Lecture 11

Pei-Chen Chen

Nov 4, 2013

1 Homework Related

- HW2 Due on Nov 13 (Wednesday)
- Use Python for the problem on MapReduce in HW2

2 Python

- Python Version: Python 2: 2.6 and 2.7 (version on Gauss)
  Python 3: 3.2 and 3.3

- It does not matter which version you choose but recommend to use the same version across machines. (Prof. Baines will use Python 2 in class.)
- Recommend Software: PyCharm
- Python examples

1. Combine two strings
   foo = 10.0
   bar = "hello"
   print bar
   baz = "class"
   foo2 = bar +baz
   # foo2 puts 2 strings together

2. Conditional Statement
   if (i > 0):
   [tab] print("i is...

[tab] ...
[tab] ...
#the code has to be tab indented
#Don't recommend to copy and paste in Python because it may change the indentation.

3. For loop
   for i in 1:n: is equal to **for i in range(0, n):**

4. foo = 1,2,3,4,5
   foo = list(foo)
   print foo   (1,2,3,4,5)
   print type(foo) $<$ type 'list' $>$
   print foo[0:3]   (1, 2, 3)
   print foo[1:]   (2, 3, 4, 5)
   print foo[:1]   (1,)
   bar=foo[2:]
   print bar   [3, 4, 5]
   foo[3]=100
   print foo   [1, 2, 3, 100, 5]

5. Convert between string, int and float
   print ”float('6.5')” +str(float('6.5')) $\rightarrow$float('6.5')6.5
   print ”int('6.5')” + str(int('6.5')) $\rightarrow$ shows error (can't print)
   print ”int(float('6.5')” + str(int(float('6.5'))) $\rightarrow$ int(float('6.5')6

3 Bayes + Big Data

- Bayes Recap:
  $X_1, ..., X_n \overset{i.i.d}{\sim} N(\mu, \sigma^2)$, where σ^2 known and $\mu \sim N(\mu_0, \sigma_0^2)$

$$\mu|\vec{X} \sim N(\frac{\frac{\mu_0}{\sigma_0^2} + \frac{\sum x_i}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}}, \frac{1}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}})$$

  General: Likelihood $p(\vec{x}|\theta)$;   Prior $p(\theta) \rightarrow$ Posterior: $p(\theta|\vec{x}) \propto p(\theta)p(\vec{x}|\theta)$
  If $p(\theta|\vec{x})$ has no closed form, then use MCMC.

- For big data, when the $\vec{x}$ in $p(\vec{x}|\theta)$ is massive, applying MCMC directly to the full dataset is not computably feasible.

- Idea: "Chuck" the full data set into a series of smaller datasets, sample for the posterior of each smaller dataset and "combine" results. (Treat each one as a separate model and pull together to estimate the posterior distributions.)

- $p(\theta|\vec{x}) \propto p(\theta)p(\vec{x}|\theta) \rightarrow$ divide the $\vec{x}$ from $p(\vec{x}|\theta)$ to different dataset.
  Assume X's are conditionally independent given θ.

$$\prod_{j=1}^{s}\{p(\theta)^{1/s}p(\vec{x}_j|\theta)\}$$

  (If things are conditionally independent.)

- But it is still not the same as if we sample from the full data. Thus...
  Distribute:
  $\vec{X}_n$ (full data) to

$$\begin{array}{ccccc} \vec{X}_1 & \vec{X}_2 & ... & \vec{X}_s & \textbf{(data)} \\ \pi(\theta)^{q_1} & \pi(\theta)^{q_2} & ... & \pi(\theta)^{q_s} & \textbf{(prior)} \end{array}$$

  (Note: $q_1...q_s$ don't have to have the same amount.)

  Where $\sum_{j=1}^{q} q_j = 1, q_j > 0$, for all j , $\cup_j \vec{X}_j = \vec{X}, \cap_j \vec{X}_j = \emptyset$

- Treat each $\vec{X}_j$ and $\pi(\theta)^{q_j}$ as a full model and run MCMC.
  Aside,

$$\mu \sim N(\mu_0, \sigma_0^2)$$
$$p(\mu)^{q_j}, p(\mu) \propto exp\{-\frac{1}{2\sigma_0^2}(\mu - \mu_0)^2\}^{q_j}$$

- Note: In the google talk, if $\theta^{(t)}$ is discrete we can't do weight average. We have to do the density and average over the density.

- For weights, it turns out that a good choice is:

$$w_j = Var^{-1}(\theta|\vec{X}_j)$$

- EX: $X_{ij}, ..., Xijk_j \overset{i.i.d}{\sim} N(\mu, \sigma^2), \qquad \mu \sim N(\mu_0, \sigma_0^2)$
  Chunk to $\vec{X_j} = \{X_{ij_1}, X_{ij_2}, ..., X_{ijk_j}\}$where$\{i_{j1}, ..., i_{jk_j}\}$are indices $\leftarrow$responding to the subject.

$$Modelj : X_{ij}, ..., X_{ijk_j} \overset{i.i.d}{\sim} N(\mu, \sigma^2), \qquad \mu \sim N(\mu_0, \frac{\sigma_0^2}{q_j})$$

$$\mu|\vec{X_j} \sim N(\frac{\frac{q_j\mu_0}{\sigma_0^2} + \frac{\sum x_j}{\sigma^2}}{\frac{q_j\sigma_0^2}{\sigma_0^2} + \frac{k_j}{\sigma^2}}, \frac{1}{\frac{q_j}{\sigma_0^2} + \frac{k_j}{\sigma^2}})$$

Combine: $\mu = \frac{\sum w_j\mu_j}{\sum w_j}, w_j = \frac{q_j}{\sigma_0^2} + \frac{k_j}{\sigma^2}$

$$\rightarrow \sum w_j = \frac{q_j}{\sigma_0^2} + \frac{k_j}{\sigma^2} = \frac{1}{\sigma_0^2} + \frac{n}{sigma^2}$$

$$Var(\mu) = \frac{1}{(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2})^2} \sum_j w_j^2 Var(\mu_j)$$

where $\sum_j w_j^2 Var(\mu_j)$ reduce to$\sum_j w_j$

$$E[\mu] = \frac{\sum_j w_j E[\mu_j]}{\sum w_j} = \frac{\sum_j \frac{q_j\mu_0}{\sigma_0^2} + \sum\sum x_j}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}} = \frac{\frac{\mu_0}{\sigma_0^2} + \frac{\sum x}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}}$$

where $\sum x$ in the last term contains the full sum of data points

$$\mu \sim N(E[\mu], Var[\mu])$$

- Comments:

1. It's not gonna work if the data is not normal but suppose n is large and sub data size $= \{d_1, ...d_s\}$. If the $d_{j's}$ are large what happens to $p(\theta|\vec{X_j})$?
   $\rightarrow$ Become normal (Posterior converge to normal: CLT result)

2. If the $d_{j's}$ are large. They are approximately normal $\rightarrow$ consensus MC will still work well!!