Notes on the BLCOP Package

Francisco Gochez, Mango Solutions

August 20, 2008

1 Introduction

The BLCOP package is an implementation of the Black-Litterman and copula opinion pooling frameworks. The current release (0.2.2) should be considered as a beta version, and the main purpose of this release is to allow the community to use it and suggest improvements (any feedback would be greatly appreciated). This vignette gives an overview of these two opinion-blending methods, briefly shows how they are implemented in this package, and closes with a short discussion of how the package may evolve in the future.

2 Overview of the Black-Litterman model

The Black-Litterman model was devised in 1992 by Fisher Black and Robert Litterman. Their goal was to create a systematic method of specifying and then incorporating analyst/portfolio manager views into the estimation of market parameters. Let $A = \{a_1, a_2, ..., .a_n\}$ be a set of random variables representing the returns of n assets. In the BL approach, the joint distribution of A is taken to be multivariate normal, i.e. $A \sim N(\mu, \Sigma)$. The problem they then addressed was that of incorporating an analyst's views into the estimation of the market mean μ [1]. Suppose that we take μ itself to be a random variable which is itself normally distributed, and moreover that its dispersion is proportional to that of the market. Then

$$\mu \sim N(\pi, \tau\Sigma),$$

and π is some underlying parameter which can be determined by the analyst using some established procedure. Black and Litterman argued from equilibrium considerations that this should be obtained from the intercepts of the capital-asset pricing model.

Next, the analyst forms subjective views on the actual mean of the returns for the holding period. This is the part of the model that allows the analyst/portfolio manager to include his or her views. BL proposed that views should be made on linear combinations (i.e. portfolios) of the asset return variable means μ. Each view would take the form of a "mean plus error". Thus for example, a typical view would look as follows:

$$p_{i1}\mu_1 + p_{i2}\mu_2 + ... + p_{in}\mu_n = q_i + \epsilon_i$$

, where $\epsilon_i \sim N(0, sigma_i^2)$. The standard deviations σ_i^2 of each view could be taken as controlling the confidence in each view. Collecting these views into a matrix we will call the "pick" matrix, we obtain the "general" view specification

$$P\mu \sim N(\mu, \Omega).$$

Ω is the diagonal matrix $diag(\sigma_1^2, \sigma_2^2, ..., \sigma_2^n)$. It can be shown (c.f. [Me08], p.5 and appendix), based on Bayes' Law, that the posterior distribution of the market mean conditional on these views is

$$\mu|_{q;\Omega} \sim N(\mu_{BL}, \Sigma_{BL}^{\mu})$$

where

$$\begin{aligned}
\mu_{BL} &= ((\tau\Sigma)^{-1} + P^T\Omega^{-1}P)^{-1}((\tau\Sigma)^{-1}\pi + P^T\Omega^{-1}q) \\
\Sigma_{BL}^{\mu} &= ((\tau\Sigma)^{-1} + P^T\Omega^{-1}P)^{-1}
\end{aligned}$$

We can then obtain the posterior distribution of the *market* by taking $A|_{q,\Omega} = \mu|_{q,\Omega} + Z$, and $Z \sim N(0, \Sigma)$ is independent of μ. One then obtains that $E[A] = \mu_{BL}$ and $\Sigma_{BL} = \Sigma + \Sigma_{BL}^{\mu}$ ([Me08], p. 5).
Let us now see how these ideas are implemented in the `BLCOP` package.

[1] A. Meucci has reformulated the model in terms of forming views directly on market realization rather than the mean, and in my opinion this formulation is considerably clearer. See [Me08]

3 Using the Black-Litterman model in BLCOP

The implementation of the Black-Litterman model in BLCOP is based on objects that represent views on the market and objects that represent the posterior distribution of the market after blending the views. We will illustrate this with a simple example. Suppose that an analyst wishes to form views on 6 stocks, 3 of which are technology stocks and the other 3 of which are from the financial sector. Intially, she has a view on the technology sector and believes that the average of 2 of the stocks will outperform the third, say $\frac{1}{2}(\mathrm{MS} + \mathrm{IBM}) - \mathrm{DELL} \sim N(0.06, 0.01)$. We will create a `BLViews` class object with the `BLViews` constructor function. Its arguments are the "pick" matrix, a vector of confidences, the vector "q", and the the names of the assets in one's "universe".

```
> pickMatrix <- matrix(c(rep(1/2, 2), -1,  rep(0, 3)), nrow = 1, ncol = 6 )
> views <- BLViews(P = pickMatrix, q = 0.06,confidences =  100,
+ assetNames = colnames(monthlyReturns))
> views

1 : 0.5*IBM+0.5*MS+-1*DELL=0.06 . Confidence: 100
```

