Solutions - Practical Lesson 7

Matteo Sani
matteosan1@gmail.com

November 11, 2019

1 Solutions

1.1 Exercises

1.1.1 Exercise 7.2

Applying a bootstrapping technique (outlined in lesson 5) derive the a credit curve from the following CDS market quotes:

```
pricing_date = date(2019, 11, 6)

cds_quotes = [
    {'maturity': 12, 'spread':0.0149},
    {'maturity': 24, 'spread':0.0165},
    {'maturity': 36, 'spread':0.0173},
    {'maturity': 69, 'spread':0.0182},
    {'maturity': 120, 'spread':0.0183},
    {'maturity': 240, 'spread':0.0184},
]
```

Hint

- create a CDS contract for each input market quote;
- implement an objective function to minimize the squared sum of the CDS npvs, the function has to implement also a `CreditCurve` with the unknown (to be detrmined by the bootstrap survival probabilities) and a list of pillars corresponding to the CDS maturities;
- remember to set initial guesses and boundary conditions for the unknown parameters (the first survival probability has to be set to 1 since no default happened !, the others free to move between [0.01 to 1]);
- using the `scipy.optimize.minimize` find the curve.

[1]:
```
from finmarkets import generate_swap_dates, CreditDefaultSwap, CreditCurve
from curve_data import discount_curve
from datetime import date
from scipy.optimize import minimize

pricing_date = date(2019, 11, 6)
```

```python
cds_pillar_dates = [pricing_date]

cds_quotes = [
    {'maturity': 12, 'spread':0.0149},
    {'maturity': 24, 'spread':0.0165},
    {'maturity': 36, 'spread':0.0173},
    {'maturity': 69, 'spread':0.0182},
    {'maturity': 120, 'spread':0.0183},
    {'maturity': 240, 'spread':0.0184},
]

creditdefaultswaps = []
for quote in cds_quotes:
    creditdefswap = CreditDefaultSwap(
        1,
        pricing_date,
        quote['maturity']//12,
        quote['spread']
    )
    creditdefaultswaps.append(creditdefswap)
    cds_pillar_dates.append(creditdefswap.payment_dates[-1])

cds_pillar_dates = sorted(cds_pillar_dates)
n_cds_vector = len(cds_pillar_dates)

def obj_function(unknown_ndps):
    curve_c = CreditCurve(
        cds_pillar_dates,
        unknown_ndps
    )

    sum_sq = 0.0
    for creditdefswap in creditdefaultswaps:
        sum_sq += creditdefswap.npv(discount_curve, curve_c) ** 2

    return sum_sq

x0_guess = [0.001 for i in range(n_cds_vector)]

bounds_credit_curve = [(0.01, 1) for i in range(n_cds_vector)]
bounds_credit_curve[0]=(1,1)
results = minimize(obj_function, x0_guess, bounds=bounds_credit_curve)

print (cds_pillar_dates)
print (results.x)
```

```
[datetime.date(2019, 11, 6), datetime.date(2020, 11, 6), datetime.date(2021, 11,
6), datetime.date(2022, 11, 6), datetime.date(2024, 11, 6), datetime.date(2029,
11, 6), datetime.date(2039, 11, 6)]
[1.         0.90681257 0.80416782 0.70889095 0.54479309 0.29706549
 0.08665777]
```

1.1.2 Exercise 7.3

Using the above `Credit Curve` and the `DiscountCurve` already defined in lesson 5, price the following CDS:

```
cds_to_price = [
    {'nominal': 5000000, 'maturity': 18, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 30, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 42, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 72, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 108, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 132, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 160, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 184, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 210, 'spread': 0.02}
]
```

[2]:

```python
from curve_data import discount_curve
from finmarkets import CreditDefaultSwap, CreditCurve
import datetime

pricing_date = datetime.date(2019, 11, 6)
curve_pillar_dates = [datetime.date(2019, 11, 6), datetime.date(2020, 11, 6),␣
 ↪datetime.date(2021, 11, 6), datetime.date(2022, 11, 6), datetime.date(2024,␣
 ↪11, 6), datetime.date(2029, 11, 6), datetime.date(2039, 11, 6)]

surv_prob = [1, 0.90681256, 0.80416782, 0.70889095, 0.54479309, 0.29706549, 0.
 ↪08665777]

cds_to_price = [
    {'nominal': 5000000, 'maturity': 18, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 30, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 42, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 72, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 108, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 132, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 160, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 184, 'spread': 0.02},
    {'nominal': 5000000, 'maturity': 210, 'spread': 0.02},
]

credit_curve = CreditCurve(curve_pillar_dates, surv_prob)
```

```
npv_cds_to_price = []
for quote in cds_to_price:
    creditdefswapprice = CreditDefaultSwap(
        quote['nominal'],
        pricing_date,
        quote['maturity']//12,
        quote['spread']
    )

    npvcdstoprice = creditdefswapprice.npv(discount_curve, credit_curve)

    npv_cds_to_price.append(npvcdstoprice)

print (npv_cds_to_price)
```

```
[-96879.21494231955, -125875.49083899031, -137489.2485942383,
-150388.56810656283, -184387.14756564656, -199054.44398897374,
-208689.7770410725, -216150.51575594302, -221690.38180709025]
```