Metodi Stocastici per la Finanza

Tiziano Vargiolu
vargiolu@math.unipd.it[1]

[1]Università degli Studi di Padova

Anno Accademico 2011-2012
Lezione 6

Indice

Il metodo bootstrap

Il metodo bootstrap permette di ricavare i prezzi degli zero-coupon bond dai prezzi di altri titoli, che sono tipicamente funzioni lineari di zero-coupon bond, quali obbligazioni o swap.
Nel seguito, analizziamo in dettaglio il bootstrap eseguito da prezzi di obbligazioni, che sono i dati che più comunemente si riescono a trovare gratis su Internet.
Supponiamo di avere al tempo t i prezzi di N obbligazioni $O_1, \ldots, O_N$ a tasso fisso, ciascuna con maturità $T_1 < \ldots < T_N$, dove le date di stacco cedole delle obbligazioni coincidono con maturità di obbligazioni precedenti, e di voler ricavare i prezzi dei corrispondenti zero-coupon bond $P(t, T_1), \ldots, P(t, T_N)$.
Abbiamo visto che il prezzo di O_n è dato da

$$O_n = \sum_{i=1}^{n} \rho_n (T_i - T_{i-1}) P \cdot P(t, T_i) + P \cdot P(t, T_n)$$

dove poniamo $T_0 := t$.

Bootstrap e algebra lineare

Supponiamo di avere i prezzi di N obbligazioni $O_1, \ldots, O_N$ con maturità rispettivamente $T_1 < \ldots < T_N$ tali che le cedole vengano pagate esattamente in queste maturità con tasso fisso rispettivamente $\rho_1, \ldots, \rho_n$: chiaramente se uno di questi tassi è nullo l'obbligazione corrispondente sarà uno zero-coupon bond. Allora i prezzi $(O_1, \ldots, O_n)$ soddisfano il sistema

$$\begin{pmatrix} O_1/P \\ \vdots \\ O_N/P \end{pmatrix} = A \begin{pmatrix} P(t, T_1) \\ \vdots \\ P(t, T_N) \end{pmatrix}$$

dove (I_N = matrice identità su $\mathbb{R}^N$):

$$A := I_N + \begin{pmatrix} \rho_1(T_1 - t) & 0 & \ldots & 0 \\ \rho_2(T_1 - t) & \rho_2(T_2 - T_1) & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \rho_N(T_1 - t) & \ldots & \ldots & \rho_N(T_N - T_{N-1}) \end{pmatrix}$$

Invertire il sistema

Dobbiamo risolvere il sistema

$$\begin{pmatrix} O_1/P \\ \vdots \\ O_N/P \end{pmatrix} = A \begin{pmatrix} P(t,T_1) \\ \vdots \\ P(t,T_N) \end{pmatrix} \tag{1}$$

La matrice A è triangolare inferiore ed invertibile per ogni valore di $\rho_i > 0$ e per ogni $t < T_1 < \ldots < T_N$.
Per trovare i $(P(t,T_1),\ldots,P(t,T_N))$ basta quindi risolvere il sistema (1).
Nel linguaggio di programmazione MATLAB (e quindi anche nel suo clone Octave) è molto facile risolvere numericamente sistemi lineari. Si può quindi usare la formulazione più astratta del metodo bootstrap, costruendo la matrice A e poi risolvendo il sistema (1).

Funzioni MATLAB

Una peculiarità di MATLAB è quella che le sue funzioni, a differenza di molti linguaggi di programmazione, possono avere come output più di una variabile.
La sintassi tipica per definire una funzione è

```
function [lista output] = nome(lista input)
    (istruzioni)
endfunction
```

- `nome`: nome con cui si vuole richiamare la funzione;
- `lista input`: variabili separate da virgole;
- `[lista output]`: nome o nomi delle variabili che si vogliono come output. In particolare, se è sufficiente una sola variabile (che può essere anche un vettore), è sufficiente mettere il nome; se invece si vogliono più variabili con nomi diversi, i nomi vanno messi in sequenza, senza separatori, tra due parentesi quadre.

Funzione bootstrap

```
function [B R] = bootstrap(Obbl,rho,T)

[m n] = size(rho);

for i = 1:m
  for j = 1:i
    A(i,j) = rho(i)*(T(j+1)-T(j));
  end
  A(i,i) = A(i,i) + 1;
end

B = A\Obbl;

R = [0;-log(B/100)]./(T-T(1));

R(1) = R(2);

endfunction
```

Implementazione MATLAB

La routine `bootstrap.m` accetta come input:

- il vettore m-dimensionale `O` costituito dai prezzi delle obbligazioni, dove il valore nominale di ognuna si suppone uguale a 100;
- il vettore m-dimensionale `rho` costituito dalle loro cedole, in numero puro (quindi un tasso del 4% deve essere scritto come 0.04);
- infine, il vettore $(m+1)$-dimensionale `T` costituito dallo scadenzario, in anni.

