Homework 6

Handed out: Tuesday, November 20, 2018

Due: , December , 2018 Midnight

Notes:

- We *highly* encourage typed (Latex or Word) homework. Compile as single report containing solutions, derivations, figures, etc.
- Submit all files including report pdf, report source files (e.g. .tex or .docx files), data, figures produced by computer codes and programs files (e.g. .py or .m files) in a **.zip** folder. Programs should include a Readme file with running instructions.
- Zipped folder should be turned in on Sakai with the following naming scheme: **HW6_LastName_FirstName.zip**
- Collaboration is allowed however all submitted reports, programs, figures, etc. should be an individual student's write ups. Direct copying could be considered cheating.
- Software resources for this homework set can be downloaded from website or on Sakai under the Resource folder.

Problem 1-20pts

EM algorithm

Implement the EM algorithm for estimating the parameters of a mixture of Gaussians with isotropic covariances using the data provided on data resources. There are two datasets each of which is two-dimensional. You can write your own or use any available code for mixture of Gaussians (e.g. you can use the code in the code directory with some changes to account for the isotropic covariances. Also see the accompanying paper Unsupervised Learning of Finite Mixture Models, M. Figueiredo and A.K. Jain.)

- Experiment with the number of mixtures and comment on the tradeoff between the number of mixtures and goodness of fit (i.e. log-likelihood) of the data. Plot the log-likelihood as a function of the number of components of a mixture of Gaussians to support your argument.
- Find a fixed number of Gaussians that works well for each data set.
- Plot the estimated Gaussians as one-sigma contours of each mixing component on top of the training data.
- List the mean, covariance and mixing weights of each mixture component.

Problem 2-20pts

Particle Filter

A cable layer ship is lost at the arctic. The ship can measure the distance from the bottom of the sea using sonar sensors that are prone to error (Gaussian, $\sigma = 0.1$). Considering that we do have bathymetric map, use particle filter method to find the location of the ship. The movement of the ship is constrained to 1D, starting from $[4, -25]$ corresponding to a point on the bottom of the ocean and with speed 1. Note that the speed measurement is also prone to error (Gaussian, $\sigma = 1.0$) and the function modeling submerged terrain is provided in the recourse code. Starting with 200 particles, plot particles form $t = 1$ until $t = 60$ with the size of the particle demonstrating it's corresponding weight and generate an animation from these plots. In your report, turn in plots for $t = 1, 15, 30, 45$, and 60.

Problem 3-20pts

Kalman Filter

Suppose we want to track the position of a drone in 2D plane. Our state is

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$$

where $\mathbf{x} \in \mathbb{R}$ is the position from starting point. To do this, we are using data provided by a GPS attached to the drone. This GPS, however, reports the location of the drone with some error (Gaussian, $\sigma = 4$). A way of getting the errors and noise out of our measurement is by using Kalman filtering. In this problem, the state model is defined as:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + Bu$$

where A and B denote state transition and control matrices, respectively, which are set to:

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Using the data provided in the code, implement a Kalman filter to track the position of the drone for 2500 time steps. In your report, include the true path of the drone, noisy GPS observations, and the result after Kalman filter. Include another plot with these information but only for time steps 300 to 600.

Problem 4-20pts

Sampling Importance Resampling

We are designing a self driving car that rides in the city using computer vision to detect obstacles, pedestrians, and other landmarks. The control system maneuvers by pivoting the front wheels which makes the car pivot around the rear axle while moving forward. This nonlinear behavior can be modeled using following equations:

$$d = v\Delta t$$

$$x = x + \cos(\theta + \alpha)d$$

$$y = y + \sin(\theta + \alpha)d$$

$$\theta = \theta + \alpha$$

where θ and α denote vehicle orientation and steering angle, respectively, and v is the forward velocity. We will maintain the position x, y and orientation θ of the vehicle. We consider that the vehicle starts from $[0, 0]$ and moves on a straight line moving 1 unit in x and 1 unit in y direction per time step. The landmarks are located at $[-1, 2]$, $[5, 10]$, $[12, 14]$, $[18, 21]$ and the vision mechanism measures distance to visible landmarks. Both the computer vision and control mechanism are prone to error. σ for the error in the turn is 0.2, for the distance is 0.05, and for vision sensor is 0.1.

- Implement a Sampling Importance Resampling filter to track the location of the car.
- Start from 5000 particles uniformly distributed on a 10×10 domain.
- Utilize systematic resampling.
- Relive degeneracy when ESS is lower than $N/2$.

Problem 5-20pts

Resampling

Randomly generate 100 particles x^i from some distribution π of your choice, and 100 (positive) weights w^i. Normalize the weights such that $\sum_i w^i = 1$, and use the weighted samples $\{x^i, w^i\}$ to estimate the mean m of π, and denote this estimate by $\hat{m}$.

- Resample the particles x^i (from the weights w^i) using multinomial resampling, and estimate the mean from the resampled (now equally weighted) samples. Denote this estimate $\hat{m}_m$.
- Repeat this for systematic resampling, and denote this estimate $\hat{m}_s$.
- Repeat this for stratified resampling, and denote this estimate $\hat{m}_t$.

Repeat the items above multiple times, and report an estimate of the variance for $\hat{m} - \hat{m}_m$, $\hat{m} - \hat{m}_s$, and $\hat{m} - \hat{m}_t$ respectively, conditionally on $\hat{m}$ (that is, do not sample new particles from π, but only repeat the resampling step). Which resampling scheme appears to be the preferred one, in terms of variance?

Problem 6-20pts

EM algorithm

Consider data,

$$D = \left\{ \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 5 \end{pmatrix}, \begin{pmatrix} 2 \\ \star \end{pmatrix} \right\}$$

sampled from a two-dimensional (separable) distribution $p(x_1, x_2) = p_{x_1}(x_1) p_{x_2}(x_2)$ where:

$$p_{x_1}(x_1) = \begin{cases} \frac{1}{\theta_1} \exp(-\frac{x_1}{\theta_1}) & \text{if } x_1 \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$p_{x_2}(x_2) = \begin{cases} \frac{1}{\theta_2} & \text{if } 0 \leq x_2 \leq \theta_2, \\ 0 & \text{otherwise,} \end{cases}$$

and $\star$ in the dataset indicates a missing value.

- What can you infer from θ_2 by looking at D?
- Start with an initial estimate $\boldsymbol{\theta}^0 = \begin{pmatrix} 3 \\ 6 \end{pmatrix}$ and analytically calculate $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^0)$. This is the expected joint data log-likelihood considered in class. For this problem to compute it, you effectively have to marginalize out the missing values. This is the estimate/expectation step in the EM algorithm.
- Find the $\boldsymbol{\theta}$ that maximizes your $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^0)$. This is the maximization step of the EM algorithm.