Next, we need to determine the "prior" distribution of these assets. The analyst may for instance decide to set these means to 0, and then calculate the variance-covariance matrix of these through some standard estimation procedure (e.g. exponentially weighted moving average). Here we use `cov.shrink` from the `corpcov` package.

```
> priorMeans <- rep(0, 6)
> priorVarcov <- unclass(cov.shrink(monthlyReturns))

Estimating optimal shrinkage intensity lambda.var (variance vector): 0.2901
Estimating optimal shrinkage intensity lambda (correlation matrix): 0.0969
```

We can now calculate the posterior market distribution using the `posteriorEst`. This takes as parameters the view object, the prior covariance and mean, and "tau" [2]. The procedure for setting τ is the subject of some controversy in the literature, but here we shall set it to 1/2.

```
> marketPosterior <- posteriorEst(views = views, sigma = priorVarcov,
+  alpha = priorMeans, tau = 1/2)

Prior means:
 IBM   MS DELL    C  JPM  BAC
   0    0    0    0    0    0
Posterior means:
         IBM           MS         DELL            C          JPM          BAC
 0.004182604  0.007079224 -0.027220512  0.003876496  0.003686935  0.002247500
Posterior covariance:
             IBM          MS        DELL           C         JPM         BAC
IBM  0.023465258 0.009561087 0.012734802 0.009014684 0.010499443 0.006264828
MS   0.009561087 0.032600588 0.014685795 0.013606881 0.014508407 0.009583470
DELL 0.012734802 0.014685795 0.038300046 0.007808936 0.009173319 0.005893862
C    0.009014684 0.013606881 0.007808936 0.020016944 0.011931049 0.008204580
JPM  0.010499443 0.014508407 0.009173319 0.011931049 0.029266751 0.013394136
BAC  0.006264828 0.009583470 0.005893862 0.008204580 0.013394136 0.016143364
attr(,"lambda")
[1] 0.0968954
attr(,"lambda.estimated")
[1] TRUE
```

[2] An additional parameter called `kappa` will be discussed shortly

```
attr(,"lambda.var")
[1] 0.2900958
attr(,"lambda.var.estimated")
[1] TRUE
```

Now suppose that we wish to add another view, this time on the average of the three financial stocks. This can be done conveniently with `addBLViews` as in the following example:

