Using the *schwartz97* package

Philipp Erb, David Lüthi & Simon Otziger

October 12, 2010

Abstract

The purpose of this document is to show how the R package *schwartz97* can be used. This is done by numerous examples and intuitive explanations.

1 Introduction

The package *schwartz97* provides a set of functions to work with the two-factor model of Gibson and Schwartz (1990)[1]. The two-factor model describes the joint dynamics of the state variables *spot price* and *spot convenience yield* (later simply called *convenience yield*).

We believe that the value of this package primarily lies in the parameter fitting routine `fit.schwartz2f`. Once the parameters of the two-factor model are estimated, a number of functions can be used to, e.g., draw samples from the model, price derivatives as European options, or filter a futures price series to get an estimate of the underlying state variables. The document is organized as follows: Section 2 briefly describes the model and section 3 gives an overview of the classes and functions. Then, section 4 to 8 give examples and a case study.

2 The Schwartz Two-Factor Model

This model assumes the spot price of the commodity and the instantaneous convenience yield to follow the joint stochastic process:

$$\begin{aligned} dS_t &= (\mu - \delta_t)S_t dt + \sigma_S S_t dW_S \quad (1) \\ d\delta_t &= \kappa(\alpha - \delta_t)dt + \sigma_\epsilon dW_\epsilon, \quad (2) \end{aligned}$$

with Brownian motions W_S and W_ϵ under the objective measure $\mathbb{P}$ and correlation $dW_S dW_\epsilon = \rho dt$. Under the pricing measure $\mathbb{Q}$ the dynamics are of the form

$$\begin{aligned} dS_t &= (r - \delta_t)S_t dt + \sigma_S S_t d\widetilde{W}_S \quad (3) \\ d\delta_t &= [\kappa(\alpha - \delta_t) - \lambda]dt + \sigma_\epsilon d\widetilde{W}_\epsilon, \quad (4) \end{aligned}$$

where the constant λ denotes the market price of convenience yield risk and $\widetilde{W}_S$ and $\widetilde{W}_\epsilon$ are $\mathbb{Q}$-Brownian motions. It may be handy to introduce a new mean-level for the convenience yield process under $\mathbb{Q}$

$$\tilde{\alpha} = \alpha - \lambda/\kappa. \quad (5)$$

The dynamics is then

$$d\delta_t = \kappa(\tilde{\alpha} - \delta_t)dt + \sigma_\epsilon d\widetilde{W}_\epsilon. \quad (6)$$

For more information on the model and pricing formulas we refer to the other package vignette *Technical Document*, or to Schwartz (1997) and Hilliard and Reis (1998).

3 Package Overview

This section gives an overview of the functions and classes contained in the package *schwartz97*. In addition, the object-oriented programming approach followed in this package is explained.

3.1 Functions

The core of the package *schwartz97* is built by the following functions[2]:

R-function	Description
`dstate`	Density of the bivariate state vector.
`pstate`	Distribution of the bivariate state vector.
`qstate`	Quantile of the bivariate state vector.
`rstate`	Sample from the state distribution at some future time.
`simstate`	Generate trajectories from the bivariate state vector.
`dfutures`	Density of the futures price.
`pfutures`	Distribution of the futures price.
`qfutures`	Quantile of the futures price.
`rfutures`	Sample from the futures price distribution.
`pricefutures`	Calculate the futures price.
`priceoption`	Calculate the price of European call or put options.
`filter.schwartz2f`	Filter a futures price series.
`fit.schwartz2f`	Fit the two-factor model to data.

Except the function `fit.schwartz2f` all the above functions are set to generic and accept three different signatures (see section 3.3).

[1]Because the model was extended in Schwartz (1997) and Miltersen and Schwartz (1998) we call it the *Schwartz two-factor model* hereafter.

