Project
The Information Content of Options

Michael Rockinger*

January 2025

To run this program you need a working version of Matlab (I tested all versions up to 2024 and they work). You need the optimizer package and a routine to do OLS regressions. E.g. Kevin Shepard or LeSage's toolboxes. I can't help with those external packages.

1 Part I: Smiles and Volatility surfaces

The following case aims at taking you through the various steps that you may have to follow if you are asked one day to estimate risk neutral densities. This case cannot be solved in a single evening. Actually, since there is still ongoing research in this domain, it may serve you as a laboratory within which you can experiment new techniques to check their actual working. As you will notice, formulae may look simple, but when you actually have to estimate RNDs, things may turn out to be rather hard.

In this case, I propose that you work with FTSE 100 options data. I took that data out of the Financial Times of March 26, 2004. If you are done with this case, you can replace this data by your own data. You may obtain you own data either by consulting the websites of various trading places, or by reading a good newspaper such as the NZZ. The data is for European options. The underlying asset is the FTSE 100. All option prices are settlement prices. The level of the index, 4357.5, was also obtained at the close of the market. The options are exercised at mid-month and the expirations are April, May, June, July, and September. We assume that there are 365 days per year. You will find an Excel sheet labeled `FTSE100Mar2604.XLS` containing this data. It is found that the interest rates for one month are $4\frac{7}{32} - 4\frac{1}{8}$, for three months $4\frac{3}{8} - 4\frac{9}{32}$, and for six months, they are $4\frac{1}{2} - 4\frac{13}{32}$. There are 8 different strike prices ranging from 4125 up to 4825 by increments of 100.

Quest 1: Compute for the various interest rates the mid-quote. Verify that this mid-quote corresponds to the rates that you were actually given.

Quest 2: You are given a matlab file `FTSEStart.m` that contains a few lines of code. These lines input the raw data and do some transforms to put the data in a nicer format. You are asked to print the info contained in the variable `AllInfo`. Verify the correctness of the data.

Quest 3: It is assumed that the model describing the FTSE is Black-Scholes for a dividend paying asset. We assume that the dividend payment is continuous, at some rate that we denote by δ. We recall that the corresponding option pricing formula may be found in the lecture notes. Obviously, the dividend rate is unknown, and therefore this rate has to be estimated. One way of doing so is to use all the call and put options and to extract that volatility and that dividend rate that minimizes the distance of all call and put options with their empirical equivalents. This means that if NC and NP are the number of calls and puts, if $\widehat{C}_i$ and $\widehat{P}_i$ are the various option prices, we seek

$$(\delta, \sigma) \in \arg\min \sum_{i=1}^{NC} (C_i(\delta, \sigma) - \widehat{C}_i)^2 + \sum_{i=NC+1}^{NC+NP} (P_i(\delta, \sigma) - \widehat{P}_i)^2.$$

Complete your program with an implementation of this minimization. You should use the instruction `fmincon` to do the actual minimization. You should also create two external functions that you will call `BSCALLD.m` and `BSPUTD.m` implementing the Black-Scholes formula for a dividend paying asset. What de you find in terms of implied volatility and implied dividend? Plot the implied volatility against time.

Quest 4: In the following we will assume that the dividend payments can get neglected, hence $\delta = 0$. We will also continue to work with the actual interest rate rather than the implicit interest rate. At this stage you are asked to compute for each of the options the implied volatility and trace the volatility surface for that day. This means that you should compute for each option that volatility solving

$$\sigma \in \arg\min \ (C_i(\sigma) - \widehat{C}_i)^2$$

for a given call option i and similarly for the various put options. Since there are call and put options for a given maturity and especially for a given strike there are various ways to construct "the" volatility for a given strike. Trace the volatility surface under the following scenarii

1) only call option volatilities

2) only put option volatilities

3) the average of put and call options volatilities

4) the implied volatility for in-the-money options only.