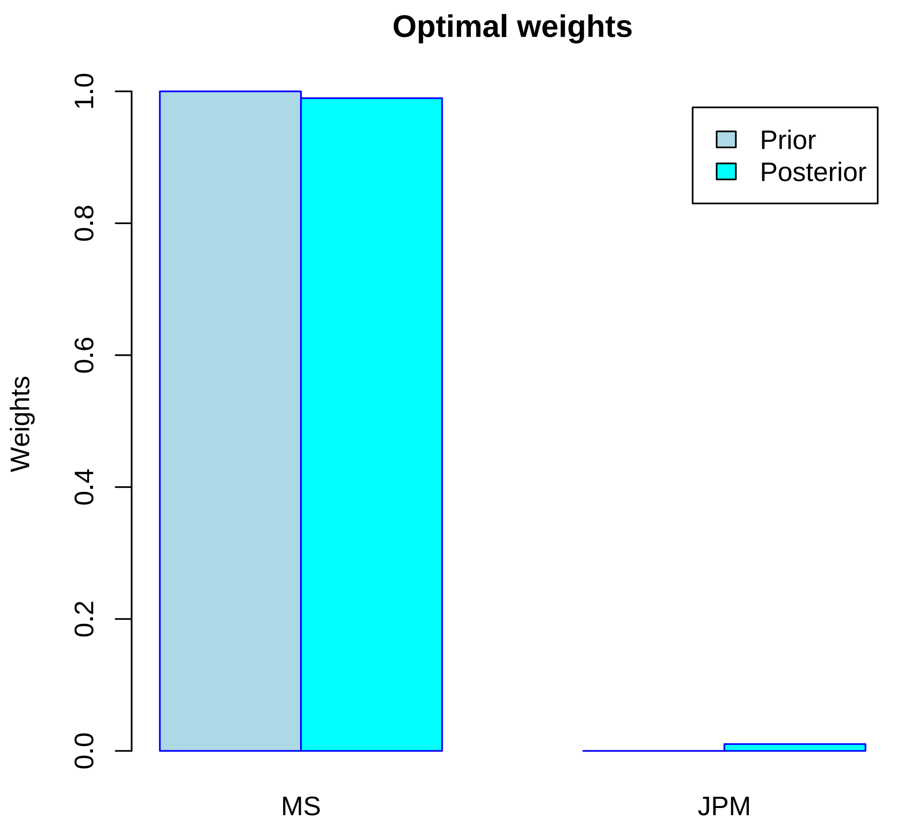
```
> finViews <- matrix(ncol = 3, nrow = 1, dimnames = list(NULL, c("C","JPM","BAC")))
> finViews[,1:3] <- rep(1/3,3)
> views <- addBLViews(finViews, 0.15, 90, views)
> views

1 : 0.5*IBM+0.5*MS+-1*DELL=0.06 . Confidence: 100
2 : 0.333333333333333*C+0.333333333333333*JPM+0.333333333333333*BAC=0.15 . Confidence: 90
```

We will now recompute the posterior, but this time using the captial asset pricing model to compute the “prior” means. Rather than manually computing these, it is convenient to use the `BLPosterior` wrapper function. It will compute these “alphas”, as well as the variance-covariance matrix of a returns series, and will then call `poseriorEst` automatically.

```
> marketPosterior <- BLPosterior(as.matrix(monthlyReturns), views, tau = 1/2,
+         marketIndex = as.matrix(sp500Returns),
+         riskFree = as.matrix(US13wTB))

Estimating optimal shrinkage intensity lambda.var (variance vector): 0.2901
Estimating optimal shrinkage intensity lambda (correlation matrix): 0.0969

Prior means:
        IBM          MS        DELL           C         JPM         BAC
0.020883598 0.059548398 0.017010062 0.014492325 0.027365230 0.002829908
Posterior means:
       IBM         MS       DELL          C        JPM        BAC
0.04684788 0.09800115 0.02453154 0.05360993 0.07975550 0.03889862
Posterior covariance:
             IBM          MS        DELL           C         JPM         BAC
IBM  0.022971538 0.008841343 0.012223893 0.008239472 0.009441728 0.005532868
MS   0.008841343 0.031551347 0.013940993 0.012476779 0.012966473 0.008516421
DELL 0.012223893 0.013940993 0.037771349 0.007006734 0.008078779 0.005136419
C    0.008239472 0.012476779 0.007006734 0.018799748 0.010270283 0.007055297
JPM  0.009441728 0.012966473 0.008078779 0.010270283 0.027000770 0.011826033
BAC  0.005532868 0.008516421 0.005136419 0.007055297 0.011826033 0.015058206
attr(,"lambda")
[1] 0.0968954
attr(,"lambda.estimated")
[1] TRUE
attr(,"lambda.var")
[1] 0.2900958
attr(,"lambda.var.estimated")
[1] TRUE
```

Both `BLPosterior` and `posteriorEst` have a `kappa` parameter which may be used to replace the matrix Ω of confidences in the posterior calculation. If it is greater than 0, then Ω is set to $\kappa P^T \Sigma P$ rather than $diag(\sigma_1^2, \sigma_2^2, ..., \sigma_2^n)$. This choice of Ω is suggested by several authors, and it leads to the confidences being determined by volatilities of the asset returns. A user may also be interested in comparing allocations that are optimal under the prior and posterior distributions. A helper function is provided in the BLCOP package for doing this, and it can use any optimization routine that relies on a distribution's first and second moments. This helper function is optimalPortfolios. If no optimizing function is passed in, it will use solve.QP to perform basic Markowitz optimization (as shown below). A barplot that compares the portfolio weights is optionally produced.

