Autor: José Pedro García Sabater

Fecha: 28/12/2012

Palabras Clave: Ley de Zipf, Análisis ABC, Generación de Ejemplares

Título

Generación de ejemplares con Distribución irregular.

Motivación

Cuando se generan ejemplares para experimentos, se establece que la presencia de los productos (u otros elementos) oscila en un cierto rango de una manera uniforme En algunos casos concretos eso genera un tipo de distribución P/Q (Producto/Cantidad) que se aleja bastante de la realidad. En general un análisis ABC o análisis de Pareto nos deja claro que no todos los productos tienen la misma presencia.[1],[2]. Para resolver este problema se propone el siguiente procedimiento para asignar cantidades (por ejemplo de demanda o de ventas… a productos.

Desarrollo.

Supongamos que hemos de preparar un ejemplar de un problema para experimentar. El ejemplar tiene que considerar un número de productos y una demanda de esos productos. El modo convencional de preparar ese ejemplar es que para cada producto se le asigna demanda según una función de probabilidad, que en general, se considera uniforme porque es la más fácil de exponer en el paper.

La distribución de la demanda en ese caso se reparte homogéneamente por los productos, unos tienen más presencia y otros menos, pero en la misma cantidad. La realidad no suele ser así, sino que unos pocos productos acumulan casi toda la demanda y otros muchos tienen cantidades residuales.

A continuación se propone un procedimiento sencillo que permite analizar la situación. El procedimiento está basado en la conocida como ley de Zipf (Zipf, 1949). Es una versión particular de las denominadas leyes de potencia[3]. La Ley de Zipf establece que, si ordenamos los elementos de mayor a menor presencia, como para hacer un análisis ABC, la probabilidad del n-simo elemento será $P_n = \frac{1}{n^a}$.

El parámetro *a* marcará que la distribución sea más o menos picuda. Así un *a=0* hace una distribución homogénea donde todos los productos, mientras que una a=2 establece una distribución donde el décimo producto ya sólo tiene una presencia del 1 por mil.

Para valores de *a* superiores a 1 un procedimiento de reparto es el descrito a continuación.

---

[1] Por ejemplo al emular la “popularidad” de un sitio en internet, el reparto de riqueza entre los habitantes del planeta, la población de las ciudades(Soo, 2005). o el número de veces que una determinada palabra aparece en un texto. Más ligado a nuestra actividad se da en la presencia de tipos de coches (importante en la secuenciación de unidades homogéneas en líneas de montaje). Pero también en la presencia de productos en un almacén, o en la selección de referencias (y cantidades en una hoja de peido).

[2] Tengo la impresión (que debería ser demostrada) que los experimentos a diseñar no tienen en cuenta estas distribuciones los resultados obtenidos en la resolución no tienen porqué ajustarse a la realidad posterior.

[3] aplican por ejemplo a los tamaños de las ciudades

Sea la lista que contiene los productos a los que les queremos asignar cantidad. Dicha lista puede estar ordenada según algún criterio u ordenada aleatoriamente. La lista contiene N elementos. La cantidad de unidades asignadas al elemento i se almacena en la variable q(i). Sea T la cantidad total de unidades que deben asignarse.

Suma=0

$$Z = \sum_i \frac{1}{i^a}$$

For i=0 to N-2

$$q(i+2) = redondear\left(\frac{T}{Z*(i+2)^a}\right)$$

suma=suma+q(i+2)

Next i

q(1)=T-suma

El cálculo de Z permite que la suma de las presencias se ajuste a 1, de tal manera que luego la suma total de elementos sea T.

El último de los pasos pretende que la cantidad total se ajuste a la definida. En algunos casos eso puede alterar sustancialmente la cantidad del primero y convertirlo en el segundo de la lista. Pero no creo que eso sea un inconveniente que exija modificar el problema. (De hecho en la realidad los primeros productos nunca cumplen bien la ley de Zipf)

La distribución para 100 productos de un total de 10.000 unidades quedaría así para valores de a de 1, 1'2 y 1'5

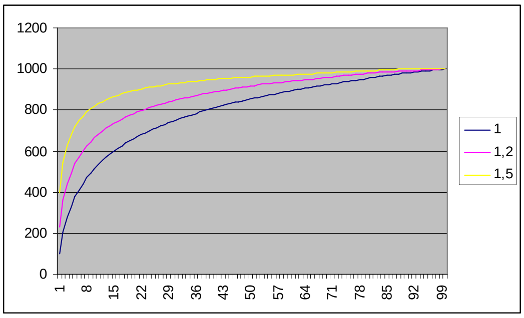

Ilustración 1. Gráfico ABC sin aleatorios.

