mui_240a2_pset1_ipynb

October 25, 2015

ECON-240A, Problem Set 1
Preston Mui

In [1]:
```
# Python stuff
import numpy as np
import pandas as pd
import math, random
from scipy.special import comb
from scipy.stats import norm, beta
%matplotlib inline
import matplotlib.pyplot as plt
random.seed(813558889)
```

1 The Binomial Distribution

1. Derive a formula that can be used to calculate the ex ante probability of $Z_N < z$ for any $z \in \{1, 2, \cdots, N\}$:

$$P(Z_N < z) = \sum_{k=0}^{z-1} \binom{n}{k} \theta^k (1-\theta)^{N-k} \tag{1}$$

For any given $k \in 0, 1, \cdots, N$, $P(Z_N = k) = \binom{n}{k}\theta^k(1-\theta)^{N-k}$. Since the support of the binomial distribution is the positive integers from 0 to N, the cumulative distribution function of Z_N is the sum of $\binom{n}{k}\theta^k(1-\theta)^{N-k}$ for $k = 0$ through $k = z$.

2. Provide an expression that can be used to calculate the ex ante probability of the event $\frac{\sqrt{N}(\bar{Y}_N - \theta)}{\sqrt{\theta(1-\theta)}} < c$:

$$P\left(\frac{\sqrt{N}(\bar{Y}_N - \theta)}{\sqrt{\theta(1-\theta)}} < c\right) = P\left(\bar{Y}_N < \theta + \frac{\sqrt{\theta(1-\theta)}c}{\sqrt{N}}\right) \tag{2}$$

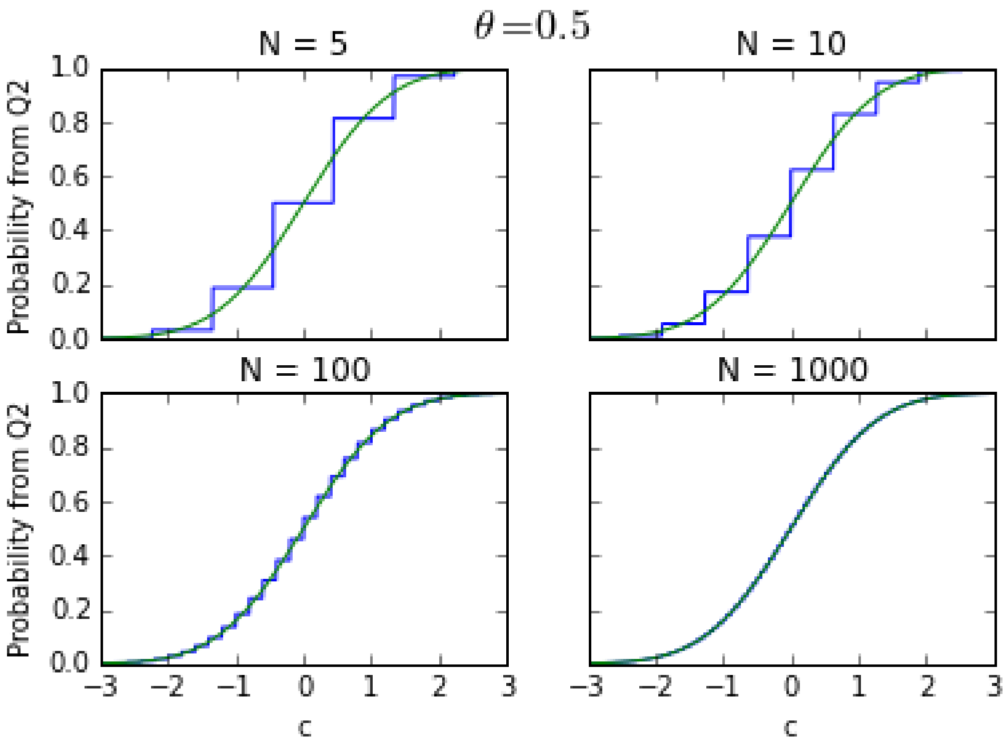
$$= P\left(Z_N < N\theta + \sqrt{N\theta(1-\theta)}c\right) \tag{3}$$

$$= \sum_{k=0}^{\left\lceil N\theta + \sqrt{N\theta(1-\theta)}c \right\rceil} \binom{n}{k} \theta^k (1-\theta)^{N-k} \tag{4}$$