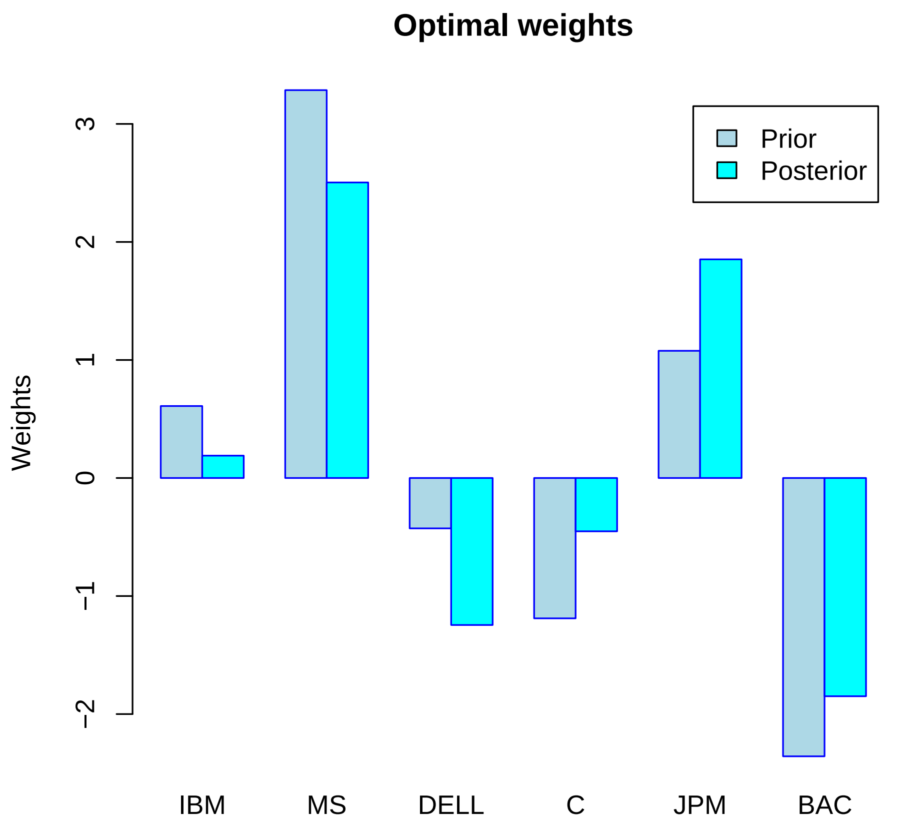
```
> optPorts <- optimalPortfolios(marketPosterior, doPlot = TRUE)

$priorPfolioWeights
 IBM   MS DELL    C  JPM  BAC 
   0    1    0    0    0    0 

$postPfolioWeights
       IBM         MS       DELL          C        JPM        BAC 
0.00000000 0.98960050 0.00000000 0.00000000 0.01039950 0.00000000 
```



Additional parameters may be passed into one's optimizer. In the case of the default optimizer, one can pass in a constraints argument that is used to control the execution of solve.QP. Thus we can remove the "long only" constraint as in the following example:

