Bayesian Factor Zoo package

Christian Julliard

2025-01-07

Installation

This package implements all linear SDF estimation methods in Bryzgalova, Huang, and Julliard (2023) as well as the Bayesian Fama-MacBeth regressions of Bryzgalova, Huang, and Julliard (2024)

The package, and its documentation, are available here: https://cran.r-project.org/web/packages/BayesianFactorZoo/index.html.

It can be automatically installed into R with the command:

```
install.packages("BayesianFactorZoo")
```

And to use it with your code just use the command:

```
library(BayesianFactorZoo)
```

Here we'll focus on practical applications. For the theory behind the methods see the papers.

The R Markdown file with the example codes, as well as introductory teaching (Latex) slides, and Julia and Python implementations of the methodology, are available at: https://christianjulliard.net/bayesian-factor-zoo-package

Bayesian Fama-MacBeth Regressions

Let's start with the simple Bayesian estimation of Fama-MacBeth regressions with **BayesianFM().**

This estimates observable factor **risk premia** (recall that a factor might have a non-zero risk premium even if it's not a fundamental source of risk, just because it correlates with the latter).

For more details and examples in R run:

```
?BayesianFM
```

Let's load the example data and a few libraries we'll use:

```
library(reshape2)
library(ggplot2)
library(timeSeries)

# Load Data
data("BFactor_zoo_example")
lambda_ols <- BFactor_zoo_example$lambda_ols # pseudo-true values of the risk premia
R2.ols.true <- BFactor_zoo_example$R2.ols.true # pseudo-true cross-sectional R^2
sim_f <- BFactor_zoo_example$sim_f # the strong, psuedo-true, factor in the examples
sim_R <- BFactor_zoo_example$sim_R # the sample test assets
```

Missing libraries can be installed with:

```
install.packages("The name of the library you want to install")
```

BFM continued: Case 1 – a strong factor

Lets consider first a strong factor (one that does not cause identification problems). And let's also perform the frequentist Fama-MacBeth estimation for comparison.

- sim_f: simulated factor, sim_R: simulated return
- sim_f is the useful (i.e., strong) factor

```
# the Frequentist Fama-MacBeth
results.fm <- Two_Pass_Regression(sim_f, sim_R)
# the Bayesian Fama-MacBeth with 10000 simulations
results.bfm <- BayesianFM(sim_f, sim_R, 10000) # 10000 is the number of MCMC draws
```

BFM Case 1 – a strong factor, cont'd

Note that the first element of the estimated parameters corresponds to lambda of the constant term.

So, we set k=2 to get the lambda of the strong factor

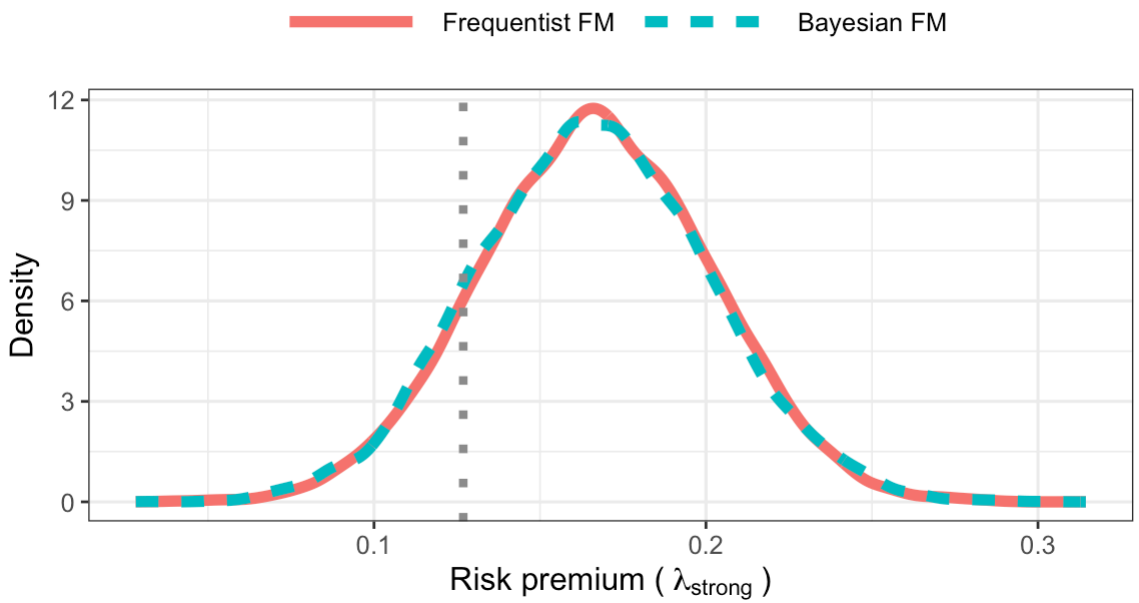
```
k <- 2
m1 <- results.fm$lambda[k]
sd1 <- sqrt(results.fm$cov_lambda[k,k])
bfm<-results.bfm$lambda_ols_path[1001:10000,k] # for good practice, we are dropping the starting MCMC draws
fm<-rnorm(20000,mean = m1, sd=sd1) # the Gaussian asymptotic benchmark

# and put the two types of estimates together
data<-data.frame(cbind(fm, bfm))
colnames(data)<-c("Frequentist FM", "Bayesian FM")
data.long<-melt(data)
```

