Homework 4: Inference & Learning for Temporal State Space Models

University of Arizona CSC 535: Probabilistic Graphical Models

Homework due at 11:59pm on November 16, 2020

In this problem set, we explore inference and learning algorithms for the *linear dynamical system* (LDS), which is summarized by the graphical model of Fig. 1. Our observations are a sequence of real-valued vectors, $y_t \in \mathbb{R}^p$, which depend on corresponding latent state vectors $x_t \in \mathbb{R}^d$. The joint distribution is a liear-Gaussian model given by,

$$
\begin{aligned}
p(x_t \mid x_{t-1}) &= \mathcal{N}(x_t \mid Ax_{t-1}, Q) \\
p(y_t \mid x_t) &= \mathcal{N}(y_t \mid Cx_t, R)
\end{aligned}
$$

Model parameters are denoted by the set $\{A, C, Q, R\}$. We assume that the initial state is distributed via $p(x_1) = \text{Norm}(x_1 \mid m, S)$ for some initial state mean m and covariance S. (Often, but not always, $m = 0$ and $S = I_d$.)

In general, many different state space models may assign equal likelihood to a given dataset, due to generalized versions of the rotational ambiguities underlying PCA and factor analysis models. For all models we consider, we thus constrain $C = [I_p \ O]$, where O is a $p \times (d-p)$ matrix of zeros, and I_p is a $p \times p$ identity matrix. When $d = p$, we have $C = I_p$.

For some experiments, we define our state space model parameters to take one of a few canonical forms. The *constant position* model takes $d = p$, and

$$
A = I_p, \quad Q = \sigma_x^2 I_p, \quad R = \sigma_y^2 I_p,
$$

so that the observations are noisy estimates of an underlying random walk. The *constant velocity* model generates more smooth motions by taking $d = 2p$, and

$$
A = \begin{bmatrix} I_p & I_p \\ O_p & I_p \end{bmatrix}, \quad Q = \begin{bmatrix} \sigma_x^2 I_p & O_p \\ O_p & \sigma_v^2 I_p \end{bmatrix}, \quad R = \sigma_y^2 I_p,
$$

where O_p is a $p \times p$ matrix of zeros. More generally, we will use the expectation maximization (EM) algorithm to learn model parameters $\{A, Q, R\}$ from observation sequences.

Question 1: Tracking and Kalman Filters

In the handout code, see `kalman_smoother.m` for a partial implementation of a Kalman smoother. The Kalman smoother is equivalent to Gaussian BP, and computes full posterior marginals at each time in a Gaussian LDS.

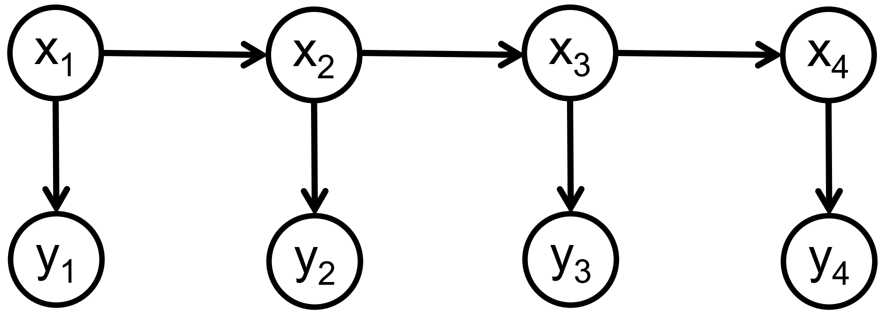

Figure 1: A directed graphical model representing a linear dynamical system (LDS). At each time step continuous state x_t evolves conditional on x_{t-1} according to dynamics and generates observations y_t.