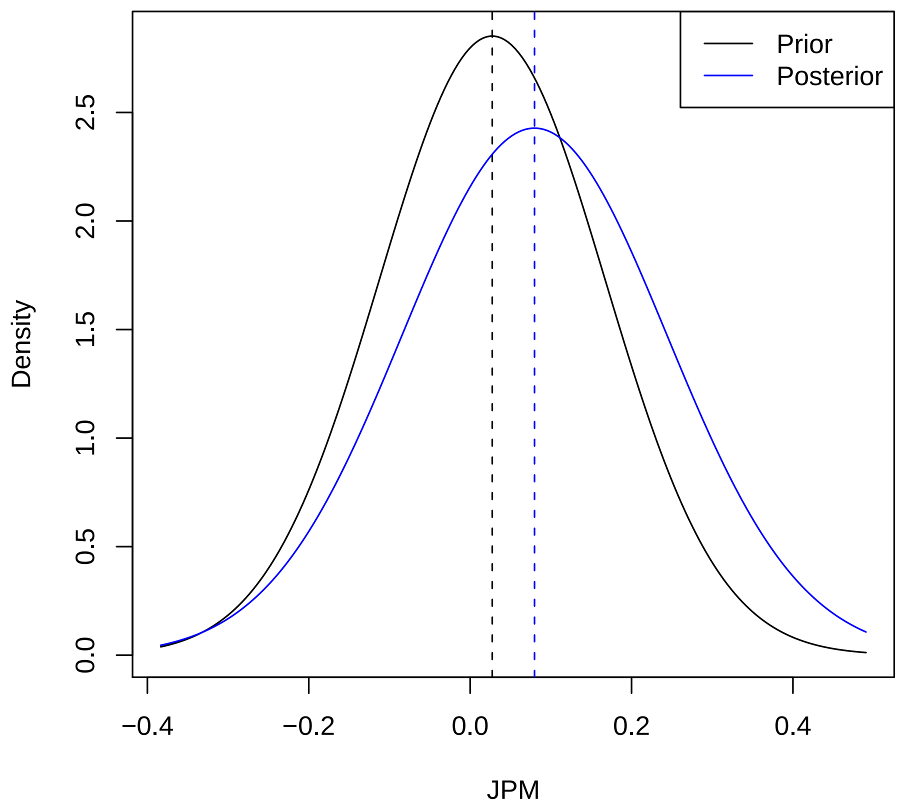
```
> constr <- list()
> constr$bvec <- 1
> constr$meq <- 1
> constr$Amat <- matrix(rep(1, 6), ncol = 1, nrow = 6)
> optimalPortfolios(marketPosterior, constraints = constr)
```

```
$priorPfolioWeights
       IBM         MS       DELL          C        JPM        BAC 
 0.6098815  3.2863069 -0.4268998 -1.1890440  1.0776810 -2.3579256 

$postPfolioWeights
       IBM         MS       DELL          C        JPM        BAC 
 0.1893273  2.5040596 -1.2453713 -0.4516378  1.8526563 -1.8490341 
```



Finally, density plots of marginal prior and posterior distributions can be generated with `densityPlots`. As we will see in the next section, this gives more interesting results when used with copula opinion pooling.

```
> densityPlots(marketPosterior, assetsSel = "JPM")
```



4 Overview of Copula Opinion Pooling

Copula opinion pooling is an alternative way to blend analyst views on market distributions that was developed by Attilio Meucci towards the end of 2005. It is similar to the Black-Litterman model in that it also uses a "pick" matrix to formulate views. However it has several advantages including the following:

- Views are made on realizations of the market, not on market parameters as in the original formulation of BL
- The joint distribution of the market can be any multivariate distribution
- Views are not restricted to the normal distribution
- The parameters in the model have clearer meanings
- The model can easily be generalized to incorporate the views of multiple analysts

Nevertheless, all of this comes at a price. We can no longer use closed-form expressions for calculating the posterior distribution of the market and hence must rely on simulation instead. Before proceeding to the implementation however let us look at the theory. Readers are referred to [Me05] for a more detailed discussion.
As before, suppose that we have a set of n assets whose returns are represented by a set of random variables $A = \{a_1, a_2, ..., a_n\}$. As in Black-Litterman, we suppose that A has some prior joint distribution

whose c.d.f we will denote by Φ_A. Denote the marginals of this distribution by ϕ_i. An analyst forms his views on linear combiniations of future *realizations* of the values of A by assigning subjective probability distributions to these linear combinations. That is we form views of the form $p_{i,1}a_1+p_{i,2}a_2+...+p_{i,n}a_n \sim \theta_i$, where θ_i is some distribution. Denote the pick matrix formed by all of these views by P once again. Now, since we have assigned some prior distribution Φ_A to these assets, it follows that actually the product $V = PA$ inherits a distribution as well, say

$$v_i = p_{i,1}a_1 + p_{i,2}a_2 + ... + p_{i,n}a_n \sim \theta'_i$$

. In general $\theta_i \neq \theta'_i$ unless one's views are identical to the market prior. Thus we must somehow resolve this contradiction. A straightforward way of doing this is to take the weighted sum of the two marginal c.d.fs, so i.e. $\hat{\theta_i} = \tau_i\theta_i + (1-\tau_i)\theta'_i$, and $\tau_i \in [0,1]$ is a parameter representing our confidence in our subjective views. This is the actual marginal distribution that will be used to determine the market posterior.