BFM Case 1 – a strong factor, cont'd

Let's plot the estimates

```
p <- ggplot(aes(x=value, colour=variable, linetype=variable), data=data.long)
p+ stat_density(aes(x=value, colour=variable), geom="line",position="identity", size = 2, adjust=1) +
geom_vline(xintercept = lambda_ols[2], linetype="dotted", color = "#8c8c8c", size=1.5)+
guides(colour = guide_legend(override.aes=list(size=2), title.position = "top", title.hjust = 0.5,nrow=1,byrow=TRUE))+ theme_bw()+
labs(color=element_blank()) + labs(linetype=element_blank()) + theme(legend.key.width=unit(4,"line")) +
theme(legend.position="top")+ theme(text = element_text(size = 12))+ xlab(bquote("Risk premium ("~lambda[strong]~")")) + ylab("Density" )
```



Pretty similar - as it should be!

BFM Case 2 – a useless (weak) factor

Let's repeat the exercise with a useless factor (that is not correlated with the test assets)

- uf is the useless factor

As before, let's perform frequentist and Bayesian estimations and store the results

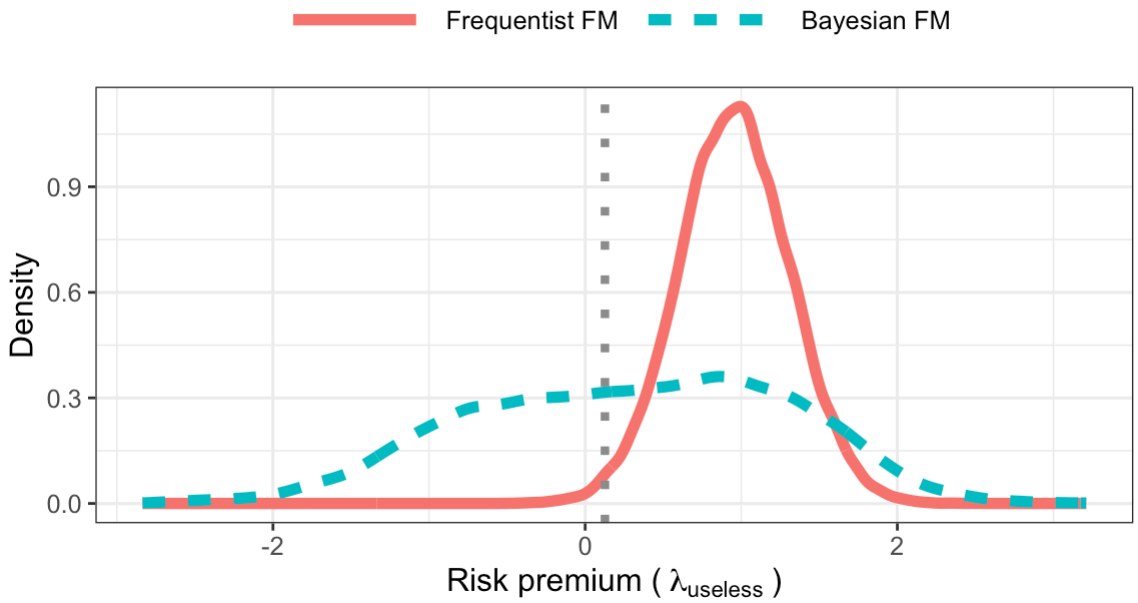
```
uf <- BFactor_zoo_example$uf # the useless (weak) factor in the examples

results.fm <- Two_Pass_Regression(uf, sim_R) # Frequentist benchmark
results.bfm <- BayesianFM(uf, sim_R, 10000) ## BFM estimation

k <- 2
m1 <- results.fm$lambda[k]
sd1 <- sqrt(results.fm$cov_lambda[k,k])
bfm<-results.bfm$lambda_ols_path[1001:10000,k]
fm<-rnorm(20000,mean = m1, sd=sd1)
data<-data.frame(cbind(fm, bfm))
colnames(data)<-c("Frequentist FM", "Bayesian FM")
data.long<-melt(data)
```