Modificando el orden de la lista se obtienen diferentes ejemplares del mismo problema (si los ejemplares son diferentes entre sí pero tienen componentes o características relacionadas) los problemas generados serán suficientemente distintos.

También se podría asignar una cierta aleatoriedad a la cantidad q, aunque habría que establecer mecanismos más complejos de control probablemente y probablemente no aporten mucho al análisis del problema.

A continuación establezco una versión que no funciona mal. Le asigna una aleatoriedad del 10% a la cantidad q calculada4.

Suma=0

For i=0 to N-2

$$Z = \sum_i \frac{1}{i^a}$$

$$q(i+2) = redondear\left(\frac{T}{Z}\left(1+\frac{(aleatorio()-0{,}5)}{5}\right)\frac{1}{(i+2)^a}\right)$$

suma=suma+q(N-i)

Next i

q(1)=T-suma

En la gráfica siguiente represento las mismas series de valores. Como se aprecia no hay grandes diferencias entre una gráfica y la otra.

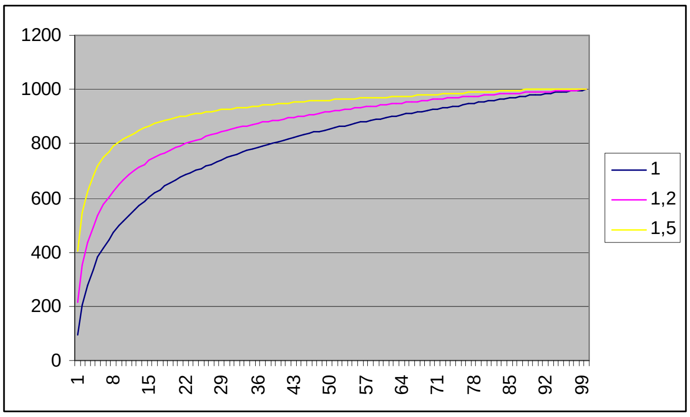

Ilustración 2. Gráfico ABC incorporando un 10% de aleatoriedad

Anexo 1. Código en Java para el caso no aleatorio

```
public static void main(String[] args) {
      double p []=new double[100];
      long x []=new long[100];
      long suma=0;
      double suma2=0;
      int N=100;
      int T=1000;
      double a=1.2;
      for(int i=0;i<=N-1;i++){
            p[i]=  1/(Math.pow(N-i, a));
            suma2=suma2+p[i];
```

[4] Ese 10% está obtenido de trabajos que explican que la distribución del tamaño de las ciudades en el mundo se ajusta a una ley potencial con oscilaciones del 10%(Simon, 1989)

```
			}
		for(int i=0;i<=N-2;i++){
			x[i]=  Math.round(T*p[i]/z);
			suma=suma+x[i];
			}
		x[0]=T-suma;

	}
```

Anexo 2. Código en Java para el caso aleatorio con un desajuste aleatorio de 10% para presencia.

```
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		double p []=new double[100];
		long x []=new long[100];
		long suma=0;
		double =0;
		int N=100;
		int T=1000;
		double a=1;
		for(int i=0;i<=N-1;i++){
			p[i]=  1+((Math.random()-0.5)/5);
			p[i]=1/(Math.pow(N-i, a))*p[i];
			z=z+p[i];
			}
		for(int i=0;i<=N-2;i++){
			x[i]=  Math.round(T*p[i]/z);
			suma=suma+x[i];
			}
		x[0]=T-suma;
```

References

Simon, H. A. (1989). Los Tamaños de las Cosas,La estadística. Una guía de los desconocido., Alizanza Editorial, Madrid.

Soo, K. T. (2005). Zipf's Law for cities: a cross-country investigation". Regional Science and Urban Economics, Vol. 35, nº. 3, pp. 239-263.

Zipf, G. K. (1949). Human Behaviour and the Principle of Least Effort. , Addison-Wesley, Reading, MA. Addison-Wesley