We store the given information characterizing the various options as well as the implied volatilities computed, for instance as the average volatilities (3 above), in a matrix named `AllInfo1.mat`. You will use this matrix in the following parts.

2 Part II: Estimating RNDs

In this second part of the case we will implement the estimation of several techniques that yield RNDs for various maturities. In practice, as you will discover, this estimation is rather delicate since it often involves polynomial approximations that may become negative unless correctly twisted. Other techniques require good starting values. Also the data may require some weeding in order to eliminate those options that appear mispriced. For this reason you should view the following as a first step to a "professional" implementation of the RND computation rather than the "ultimate thing" running on a trading floor.

Obviously, we suggest you continue to work with the same data as in the first part of the case study. You will need the values of the implied volatilities that you should have stored in some matlab matrix, say `AllInfo1.mat`.

Question 1: Create a program that loads the matrix `AllInfo1.mat`. Next construct the support of the risk neutral density. This support is assumed to range over the interval 3800 to 5100 with a step length of 10. In the other methods you will also use this support. To obtain benchmark RNDs, we consider log-normals where the density is defined as

$$f(S_T) = \frac{1}{\sqrt{2\pi}\sigma\sqrt{T}} \cdot \frac{1}{S_T} \cdot \exp[-\frac{1}{2}(\frac{\log(S_T) - m}{\sigma\sqrt{T}})^2],$$

and

$$m = \log(S_0) + (r - \frac{1}{2}\sigma^2)T,$$

the only unknown being σ. We suggest you take, for a given maturity, both the put and call implied volatilities, compute their average, and eventually take the smallest implied volatility over the various strikes. This smallest volatility tends to be associated with the most liquid options and is, therefore, a good representative of volatility. Actually, some stock exchanges publish a series of implied volatilities. Often it is this "at-the-money" minimal volatility that is published. Once you have constructed the RNDs for the various maturities, you may wish to save them in a matrix we will call `BenchRND.mat`.

Question 2: Presently you will fit the mixture-of-distributions RND to the various in the money options for a given maturity. As a reminder, this model assumes that call options (and similarly put options) can be written as

$$C(a, \mu_1, \mu_2, \sigma_1, \sigma_2) = a \cdot C^{LN}(\mu_1, \sigma_1) + (1-a) C^{LN}(\mu_2, \sigma_2)$$

where

$$\begin{aligned} C^{LN}(\mu, \sigma) &= e^{-rT} \left(S_0 \exp(\mu T) \Phi(d_1) - K \Phi(d_2) \right), \\ d_1 &= \frac{\log(S_0/K) + (\mu + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}, \\ d_2 &= d_1 - \sigma\sqrt{T}. \end{aligned}$$

The estimation should also impose the martingale condition. This condition can be imposed by setting in the estimation

$$S_0 = a C^{LN}(\mu_1, \sigma_1) \mid_{K=0} + (1-a) C^{LN}(\mu_2, \sigma_2) \mid_{K=0} .$$

Since the Black-Scholes formula is not defined for $K = 0$, this condition can be imposed by setting $K = 0.1$. This is considered to be small in comparison with the other strikes. Overall, the minimization seeks those $a, \mu_1, \mu_2, \sigma_1, \sigma_2$ that minimize

$$\begin{aligned} \sum_{i=1}^{NC} (\widehat{C}_i - C_i(\theta))^2 \quad &+ \sum_{i=NC}^{NC+NP} (\widehat{P}_i - P_i(\theta))^2 + \\ &\left[S_0 - a C^{LN}(\mu_1, \sigma_1) \mid_{K=0} - (1-a) C^{LN}(\mu_2, \sigma_2) \mid_{K=0} \right]^2 \end{aligned}$$

where $\theta = (a, \mu_1, \mu_2, \sigma_1, \sigma_2)$ and $\widehat{C}, \widehat{P}$ are the actual option prices.

We recommend that, as a first step, you take a grid for a such as $a = 0.01 : 0.1 : 0.99$. For each a compute the optimizing parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2)$. In a second step, start with the optimal a and the associated parameters and perform a global estimation involving all the parameters: $a, \mu_1, \mu_2, \sigma_1$, and σ_2.

