

Homework 5: Monte Carlo Methods

University of Arizona CSC 535: Probabilistic Graphical Models

Homework due at 11:59pm on December 7, 2020

Question 1: Particle Filters

The goal of this question is to familiarize yourself with the bootstrap particle filter. To simplify things we will use the standard linear dynamical system from Homework 4. Recall that The joint distribution of the LDS with latent state $x_t \in \mathbb{R}^d$ and observations $y_t \in \mathbb{R}^p$ is given by,

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

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

- a) The basic “bootstrap” particle filter proposes particles from the state transition distribution $p(x_t | x_{t-1})$, reweights with the observation likelihood $p(y_t | x_t)$, and resamples weighted particles (with replacement) at each time step. `particle_filter.m` provides a partial implementation of the particle filter for a Gaussian state space model, but is missing three critical steps: (i) particle resampling (ii) particle propagation (iii) weight updating. Implement these steps, which are denoted by `(**)` in the code, to complete the particle filter.
- b) Consider the $p = 1$ -dimensional observation sequence we provide in `track`. It was sampled from a constant velocity model with,

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}, \quad R = \sigma_y^2, \quad C = (1, 0)$$

where $d = 2$, $\sigma_v^2 = 0.01$, $\sigma_x^2 = 0.01/3$, $\sigma_y^2 = 20$. Apply your particle filter to this model and data using 100 particles. At each time step, estimate the posterior mean as the weighted mean (before resampling) of the current particle locations. Repeat this experiment three times, and on a single set of axes, plot the three particle filter mean estimates together with the true posterior means as computed by the Kalman filter. **Note:** We have provided an implementation of the Kalman filter/smoothen from HW 4 in `kalman_smoother.m` though feel free to use your own implementation.

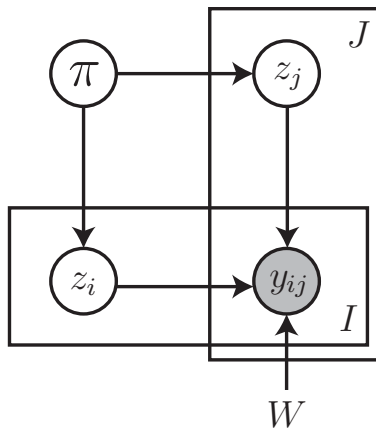


Figure 1: **Stochastic Block Model** The PGM uses overlapping plates to denote dependence of observation y_{ij} on pairs of elements $\{z_i, z_j\}$. This notation is not common, but this type of dependence cannot be encoded using a single plate.

- c) Repeat part (b) using only 20 particles. Discuss the reliability of the particle filter, compared to the Kalman filter.
- d) We now explore 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 $y_t \sim \text{Norm}(0, 40^2)$. Apply, both, the Kalman filter and particle filter (with 100 particles) to this corrupted data, with the (now inaccurate) original observation likelihood. Plot the corrupted observation data, as well as the mean estimates for both methods, and discuss.
- e) Now, we finish by investigating the effect of particle resampling on this data. Disable the particle resampling stage in your particle filter code and repeat parts (b) and (c) along with associated plots. Plot the histogram of weight values at the final stage (T). What behavior do you observe?

Question 2: Stochastic Block Models & Gibbs Sampling

Relational models allow inferences on unknown relationships between observable quantities, such as in biological or social networks. Consider a set of N nodes, representing entities to be modeled. For node pairs $i \neq j$, we let $y_{ij} = 1$ if there is a relationship (e.g., friendship in a social network) from entity i to entity j , and $y_{ij} = 0$ otherwise.