[2]There are also a number of utility functions as *coef*, *mean*, *vcov*, *plot*, *resid*, and *fitted*.

```
----------------------------------------------------------------
Schwartz97 two-factor model:

SDE
d S_t     = S_t   * (mu - delta_t)    * dt + S_t * sigmaS * dW_1
d delta_t = kappa * (alpha - delta_t) * dt + sigmaE * dW_2
E(dW_1 * dW_2) = rho * dt

Parameters
s0    : 100
delta0: 0
mu    : 0.1
sigmaS: 0.2
kappa : 1
alpha : 0.1
sigmaE: 0.3
rho   : 0.4
----------------------------------------------------------------
```

Objects of class `schwartz2f.fit` are constructed via the function `fit.schwartz2f` (see section 8).

5 Working with the state variables

As soon as a `schwartz2f` object is initialized, it can be passed to the functions `dstate`, `pstate`, `qstate`, `rstate`, and `simstate`. The distribution of the state variables depend on the horizon. Once this point in time is defined the above functions can be used like the standard R distribution functions for, e.g., the normal distribution (`dnorm`, `pnorm`, `qnorm`, `rnorm`).

In this example a sample of the spot price and the convenience yield in five years is generated by the function `rstate`. Then, the probability that the spot price is below 150 and the convenience yield is lower than 0 in five years is computed. The mean of the state variables in one and ten years is calculated next. Finally, trajectories of the state variables are plotted (see fig. 1).

```
> time <- 5
> sample.t <- rstate(n = 2000, time, obj)
> pstate(c(0, -Inf), c(150, 0), time, obj)
```

```
[1] 0.2243732
attr(,"error")
[1] 1e-15
attr(,"msg")
[1] "Normal Completion"
```



```
> mean(obj, time = c(1, 10))
```

```
          s.t    delta.t
[1,] 106.3906 0.06321206
[2,] 130.5386 0.09999546
```

```
> plot(obj, n = 50, time = 5, dt = 1/52)
```

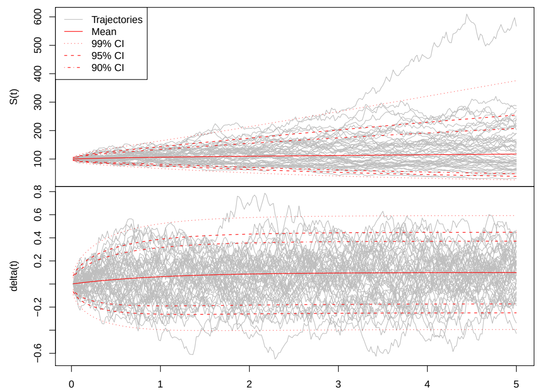

Figure 1: Fifty trajectories of the state variables are plotted on a weekly interval and a five years horizon. The initial values of the state variables are 100 for the spot price (s_0) and 0 for the convenience yield (δ_0). The spot price has a drift μ of 10% and a volatility σ_S of 20%. The *speed of mean-reversion* parameter κ of the convenience yield process is 1, and the long-term mean (α) is 10%. The volatility of the convenience yield σ_ϵ is 30% and the correlation ρ between the Brownian motions driving the state variables is 40%.

6 Working with derivatives

In this example we calculate some futures prices and plot the dynamics of the term structure (“forward curve”). In addition prices of European options are computed. We work through this section by looking at corn and assuming all the parameters are known.