BFM Case 2 – a useless (weak) factor, cont'd

And let's plot the results (same code as before)



Pretty different! The frequentist estimator is away from zero and significant, while BFM is well centered around the pseudo-true value

BFM extensions

The Bayesian Fama-MacBeth estimator can also

- perform GLS estimation
- account for omitted variables (same spirit as Giglio and Xiu (2021))

See the paper Bryzgalova, Huang, and Julliard (2024) for details.

An independently mantained Python implementation of these methods is available here: https://github.com/gusamarante/bayesfm

Bayesian estimation of Linear SDF (B-SDF)

The **BayesianSDF()** function provides the Bayesian estimates of factors' **risk prices** (market prices of risk – i.e., what we care about in the quest for fundamental sources of risk) for a **single model** (see Definitions 1 and 2 in Bryzgalova, Huang, and Julliard (2023)).

For more details and examples:

```
?BayesianSDF
```

We'll now perform an example of risk prices and R^2 estimation (same example as in "Section III. Simulation" of the paper)

```
# Load additional example data
W_ols <- BFactor_zoo_example$W_ols # Identity weighting matrix for GMM-OLS
```

```
## Cross-section: Fama-French 25 size and value portfolios
```

```
## True pricing factor in simulations: HML
```

```
## Pseudo-true cross-sectional R-squared: 0.438728
```

```
## Pseudo-true (monthly) risk price: 0.1268817
```

B-SDF continued

We'll use GMM as frequentist example estimator for comparison

```
sim_result <- SDF_gmm(sim_R, sim_f, W_ols) # GMM estimation
two_step <- BayesianSDF(sim_f, sim_R, sim_length = 10000, psi0 = 5, d = 0.5) ## estimate using the Bayesian method
```

The first entry that we pass to the function is the matrix of factors, the second is the set of test assets, the third is the number of MCMC draws to compute, and the remaining are prior parameters (discussed later).

Extracting the posterior mean and CIs for the Bayesian estimates is straightforward:

```
print(mean(two_step$lambda_path[,k])) # again, the first parameter is the intercept, so we look at k=2
```

```
## [1] 0.1633438
```

```
print(quantile(two_step$lambda_path[,k], probs = c(0.05, 0.5, 0.95))) # summarize the posterior of the MPR
```

```
##         5%       50%       95%
## 0.1060847 0.1632952 0.2219416
```

```
print(quantile(two_step$R2_path, probs = c(0.05, 0.5, 0.95))) # summarize the posterior of the R^2
```

```
##         5%       50%       95%
## 0.3013073 0.5402523 0.7053578
```

