Tarea 3: Aplicación de Prueba Chi-Cuadrado

Instituto Tecnológico Autonómo de México - ITAM
Maestría en Ciencias en Computación
Karen L. Poblete 116452
Complejidad y computabilidad COM-31704

20 de septiembre de 2014

Se implementó un programa en *Java*, que realiza las siguientes funciones:

1. Dado el número de cuantiles en que se desea dividir una muestra de variables normales se calculan los valores de los limites de dichos cuantiles.

2. Dada una función de probabilidad se generan n muestras cada una de m variables aleatorias. Se calcula $\overline{X}$ arriba de cada muestra, además calcular σ y μ del conjunto de muestras.

3. Aplicar la prueba Chi-cuadrado a los promedios obtenidos de las muestras del punto anterior para comprobar la similitud de la distribución de dichos valores con la distribución normal con $\sigma = 1$ y $\mu = 0$. Encontrar el número mínimo de muestras que se requieren para obtener valores de $\chi = 4$. Se tiene la restricción de que por lo menos existan 5 eventos en cada cuantil, de lo contrario la prueba no es válida.

El código abarca los tres puntos anteriores con la restricción de que las funciones ingresadas para generar las m muestras de variables aleatorias deben ser previamente analizadas por el usuario y transformadas en la función que el sistema requiere para su correcto funcionamiento. En la sección de generación de variables aleatorias de acuerdo a una distribución dada, se explica a profundidad dicho proceso.

Se adjunta la clase en JAVA llamada PruebaChi.java a éste reporte. El programa calcula los valores de μ y σ del grupo de m muestras de n variables aleatorias de la función ejemplo $x^2/4$ evaluada de [0,2]. Se calcula el sesgo entre μ y σ contra los valores calculados por medio de métodos matemáticos para la media y la desviación estándar de la población real. El programa regresa la tabla de frecuencias en los cuantiles de la prueba Chi-cuadrada junto con el valor χ de la prueba.

1. Cuantiles

Los cuantiles son los valores de una función de distribución que la dividen en intervalos iguales [1]. Por ejemplo, los *cuartiles* que dividen la distribución en 4 partes iguales, es decir, en cada cuartil se acumula el 25 % del área que se forma debajo de la curva de la función de probabilidad acumulada. Hasta el primer cuartil se acumula el 25 % de la probabilidad, en el segundo el 50 % y en el tercero el 75 %. De manera analóga se definen los deciles, es decir, un cuantil por cada 10 % del área bajo la curva de la función de probabilidad acumulada.

Existen diferentes métodos para calcular los valores que limitan cada cuantil. En la clase adjunta se agrega un método para devolver éstos valores. El número de cuantiles puede variar. En el ejemplo de usan los deciles, pero es posible utilizar diferente número de cuantiles libremente. Para realizar la prueba χ^2, objetivo de dicho trabajo, debemos obtener los valores de los limites de los cuantiles de la distribución normal con media 0 y desviación estándar 1. Para hacer esto se deben de generar variables aleatorias suficientes con distribución normal. A continuación se enlistan los pasos para calcular dichos cuantiles.

Prob. Acum.	Decil Experiental	Decil Referencia
0.1	-1.2858507199155866	-1.28
0.2	-0.8344599160385702	-0.84
0.3	-0.5095425487600069	-0.52
0.4	-0.24096137391684813	-0.25
0.5	9.428888527320822E-4	0
0.6	0.2552393573088336	0.26
0.7	0.538252423345579	0.52
0.8	0.8558173062601417	0.85
0.9	1.280967183792776	1.29

Cuadro 1: Deciles experimentales y de referencia para distribución normal

1. Generar N variables aleatorias suficientes con distribución normal con $\mu = 0$ y $\sigma^2 = 1$. En el código se usan 6000. A mayor cantidad de variables aleatorias generadas se adquiere mayor presición. El método para la generación de números normales se describe posteriormente.

