 'clean', 'shop', 'walk', 'walk', 'clean', 'shop.'

Generally we would like to answer the following questions:

1. Evaluation Find $\pi(x|\theta)$ for a $\theta = (\pi, T, A)$.
2. Filtering, Smoothing, Decoding: Find $\pi(z_t|x_{1:t}, \theta)$ and $\pi(z_t|x_{1:T}, \theta)$ as before. Also, find an optimal state sequence $z = \{z_1, \ldots, z_T\}$ which best explains the observations x.
3. Learning Problem: Draw statistical inference on θ based on x.

We will suppress θ in the notation $\pi(x) = \pi(x|\theta)$ for questions (1) and (2).

Forward algorithm

Evaluating $\pi(x)$ requires summing out all K^T possible z paths!

Write $\pi(z_t = j | x_{1:t-1}) = \sum_i \pi(z_t = j | z_{t-1} = i)\pi(z_{t-1} = i | x_{1:t-1})$ and let

$$\begin{aligned} \alpha_t(j) &= \pi(z_t = j | x_{1:t}) = \pi(z_t = j | x_t, x_{1:t-1}) \\ &= \frac{1}{Z_t}\pi(x_t | z_t = j, \cancel{x_{1:t-1}})\pi(z_t = j | x_{1:t-1}) \\ Z_t &= \pi(x_t | x_{t-1}) = \sum_j \pi(x_t | z_t = j)\pi(z_t = j | x_{1:t-1}) \end{aligned}$$

So $\alpha_t(j)$ provides filtering and evaluation of $\pi(x_t | x_{t-1})$ in $O(TK^2)$ cost.

Note that (aka dynamic programming)

$$\begin{aligned} \alpha_t(j) &= \frac{1}{Z_t}\pi(x_t | z_t = j)\pi(z_t = j | x_{1:t-1}) \\ &= \frac{1}{Z_t}\pi(x_t | z_t = j)\sum_i \pi(z_t = j | z_{t-1} = i)\,\alpha_{t-1}(j) \end{aligned}$$

Forward Backward Algorithm

Regarding smoothing, Note that we can write

$$\begin{aligned}\gamma_j(t) &= \pi(z_t = j | x_{1:T}) = \frac{\pi(z_t = j, x_{1:t})\pi(x_{t+1:T} | z_t = j, x_{1:t})}{\pi(x_{1:T})} \\ &= \frac{\pi(z_t = j, x_{1:t})\pi(x_{t+1:T} | z_t = j)}{\pi(x_{1:T})} = \frac{\alpha_j(t)\beta_j(t)}{L},\end{aligned}$$

where $\beta_j(t)$ is a backward recursion

$$\beta_j(t) = \pi(x_{t+1:T} | z_t = j) = \sum_i T_{ij}\beta_i(t+1)A_i(x_{t+1})$$

This is known as the forward-backward algorithm, which is a special case of the sum-product aka belief propagation algorithm that applies to more general graphs with tree-structure.

Viterbi Encoding

- The $\gamma_j(t)$'s provide the posterior over states at each time.
- An optimal state path can be obtained by choosing the state $j^*(t)$ with the largest $\gamma_j(t)$ at each time.
- But it is not the path with the highest probability of generating the data. In fact it may be a path of probability zero!
- To find the single best path, we use the Viterbi decoding algorithm, a special case of Bellman's dynamic programming.
- The recursions look the same as forward-backward, except with max instead of sum. A special case of the max-sum algorithm.

Statistical inference on θ

Frequentist inference can proceed as before:

- Use forward algorithm, compute $\pi(x|\theta)$ via the $\alpha_j(t)$'s and apply direct maximisation.
- Use the forward backward algorithm to get $\gamma_j(t)$'s and apply the EM algorithm.

Those algorithms can also be used for Bayesian inference. Perhaps the simplest approach is to use Metropolis Hastings or Hamiltonian MCMC equipped with the forward algorithm calculation of $\pi(x|\theta)$.

Today's lecture - Reading

Murphy: 17.3 17.4.1-4 17.5.1-3 18.1-18.5

Bishop: 13.2 13.3