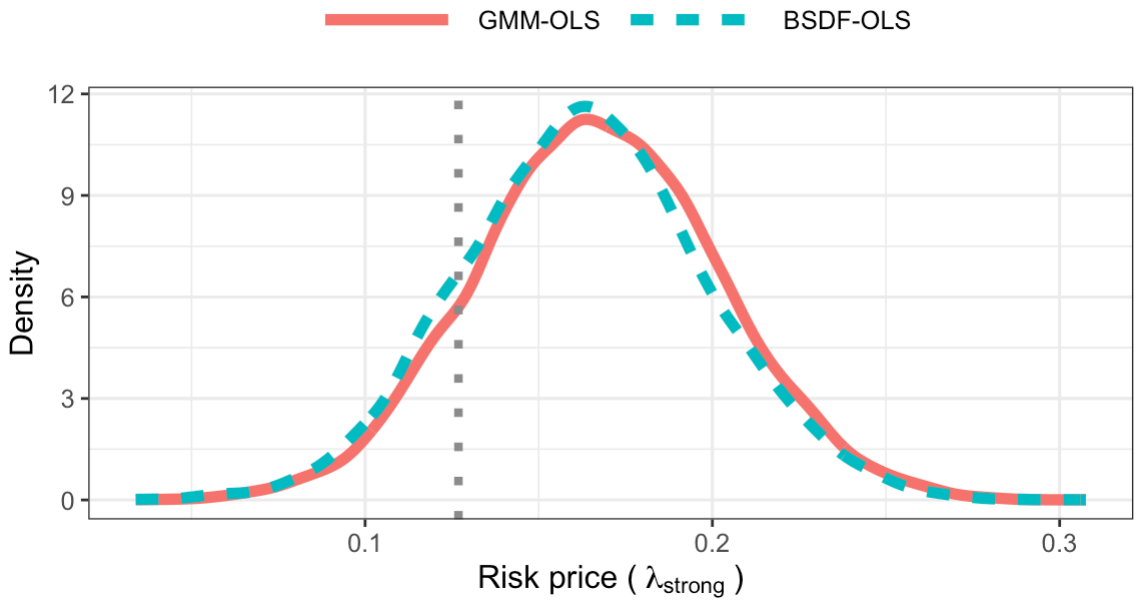
B-SDF with a strong factor – cont'd

Let's extract the results …

```
bsdf<-two_step$lambda_path[1001:10000, k] # again, we look at k=2, as the first entry is the intercept, and we drop the starting MCMC draws
m1 <- sim_result$lambda_gmm[k]
sd1 <- sqrt(sim_result$Avar_hat[k,k])
gmm<-rnorm(10000,mean = m1, sd=sd1) # the frequentist Gaussian asymptotic benchmark
data<-data.frame(cbind(gmm, bsdf))
colnames(data)<-c("GMM-OLS", "BSDF-OLS")
data.long<-melt(data)
```

B-SDF with a strong factor – cont'd

… and plot them (using the same code as before)



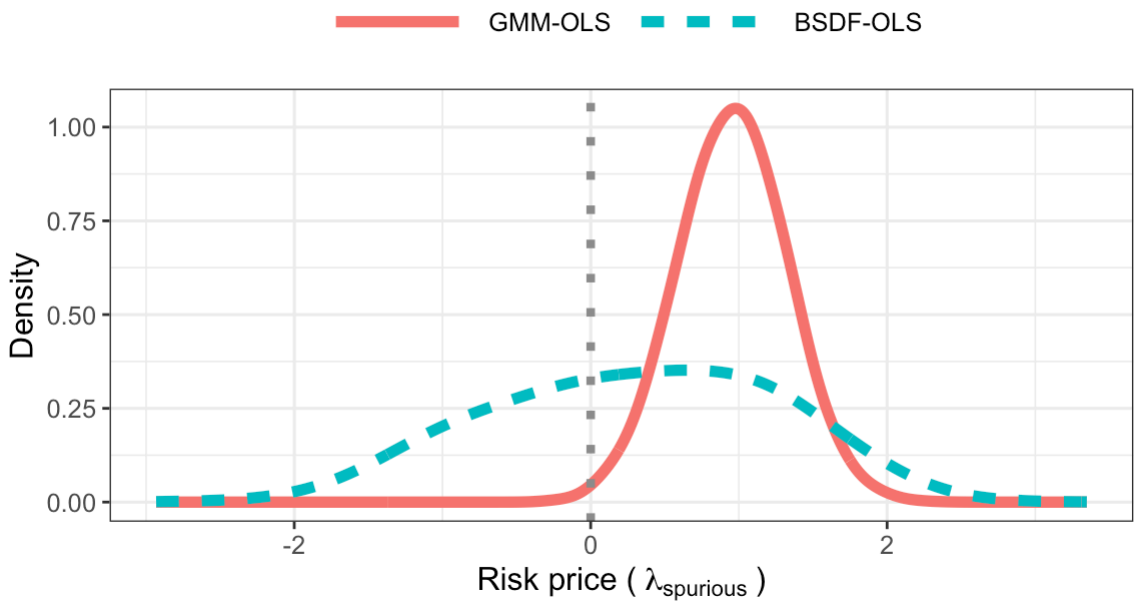
Pretty close - as it should be!