The market posterior is actually determined by setting the marginals of distributions of V to $\hat{\theta_i}$, while using a copula to keep the dependence structure of V intact. Let $V = (v_1, v_2, ..., v_k)$, where k is the number of views that the analyst has formed. Then $v_i \sim \theta'_i$. Let C be the copula of V so that C is the joint distribution of

$$(\theta'_1(v_1), \theta'_2(v_2), ..., \theta'_k(v_k)) = (C_1, C_2, ..., C_k)$$

if we now take the θ'_i to be cumulative density functions. Next set $\hat{V}$ as the random variable with the joint distribution $(\hat{\theta_1}^{-1}(C_1), \hat{\theta_2}^{-1}(C_2), ..., \hat{\theta_k}^{-1}(C_k))$. The posterior market distribution is obtained by rotating $\hat{V}$ back into market coordinates using the orthogonal complement of P. See [Me05], p. 5 for details.

5 COP in BLCOP

Let us now work through a brief example to see how these ideas are implemented in the BLCOP package. First, one again works with objects that hold the view specification, which in the COP case are of class `COPViews`. These can again be created with a constructor function of the same name. However a significant difference is the use of `mvdistribution` and `distribution` class objects to specify the prior distribution and view distributions respectively. We will show the use of these in the following example, which is based on the example used in [?], p.9. Suppose that we wish to invest in 4 market indices (S&P500, FTSE, CAC and DAX). Meucci suggests a multivariate Student-t distribution with $\nu = 5$ degrees of freedom and dispersion matrix given by:

$$10^{-3}\begin{pmatrix} .376 & .253 & .333 & .397 \\ . & .360 & .360 & .396 \\ . & . & .600 & .578 \\ . & . & . & .775 \end{pmatrix}.$$

He then sets $\mu = \delta\Sigma w_{eq}$ where w_{eq} is the relative capitilization of the 4 indices and $\delta = 2.5$. For simplicity we will simply take $w_{eq} = (1/4, 1/4, 1/4, 1/4)$.

```
> dispersion <- c(.376,.253,.360,.333,.360,.600,.397,.396,.578,.775) / 1000
> sigma <- BLCOP:::.symmetricMatrix(dispersion, dim = 4)
> caps <- rep(1/4, 4)
> mu <- 2.5 * sigma %*% caps
> dim(mu) <- NULL
> marketDistribution <- mvdistribution("mt", mean = mu, S = sigma, df = 5 )
> class(marketDistribution)
```

```
[1] "mvdistribution"
attr(,"package")
[1] "BLCOP"
```

The class `mvdistribution` works with R multivariate probabilty distribution "suffixes". `mt` is the R "name"/"suffix" of the multivariate Student-t as found in the package `mnormt`. That is, the sampling function is given by `rmt`, the density by `dmt`, and so on. The other parameters are those required by the these functions to fully parametrize the multivariate Student-t. The `distribution` class works with univariate distributions in a similar way and is used to create the view distributions. We continue with the above example by creating a single view on the DAX.