a) *Given a sequence of T observations* `kalman_smoother.m` *should produce a $d \times T$ matrix of posterior mean estimates, and a $d \times d \times T$ array of posterior covariance estimates. The provided code fully implements the backwards smoothing pass, but the forward filter needs to be implemented. Using the lecture slides and textbooks (Murphy Ch. 18, Bishop 13.3) as references, implement the forward pass of the Kalman filter. Specifically, implement the prediction and measurement updates on lines marked by (**).*

b) *Consider the $p = 1$-dimensional observation sequence we provide in* `track`*. It was sampled from a constant velocity model with $d = 2, \sigma_v^2 = 0.01, \sigma_x^2 = 0.01/3, \sigma_y^2 = 20$. Apply the Kalman filter code using this correct model. On one set of axes, plot the observation sequence, the posterior mean of the first (position) component of the state, and posterior confidence intervals. The confidence intervals should be determined as the posterior mean plus or minus two times the posterior standard deviation of the first state component.*

c) *Suppose you incorrectly assumed the* `track` *data was generated by a constant-position model with $d = 1$ and $\sigma_y^2 = 20$. Consider two possible state-transition noise levels, $\sigma_x^2 = 0.01/3$ and $\sigma_x^2 = 10$. For each of these alternative models, apply the Kalman filter code and plot your results as in part (a). Discuss differences from the results in part (a).*

d) *We finish by exploring robustness to outliers. Again consider the* `track` *data and independently at each time step t, with probability 0.1 replace the true observation by a sample $y_t \sim Norm(0, 40^2)$. Apply the Kalman filter to this corrupted data, with the (now inaccurate) original observation likelihood. Plot the corrupted observation data, as well as the mean estimates and discuss.*

Question 2: Learning, Kalman Smoothers, & Expectation Maximization (EM)

In the handout code, see `em_slds.m` for an implementation of the EM algorithm for learning state space models.

a) `kalman_smoother` *returns marginal moments for, both, the filter (forward pass only) and smoother (forward and backward passes). Apply the Kalman smoother to the constant velocity model and* `track` *data from part 1(a). On one set of axes, plot the observation* *sequence, the posterior mean of the first (position) component of the state, and posterior confidence intervals. Compare to the estimates produced by the Kalman filter and explain differences.*

b) *The marginal log-likelihood of an observation sequence y, integrating over states x for some fixed state space model parameters, can be written as follows:*

$$\begin{aligned}\log p(y) &= \log p(y_1) + \sum_{t=1}^{T-1} \log p(y_{t+1} \mid y_1, \ldots, y_t) \\ &= \log \int_{\mathcal{X}_1} p(y_1 \mid x_1) p(x_1)\, dx_1 + \sum_{t=1}^{T-1} \log \int_{\mathcal{X}_t} p(y_{t+1} \mid x_t) p(x_t \mid y_1, \ldots, y_t)\, dx_t\end{aligned}$$

Provide a formula for evaluating the marginal log-likelihood. Your answer should be an explicit function of the Kalman filter mean vectors and covariance matrices. ***Hint:*** *Useful identities for multivariate Gaussian distributions are in Murphy Sec. 4.4 and in Bishop Sec. 2.3.3.*

c) *Implement the body of* `compute_lds_bound.m` *so that it computes the log-likelihood bound derived in part (b).* ***Hint:*** *If your implementation is correct, the log-likelihoods computed during any execution of the EM algorithm should be monotonically increasing.*

d) *Consider the $p = 2$-dimensional observation sequence we provide in* `spiral`*, and use the EM algorithm to learn a $d = 2$-dimensional state space model. Initialize with a constant position model with $\sigma_x^2 = 1, \sigma_y^2 = 1$. Run the EM algorithm for 100 iterations, plot the marginal log-likelihood versus iteration, and report the learned parameters after the final iteration. Plot the observation sequence, and the output of the Kalman smoother after the final iteration, as overlaid 2-dimensional curves.*

e) *Should we expect the learned parameters to be the exact maximum likelihood estimates for this model? Why or why not?*