B-SDF with a useless factor

Let's repeat the estimations with the usless factor (uf):

```
sim_result <- SDF_gmm(sim_R, uf, W_ols) # GMM estimation
## Now estimate the model using the Bayesian method
two_step <- BayesianSDF(uf, sim_R, sim_length = 10000, psi0 = 5, d = 0.5)
```

and extract and plot exactly as before



The frequentist estimate is sigificantly away from zero, while the B-SDF posterior is diffused and centered at 0.

SDF model selection and aggregation with continuous spike-and-slab prior

```
?continuous_ss_sdf
```

This function provides SDF model selection using the continuous spike-and-slab prior (Propositions 3 and 4 in Bryzgalova, Huang, and Julliard (2023)). It considers all the possible models obtainable with the provided factors.

We'll discuss the formulation of the prior later. Here we'll just use a function (discussed later) to set the key prior value psi_hat.

```
psi_hat <- psi_to_priorSR(sim_R, cbind(sim_f,uf), priorSR=0.1) # changing the prior (discussed later)
```

Let's consider both the strong, sim_f, and the useless, ul, factors (first entry provided to the function):

```
shrinkage <- continuous_ss_sdf(cbind(sim_f,uf), sim_R, 10000, psi0=psi_hat, r=0.001, aw=1, bw=1)
```

```
## Null hypothesis: lambda = 0 for each factor
```

```
cat("Posterior probabilities of rejecting the above null hypotheses are:",
       colMeans(shrinkage$gamma_path), "\n")
```

```
## Posterior probabilities of rejecting the above null hypotheses are: 0.9457 0.5175
```

Hence, with the Bayesian method one would select the pseudo-true factor and discard the useless and the spurious (note that the probability for the uselsses factor is reverting to the 50% prior, as it should).

SDF model selection and aggregation cont'd

We also have the posterior draws of the SDF:

$$m_t = 1 - (f_t - \mu_f)^\top \lambda_f$$

```
        sdf_path <- shrinkage$sdf_path
```

As well as the Bayesian model averaging of the SDF (BMA-SDF) of all possible models obtainable with the factors provided

```
        bma_sdf <- shrinkage$bma_sdf
```

We can further estimate the posterior distributions of the model-implied Sharpe ratios (across all possible models given the factors provided):

```
        cat("The 5th, 50th, and 95th quantiles of model-implied Sharpe ratios:",
                quantile(colSds(t(sdf_path)), probs=c(0.05, 0.5, 0.95)), "\n")
```

```
## The 5th, 50th, and 95th quantiles of model-implied Sharpe ratios: 0.004386482 0.1067467 0.1720776
```

SDF with spike-and-slab cont'd

The _v2 version of the function: unlike with continuous_ss_sdf, tradable factors are treated as additional test assets (see Propositions 3 and 4 in Bryzgalova, Huang, and Julliard (2023), and also Barillas and Shanken (2016))

```
?continuous_ss_sdf_v2 # for more details
```

Let's use it for the next examples.

The syntax of **continuous_ss_sdf_v2()** sets the first 3 entries as:

- f1 = A matrix of nontradable factors
- f2 = A matrix of tradable factors (for which we'll require self-pricing)
- R = A matrix of test assets (not including f2)

We include the first test asset, sim_R[,1], into the set of factors, so f2 = sim_R[,1,drop=FALSE].

We'll have to remove it from the set of simple test assets, so R = sim_R[,-1]

```
shrinkage <- continuous_ss_sdf_v2(cbind(sim_f,uf), sim_R[,1,drop=FALSE], sim_R[,-1], 10000,
                                  psi0=psi_hat, r=0.001, aw=1, bw=1)
```

SDF with spike-and-slab cont'd

```
cat("Null hypothesis: lambda =", 0, "for each of these three factors", "\n")
```

```
## Null hypothesis: lambda = 0 for each of these three factors
```

```
cat("Posterior probabilities of rejecting the above null hypotheses are:",
    colMeans(shrinkage$gamma_path), "\n")
```

```
## Posterior probabilities of rejecting the above null hypotheses are: 0.948 0.491 0.472
```

Again appropriate selection.

```
sdf_path <- shrinkage$sdf_path
cat("The 5th, 50th, and 95th quantiles of model-implied Sharpe ratios:",
    quantile(colSds(t(sdf_path)), probs=c(0.05, 0.5, 0.95)), "\n")
```

```
## The 5th, 50th, and 95th quantiles of model-implied Sharpe ratios: 0.02363795 0.09641932 0.1577847
```

Hence the posterior estimate of the SR achievable in the economy has not changed (as it should be).

