Bayesian Hierarchical Models in Finance

Yun Yan (yy1533@nyu.edu)

Contents

The proposal can be also accessed at https://puriney.github.io/gsoc-r-2017-bayes-finance/yy_proposal.html which has same content as this Google doc.

My GitHub repository for this project has already been set at https://github.com/Puriney/gsoc-r-2017-bayes-finance. It contains the Rmarkdown source codes for both proposal and solution, and it is going to host all the source codes for this project, and has the full track of my role in GSoC summer project. Status of the repository will be changed to public after the student application deadline (Apr 3).

1 PROJECT INFO

Project title: Bayesian Hierarchical Models in Finance

Project short title (30 characters): Implement R and RStan solutions to perform Bayesian hierarchical modeling with interest to reproduce the results of three or more financial research papers.

URL of project idea page: https://github.com/rstats-gsoc/gsoc2017/wiki/Bayesian-Hierarchical-Models-in-Finance

2 BIO OF STUDENT

- Yun has valid background in data science and machine learning. He is a 2nd-year graduate student majoring in Computer Science at New York University, where he took related graduate-level courses: foundation of data science, machine learning, algorithm and data structure, computer vision, big data analytic etc.
- Yun has strong interest of pattern discovery for data, in particular in context of academic research. For example, since July 2016 Yun has been involved in researches at Institute for Computational Medicine of NYU. The research is about analyzing the single-cell RNA-seq data-set where rows are cells (samples) and columns are genes (features). Since this project is about research paper, his experience and personal passion for research push him to submit this specific application.
- Another motivation is to have furthermore hands-on experience in Bayesian analysis. This is because recently there emerges a strong trend to apply Bayesian analysis in his research field, e.g. BASiCS(Bayesian Analysis of Single-Cell Sequencing Data). It is appealing to him. He is not satisfied at how to use the established R pipeline for performing Bayesian analysis, but furthermore he has a strong interest of getting hands wet – figure out the actual implementation for solving real-world challenges. Reproducing the results of finance research paper using Bayesian models achieves a good combination of depth into Bayes and feasibility for implementation.
- Yun is an active R user who has been participating in developing R package. For example, ChIPseeker, a Bioconductor package for bioinformatics analysis; honfleuR, an extension supporting object-oriented programming in R S4 methods for an existed R package to analyze single-cell sequencing data.
- Yun has already finished the free-part of on-line course Introduction to Portfolio Analysis in R at DataCamp in order to have basic ideas of portfolio analysis and possible common language about topic contexts with mentors.

Therefore his strong interest for Bayesian analysis in academic research and 5-year experience in R language make him a self-motivated and competent participant for this project.

3 CONTACT INFORMATION

Student name: Yun Yan

Student postal address: 564 First Ave, Apt 11X, New York, NY, 10016

Telephone(s): +1 917-756-3868

Email(s): yy1533@nyu.edu (primary account); youryanyun@gmail.com (acount for sharing draft).

3.1 Student affiliation

Institution: New York University

Program: Master program, Computer Science, Tandon School of Engineering

Stage of completion: 2015.09 - 2017.06. As F-1 student, I have OPT sponsored for participating in GSoC this summer.

Contact to verify: Office of Global Services (OGS), 5 MetroTech Center, Room 259, Brooklyn, NY 11201. Tel: (646) 997-3805

3.2 Schedule Conflicts:

Off-keyboards on Sundays, otherwise there is no time schedule conflicts. I am dedicated to GSoC this summer.

4 MENTORS

4.1 Mentor names

- Brian Peterson (brian@braverock.com)
- Michael Weylandt

4.2 Contact with Mentors

No direct contact. The project idea list and the guide manual for research replication are self-explaining to help me grasp the goal of this project.

5 CODING PLANS AND METHODS

The project is about implementing RStan solutions for reproducing finance research paper. In principle, the replication should follow the format and instruction (see here) outlined by mentor:

- Summarize the essence of the paper by giving the bullets;
- Extract its core hypothesis;
- Fetch, clean, and prepare data-set;
- Code development for repeating the results;
- Extend data analysis to conceive new hypothesis, and find appropriate data to valid new hypothesis.