L'output è costituito dai due vettori, rispettivamente m-dimensionale ed $(m+1)$-dimensionale, `B` ed `R`, costituiti rispettivamente dai prezzi degli zero-coupon bonds di maturità $T_1, \ldots, T_m$ (sempre con valore nominale 100), e dai tassi spot di maturità $T_0, \ldots, T_m$ (sempre in numeri puri).
Il primo tasso $\texttt{R(1)} = R(T_0, T_0)$ viene posto artificialmente uguale a $R(T_0, T_1)$ come da prassi di mercato.

Scrivere o caricare i dati

Proviamo la funzione `bootstrap` con i seguenti dati:

```
Obbl = [100.38; 101.88; 103.34; 105.62; 108.66];

rho = [4; 3.5; 3.625; 4.375; 5.125]/100;

T = [0; .0959; .5973; 1.0959; 1.5973; 2.1370];
```

che possono essere scritti direttamente da riga di comando, oppure caricati da un file script:

```
> datibootstrap
> [B R] = bootstrap(Obbl,rho,T)
```

A questo punto si può visualizzare il risultato, in termini di tassi zero-coupon, con il comando `stem(T,R)`.

Interpolazione

Il metodo bootstrap ci permette di trovare i tassi $R(t, T_i)$ a partire da prezzi di obbligazioni con maturità T_i, non necessariamente a cedola nulla. Spesso però si pone il problema di conoscere tassi $R(t, T)$ (o equivalentemente prezzi di zero-coupon bond) per $T \neq T_i$, cioè per maturità per cui sul mercato non sono presenti strumenti.

Supponiamo infatti di avere trovato vettori di maturità e tassi spot, ad esempio con la precedente procedura di bootstrap. Tutto quello che sappiamo, allora, sono i tassi con *quelle* maturità (`stem(T,R)`), mentre invece potremmo avere bisogno di una curva di tassi continua (`plot(T,R)`).

Quello che vogliamo fare è quindi una *interpolazione* della curva dei tassi, cioè una ricostruzione dell'intera funzione $R(t, \cdot)$ a partire da pochi valori noti $R(t, T_i) = R_i$. Ci limitiamo qui a citare pochi metodi semplici.

File datitassi.m

```
T = [0; 1/365; 7/365; 1/12; 1/6; 0.25; 0.5; 0.75; 1; 2; 3;
8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22;
26; 27; 28; 29; 30; 35; 40; 45; 50];

R = [3.99; 3.99; 4.23; 4.37; 4.68; 4.86; 4.92; 4.99; 5.04;
4.66; 4.62; 4.61; 4.63; 4.66; 4.69; 4.73; 4.77; 4.81; 4.84
4.91; 4.92; 4.93; 4.93; 4.95; 4.94; 4.94; 4.94; 4.93; 4.94
4.91; 4.91; 4.91; 4.88; 4.85; 4.83; 4.81]/100;
```

Interpolazione lineare *raw*

Questa interpolazione consiste nell'interpolare i valori di $R(t,\cdot)$ nell'intervallo (T_i, T_{i+1}) con un segmento di estremi (T_i, R_i) e (T_{i+1}, R_{i+1}). Questo può essere fatto con le seguenti istruzioni Octave:

```
> xf = [0:1/12:50];
```

definisce un vettore di tempi `x` con componenti da 0 a 50 (anni) intervallate da 1/12 (un mese);

```
> lin = interp1(T,R,xf);
```

interpola linearmente i punti $(T, R) = ((T_i, R_i))_i$ rispetto ai nuovi punti della variabile indipendente contenuti in `xf` ;

```
> plot(xf,lin)
```

disegna il grafico ottenuto, con lo stesso risultato del comando `plot` senza interpolazione (difatti l'interpolazione standard di `plot` è quella lineare).

Interpolazione con *spline* cubiche

Questa interpolazione consiste nell'interpolare i valori di $R(t, \cdot)$ nell'intervallo (T_i, T_{i+1}) con un polinomio di 3. grado, in modo che la funzione risultante sia globalmente di classe C^2. Questo può essere fatto con le seguenti istruzioni Octave:

```
> xf = [0:1/12:50];
```
come sopra;

```
> spl = interp1(T,R,xf,'spline');
```
interpola i punti $(T, R) = ((T_i, R_i))_i$ rispetto ai nuovi punti della variabile indipendente contenuti in `xf`, usando stavolta il metodo delle spline cubiche;

```
> plot(xf,spl)
```
disegna il grafico ottenuto.

Interpolazione parametrica

A seconda dei modelli, questa potrebbe non essere una vera e propria interpolazione: questo metodo consiste in fatti nel supporre che $R(t,\cdot)$ appartenga ad una classe di funzioni parametrizzata da un vettore di parametri θ. Se la classe è sufficientemente ricca, può accadere che esista un θ^* tale che $R(t, T_i; \theta^*) = R_i$ per ogni scelta di T_i, R_i, $i = 1, \ldots, n$. Se questo non è possibile, il tipico criterio è quello di realizzare le uguaglianze ai minimi quadrati, cioè di scegliere il θ^* che minimizza l'espressione

$$\min_{\theta} \sum_{i=1}^{n} (R(t, T_i; \theta) - R_i)^2 \tag{2}$$

In questo caso non si parla più di interpolazione in senso stretto, ma piuttosto di *calibrazione* della curva zero-coupon ai dati di mercato.