2. Ordenar las N variables aleatorias normales desde la de menor valor a la más grande. En el código adjunto se utiliza el método *Bubble Sort*, por ser sencillo de implementar aunque en casos grandes puede tomar mayor tiempo.

3. Se calcula la posición de la variable en la lista de variables aleatorias ordenadas que éste en el 10 %, 20 %, y así hasta el 90 % de la lista. En el algoritmo anexo se guardan también el primer y último valor de las variables ordenadas para usarlos posteriormente en la implementación de la prueba χ^2. Se divide N entre γ y se multiplica por la variable entera i, donde i va de 1 a N-1 y γ define el número de cuantiles que se requieren: $(N * i)/\gamma$

En el ejemplo se usan deciles, es decir, se divide la distribución de probabilidad en 10. Corriendo el código, se obtuvieron los valores que se presentan en la tabla 1. En la misma tabla se muestran los valores tomados de [4]. Con lo que queda comprobado que el algoritmo es correcto 1.

```
public static double[] getNcilesValue(double[] vals, int nciles) {
    double[] valores = new double[nciles + 1];
    valores[0] = vals[0];
    for (int i = 1; i < nciles; i++) {
        double indexD = (vals.length * (double) i) / nciles;
        int index = new Integer((int) indexD);
        valores[i] = vals[index];
    }
    valores[nciles] = vals[vals.length - 1];
    return valores;
}
```

Figura 1: Código para calcular los valores de los deciles.

Generación de variables aleatorias con distribución normal

Java cuenta con una función generadora de valores aleatorios entre 0 y 1 con distribución uniforme, *Random.NextDouble*. Dicha función 2 se puede aprovechar para producir variables aleatorias con diferentes distribuciones como la distribución normal por medio de la siguiente transformación:

```java
public static double[] generaNormales(int raiz, int num, double mu, double sigma) {
    double[] valores = new double[num];
    Random aleatorio = new Random(raiz);
    double y;
    for (int j = 0; j < num; j++) {
        y = 0.0;
        for (int i = 0; i < 12; i++) {
            y = y + aleatorio.nextDouble();
        }
        valores[j] = sigma * (y - 6) + mu;
    }
    return valores;
}
```

Figura 2: Código de generación de variables normales.

$$X := \sum_{i=1}^{1} 2U_i - 6, for U_i \sim U[0,1) \tag{1}$$

2. Generación de variables aleatorias de una función de probabilidad dada

En diferentes áreas del conocimiento y ciencias es necesario realizar simulaciones de fenómenos que parten de diferentes distribuciones fuera de las comunes. Existen diversos métodos para generar variables aleatorias con distribuciones específicas por ejemplo: método de la función inversa, montecarlo y métodos que implican la función cuantil de la distribución entre otros. El método utilizado en el código del anexo usa la función de probabilidad de la distribución objetivo. Ésta función debe de cubrir ciertas restricciones para poder ser una función de distribución de probabilidad válida: la integral de la función dentro del intervalo definido debe ser 1, no se permiten valores negativos para las probabilidades y los valores del rango de la función se encuentran dentro de 0 a 1.

En el código anexo no se implementó un interprete de ecuaciones matemáticas, por lo que el usuario debe de hacer el ajuste de la función objetivo manualmente, es decir, se debe encargar de que las tres condiciones anteriores se cumplan 3.

En el código anexo se utiliza la fdp $y = x/2$ cuya cdf es $F_X(x) = x^2/4$ en el intevalo $[0, 2]$. La función $F_X(x)$ es equivalente a $P[X \leq x]$ con x de $[0, 2]$. Ésta función debe de ser discretizada para poder ser manejada. En el código ejemplo se discretiza a 2000 puntos, es decir, se define un valor en intervalos de 0,001. Cada uno de los puntos generados es evaluado en la función $F_X(x)$ y los valores resultantes son la probabilidad que se acumula hasta ese valor.