3. Plot $P\left(\frac{\sqrt{N}(\bar{Y}_N - \theta)}{\sqrt{\theta(1-\theta)}} < c\right)$ as a function of c for $N = 5, 10, 100, 1000$ and $\theta = 1/2$., and 4. Plot the normal cdf on top:

```
In [2]: c = np.linspace(-3,3,1000)
        Nvalues = [5,10,100,1000]

        def binomialMeanCdf(N,theta,c):
            cdf = 0
            upperLimit = int(math.ceil(N * theta + np.sqrt(N * theta * (1-theta)) * c))
            for k in range(upperLimit):
                cdf = cdf + comb(N,k) * theta**k * (1-theta)**(N-k)
            return cdf

        CDFvalues50 = np.zeros((len(Nvalues),len(c)))
        for i in range(len(Nvalues)):
            for j in range(len(c)):
                CDFvalues50[i,j] = binomialMeanCdf(Nvalues[i],0.5,c[j])
```

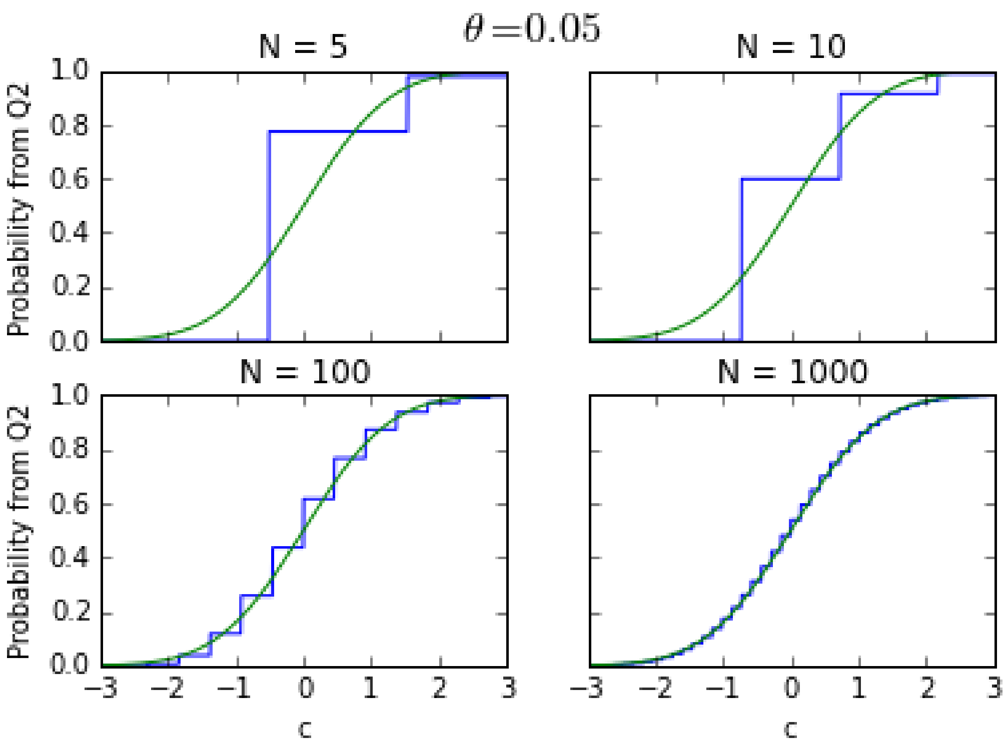
```
In [3]: normCdf = norm.cdf(c)

        plt.close('all')
        f, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, sharex='col', sharey='row')
        ax1.set_ylabel('Probability from Q2')
        ax1.plot(c,CDFvalues50[0,:])
        ax1.plot(c,normCdf)
        ax1.set_title('N = 5')

        ax2.plot(c,CDFvalues50[1,:])
        ax2.plot(c,normCdf)
        ax2.set_title('N = 10')

        ax3.set_xlabel('c')
        ax3.set_ylabel('Probability from Q2')
        ax3.plot(c,CDFvalues50[2,:])
        ax3.plot(c,normCdf)
        ax3.set_title('N = 100')

        ax4.set_xlabel('c')
        ax4.plot(c,CDFvalues50[3,:])
        ax4.plot(c,normCdf)
        ax4.set_title('N = 1000')

        plt.suptitle(r'$\theta = 0.5$', size = 16)
        plt.show()
```



5. Repeat the above with $\theta = 1/20$:

In [4]:
```
CDFvalues05 = np.zeros((len(Nvalues),len(c)))
for i in range(len(Nvalues)):
    for j in range(len(c)):
        CDFvalues05[i,j] = binomialMeanCdf(Nvalues[i],0.05,c[j])

plt.close('all')
f, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, sharex='col', sharey='row')

ax1.set_ylabel('Probability from Q2')
ax1.plot(c,CDFvalues05[0,:])
ax1.plot(c,normCdf)
ax1.set_title('N = 5')

ax2.plot(c,CDFvalues05[1,:])
ax2.plot(c,normCdf)
ax2.set_title('N = 10')

ax3.set_xlabel('c')
ax3.set_ylabel('Probability from Q2')
ax3.plot(c,CDFvalues05[2,:])
ax3.plot(c,normCdf)
ax3.set_title('N = 100')

ax4.set_xlabel('c')
```

```
ax4.plot(c,CDFvalues05[3,:])
ax4.plot(c,normCdf)
ax4.set_title('N = 1000')

plt.suptitle(r'$\theta = 0.05$', size = 16)
plt.show()
```



Comments: With both $\theta = 0.5$ and $\theta = 0.05$, the normal c.d.f is a good approximation to the binomial distribution the larger N is. In particular, it seems to start being a good approximation at $N = 100$. However, for any given N, the normal distribution is a better approximation when $\theta = 0.5$ rather than $\theta = 0.05$.

