Mathematical Finance

Solution sheet 13

Define the convex conjugate of U as

$$V(y) = \sup_{x \in \mathrm{dom}(U)} \{U(x) - xy\}, \; y > 0.$$

By the Fenchel inequality, $U(x) \leqslant V(y) + xy$ for all $x \in \mathrm{dom}(U)$ and $y > 0$. Hence,

$$E[U(X_T)] \leqslant E\left[V\left(y\frac{dQ}{dP}\right)\right] + E_Q[yX_T] \leqslant E\left[V\left(y\frac{dQ}{dP}\right)\right] + yx,$$

for every $x \in \mathrm{dom}(U)$ and $y > 0$. The wealth process X is a Q-supermartingale for every $\varphi \in \mathcal{A}$ and the equality is achieved for the pair $(\widehat{y}, \widehat{X})$ for which

$$\widehat{y}\frac{dQ}{dP} = U'(\widehat{X}_T) \text{ and } E_Q[\widehat{X}_T] = x.$$

Since U is strictly increasing and strictly concave, we have $\widehat{y} > 0$ for $\widehat{y}\frac{dQ}{dP} = U'(\widehat{X}_T)$.

Solution 13.1 For $U(x) = -e^{-\gamma x}$, the inverse of the marginal utility is $(U')^{-1}(y) = -\frac{1}{\gamma}\log(\frac{y}{\gamma})$ and we have

$$x = E_Q\left[\widehat{X}_T\right] = E\left[\frac{dQ}{dP}(U')^{-1}\left(\widehat{y}\frac{dQ}{dP}\right)\right] = E\left[\frac{dQ}{dP}\left(-\frac{1}{\gamma}\log\left(\frac{\widehat{y}}{\gamma}\frac{dQ}{dP}\right)\right)\right],$$

so,

$$\widehat{y} = \gamma e^{-\gamma x - E[\frac{dQ}{dP}\log(\frac{dQ}{dP})]}.$$

In the Black-Scholes, the equivalent martingale measure is given by

$$\frac{dQ}{dP} = \exp\left(-\int_0^T \frac{\mu - r}{\sigma} dW_t - \int_0^T \frac{(\mu - r)^2}{2\sigma^2} dt\right). \tag{1}$$

We have

$$\begin{aligned}
\widehat{X}_T = x + \int_0^T \varphi_t dS_t &= -\frac{1}{\gamma}\log(\frac{\widehat{y}}{\gamma}\frac{dQ}{dP}) = -\frac{\log(\widehat{y}/\gamma)}{\gamma} + \int_0^T \frac{(\mu - r)}{\gamma\sigma} dW_t + \int_0^T \frac{(\mu - r)^2}{2\gamma\sigma^2} dt \\
&= x + \int_0^T \frac{\mu - r}{\gamma\sigma^2}\frac{1}{S_t}(S_t\sigma dW_t + S_t(\mu - r)dt) = x + \int_0^T \frac{\mu - r}{\gamma\sigma^2}\frac{1}{S_t} dS_t.
\end{aligned}$$

Thus, $\widehat{\varphi}_t = \frac{\mu - r}{\gamma\sigma^2}\frac{1}{S_t}$, so, the amount of money invested in the stock $\widehat{\alpha}_t = \frac{\mu - r}{\gamma\sigma^2}$ is constant.

Solution 13.2 For $U(x) = \frac{x^\gamma}{\gamma}$, the inverse of the marginal utility is $(U')^{-1}(y) = y^{-\delta}$ with $\delta = \frac{1}{1-\gamma}$.

Denote by Z the density process of Q w.r.t. P. The optimal wealth process is

$$\begin{aligned}\widehat{X}_t &= E_Q[\widehat{y}^{-\delta} Z_T^{-\delta} \mid \mathcal{F}_t] = \widehat{y}^{-\delta} E_Q\left[\exp\left(\int_0^T \frac{\mu - r}{\sigma}\delta dW_t - \frac{1}{2}\int_0^T \frac{(\mu-r)^2}{\sigma^2}\delta dt\right) \mid \mathcal{F}_t\right] \\ &= \widehat{y}^{-\delta}\exp\left(\frac{1}{2}\int_0^T \frac{(\mu-r)^2}{\sigma^2}\delta(\delta-1)dt\right) E_Q\left[\exp\left(\int_0^T \frac{\mu-r}{\sigma}\delta dW_t - \frac{1}{2}\int_0^T \frac{(\mu-r)^2}{\sigma^2}\delta^2 dt\right) \mid \mathcal{F}_t\right] \\ &= \widehat{y}^{-\delta}\exp\left(\frac{1}{2}\int_0^T \frac{(\mu-r)^2}{\sigma^2}\delta(\delta-1)dt\right)\exp\left(\int_0^t \frac{\mu-r}{\sigma}\delta dW_u - \frac{1}{2}\int_0^t \frac{(\mu-r)^2}{\sigma^2}\delta^2 du\right).\end{aligned}$$