The current price (s_0) of 1000 bushels of corn is assumed to be 80 USD. The convenience is yield (δ_0) is zero at the moment but it’s long-term mean

```
> ttm <- 0:4
> pricefutures(ttm, s0 = s0, delta0 = 0, sigmaS = sigmaS,
+     kappa = kappa, sigmaE = sigmaE, rho = rho,
+     r = r, alphaT = 0)

[1] 1.000000 1.021605 1.054220 1.099526 1.152367

> pricefutures(ttm, s0 = s0, delta0 = 2 * r, sigmaS = sigmaS,
+     kappa = kappa, sigmaE = sigmaE, rho = rho,
+     r = r, alphaT = 0)

[1] 1.0000000 0.9835835 1.0009214 1.0385929 1.0864517

> pricefutures(ttm, s0 = s0, delta0 = r, sigmaS = sigmaS,
+     kappa = kappa, sigmaE = sigmaE, rho = rho,
+     r = r, alphaT = 2 * r)

[1] 1.0000000 0.9805302 0.9595804 0.9449587 0.9335775

> pricefutures(ttm, s0 = s0, delta0 = -r, sigmaS = sigmaS,
+     kappa = kappa, sigmaE = sigmaE, rho = rho,
+     r = r, alphaT = 2 * r)

[1] 1.0000000 1.0184332 1.0106773 1.0003988 0.9902179
```


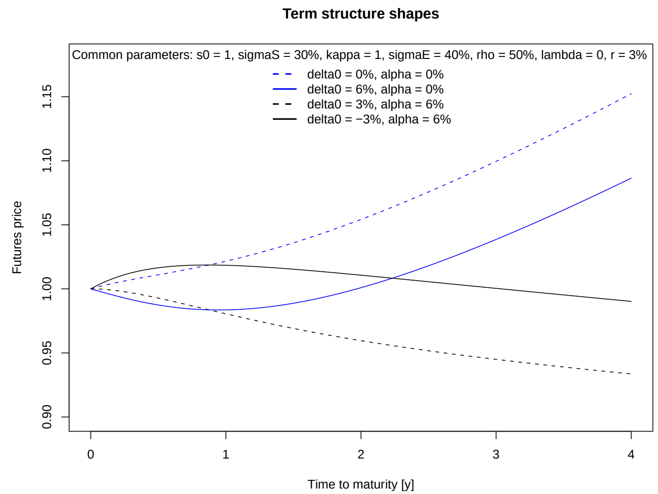

Figure 3: Four term structure shapes which can be generated by the Schwartz two-factor model. For a given risk-free interest rate r, the shape is determined by the initial value of the convenience yield δ_t and by the mean level of the convenience yield $\tilde{\alpha}$ under the pricing measure $\mathbb{Q}$ (see (3) and (4)).

```
+     silent = TRUE)
> wheat.fit

----------------------------------------------------------------
Fitted Schwartz97 two-factor model:

SDE (P-dynamcis)
d S_t     = S_t   * (mu - delta_t) * dt    + S_t * sigmaS * dW_1
d delta_t = kappa * (alpha - delta_t) * dt + sigmaE * dW_2
E(dW_1 * dW_2) = rho * dt

SDE (Q-dynamcis)
d S_t     = S_t   * (r - delta_t) * dt      + S_t * sigmaS * dW*_1
d delta_t = kappa * (alphaT - delta_t) * dt + sigmaE * dW*_2
alphaT = alpha - lambda/kappa

Parameters
s0    : 395.5
delta0: 0
mu    : 0.25026786548102
sigmaS: 0.376161779933003
kappa : 0.00499248996853649
alpha : -0.12065663312003
sigmaE: 0.159902538320832
rho   : 0.904453926049466
r     : 0.03
lambda: 0
alphaT: -0.12065663312003
----------------------------------------------------------------
Optimization information
Converged:            FALSE
Fitted parameters:  mu, sigmaS, kappa, alpha, sigmaE, rho, meas.sd1; (Number: 7)
log-Likelihood:       -3343706937
Nbr. of iterations: 302
----------------------------------------------------------------

> plot(wheat.fit, type = "trace.pars")
```

Let's discuss the parameters: A μ of 25% is probably not far off and the spot price volatility σ_S of 37% seems to be fine too. The speed of mean reversion of the convenience yield process κ is alarmingly close to zero which means that the convenience yield can drift far away from it's mean. The mean level of the convenience yield α of -12% is not intuitive to say the least. The correlation ρ of 90% is very high, making the parameters even

