Estimation Candidates

1. How do you test whether a given set of real-valued observations are drawn from a Gaussian distribution?

   I use P-value approach. we already know the target Gaussian distribution, noted as $\mathcal{N}\sim(\mu_0, \sigma_0^2)$, and a given set of real-valued observations as $(x_1, x_2 \dots x_n)$.

   For set of observations, note its mean is μ, its standard deviation is σ, and number of observations is n.

   1) Test for mean. (Student's t-test)

   $H_0: \mu = \mu_0$, $H_1: \mu \neq \mu_0$, and confidence level is 95%.

   We can derive that confidence interval is

   $$U = \left(\mu_0 - 1.96\frac{\sigma}{\sqrt{n}}, \mu_0 + 1.96\frac{\sigma}{\sqrt{n}} \right)$$

   if μ is in confidence interval U, that we can say H_0 is true that its mean is μ_0. Otherwise, this set of observations are not from this Gaussian distribution.

   2) Test for standard deviation. (Chi-squared test)

   $H_0: \sigma = \sigma_0$, $H_1: \sigma \neq \sigma_0$, and confidence level is 95%.

   Note that

   $$s^2 = \frac{\sum_{i=1}^{n}(x_i - \mu_0)^2}{n-1}$$

   Test statistics is that

   $$\chi^2 = \frac{(n-1) \cdot s^2}{\sigma_0^2}$$

   We know that when confidence level is 95%, confidence interval for χ^2 is U'

   $$U' = (12.40\,, 39.36\,)$$

   If χ^2 is in the confidence interval U', that we can say H_0 is true that its standard deviation is σ_0. Otherwise, this set of observations are not from this Gaussian distribution.

2. Given a streaming time series of real-valued, scalar observations, describe an online algorithm for determining a linear Gaussian model of these observations as a function of time step.

Notations: at time step n, we have mean μ_n, standard deviation σ_n and input at step n is x_n.

$$\mu_n = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\sigma_n^2 = \frac{\sum_{i=1}^{n}(x_i - \mu_n)^2}{n-1}$$

Therefore, at time step (n-1), we have

$$\mu_{n-1} = \frac{\sum_{i=1}^{n-1} x_i}{n-1}$$

$$\sigma_{n-1}^2 = \frac{\sum_{i=1}^{n-1}(x_i - \mu_{n-1})^2}{n-2}$$

For mean at time step n, we have

$$\mu_n = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\mu_n = \frac{n-1}{n} \cdot \frac{\sum_{i=1}^{n-1} x_i}{n-1} + \frac{1}{n} x_n$$

$$\mu_n = \frac{n-1}{n} \mu_{n-1} + \frac{1}{n} x_n$$

For standard deviation at step n, we have

$$\sigma_n^2 = \frac{\sum_{i=1}^{n}(x_i - \mu_n)^2}{n-1}$$

$$\sigma_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} ((x_i - \mu_{n-1}) + (\mu_{n-1} - \mu_n))^2$$

$$\sigma_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu_{n-1})^2 + 2(x_i - \mu_{n-1})(\mu_{n-1} - \mu_n) + (\mu_{n-1} - \mu_n)^2$$

$$\sigma_n^2 = \frac{1}{n-1} \left(\sum_{i=1}^{n-1} (x_i - \mu_{n-1})^2 + (x_n - \mu_{n-1})^2 + \sum_{i=1}^{n} 2(x_i - \mu_{n-1})(\mu_{n-1} - \mu_n) + \sum_{i=1}^{n} (\mu_{n-1} - \mu_n)^2 \right)$$

$$\sigma_n^2 = \frac{1}{n-1} ((n-2)\sigma_{n-1}^2 + n^2(\mu_n - \mu_{n-1})^2 + 2(n \cdot \mu_n - n \cdot \mu_{n-1})(\mu_{n-1} - \mu_n) + n(\mu_{n-1} - \mu_n)^2)$$

$$\sigma_n^2 = \frac{1}{n-1} \left((n-2)\sigma_{n-1}^2 + n^2(\mu_n - \mu_{n-1})^2 - 2n(\mu_{n-1} - \mu_n)^2 + n(\mu_{n-1} - \mu_n)^2 \right)$$

$$\sigma_n^2 = \frac{1}{n-1} \left((n-2)\sigma_{n-1}^2 + n(n-1)(\mu_n - \mu_{n-1})^2 \right)$$

$$\sigma_n^2 = \frac{n-2}{n-1} \sigma_{n-1}^2 + n(\mu_n - \mu_{n-1})^2$$

Therefore, we have

$$\mu_n = \frac{n-1}{n}\mu_{n-1} + \frac{1}{n}x_n$$

$$\sigma_n^2 = \frac{n-2}{n-1}\sigma_{n-1}^2 + n(\mu_n - \mu_{n-1})^2$$

We can write the algorithm in Java as following:

```
public static void main(String[] args){
        double u = 0, r = 0, n = 0;
        Scanner scan = new Scanner(System.in);
        double x1 = Double.parseDouble(scan.nextLine());
        double x2 = Double.parseDouble(scan.nextLine());
        n=2;
        u = (x1+x2)/2;
        r = Math.sqrt((x1-u)*(x1-u)+(x2-u)*(x2-u))
        while(in.hasNextLine()){
                //time step
                n+=1;
                double x_n = Double.parseDouble(scan.nextLine());
                double u_n = (n-1)/n*u+1/n*x_n;
                double r_n = Math.sqrt((n-2)/(n-1)*r*r+n*(u-u_n)*(u-u_n));
                u = u_n;
                r = r_n;
        }
        System.out.println("The mean of this linear Gaussian model is "+u);
        System.out.println("The standard deviation of this linear Gaussian model is "+r);
}
```