I have already found two related research papers:

1. Avramov, D. & Zhou, G. Bayesian Portfolio Analysis. Annu. Rev. Financ. Econ. 2, 25–47 (2010).
2. Greyserman, A., Jones, D. H. & Strawderman, W. E. Portfolio selection using hierarchical Bayesian analysis and MCMC methods. J. Bank. Financ. 30, 669–678 (2006)

First one is good review paper for me to know the overall research interest since it is published on Annual Review of Financial Economics.

Second one is good candidate paper that I can give a try to reproduce, because its essence is to highlight the advantage of Bayesian hierarchical models in finance, in particular portfolio selection. I still look forward to discussing with mentor to see whether the paper I chose fits for project.

I need more help from mentors about choosing appropriate paper along these months: two are straightforward for repeating, and last one is worthwhile to go beyond the original hypothesis of the paper and try some novel analysis.

6 TIMELINE

6.1 Pre-coding

- 5/04 ~ 5/09: setup GitHub and communication tools, e.g. Skype, Slack; Also discuss choosing the possible first paper to be reproduced;
- 5/10 ~ 5/17: My final exam season at New York University;
- 5/18 ~ 5/30: Read the review paper (Avramov and Zhou 2010);

6.2 Coding

- 5/30 ~ 6/26: Working on the first paper that I selected (Greyserman, Jones, and Strawderman 2006) or the one after discussion with mentors. It would takes ~1 month since it is a starting period for us.
- 6/26 ~ 6/30: Evaluation and discuss with mentor to choose second paper, and third if possible. I might come up with my own choice after the first month experience.
- 7/01 ~ 7/21: Working on second paper. And consider choosing final paper.
- 7/22 ~ 8/21: Working on third paper which is expected to be most difficult as it involves in extending hypothesis.

6.3 Submission

- 8/21 ~ 8/29: Final submission

7 MANAGEMENT OF CODING PROJECT

- Code: GitHub repo https://github.com/Puriney/gsoc-r-2017-bayes-finance
- Communication: Slack
- Task assignment: Trello (https://trello.com/)

8 SOLUTION

Solution was written in Rmarkdown and the compiled HTML file was deployed on GitHub: https://puriney.github.io/gsoc-r-2017-bayes-finance/index.html. The following results are same as displayed on website thus I would suggest visit the HTML page for better experience.

I am going to apply Bayesian hierarchical models for solving following example problems as solutions to hopefully pass the project tests: multivariate regression.

8.1 Env Setup

Before starts, here are the working environment setup and library dependencies.

```r
library(rstan)
library(ggplot2)
library(dplyr)
library(readr)
library(readxl)
library(scales)
# library(hrbrthemes)
# library(extrafont)
theme_set(theme_minimal())
# hrbrthemes::import_roboto_condensed()
# tmp <- list.dirs(.libPaths(), recursive = F)
# tmp <- file.path(tmp[grepl('hrbrthemes', tmp)][1], 'fonts'); extrafont::font_import(paths=tmp, prompt
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

```r
DIRDATA <- './data'
DIRFIG  <- './fig'
DIRSTAN <- './stan'
DIRRES  <- './res'
if (!dir.exists(DIRFIG)) {dir.create(DIRFIG)}
if (!dir.exists(DIRDATA)) {dir.create(DIRDATA)}
if (!dir.exists(DIRSTAN)) {dir.create(DIRSTAN)}
if (!dir.exists(DIRRES)) {dir.create(DIRRES)}
```

8.2 Perform multivariate regression on Istanbul Stock Exchange Dataset

Import and pre-process the stock data downloaded from UCI-ML database.