Ahora bien, usando el generador de variables uniformes de java en el intervalo de $[0, 1)$ se produjeron variables uniformes y se buscaron cada uno de los valores en la tabla de valores discretos de $F_X(x)$. La busqueda termina cuando se encuentra el intervalo al que pertenece la variable uniforme generada y se regresa el valor $F_X(x)^-1$. Es así como se garantiza que las variables que se generan cuadren con esa distribución objetivo.

```java
//Funcion que calcula la frecuencia de las variables en el decil correcto
public static double[] getValoresDeCuantil(int nciles,
        double ncilesValNorm[], double medias[], double muMues,
        double sigmaMues) {
    double valoresRes[] = new double[nciles];
    //sumar las apariciones por cuantil, limites de cuantil
    for (int i = 0; i < medias.length; i++) {
        int j = 0;
        while (j < ncilesValNorm.length - 1) {
            double aux = (medias[i] - muMues) / sigmaMues;
            if (ncilesValNorm[j] < aux && ncilesValNorm[j + 1] > aux) {
                valoresRes[j] = valoresRes[j] + 1;
                j = ncilesValNorm.length;
            }
            if (aux < ncilesValNorm[0]) {
                valoresRes[0] = valoresRes[0] + 1;
                j = ncilesValNorm.length;
            }
            if (aux > ncilesValNorm[ncilesValNorm.length-1]) {
                valoresRes[nciles-1] = valoresRes[nciles-1] + 1;
                j = ncilesValNorm.length;
            }
            j++;
        }
    }
    return valoresRes;
}
```

Figura 3: Clasificación de variables en los cuantiles.

Teniendo listo el generador de variables aleatorias de dicha distribución se generan m muestras de n variables cada una. Para cada una de las muestra se calcula su promedio. Sobre el conjunto de promedios de muestra se calcula $\overline{X}$ y *sigma*.

$$\overline{X} = \frac{\sum_{i=0}^{N-1} X_i}{N} \tag{2}$$

$$\sigma = \sqrt{\frac{\sum_{i=0}^{N-1} X_i - \overline{X}}{N-1}} \tag{3}$$

El valor esperado $E[X]$ y desviación estándar $STD[X]$ teóricas para $f_X(x) = x/2$ se presentan a continuación, las formulas usadas fueron tomadas de [1]. Cuando se hacen las muestras de n variables con ésta distribución, sus promedios rondan cerca de $E[X]$.

$$\begin{aligned} E[X] &= \int_0^2 x\, f_X(x)\, dx \\ &= \int_0^2 x\, \tfrac{x}{2}\, dx \\ &= \int_0^2 \tfrac{x^2}{2}\, dx \\ &= \tfrac{x^3}{6}\Big]_0^2 = 1.\overline{33} \end{aligned} \tag{4}$$

$$\begin{aligned} VAR[X] &= E[X^2] - E[X]^2 \\ &= \int_0^2 \tfrac{x^3}{2}\, dx - (1.\overline{33})^2 \\ &= \tfrac{x^4}{8}\Big]_0^2 - 1.\overline{77} \\ &= 2 - 1.\overline{77} = 0.\overline{22} \end{aligned} \tag{5}$$

En el código se genera un arreglo con todos los valores de cada una de las medias μ obtenidas de cada muestra de n variables con ésta distribución.

Intervalo	Frec. Esperada	Frec. Experimental
$(\infty, -1{,}28]$	40	45.0
$(-1{,}28, -0{,}84]$	40	35.0
$(-0{,}84, -0{,}52]$	40	44.0
$(-0{,}52, -0{,}25]$	40	34.0
$(-0{,}25, 0]$	40	40.0
$(0, 0{,}26]$	40	46.0
$(0{,}26, 0{,}52]$	40	35.0
$(0{,}52, 0{,}85]$	40	31.0
$(0{,}85, 1{,}29]$	40	48.0

Cuadro 2: Deciles experimentales y de referencia para distribución normal

3. Prueba Chi-cuadrada

En el área de las estadísticas existen diversos métodos para comprobar y rechazar hipótesis acerca de la distribución de un conjunto de variables aleatorias cuya distribución original es desconocida. Como es bien sabido las distribuciones más conocidas son: la uniforme, normal, binomial, exponencial, betta, gamma, etc. Es de gran utilidad poder descartar si la población muestral generada con la función desconocida tiene alguna similitud con las anteriores. Las pruebas más utilizadas son: chi-cuadrada y la prueba de Kolmogorov-Smirnov [3].