Eventually, for each maturity separately, construct the risk neutral densities and save them in a matrix `MixRND.mat`.

You may have to play with various initial values to obtain "reasonable" estimates. Another trick consists in performing the estimations with a constraint on the volatility parameters such as $\sigma_1 > \sigma_2$. This is easy to implement with the `fmincon` instruction.

Question 3: Construct a program that constructs the support of the densities. Then this program should load the various risk neutral densities. Load the benchmark and mixtures-of-distributions RNDs and compare the densities. What do you notice?

Question 4: You also wish to estimate the GB2 density proposed by Bookstaber and McDonald. We remind that the GB2 density is defined by

$$g_{GB2}(x|a,b,p,q) = \frac{a}{b^{ap}B(p,q)} x^{ap-1}[1+(\frac{x}{b})^a]^{-(p+q)}, \quad x > 0,$$

where

$$B(p,q) \equiv \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$$

is the beta function and where $\Gamma(\cdot)$ is the gamma function.

The martingale condition is

$$S_0 = \frac{b \cdot B(p+\frac{1}{a}, q-\frac{1}{a})}{B(p,q)}.$$

The theoretical option pricing formula depends on the cumulative distribution function of the GB2 density. This cdf will be denoted G_{GB2}. This cdf is related to the cdf of the beta distribution, denoted by G_β by

$$z(x,a,b) = (x/b)^a / \left(1 + (x/b)^a\right), \text{ and}$$

$$G_{GB2}(x|a,b,p,q) = G_\beta(z(x,a,b) \mid p,q).$$

The European call option price is

$$C(K) = S_0[1 - G_\beta(z(K,a,b) \mid p + \frac{1}{a}, q - \frac{1}{a}] - Ke^{-rT}[1 - G_\beta(z(K,a,b) \mid p,q)].$$

To get a first feel for this density, create a program that traces the g_{GB2} density as a function of the various parameters. Play with the parameters till you find a density that is located over the support of the RNDs (3800:10:5100) and that looks roughly like a normal. Verify, using a numerical integration that the construction of the cdf is correct. In other words, that the probability mass integrates up to one, see also below how to do this integration.

Then, create a program that implements the call and put option formula as well as the martingale condition. Formally estimate the parameters for the various maturities. Save the RNDs under the name `GB2RND.mat`. Compare these densities with the benchmark and mixtures densities.

Notice that the estimation of the parameters along this method does not come without surprises. The density is only well defined if the parameters a, b, p, q are positive and if $q - \frac{1}{a} > 0$. This latter condition can be imposed as a linear constraint by estimating the parameter $\frac{1}{a}$ rather than a. Also, a good idea is to keep reasonable parameters (once you have found some) and to test their stability by considering perturbations of these initial conditions.

Question 5: you may implement some other estimation. Possible choices are the semiparametric model of Abken, Madan, and Ramamurty, the entropy method of Buchen and Kelly, the spline method of Shimko or the more modern versions thereof such as by Campa, Chan, Reider 1997 (See also Bliss and Panigirtzoglou). Answer this question only if you have the time to do so. Good luck!

3 Part III: Estimation of the subjective density.

In this last section of the case study, you will construct the subjective real world density using a simulation based on a GARCH with asymmetries. You will also estimate the relative risk aversion function and the risk aversion function.

This part is in itself composed by two parts. In a first part you have to estimate the parameters of the GARCH with asymmetries and Student-t distributed residuals. In a second part, you are asked to simulate trajectories and to construct the real density by using a kernel smoother.

Question 1: Create a matlab program that loads the Excel matrix of data named `StkIdx.XLS`. The fourth column of that matrix contains the values of the FT100 ranging between the dates Jan 1, 1970 and April 8, 2004. Using Excel, observe the values of this stock-market index. Verify under Matlab the correctness of your data. Construct daily returns using the formula $r_t = 100 * \log(S_t/S_{t-1})$. Save these returns under the name `rtFT.mat`.