SDF with spike-and-slab cont'd

Finally, we can estimate the posterior distribution of model dimensions:

```
cat("The posterior distribution of model dimensions (= 0, 1, 2, 3):",
    prop.table(table(rowSums(shrinkage$gamma_path))), "\n")
```

```
## The posterior distribution of model dimensions (= 0, 1, 2, 3): 0.0109 0.2894 0.4775 0.2222
```

Note that the market prices of risk for the useless (3rd entry in the vector of parameters) and the portfolios added as factor (4th entry) have no impact on the BMA-SDF as their MPRs estimates are nicely concetrated around zero.

```
print(quantile(shrinkage$lambda_path[,3], probs = c(0.05, 0.5, 0.95))) ## Posterior distribution of ul MPR
```

```
##            5%           50%           95%
## -2.343145e-03  2.078991e-07  2.397930e-03
```

```
print(quantile(shrinkage$lambda_path[,4], probs = c(0.05, 0.5, 0.95))) ## Posterior distribution of f2 MPR
```

```
##            5%           50%           95%
## -0.0451431488 -0.0002434319  0.0258111867
```

While the posterior of the pseudo-true factor is away from zero and well centred

```
##          5%         50%         95%
## 0.002142331 0.093257061 0.154142008
```

SDF with spike-and-slab cont'd

Let's repeat the experiment using the 17th test asset as tradable factor, so f2 = sim_R[,17,drop=FALSE].

And we should remember to remove it from the test assets, so R = sim_R[,-17]

```
shrinkage <- continuous_ss_sdf_v2(cbind(sim_f,uf), sim_R[,17,drop=FALSE], sim_R[,-17],
                                  10000, psi0=psi_hat, r=0.001, aw=1, bw=1)
```

```
## Null hypothesis: lambda = 0 for each of these three factors
```

```
cat("Posterior probabilities of rejecting the above null hypotheses are:",
    colMeans(shrinkage$gamma_path), "\n")
```

```
## Posterior probabilities of rejecting the above null hypotheses are: 0.9619 0.4814 0.5188
```

Again, appropriate selection.

SDF point hypothesis testing with Dirac spike-and-slab prior

```
?dirac_ss_sdf_pvalue
```

This function tests the null hypothesis $H_0: \lambda = \lambda_0$ when $\gamma = 0$ (the spike).

When $\lambda_0 = 0$, we compare factor models using the algorithm in Proposition 1 of Bryzgalova, Huang, and Julliard (2023).

When $\lambda_0 \neq 0$, this function corresponds to Corollary 2 in Section II.A.2 of the paper.

The function can also be used to compute the posterior probabilities of all possible models with up to a **given maximum number of factors** (see examples)

This function evaluates all models individually, rather than running a Markov Chain over the space of all possible models (as the continuous_ss_sdf() functions do), hence it is less suited for handling quadrillion of models (but can be feasibly used for millions and billions of specifications)

In this example we'll use the calibrated HML as the pseudo-true factor

```
HML <- BFactor_zoo_example$HML
```

SDF with Dirac spike-and-slab prior cont'd

Now we estimate the **Bayesian p-values** defined in Corollary 2 for the MPR of the factors.

Let's test the null of the MPR for HML (the strong factor) being equal to the pseudo-true value (matrix(lambda_ols[2]*sd(HML))

```
shrinkage <- dirac_ss_sdf_pvalue(sim_f, sim_R, 10000, matrix(lambda_ols[2]*sd(HML),ncol=1))
cat("Null hypothesis: lambda =", matrix(lambda_ols[2]*sd(HML)), "\n")
```

```
## Null hypothesis: lambda = 0.1268817
```

```
cat("Posterior probability of rejecting the above null hypothesis is:",
    mean(shrinkage$gamma_path), "\n")
```

```
## Posterior probability of rejecting the above null hypothesis is: 0.0153
```

Hence we'd be unlikely to reject it.