We model such relational data using the *stochastic block model* in Fig. 1 (See Murphy: Sec. 27.5.1). Data are generated from K unknown latent communities, indexed by integers $k = 1, \dots, K$. For each pair of communities k, ℓ , we define an interaction probability $W_{k\ell}$, and place a uniform prior distribution:

$$W_{k\ell} \sim \text{Beta}(1, 1), \quad 1 \leq k, \ell \leq K.$$

The stochastic block model assumes each entity i is a member of a single latent community z_i , sampled according to

$$z_i \sim \text{Cat}(\pi), \quad i = 1, \dots, N.$$

The K -dimensional distribution of community frequencies π has a symmetric Dirichlet prior:

$$\pi \sim \text{Dir}(\alpha, \dots, \alpha).$$

For all experiments we set $\alpha = 1$. For any pair of nodes $i \neq j$, their relationship link variables are generated according to

$$p(y_{ij} = 1 \mid z_i = k, z_j = \ell, W) = W_{k\ell}, \quad p(y_{ji} = 1 \mid z_i = k, z_j = \ell, W) = W_{\ell k}.$$

We must assume that only partial observations y_{ij} of these link variables are available for a (known) subset of ordered entity pairs (i, j) . Your derivations and code should support partial observations.

- a) Given fixed parameters π, W , and observations y for some subset of ordered entity pairs, derive a formula for the posterior distribution $p(z_i \mid z_{\setminus i}, y, \pi, W)$. Here, $z_{\setminus i}$ denotes the community assignments for all entities except node i . **Hint:** Remember that z_i impacts the outgoing relationships y_{ij} from entity i to other entities j , as well as the incoming relationships y_{ji} from those other entities to node i .
- b) Given fixed entity assignments z , derive formulas for the posterior distributions of the model parameters, $p(\pi \mid y, z, W)$ and $p(W_{k\ell} \mid y, z, \pi)$. **Hint:** These should work out to be $\text{Dirichlet}(\pi \mid \cdot)$ and $\text{Beta}(W_{k\ell} \mid \cdot)$ distributions, respectively.
- c) We have provided skeleton code for a block model Gibbs sampler in `sbGibbs.m`, which provides appropriate initializations and evaluates the joint log-probability at each iteration:

$$\begin{aligned} \log p(\pi, W, z, y) = & \log \text{Dir}(\pi \mid \alpha) + \sum_{k=1}^K \sum_{\ell=1}^K \log \text{Beta}(W_{k\ell} \mid 1, 1) \\ & + \sum_{i=1}^N \log \text{Cat}(z_i \mid \pi) + \sum_{i=1}^N \sum_{j \neq i} \log \text{Ber}(y_{ij} \mid W_{z_i z_j}) \end{aligned}$$

Using the formulas from parts (a,b), implement code which resamples each of the z, W, π variables once per iteration.

- d) Test your sampler using the code in `testToy.m`, which constructs a synthetic dataset with $N = 30$ entities, 10 in each of $K = 3$ communities. Generate links y_{ij} by assuming a within-community link probability of $W_{kk} = 0.95$, and a between-community link probability of $W_{k\ell} = 0.10, \ell \neq k$. Then given only the observed links y , explore how accurately the sampler recovers the underlying W, z variables. Run your sampler for 1000 iterations from each of 5 random initializations, and plot the resulting log-likelihood curves on a single set of axes. After each iteration, compute the Rand index (see provided script) between the true assignments and the sampled z , and plot these scores versus iteration.
- e) In 1968 Samuel F. Sampson recorded social interactions among a group of 18 monks preparing to enter a monastery (Sampson, F. S., Dissertation, Cornell U., 1968). The original collected data include a variety of relations, such as liking, dislike, influence,

and so on, which we have distilled for simplification. Using the code in `testSampson.m`, apply the stochastic block model to the Sampson monk data. Allow $K = 3$ communities, run the sampler for 1000 iterations from each of 5 random initializations, and again plot log-likelihoods and Rand indexes (from the true faction labels) versus iteration.

- f) During Sampson's stay a political "crisis in the cloister" resulted in the expulsion of four monks, and voluntary departure of others, making this dataset an interesting case study for relational models. Since much of data analysis is exploratory, use any visualizations that you feel appropriate to provide an interpretation of the sampler results in part (e). Which monks belong to which communities? Are the communities stable across multiple initializations? Are some monks strongly/weakly associated with some communities?