UNIVERSITÉ PARIS - CERGY

CY Tech. Département Mathématiques

Option INGÉNIERIE FINANCIÈRE. Option ACTUARIAT.

MODEL CALIBRATION AND SIMULATION

TP1 Calibration de volatilité implicite dans le modèle de Black et Scholes. Smile de volatilité

Partie I

$\sigma^{implicite}(K,T)$ est une volatilité qui introduite dans la formule de BS donne comme prix celui du Call observé sur le marché.

Calibrer signifie de trouver $\sigma^{implicite}$ telle que

$$V^{BS}(S_0, t_0; K_i, T_i, r, \sigma^{implicite}) = V_i^{marche}(T_i, K_i)$$

A chaque couple (T_i, K_i) correspond une volatilité implicite $\sigma_i^{implicite}$

Pour calculer $\sigma_i^{implicite}$ on applique l'algorithme de Newton: on cherche le zero d'une fonction

$$F = V^{BS}(T_i, K_i, \sigma_i^{implicite}) - V_i^{marche}(T_i, K_i)$$

On simplifie les notations:

$$F(\sigma) = V^{BS}(\sigma) - V^{marche}$$

$V^{BS}(\sigma)$ est le prix le l'option Call en $t = 0$.

$$V^{BS}(S_0, 0) = S_0 N(d_1(S_0, 0)) - Ke^{-rT} N(d_2(S_0, 0))$$

$$d_1(S_0, 0) = \frac{ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2(S_0, 0) = \frac{ln(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}}$$

L'algorithme de Newton est iteratif:

$$\sigma_{n+1} = \sigma_n - \frac{F(\sigma_n)}{F'(\sigma_n)}$$

$$F'(\sigma) = \frac{\partial V^{BS}}{\partial \sigma} = S_0 \sqrt{\frac{T}{2\pi}} e^{\frac{-d_1^2}{2}}$$

Avant de commencer le calcul, vérifiez que le prix tombe bien dans lintervalle défini par les contraintes d'arbitrage:

$$max(S_0 - Ke^{-rT}, 0) < V^{marche} < S_0.$$

Pour le point du départ utiliser

$$\sigma_0 = \sqrt{2|\frac{ln(S_0/K) + rT}{T}|}$$

Pour chaque strike K_i calculer la volatilité implicite $\sigma_i^{implicite}$.

Utiliser les donnes pour les options cotées en London Internetional Financial Futures and Option Exchange (LIFFE) le 22 Aout 2001.

i	1	2	3	4	5	6	7	8
Strike K_i	5125	5225	5325	5425	5525	5625	5725	5825
Prix d'option	475	405	340	280.5	226	179.5	139	105

Utiliser aussi

$$S_0 = 5430.3, \quad T = 4/12, \quad r = 0.05$$

Partie II. Prise en compte des dividendes. Base de données SP-INDEX

Le calcul de la volatilité implicite des options est modifié par le paiement des dividendes par l'actif sous-jacent. Pour les indices on utilise généralement l'approximation de taux de dividende continu,qui conduit à une formule de Black-Scholes modifiée

$$V^{BS}(S_0, t) = S_0 e^{-q(T-t)} N(d_1(S_0, t)) - Ke^{-r\,(T-t)} N(d_2(S_0, t))$$
$$d_1(S_0, t) = \frac{ln(S_0 e^{-q(T-t)}/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$$
$$d_2(S_0, t) = \frac{ln(S_0 e^{-q(T-t)}/K) + (r - \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$$

autrement dit, nous avons remplacé S_0 par $S_0 e^{-q(T-t)}$ partout dans les formules.

La relation de parité Call-Put ($t = 0$) est également modifiée:

$$Call(S_0, T, K) - Put(S_0, T, K) = S_0 e^{-qT} - Ke^{-rT},$$

soit

$$S_0 e^{-qT} = Call(S_0, T, K) - Put(S_0, T, K) + Ke^{-rT}$$

Pour le fichier des données fournies, calculer les prix des Call et Put comme les prix mid-quote $(Bid + Ask)/2$.

1) Pour chaque maturité $T_j \quad (1 \le j \le 8)$ estimer $\hat{S}_j$.

$$\hat{S}_j = \sum_{i=1}^{N_j} (Call(T_j, K_i) - Put(T_j, K_i) + e^{-rT_j} K_i)/N_j$$

où N_j est le nombre d'options avec cette maturité.

2) Tracer le logarithme de $\hat{S}_j$ en fonction de T_j

De la relation de parité on sait que

$$ln(\hat{S}_j) = lnS_0 - qT_j$$

Donc les points expérimentales $\quad (ln(\hat{S}_j), T_j) \quad$ doivent suivre une droite.