6. What is the approximate ex ante probability that the interval $\bar{Y}_N \pm \frac{\bar{Y}_N(1-\bar{Y}_N)}{\sqrt{N}} z^{1-\alpha/2}$ contains the θ? The probability that the interval will contain θ is approximately α. In the questions above I found that the distribution of $\frac{\sqrt{N}(\bar{Y}_N-\theta)}{\theta(1-\theta)}$ is approximated by the standard normal distribution; that is,

$$Pr\left(\bar{Y}_N - \theta < \sqrt{\frac{\theta(1-\theta)}{N}} z^{1-\alpha/2}\right) \approx \Phi(z^{1-\alpha/2}) \quad (5)$$

$$\implies Pr\left(\theta \in \bar{Y}_N \pm \sqrt{\frac{\theta(1-\theta)}{N}} z^{1-\alpha/2}\right) \approx \Phi(z^{1-\alpha/2}) - \Phi(-z^{1-\alpha/2}) = \alpha \quad (6)$$

Since $\bar{Y}_N$ tends towards θ when N is reasonable large, substituting $\bar{Y}_N$ for θ in the last expression yields $Pr\left(\theta \in \bar{Y}_N \pm \sqrt{\frac{\bar{Y}_N(1-\bar{Y}_N)}{N}} z^{1-\alpha/2}\right) \approx \alpha$.

8: The Clopper Pearson proposal The proposal will set a lower bound on the likelihood that the observed sample mean differs from the theoretical mean of any $\theta \in [\underline{\theta}, \bar{\theta}]$ at α. The likelihood that the observed mean is below the expected mean for any θ in the confidence interval is bounded by $\frac{\alpha}{2}$ because of the construction of $\bar{\theta}$. Likewise, the likelihood that the observed mean is above the expected mean for any θ in the confidence interval is bounded by $\frac{\alpha}{2}$ by the construction of $\underline{\theta}$.

9. Argue that $F_B^{-1}(\frac{\alpha}{2}; Z_N, N - Z_N + 1) < \theta < F_B^{-1}(\frac{1-\alpha}{2}; Z_N + 1, N - Z_N)$ closely approximates Clopper's interval: The lower bound of this interval is $F_B^{-1}(\frac{\alpha}{2}; Z_N, N - Z_N + 1)$ which is θ such that

$$\begin{aligned}
\int_0^t f_B(\theta)d\theta &= \frac{\alpha}{2} \\
\sum_{i=Z_N}^{N} \binom{N}{i} t^i(1-t)^{N-i} &= \frac{\alpha}{2} \\
P(\hat{Z}_N \geq Z_N) &= \frac{\alpha}{2}
\end{aligned}$$

By similar reasoning, the upper bound of this interval is $F_B^{-1}(1 - \frac{\alpha}{2}; Z_N + 1, N - Z_N)$ which is θ such that

$$\begin{aligned}
\int_0^t f_B(\theta)d\theta &= 1 - \frac{\alpha}{2} \\
\sum_{i=Z_N+1}^{N} \binom{N}{i} t^i(1-t)^{N-i} &= 1 - \frac{\alpha}{2} \\
P(\hat{Z}_N \geq Z_N + 1) &= 1 - \frac{\alpha}{2} \\
P(\hat{Z}_N \leq Z_N) &= \frac{\alpha}{2}
\end{aligned}$$

Like Clopper-Pearson, this interval chooses the upper and lower bounds of θ so that the the likelihoods of drawing a $\hat{Z}_N$ lower or higher than the observed Z_N are bounded by $\frac{\alpha}{2}$, so the likelihood of Z_N differing from the true θ for all θ in the confidence interval is bounded by α.