SDF with Dirac spike-and-slab prior cont'd

Let's test whether the risk price of the strong factor, sim_f, is equal to zero

```
shrinkage <- dirac_ss_sdf_pvalue(sim_f, sim_R, 10000, 0, psi0=1)
cat("Null hypothesis: lambda =", 0, "\n")
```

```
## Null hypothesis: lambda = 0
```

```
cat("Posterior probability of rejecting the above null hypothesis is:",
    mean(shrinkage$gamma_path), "\n")
```

```
## Posterior probability of rejecting the above null hypothesis is: 0.9627
```

Hence we would reject it.

SDF with Dirac spike-and-slab prior cont'd

One can also put more than one factor into the test.

Let's consider the strong, sim_f, and useless, uf, factors jointly.

```
two_f <- cbind(sim_f,uf) # sim_f is the strong factor while uf is the useless factor
```

And test the null of the their MPR being equal to zero

```
lambda0_null_vec <- t(cbind(0,0)) # 2x1 vector
shrinkage <- dirac_ss_sdf_pvalue(two_f, sim_R, 10000, lambda0_null_vec, psi0=1)
```

```
## Null hypothesis: lambda = 0 for each factor
```

```
cat("Posterior probabilities of rejecting the above null hypothesis are:",
    colMeans(shrinkage$gamma_path), "\n")
```

```
## Posterior probabilities of rejecting the above null hypothesis are: 0.9636 0.5049
```

Again, appropriate selection

SDF with Dirac spike-and-slab prior cont'd

This function is handy for computing the posterior probabilities of all possible models with up to a **given maximum number of factors** (set by the variable max_k)

For example, we consider our two factors, but the number of factors in the SDF is restricted to be less than two (we impose this by specifying max_k=1, i.e. 1 is the maximum number of factors in each model)

```
shrinkage <- dirac_ss_sdf_pvalue(two_f, sim_R, 10000, lambda0_null_vec, psi0=1, max_k=1)
cat('Posterior model probabilities are:\n')
```

```
## Posterior model probabilities are:
```

```
print(shrinkage$model_probs)
```

```
##          models_probs
## [1,] 0 0       0.0363
## [2,] 1 0       0.9308
## [3,] 0 1       0.0329
```

Again, appropriate selection

Setting economic priors

Estimation of stand-alone models (via **BayesianFM()**, and **BayesianSDF()**) is reliable and robust even with flat (improper) priors. Nevertheless, as in the frequentist case, model comparison is unreliable under flat priors in the presence of weak factors (see Bryzgalova, Huang, and Julliard (2023)).

Intuition: an unidentified parameter generates a flat manifold for the likelihood, hence the marginal (aka, integrated) likelihood is not well defined. Hence, a non-flat prior restores integrability and, consequently, inference.

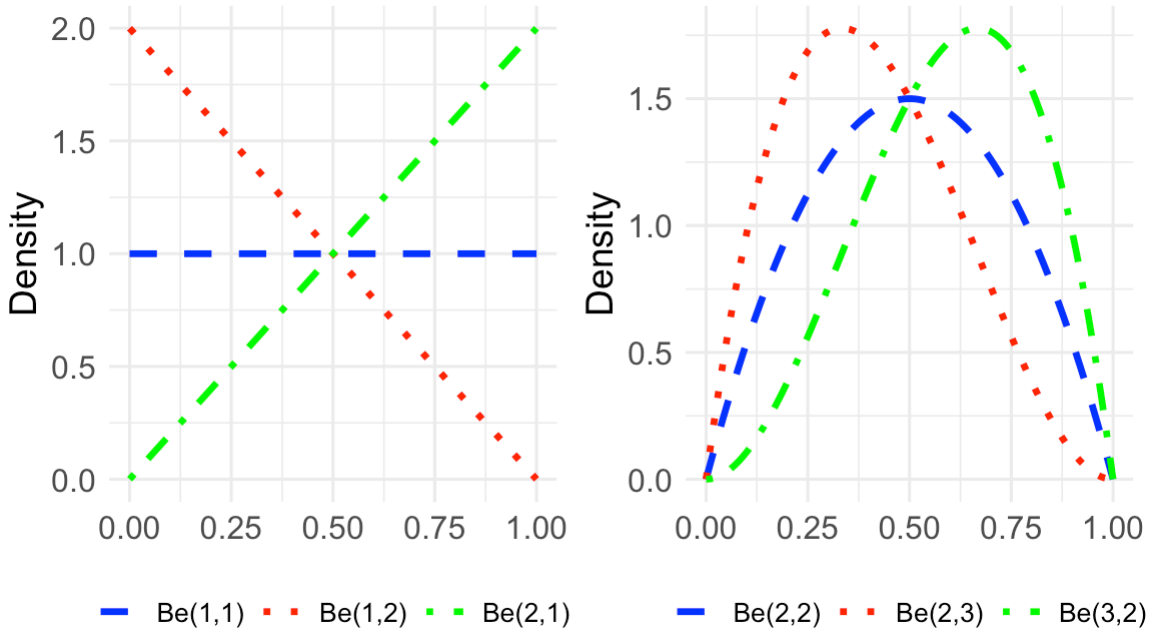
We can map economic beliefs into prior parameters (and vice versa) with the function

```
psi_to_priorSR(R, f, psi0 = NULL, priorSR = NULL, aw = 1, bw = 1)
```