Question 2: We remind that the GARCH (1,1) with asymmetric impact is given by

$$\begin{aligned} r_t &= \mu + y_t, \\ y_t &= \sigma_t \cdot \varepsilon_t, \\ \text{where } \varepsilon_t &\sim \text{Student-t}(\nu), \text{ and} \\ \sigma_t^2 &= \omega + \alpha^+ y_{t-1}^2 \cdot I(y_{t-1} > 0) + \\ & \quad \alpha^- y_{t-1}^2 \cdot I(y_{t-1 \leqslant 0}) + \beta \sigma_{t-1}^2. \end{aligned}$$

For this (last) part of the case, you should only use the last 3000 observations in this estimation unless indicated otherwise. Estimate the parameter μ

using the usual formula for an average, i.e.

$$\widehat{\mu} = \frac{1}{T}\sum_{t=1}^{T} r_t.$$

Estimate the parameters of the GARCH (1,1) using `GARCHAT.m` and $r_t - \hat{\mu} = y_t$ as input. Comment on your findings. Also comment on the size of the degrees of freedom parameter of the Student-t.

Choose different sample sizes and estimate the parameters. What do you conclude concerning the stability of the parameters?

Complete the program so that a function constructs the series $\hat{\sigma}_t$ and $\widehat{\varepsilon}_t \equiv y_t/\widehat{\sigma}_t$. Plot $\hat{\sigma}_t$ with the absolute value of the returns. Does the $\hat{\sigma}_t$ series look reasonable? Retain the value of $\widehat{\sigma}_t$ for March 26, 2004.

Question 3: For the parameters corresponding to the estimation of the GARCH with 3000 observations you will presently write a program that simulates 10'000 trajectories for a time length of 20 days. Using σ_0 and the Matlab pseudo-Student-t random-number generator `trnd`(ν, r, c) where ν is the degree of freedom parameter and r, c correspond to the number of rows and columns of data that should get generated, construct r_t. Since $S_T = S_0 \exp[(r_1 + r_2 + ... + r_T)/100]$, simulate S_T. Simulate 10'000 observations of S_T where $T = 20$. Verify the correctness of the S_T by tracing a histogram via the Matlab instruction `hist`.

Last you are asked to use the kernel estimator `kern.m` to construct the smoothed density. The kernel estimator function takes as inputs the S_T, i.e. the simulated prices, and a parameter indicating the smoothness of the kernel. Last you should provide to `kern.m` the values over which you want to estimate the density. As usual this will be $z = 3800 : 10 : 5100$. You may have to scale the smoothed curve so that its probability mass integrates up to one.

We remind that an integral may be approximated using

$$\begin{aligned}\int_b^a f(x)dx &= \sum_{i=0}^{N} f(x_i) \cdot \Delta, \text{ where } \Delta = (b-a)/N, \\ \text{and } x_i &= a + i \cdot \Delta.\end{aligned}$$

We also remind that f is a density if $\int_a^b f(x)dx = 1$ and $f(x) \geq 0$, where $[a, b]$ is the support of the density.

Call the smoothed subjective density `SubD` and save this density.

Question 4: You have presently reached the last step to evaluate the risk aversion of the representative option trader. Create a program that loads the subjective density, that we will denote by p. Also load a risk neutral density q of your choice.

The risk aversion function is defined as

$$RA(z) = \frac{1}{p(z)} \frac{dp(z)}{dz} - \frac{1}{q(z)} \cdot \frac{dq(z)}{dz} = \frac{d}{dz} \log\left(\frac{p(z)}{q(z)}\right).$$

One can also show that the relative risk aversion function is given by

$$RRA(z) = z \cdot RA(z).$$

Plot the relative risk aversion against z. What do you observe as a general shape for this function?

In practice, you would compute this function not only for one day but for many days and eventually get an overall estimate using an average of the various estimates of the risk-aversion function.