La prueba Chi-cuadrada es una prueba de ajuste a una distribución conocida de un conjunto de muestras cuya distribución se desconoce. En el código anexo se utiliza esta prueba para conocer la similitud de una muestra de promedios con una distribución de probabilidad dada por el usuario a la distribución normal.

Tomando en cuenta que se realizó previamente la obtención de m muestras de variables aleatorias conforme a la distribución definida por el usuario, y que de éstas se calculó el promedio de cada una. Además se hizo el cálculo de μ y σ para el conjunto de promedios como se explicó en el punto anterior.

Teniendo los valores de los limites de los cuantiles obtenidos de una distribución normal con media 0 y varianza 1, además sabemos que el n % del area de la distribución se encuentra entre cada una de ellas, en el caso de deciles, entre cada decil, se encuentra el 10 %. Por lo que hay el mismo número de observaciones en cada decil: $N/10$.

Sabiendo los limites de cada decil de la distribución normal queremos asignar cada una de los promedios a un intervalo de éstos deciles, es por eso que los valores de los promedios deben de normalizarse. Se quiere contar la frecuencia de los valores en cada intervalo, para compararlo con las frecuencias de los mismos intervalos en la normal.

Ésta es la fórmula que se utiliza para la normalización de los promedios. μ y σ son los valores obtenidos experimentalmente como se señaló con anterioridad.

$$\overline{Y_i} = \frac{\overline{X_i} - \mu}{\sigma} \tag{6}$$

Al terminar por ejemplo se obtienen datos como los presentados en la tabla 2. El número de promedio utilizados es 400, es decir, $N = 400$.

Y se calcula Chi2, con :

$$\chi^2 = \sum_{i=1}^{n} \frac{(FE_i - FO_i)^2}{FE_i} \tag{7}$$

FE_i es la frecuencia esperada en el decil i y FO_i es la frecuencia observada en el decil i de la muestra de promedios. Queremos encontrar el valor de n número de muestras que nos da un valor de χ menor a por ejemplo 4, un limite definido por el usuario. El programa itera hasta encontrar la solución si la hay.

4. Análisis de Datos

El valor de $\chi \leq 4$, es un valor de rechazo de la prueba de ajuste de la distribución desconocida contra la distribución normal. Éste valor puede variar de acuerdo al ajusto que se desee probar [1]. Cuando el valor es menor a el valor establecido de χ no se puede rechazar la normalidad del conjunte de variables aleatorios a las que se le aplican la prueba.

5. Conclusiones

La prueba de Chi-cuadrada es una herramienta útil para descartar que un conjunto de muestras se comporta o no de cierta distribución, en este caso de manera normal. Existen diferentes maneras de generar variables aleatorias a partir de distribuciones según se desee modelar fenómenos, una manera sencilla de hacerlo es por medio de la implementada en ésta tarea.

Referencias

[1] A. Leon-Garcia, *Probability, Statistics and Random Processes for Electrical Enginnering*, Third Edition, Pearson 2008.

[2] R. Seydel, *Tools for Computational Finance*, 2012, Springer, ISBN:978-1-4471-2992-9.

[3] E. Martinez, *Apuntes de Inferencia Computacional*, 2013.

[4] *Tablas y Formulario para el curso de estadística II*, División Académica de Actuaría, Estadística y Matemáticas, Departamento de Estadística ITAM.