- R and f, denote, respectively the test assets and factors.
- a_w and b_w are the parameters of the Beta distribution, Be(a_w, b_w), controlling the prior probability of factor inclusion. That is, a_w and b_w can be used to encode prior beliefs about the spartity of the SDF.

Setting economic priors cont'd

Be(a_w , b_w) distribution for different parameter values:



E.g., if $a_w = b_w = 1$ (the default) the prior probability of factor inclusion follows a uniform between 0 and 1, and its expected value is a $\frac{a_w}{a_w+b_w} = 50\%$ ex ante chance of selecting a factor.

Recall: if $x \sim Be(a_w , b_w)$, $E[x] = \frac{a_w}{a_w+b_w}$ and $var(x) = \frac{a_w b_w}{(a_w+b_w)^2(a_w+b_w+1)}$

Note: if $b_w >> a_w$ we favour a priori sparse models.

Setting economic priors cont'd

- **priorSR** encodes our prior belief about the Sharpe ratio achievable with the SDF that prices the cross-section of test assets (a natural benchmark is a share of the ex post maximum Sharpe ratio).

For instance, if we have a prior belief of a SDF Sharpe ratio of 0.1, and 50% prior chance of including a factor in the SDF (the default value with $a_w = b_w = 1$) , we can use the function to compute the prior parameter psi_hat:

```
psi_hat <- psi_to_priorSR(R=sim_R, f=cbind(sim_f,uf), priorSR=0.1)
```

To be fed to the estimation function and then perform inference (as before)

```
shrinkage <- continuous_ss_sdf(f=cbind(sim_f,uf), R=sim_R, 5000, psi0=psi_hat, r=0.001, aw=1, bw=1)
```

```
## Null hypothesis: lambda = 0 for each factor
```

```
## Posterior probabilities of rejecting the above null hypotheses are: 0.9468 0.526
```

- if we specify a value for **psi0**, the function gives us instead the corresponding implied prior SR:

```
priorSR<-psi_to_priorSR(sim_R, cbind(sim_f,uf), psi0=psi_hat)
print(priorSR)
```

```
## [1] 0.1
```

That is, the function returns either psi0 or priorSR depending on what we provide as input.

References

Barillas, Francisco, and Jay Shanken. 2016. "Which Alpha?" *The Review of Financial Studies* 30 (4): 1316–38. https://doi.org/10.1093/rfs/hhw101.

Bryzgalova, Svetlana, Jiantao Huang, and Christian Julliard. 2023. "Bayesian Solutions for the Factor Zoo: We Just Ran Two Quadrillion Models." *The Journal of Finance* 78 (1): 487–557. https://doi.org/10.1111/jofi.13197.

———. 2024. "Bayesian Fama-MacBeth Regressions." http://dx.doi.org/10.2139/ssrn.4989615.

Giglio, Stefano, and Dacheng Xiu. 2021. "Asset Pricing with Omitted Factors." *Journal of Political Economy* 129 (7): 1947–90. https://doi.org/10.1086/714090.