10. Find a confidence interval using Hoeffding's Inequality

$$Pr(|\bar{Y}_N - \theta| > \epsilon) \leq 2\exp\{-2N\epsilon^2\} \tag{7}$$

$$Pr(\theta \notin CI) \leq 2\exp\{-2N\epsilon^2\} \tag{8}$$

$$Pr(\theta \in CI) \geq 1 - (2\exp\{-2N\epsilon^2\}) \tag{9}$$

$$\implies \alpha = (2\exp\{-2N\epsilon^2\}) \tag{10}$$

$$\implies \epsilon = \sqrt{-\frac{\log(\frac{\alpha}{2})}{2N}} \tag{11}$$

11. Generate 1,000 samples of Bernoulli random variables.

```
In [5]: NValues = (5,10,100,1000)
        thetaValues = (0.05, 0.5)
        alphaValues = (0.05, 0.10)

        # Create an array to hold results
        results = pd.DataFrame({"N": range(0), "theta": range(0), "alpha": range(0), "Normal Approx.": 
                                "CP": range(0), "Hoeffding": range(0)})
```

```python
results = results[['N','theta','alpha','Normal Approx.', 'CP', 'Hoeffding']]

# Function to calculate CP CI
def normCI(samples,N,alpha):
    lb = samples - np.sqrt((samples * (1 - samples)) / N) * norm.ppf(1 - alpha/2)
    ub = samples + np.sqrt((samples * (1 - samples)) / N) * norm.ppf(1 - alpha/2)
    return lb, ub

# Function to calcualte Beta Distribution CI
def betaCI(samples,N,alpha):
    nsamples = len(samples)
    lb = np.zeros(nsamples)
    ub = np.zeros(nsamples)
    for i in range(0,nsamples-1):
        lb[i] = beta.ppf(alpha/2,int(N*samples[i]),N - int(N*samples[i]) + 1)
        ub[i] = beta.ppf(1 - alpha/2,int(N*samples[i]) + 1,N - int(N*samples[i]))
    return lb, ub

for N in NValues:
    for theta in thetaValues:
        # Draw 1000 samples
        sample = np.random.binomial(N,theta,size=1000) / float(N)
        for alpha in alphaValues:

            # Normal Appx. CI
            normLB, normUB = normCI(sample,N,alpha)
            normCIOutside = (sum(theta < normLB) + sum(theta > normUB)) / float(1000)

            # Beta CI
            betaLB, betaUB = betaCI(sample,N,alpha)
            betaCIOutside = (sum(theta < betaLB) + sum(theta > betaUB)) / float(1000)

            # Hoeffding CI
            hLB = sample - np.sqrt(-np.log(alpha/2)/(2*N))
            hUB = sample + np.sqrt(-np.log(alpha/2)/(2*N))
            hCIOutside = (sum(theta < hLB) + sum(theta > hUB)) / float(1000)

            results = results.append({"N": N, "theta": theta, "alpha": alpha, "Normal Approx.":
                                      "CP": betaCIOutside, "Hoeffding": hCIOutside}, ignore_inde
            # Compute confidence interval from (6)
print("Fraction of samples where theta lies outside the confidence interval")
print(results)
```

```
Fraction of samples where theta lies outside the confidence interval
      N  theta  alpha  Normal Approx.     CP  Hoeffding
0     5   0.05   0.05           0.774  0.020      0.000
1     5   0.05   0.10           0.774  0.020      0.002
2     5   0.50   0.05           0.053  0.001      0.000
3     5   0.50   0.10           0.362  0.054      0.000
4    10   0.05   0.05           0.593  0.014      0.000
5    10   0.05   0.10           0.606  0.014      0.000
6    10   0.50   0.05           0.101  0.017      0.002
7    10   0.50   0.10           0.101  0.017      0.016
8   100   0.05   0.05           0.122  0.021      0.000
```

9	100	0.05	0.10	0.144	0.067	0.000
10	100	0.50	0.05	0.064	0.038	0.005
11	100	0.50	0.10	0.101	0.102	0.009
12	1000	0.05	0.05	0.061	0.045	0.000
13	1000	0.05	0.10	0.097	0.084	0.000
14	1000	0.50	0.05	0.055	0.047	0.004
15	1000	0.50	0.10	0.106	0.092	0.010

Comments on the Table When N is small, the normal approximation approach to constructing a confidence interval is very poor. When $N = 5$ or $N = 5$, the true value of θ is outside the confidence interval far more often than α fraction of the time, and is an especially poor approximation for $\theta = 0.05$. However, when N is large, the normal approximation provides a fairly good confidence interval, as the true value lies outside the confidence interval approximately α fraction of the time.

With the Clopper-Pearson interval, the true value of θ lies outside the confidence interval less than α fraction of time time (that is, α is an upper bound on $\theta \notin CI$. The Hoeffding intervals are much more generous and it is rarely the case that $\theta \notin CI$ using the Hoeffding method.

12. What is the probability that the interval $[0.48, 0.72]$ **contains** θ**?** Without some prior on the distribution of θ, I cannot really say anything about the probability that the interval contains θ. The previous section only dealt with likelihoods (the ex ante probabilities) of the observed data given a particular θ. Luckily, the next section deals with priors. What a coincidence.