Assignment 2 – Bonus Question

Unsupervised & Probabilistic Learning

Maneesh Sahani

Due: Monday Nov 18, 2013

Note: all assignments for this course are to be handed in to the Gatsby Unit, **not** to the CS department. Assignments are due at the **beginning** of the lecture or tutorial on the due date. Late assignments (included those handed in later on the due day) will be penalised. If you are unable to attend, you may hand in your assignment to either lecturer or TA prior to the due time, or to Barry Fong in the Alexandra House 4th floor reception. Do not leave them with anyone else.

Please attempt the main coursework assignments before this bonus one.

1. **[60 points] LGSSMs, EM and SSID**.

   Download the datafiles `ssm_spins.txt` and `ssm_spins_test.txt`. Both have been generated by an LGSSM:

$$\mathbf{y}_t \sim \mathcal{N}(A\mathbf{y}_{t-1}, Q) \quad [t = 2 \dots T] \qquad \mathbf{y}_1 \sim \mathcal{N}(0, I)$$
$$\mathbf{x}_t \sim \mathcal{N}(C\mathbf{y}_t, R) \quad [t = 1 \dots T]$$

   using the parameters:

$$A = 0.99 \begin{pmatrix} \cos(\frac{2\pi}{180}) & -\sin(\frac{2\pi}{180}) & 0 & 0 \\ \sin(\frac{2\pi}{180}) & \cos(\frac{2\pi}{180}) & 0 & 0 \\ 0 & 0 & \cos(\frac{2\pi}{90}) & -\sin(\frac{2\pi}{90}) \\ 0 & 0 & \sin(\frac{2\pi}{90}) & \cos(\frac{2\pi}{90}) \end{pmatrix} \qquad Q = I - AA^{\mathsf{T}}$$

$$C = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix} \qquad R = I$$

   but different random seeds. We shall use the first as a training data set and the second as a test set.

   (a) Run the function `ssm_kalman.m` we have provided (or a re-implmentation in your favourite language if you prefer) on the training data. Warning: the function expects data vectors in columns; you will need to transpose the loaded matrices!

   Make the following plots:

```
logdet = @(A)(2*sum(log(diag(chol(A)))));
[Y,V,~,L] = ssm_kalman(X',Y0,Q0,A,Q,C,R, 'filt');
plot(Y'):
plot(cellfun(logdet,V));

[Y,V,Vj,L] = ssm_kalman(X',Y0,Q0,A,Q,C,R, 'smooth');
plot(Y'):
plot(cellfun(logdet,V));
```

   Explain the behaviour of `Y` and `V` in both cases (and the differences between them).

   [5 points]

   (b) Write a function to learn the parameters A, Q, C and R using EM (we will assume that the distribution on the first state is known *a priori*). The M-step update equations for A and C were derived in lecture. You should show that the updates for R and Q can be written:

$$R_{\text{new}} = \frac{1}{T}\left[\sum_{t=1}^{T} \mathbf{x}_t\mathbf{x}_t^{\mathsf{T}} - \Big(\sum_{t=1}^{T} \mathbf{x}_t\langle\mathbf{y}_t^{\mathsf{T}}\rangle\Big)C_{\text{new}}^{\mathsf{T}}\right]$$

$$Q_{\text{new}} = \frac{1}{T-1}\left[\sum_{t=2}^{T} \mathbf{y}_t\mathbf{y}_t^{\mathsf{T}} - \Big(\sum_{t=2}^{T} \langle\mathbf{y}_t\mathbf{y}_{t-1}^{\mathsf{T}}\rangle\Big)A_{\text{new}}^{\mathsf{T}}\right]$$

where C_{new} and A_{new} are as in the lecture notes. Store the log-likelihood at every step (easy to compute from the fourth value returned by `ssm_kalman`) and check that it increases.

[Hint: the matlab code

```
cellsum=@(C)(sum(cat(3,C{:}),3))
```

defines an inline function `cellsum()` that sums the entries of a cell array of matrices.]

Run at least 50 iterations of EM starting from a number of different initial conditions: (1) the generating parameters above (why does EM not terminate immediately?) and (2) 10 random choices.

Show how the likelihood increases over the EM iterations (hand in a plot showing likelihood vs iteration for each run, plotted in the same set of axes). Explain the features of this plot that you think are salient.

[If your code and/or computer is fast enough, try running more EM iterations. What happens?]

[25 points]

(c) Write a function to learn A, Q, C and R using Ho-Kalman SSID. You should provide options to specify the maximum lag to include in the future-past Hankel matrix (H), and the latent dimensionality. You may like to plot the singular value spectrum of H to see how good a clue it provides to dimensionality.

[Hints: if `M` is a cell array of the lagged correlation matrices M_τ, you can build the Hankel matrix using:

```
H = cell2mat(M(hankel([1:maxlag], [maxlag:2*maxlag-1])));
```

The matrices Ξ (`Xi`) and Υ (`Ups`) are found using SVD (assuming `K` latent dimensions):

```
[Xi,SV,Ups] = svds(H, K);

Xi = Xi*sqrt(SV);
Ups = sqrt(SV)*Ups';
```

and these can be used to find $\widehat{C}$ and $\widehat{A}$ and $\widehat{\Pi}$ by regression (`/` or `\` in MATLAB), e.g.:

```
Ahat = Xi(1:end-DD,:)\Xi(DD+1:end,:);
```

(The code for C and Π will take a little more thought!).

To find $\widehat{Q}$ ad $\widehat{R}$ recall that, as Π is the stationary (prior) latent covariance:

$$A\Pi A^{\mathsf{T}} + Q = \Pi$$

and

$$C\Pi C^{\mathsf{T}} + R = \mathcal{E}_{t\to\infty}\left[\mathbf{x}_t\mathbf{x}_t^{\mathsf{T}}\right]$$

Run your code on the training data using a maximum lag of 10 and the true latent dimensionality. Evaluate the likelihood of this model using `ssm_kalman.m` and show it on the likelihood figure. Now use the SSID parameters as initial values for EM and show the resulting likelihood curve.

Comment on the advantages and disadvantanges of SSID as compared to EM.

(If you have time and inclination, you might like to explore how your results vary with different choices of lag and dimensionality)

[25 points]

(d) Evaluate the likelihood of the test data under the true parameters, and all of the parameters found above (EM initialised at the true parameters, random parameters and the SSID parameters, as well as the SSID parameters without EM). Show these numbers on or next to the training data likelihoods plotted above. Comment on the results.

[5 points]