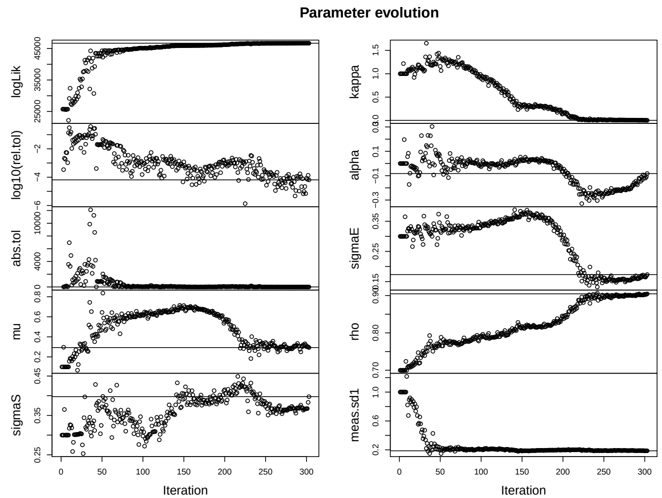

Figure 4: This figure shows the parameter evolution of the first 300 iterations of the unconstrained parameter estimation of wheat. While the relative tolerance gets below 10^{-4} after around 200 iterations the parameter values still fluctuate substantially.

harder to interpret and the process dynamic less intuitive. In addition, the parameter evolution shown in fig. 4 is concerning.

Two measures are introduced as an attempt to make parameters more appealing: κ is constrained to 1 and the measurement error standard deviations are set proportional to the average traded volumes. The average initial value of the measurement error standard deviations is set to 1%.

```
> vol.std <- colSums(futures$wheat$vol, na.rm = TRUE)/sum(futures$wheat$vol,
+     na.rm = TRUE)
> wheat.fit.constr <- fit.schwartz2f(futures$wheat$price,
+     futures$wheat$ttm/260, kappa = 1, opt.pars = c(s0 = FALSE,
+         delta0 = FALSE, mu = TRUE, sigmaS = TRUE,
+         kappa = FALSE, alpha = TRUE, sigmaE = TRUE,
+         rho = TRUE, lambda = FALSE), meas.sd = 1/vol.std/sum(1/vol.std) *
+         length(vol.std) * 0.01, deltat = 1/260,
+     control = list(maxit = 300), silent = TRUE)
> wheat.fit.constr
```

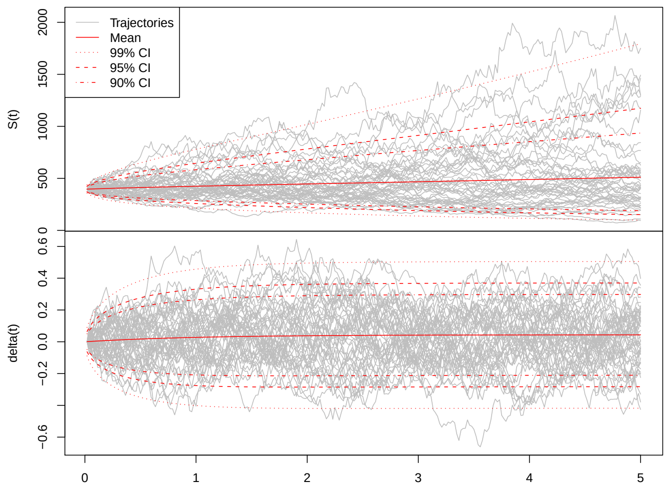

Figure 6: Fifty trajectories based on the constrained parameter estimates of wheat.

