STATS 202A, Statistical Programming Code Memo

Juan Piao

This memo is prepared based on Prof.Wu's lecture notes.

1. Homework1–Sampling

   (a) Random number generators.

      i. Uniform[0, 1], using the linear congruential method.
      The linear congruential method is to generate independent random numbers from the uniform distribution over 0, 1, ..., $M-1$.
      We start from a seed X_0 , and iterate
      $X_{t+1} = aX_t + b$ mod M
      where a, b and M are carefully chosen integers.
      Let $U_t = X_t/M$, then U_t is a sequence of independent random numbers following the uniform distribution over [0, 1].

      ii. Exponential(1), using the inversion method. We first generate $U \sim \mathcal{N}(0, 1)$, and then we let $X = F^{-1}(U) = -log(U)$ to generate $X \sim Exp(1)$.

      iii. Normal(0, 1), using the Polar method.
      First, we can generate $(U1, U2) \sim Uniform[0, 1]$ independently.
      Then we let $\theta = 2U1$,and $R = \sqrt{log(1 - U2)}$.
      Then $X = Rcos\theta$,and $Y = Rsin\theta$ are two iid copies of Normal random variables.

   (b) Monte Carlo computation of pi.
   Suppose (U, V) ∼Uniform[0, 1] independently. Then $P(U^2+V^2 <= 1) = pi/4$.
   First, generate (u_t, v_t) from unit square $[0, 1]^2$, and compute the frequency that the points fall below $u^2+v^2 = 1$. Also used Monte Carlo method to compute the volume of d-dimensional unit ball, for d = 5 and 10.

2. Homework2–Sampling2

   (a) Use the Metropolis algorithm to sample from $X \sim \mathcal{N}(0, 1)$.
   The transition probability of the Metropolis algorithm is

$$M(x, y) = B(x, y)min(1, \frac{\pi(y)}{\pi(x)})$$

   .

(b) Use the Gibbs sampler to sample from Bivariate normal with correlation ρ.
Consider the bivariate normal distribution: $X \sim N(0,1), [Y|X = x] \sim N(\rho x, 1-\rho^2)$. Since the joint density of X and Y is symmetric in (x, y), We can sample from f(x, y) using the Gibbs sampler. We start from (X_0, Y_0). Let (X_t, Y_t) be the values of (X, Y) at iteration t, then at iteration t + 1, we sample $X_{t+1} \sim N(\rho Y_t, 1-\rho^2)$, and then sample $Y_{t+1} \sim N(\rho X_{t+1}, 1-\rho^2)$.

3. Homework3–Sweep

   (a) Sweep operator. The sweep operator is a convenient tool for linear regression. Sweep operator is a space saving version of Gauss-Jordan, where we do not record the identity matrix in sweep. We construct a matrix Z = [X, Y], and let $A = Z^T Z$. Then

$$SWP[1:p]A = \begin{bmatrix} -\frac{var(\hat{\beta})}{\sigma^2} & \hat{\beta} \\ \hat{\beta}^T & RSS \end{bmatrix}$$

   where RSS is the residual sum of squares.

   (b) Linear model using the Sweep operator as the engine. The inputs are (X, Y), The outputs are $\hat{\beta}$.

4. Homework4–QR

   (a) QR decomposition. QR decomposition is to decompose a matrix X into a product X = QR where Q is an orthogonal matrix and R is an upper triangular matrix. The input is a matrix A. The outputs are Q and R.

   (b) Linear model based on QR. The inputs are (X, Y), The outputs are $\hat{\beta}$ and $|e|^2$.

5. Homework5–Eigen and PCA

   (a) Eigen decomposition based on QR. we use power method to compute the eigenvectors and eigenvalues of a symmetric matrix Σ.
   Given p vectors $V = (V_1, ..., V_p)$
   We iterate through the following two steps
   1)Compute $\tilde{V}$ orthogonalized V
   2)Update $V = \Sigma\tilde{V}$.

   (b) PCA based on eigen decomposition. Consider the nxp data matrix X.

$$X^T X = Q\Lambda Q^T$$

We can use the power method to compute Q and Λ. We can choose d < p, and represent $Xi \approx \sum_{k=1}^{k=d} z_{ik} q_k$. This is principal component analysis for dimension reduction.

6. Homework6–Logistic,Boosting

   (a) Logistic regression, based on QR code for linear regression. The inputs are (X, Y), The outputs are $\hat{\beta}$

   (b) Extreme gradient boosting, using one layer tree as base function.

   (c) Adaboost, using one layer tree as base classifier.

7. Homework7– XGBoost

   (a) Perform 5-fold validation for cancer data. output mean, std of the 5-fold validation accuracy.

   (b) Perform Grid Search for parameter max_depth and min_child_weight . Output is the grid search mean test score for each parameter combination.

8. Homework8– SVM

   (a) kernel SVM based on Gradient. Print accuracy through each iteration

   (b) kernel SVM based on Dual coordinate ascent. Print accuracy through each iteration
   The idea of support vector machine (SVM) is to find the β, so that
   (1) for positive examples $y_i = +, X_i^T \beta \geq 1$,
   (2) for negative examples $y_i = -, X_i^T \beta \leq 1$,
   The decision boundary is decided by the training examples that lies on the margin. Those are the support vectors.

9. Final Project–Lasso, NN

   (a) Lasso solution path based on coordinate descent and epsilon-boosting. The Lasso regression estimate β by

   $$\hat{\beta} = arg \min_{\beta} [\frac{1}{2} \|Y - X\beta\|_{l2}^2 + \lambda \|\beta\|_{l1}]$$

   (b) Implement Neural Network using class.