3) En utilisant les couples connus $(ln\hat{S}_j, T_j)$ on souhaite ajuster les données à la courbe de la forme:

$$f(T, \beta) = \beta^1 T + \beta^2$$

L'estimation par moindres carrés porte sur les paramètres β^1 et β^2.

Nous allons chercher β^1 et β^2 en minimisant la somme des carrés des résidus:

$$min_{\beta^1, \beta^2} \sum_j (r_j)^2, \qquad r_j = ln\, \hat{S}_j - \beta^1 T_j - \beta^2$$

4) Appliquer la regression linéaire pour estimer β_1 et β_2 ou calculer leur valeurs exacte en resonvant théoriquement le problème d'optimisation.

5) Calculer le taux de dividende $q = -\beta_1$ et le prix initiale du stock $S_0 = exp(\beta_2)$.

Montrer que

$$S_0 = 1260.36, q = 0.0217$$

6) Tracer le smile de volatilité implicite pour les CALL et pour chaque maturité, en utilisant au lieu de S_0 (du mpoblèmes sans dividendes)

$$S_0 = 1260.36 \cdot e^{-qT}$$

Indications pour MatLab et Python

```
global sp_index
load sp_index.txt
A=sp_index;

S_0=1260.36
q=0.0217
L=length(A(:,1));
```

```
T=A(1:L,1);
% il y a des dividendes de taux q=0.0217
S0=S0*exp(-q*T);
M=(A(1:L,5)+A(1:L,6))/2;
K=A(1:L,2);


figure
plot3(K, T, volatilite_implicite,'*');

ou

figure;
scatter3(K, T, volatilite_implicite);

% surface
figure
tri = delaunay(T,K);
trisurf(tri, T, K,volatilite_implicite );
```

Bases de Donnés. Base 1: SP_INDEX. Python.

```
\bigskip
\bigskip

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter
from math import log, exp, sqrt, pi, erf


def repartition(x):
    return(1.0/2*(1+erf(x/sqrt(2))))


def tracer2D_volatilite_implicite_fichier_1():
    #lien donnant l'emplacement du fichier contenant la matrice de valeurs
    link="J:/ING3-MMF-ACTU-CALIBRATION/Python/sp-index.txt" # utilisez votre link
    fichier = np.loadtxt(link)

    #Initialisation des paramètres dont certains à partir de la matrice donnée
    q=0.0217
    eps=0.0001
    t=0
```

```
S_0=1260.36

N = len(fichier)
# Programation identique à Matlab
K=np.zeros(N)
r=np.zeros(N)
T=np.zeros(N)
S0=np.zeros(N)
volatilite_impl=np.zeros(N)
for i in range (N):
    Ca = fichier[i][3]
     S0=S0*exp(-q*T[i])
    ...


volat_nules=np.where(volatilite_impl==0)
K = np.delete(K,volat_nules)
volatilite_impl = np.delete(volatilite_impl,volat_nules)
#On a enlevé les volatilités qu'on a renseignées nulles ainsi que les valeurs de strike corresp
plt.plot(K,volatilite_impl,'*')
#On trace la courbe de la volatilité en fonction de K
plt.title("Smile 2D option Call")
plt.ylabel("volatilité implicite")
plt.xlabel("S : prix de l'actif")
plt.show()


fig=plt.figure()
ax = fig.add_subplot(projection='3d') #Affichage de points en 3d
ax.scatter(K, T, volatilite_impl, c='m')
ax.set_title("Smile volatilité implicite en 3D - Option Call")
ax.set_xlabel('K')
ax.set_ylabel('T')
ax.set_zlabel('Volatilité implicite')
plt.tight_layout()
plt.show()
```

7) Tracer le smile de volatilité implicite pour les PUT et pour chaque maturité, en utilisant S_0 calculée comme une fonction du prix initiale du sous-jacent et le taux des dividendes q:

$$S_0 = 1260.36 \cdot e^{-qT}$$

Avant de tourner algorithme déduiser les contrantes d'arbitrages pour les prix d'un PUT du marché.

Partie III. Bases de Donnés. Base 2: CAC40. MatLab.

```
function[maturite,strike,Vimp]=Smiles_Q_Data()

A = importfile('spx_quotedata.csv',2,2294);

L=length(A(:,1));
M=(A(1:L,5)+A(1:L,6))/2 ;
K=A(1:L,12);
r(1:L)=0.0255;
S0(1:L)=3932.69;
\bigskip

for i=163:340
T(i)=1/365.22; % Maturity
...
```

Partie III. Bases de Donnés. Base 3: Google. MatLab.

```
Data_Google=xlsread('GoogleOrig.xlsx');

r(1:L)=0.06;
T=A(:,1);
S0(1:L)=591.66;
```