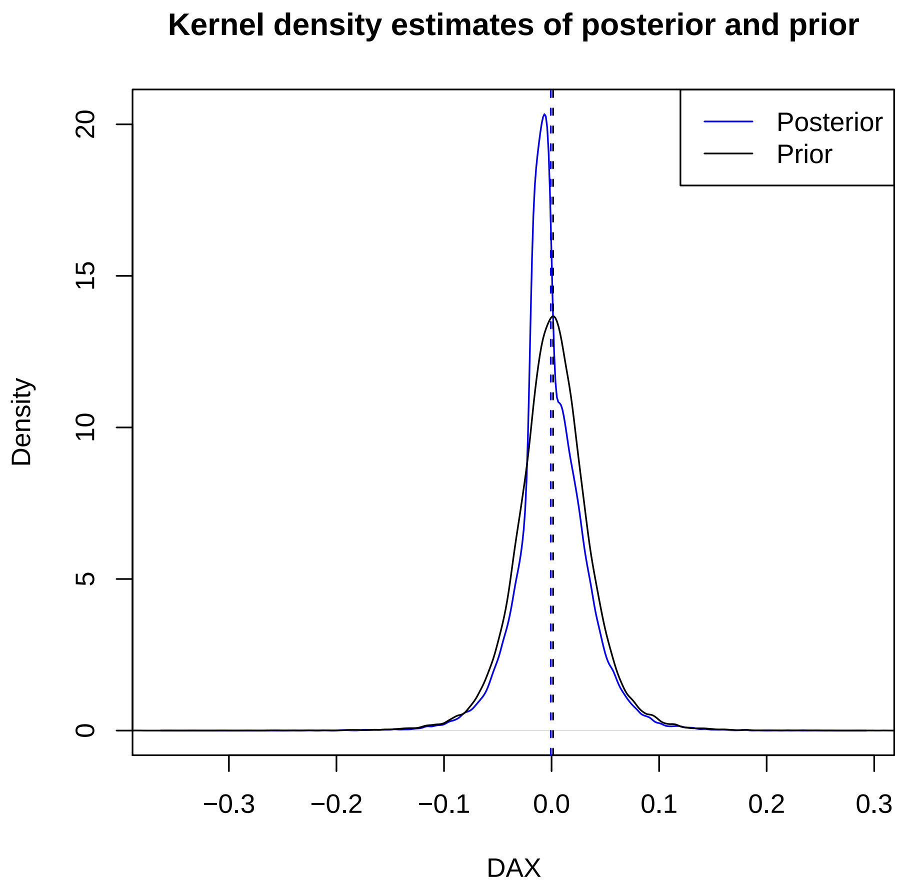
```
> pick <- matrix(0, ncol = 4, nrow = 1,
+         dimnames = list(NULL, c("SP", "FTSE", "CAC", "DAX")))
> pick[1,"DAX"] <- 1
> viewDist <- list(distribution("unif", min = -0.02, max = 0))
> views <- COPViews(pick, viewDist = viewDist, confidences = 0.2,
+         assetNames = c("SP", "FTSE", "CAC", "DAX"))
```

As can be seen, the view distributions are given as a list of `distribution class objects`, and the confidences set the *tau*'s described previously. Here we have assigned a $U(-0.02, 0)$ distribution to our view with confidence 0.2. Additional views can be added with `addCOPViews`.

```
> newPick <- matrix(0, 1, 2)
> dimnames(newPick) <- list(NULL, c("SP", "FTSE"))
> newPick[1,] <- c(1, -1) # add a relative view
> views <- addCOPViews(newPick,
+         list(distribution("norm", mean = 0.05, sd = 0.02)), 0.5, views)
```

Finally, the posterior is calculated with `COPPosterior`, and the updated marginal distributions can be visualized with `densityPlots` once again. The calculation is performed by simulation, based on the ideas described in [Me06]. The simulations of the posterior distribution are stored in the `posteriorSims` of the class `COPResult` that is returned by `COPPosterior`.

```
> marketPosterior <- COPPosterior(marketDistribution, views, numSimulations = 50000)
> densityPlots(marketPosterior, assetsSel = 4)
```



6 Future developments

While mostly stable, the code is currently in need of some minor cleanup work and refactoring (e.g. pick matrices are referred to as `P` in some places and `pick` in others) as well as improvements in the documentation and examples. Aside from this, it would be natural to implement a version of the `optimalPortfolios` function for `COPResult` class objects, possibly based on CVaR optimization. Attilio Meucci has also very recently proposed an even more general view-blending method which he calls *Entropy Pooling* and its inclusion would be another obvious extension of this package's functionality in the longer term.

References

[Me05] Meucci, Attilio. *Beyond Black-Litterman: Views on Non-Normal Markets.* November 2005, Available at SSRN: http://ssrn.com/abstract=848407

[Me06] Meucci, Attilio. *Beyond Black-Litterman in Practice: A Five-Step Recipe to Input Views on non-Normal Markets.* May 2006, Available at SSRN: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=872577

[Me08] Meucci, Attilio. *The Black-Litterman Approach: Original Model and Extensions.* April 2008, Available at SSRN: http://ssrn.com/abstract=1117574