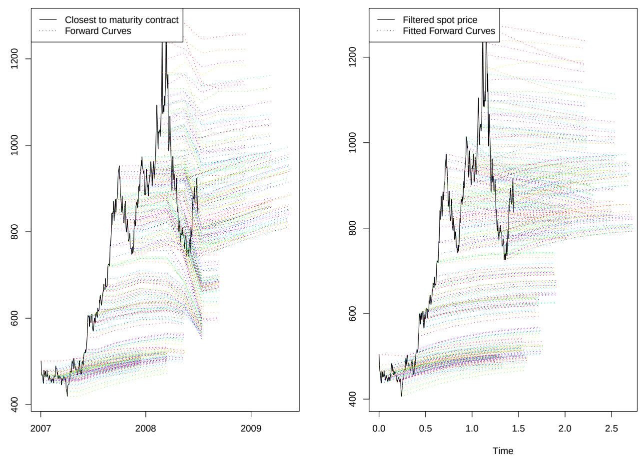

Figure 7: Real term structures (left panel) and model generated term structures (right panel) from Jan. 2007 to June 2008. The backwardation starting at mid of 2007 is not captured by the model initially, and underestimated later in 2007. The shape of the term structure at the peak starting in 2008 can not be produced by the Schwartz two-factor model. The model's prediction looks reasonable in Q1 2007 and Q2 2008.

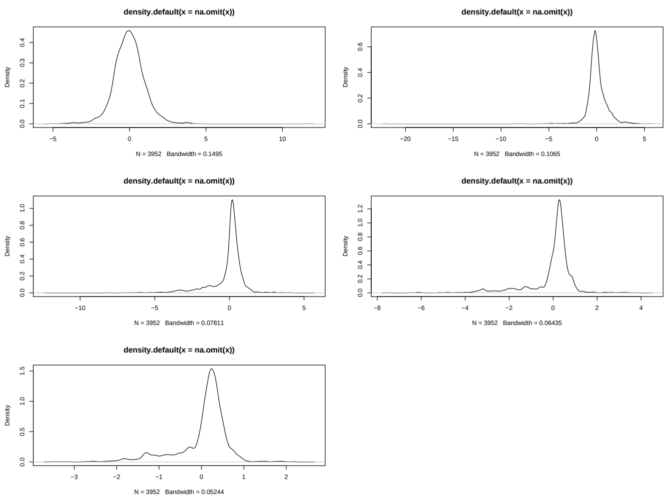

Figure 9: Non of the residual distributions look normal. The first distribution looks normally shaped but exhibits outliers on both tails. The second to fifth distributions look asymmetric and fat tailed.

8.2.2 Confidence Intervals and Value-at-Rsik

Confidence intervals and the values-at-risk are computed here regardless of the misspecification of the model estimated above. In order to estimate the most recent values of the spot price and the convenience yield the function `filter.schwartz2f` is called first. Then, the 5% and 95% quantiles are computed and plotted (see fig. 10) for a one week horizon . Note that the 5% quantile is the 95% value-at-risk.

```
> state <- filter.schwartz2f(data = futures$wheat$price,
+     ttm = futures$wheat$ttm/260, wheat.fit.constr)$state
> coefs <- coef(wheat.fit.constr)
> n <- nrow(futures$wheat$price)
> q.fut <- sapply(futures$wheat$ttm[n, ]/260, function(ttm,
+     ...) qfutures(ttm = ttm, ...), p = c(0.05,
+     0.95), time = 5/260, s0 = state[n, 1], delta0 = state[n,
+     2], mu = coefs$mu, sigmaS = coefs$sigmaS,
+     kappa = coefs$kappa, alpha = coefs$alpha,
+     sigmaE = coefs$sigmaE, rho = coefs$rho, r = coefs$r,
+     lambda = coefs$lambda)
> plot(futures$wheat$ttm[n, ], futures$wheat$price[n,
+     ], ylim = c(650, 850), type = "b", xlab = "Time to maturity [d]",
+     ylab = "Price")
> points(futures$wheat$ttm[n, ], q.fut[1, ], col = "blue",
+     type = "b")
> points(futures$wheat$ttm[n, ], q.fut[2, ], col = "blue",
+     type = "b")
> legend("topleft", c("Current observed futures price",
+     "One week ahead 90% confidence interval"),
+     fill = c("black", "blue"))
```