By writing that $\widehat{X}_0 = x$, we get $x = \widehat{y}^{-\delta}\exp\left(\frac{1}{2}\int_0^T \frac{(\mu-r)^2}{\sigma^2}\delta(\delta-1)dt\right)$, and so

$$\widehat{X}_t = x\exp\left(\int_0^t \frac{\mu-r}{\sigma}\delta dW_u - \frac{1}{2}\int_0^t \frac{(\mu-r)^2}{\sigma^2}\delta^2 du\right) =: M_t,\ 0 \leqslant t \leqslant T.$$

As in the previous exercise, identifying the dynamics of M with the dynamics of the wealth process $\widehat{X}$, we get the optimal portfolio in terms of the number of shares $\widehat{\varphi}_t = \frac{(\mu-r)\delta M_t}{\sigma^2 S_t} = \frac{\mu-r}{\sigma^2(1-\gamma)}\frac{\widehat{X}_t}{S_t}$, so, the proportion of wealth invested in the stock $\widehat{\alpha}_t = \frac{\mu-r}{\sigma^2(1-\gamma)}$ is constant.

Solution 13.3 For $U(x) = \log(x)$, the inverse of the marginal utility is $(U')^{-1}(y) = \frac{1}{y}$. Denote by Z the density process of Q w.r.t. P. The optimal wealth process is

$$\widehat{X}_t = E_Q\left[\frac{1}{\widehat{y}Z_T}|\mathcal{F}_t\right] = E\left[\frac{Z_T}{Z_t}\frac{1}{\widehat{y}Z_T}|\mathcal{F}_t\right] = \frac{1}{\widehat{y}Z_t} =: M_t,\ 0 \leqslant t \leqslant T,$$

where $\widehat{y}$ is such that $\widehat{X}_0 = x$, so, $\widehat{y} = 1/x$. Hence, $\widehat{X}_t = x/Z_t$, $0 \leqslant t \leqslant T$. Now, from (1), we deduce, by Itô's formula, that

$$dM_t = M_t\frac{\mu-r}{\sigma}dW_t.$$

As in the previous exercise, identifying the dynamics of M with the dynamics of the wealth process $\widehat{X}$, we get the optimal portfolio in terms of the number of shares $\widehat{\varphi}_t = \frac{\mu-r}{\sigma^2}\frac{\widehat{X}_t}{S_t}$, so, the proportion of wealth invested in the stock $\widehat{\alpha}_t = \frac{\widehat{\varphi}_t S_t}{\widehat{X}_t} = \frac{\mu-r}{\sigma^2}$ is constant.

Solution 13.4

```
import numpy
import matplotlib.pyplot as plt

from brownian import brownian


def main():

    # The interest rate.
    r = 0.0
    # The mean value return rate.
    mu = 0.06
    # The Wiener process parameter.
    delta = 0.4
    # Total time.
```

```
    T = 1.0
    # Number of steps.
    N = 1000
    # Time step size
    dt = T/N
    # Number of realizations to generate.
    m = 1
    # Create an empty array to store the realizations.
    x = numpy.empty((m,N+1))
    y = numpy.empty((m,N+1))
    z = numpy.empty((m,N+1))
    w = numpy.empty((m,N+1))
    # Initial values of x,y,z,w.
    x[:, 0] = 0
    y[:, 0] = 0
    z[:, 0] = 0
    w[:, 0] = 0

    # Simulate the paths
    brownian(x[:,0], N, dt, delta, out=x[:,1:])

    # Parametrize the utility functions
    alpha = 1.
    gamma = .5

    # Compute the wealth process for exp, power and log utility
    y = 1.0+(mu-r)/(alpha*delta**2.)*(x+numpy.cumsum(dt*numpy.ones((m,N+1))))
    z = numpy.exp( (mu-r)/((1.-gamma)*delta)*x+((1.-2.*gamma)/(2.*(1.-gamma**2.)))*((mu-r)**2/(delta)**2)*numpy.cumsum(dt*numpy.ones((m,N+1))))
    w = numpy.exp( (mu-r)/(delta)*x+.5*((mu-r)**2/(delta)**2)*numpy.cumsum(dt*numpy.ones((m,N+1))))

    t = numpy.linspace(0.0, N*dt, N+1)
    for k in range(m):
  plt.step(t, y[k])
    for k in range(m):
  plt.step(t, z[k])
    for k in range(m):
  plt.step(t, w[k])
    plt.show()
```