```r
## Fetch and reading stock data
file_url <- 'https://archive.ics.uci.edu/ml/machine-learning-databases/00247/data_akbilgic.xlsx'
file_path <- file.path(DIRDATA, 'istanbul.xlsx')
download.file(url = file_url,
              destfile = file_path,
              method = 'wget', quiet = T)
istanbul <- read_excel(file_path,
                       skip = 1)
colnames(istanbul)[2:3] <- c('ISE_TL', 'ISE_USD')
print(head(istanbul))
```

```
## # A tibble: 6 × 10
##         date       ISE_TL      ISE_USD           SP          DAX
##       <dttm>        <dbl>        <dbl>        <dbl>        <dbl>
## 1 2009-01-05  0.035753708  0.038376187 -0.004679315  0.002193419
## 2 2009-01-06  0.025425873  0.031812743  0.007786738  0.008455341
## 3 2009-01-07 -0.028861730 -0.026352966 -0.030469134 -0.017833062
## 4 2009-01-08 -0.062208079 -0.084715902  0.003391364 -0.011726277
## 5 2009-01-09  0.009859905  0.009658112 -0.021533208 -0.019872754
## 6 2009-01-12 -0.029191028 -0.042361155 -0.022822626 -0.013525735
## # ... with 5 more variables: FTSE <dbl>, NIKKEI <dbl>, BOVESPA <dbl>,
## #   EU <dbl>, EM <dbl>
```

```r
## Scaled dataset
istanbul2 <- dplyr::select(istanbul, -date, -ISE_TL) %>%
  scale() %>% as.data.frame()
```

Typically, building linear model is straightforward by `lm` function.

```r
## Build classical linear model
istanbul_lm <- lm(ISE_USD ~ ., data = istanbul2)
summary(istanbul_lm)
```

```
##
## Call:
## lm(formula = ISE_USD ~ ., data = istanbul2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.56399 -0.37727  0.02567  0.37178  2.45577
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.075e-17  2.803e-02   0.000 1.000000
## SP           3.661e-02  4.706e-02   0.778 0.436934
## DAX         -1.466e-01  8.303e-02  -1.765 0.078077 .
## FTSE        -1.234e-01  9.112e-02  -1.354 0.176277
## NIKKEI       2.872e-02  3.593e-02   0.799 0.424439
## BOVESPA     -1.820e-01  4.966e-02  -3.664 0.000273 ***
## EU           6.714e-01  1.317e-01   5.097 4.82e-07 ***
## EM           4.934e-01  5.601e-02   8.808  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.649 on 528 degrees of freedom
## Multiple R-squared:  0.5843, Adjusted R-squared:  0.5788
## F-statistic:   106 on 7 and 528 DF,  p-value: < 2.2e-16
```

Prepare data for running RStan.

```r
## Prepare data to Rstan
set.seed(2017)
is_trained <- sample.int(NROW(istanbul2),
                         size = round(.8 * NROW(istanbul2)),
                         replace = FALSE)
is_trained <- seq_len(NROW(istanbul2)) %in% is_trained
df_train <- istanbul2[is_trained, ]         ## 80% for building model
df_pred  <- istanbul2[!is_trained, ]        ## 20% for evaluting model
X_train <- model.matrix(~., df_train[, -1]) ## intersect item is added
y_train <- df_train$ISE_USD
X_pred  <- model.matrix(~., df_pred[, -1])
y_true  <- df_pred$ISE_USD

n_train <- NROW(X_train)
n_pred <- NROW(X_pred)
n_ft <- NCOL(X_train)
```

Bayesian hierarchical model can be expressed by factor graph:

The same model is defined by Stan language:

```r
stan_path <- file.path(DIRSTAN, 's01_lm.stan')
```

```
data {
  int N;                 // sample size of X
  int M;                 // sample size of the X_pred
  int K;                 // #features
  vector[N] y;           // response
  matrix[N,K] X;         // model matrix for training
  matrix[M,K] X_pred;    // model matrix to be predicted
}
parameters {
  vector[K] beta;        // regression associate
  real sigma;            // random noise
}
transformed parameters {
  vector[N] mu;
  mu = X * beta;
```

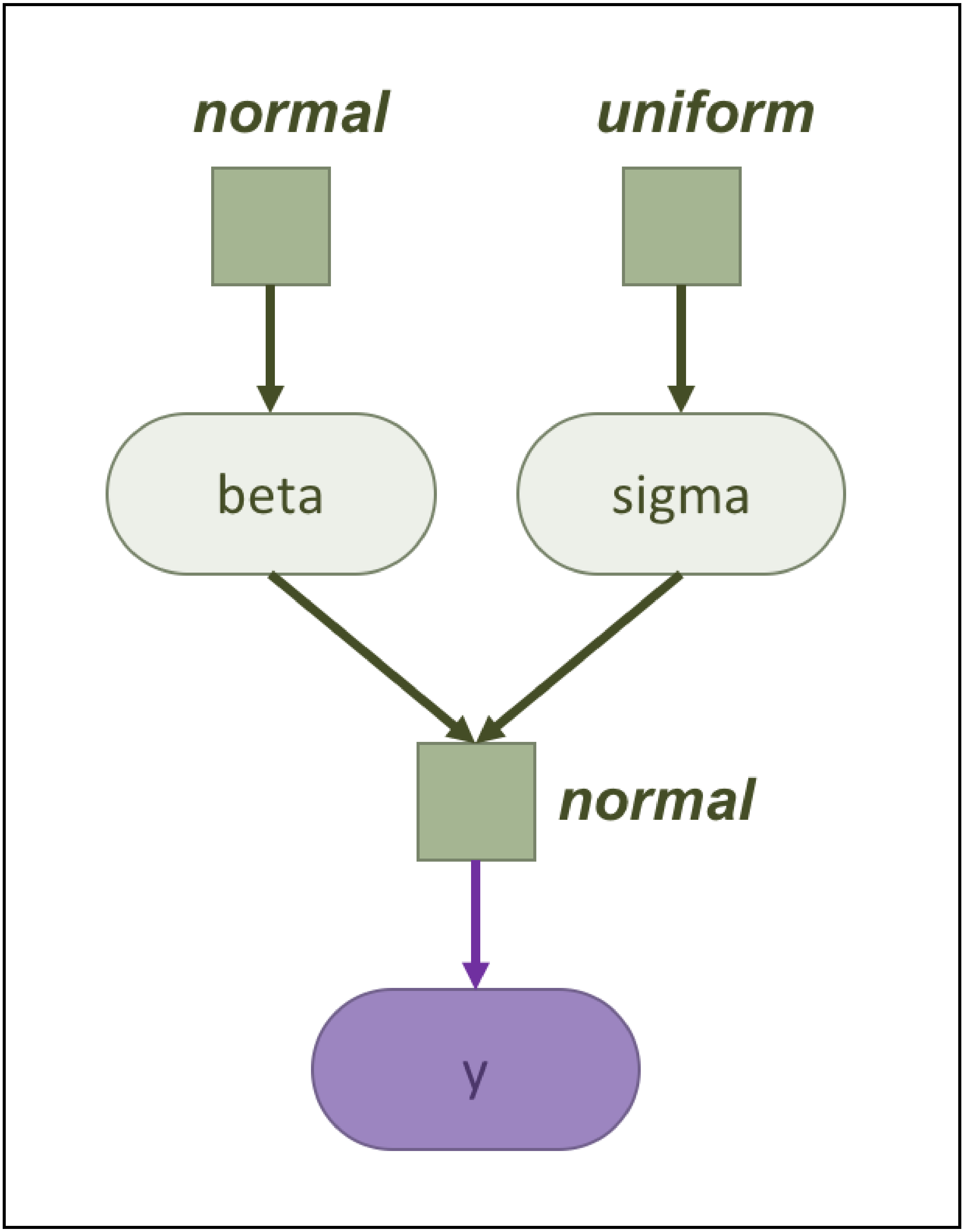


Figure 1: Factor graph

```
}
model {
  // hyperparameters
  sigma ~ uniform(-10, 10);
  beta ~ normal(0, 10);
  // parameter
  y ~ normal(mu, sigma);
}
generated quantities {
  vector[M] y_pred;
  y_pred = X_pred * beta;
}
// Reference: https://datascienceplus.com/bayesian-regression-with-stan-part-1-normal-regression/
```

Run RStan, and the resulted model is exported to RDS file (see `rds_path`).

```
## Run Rstan
rds_path <- file.path(DIRRES, 'istanbul_bayes_lm.rds')
if (file.exists(rds_path)){
  istanbul_model <- read_rds(rds_path)
} else{
  istanbul_model <- stan(file = stan_path,
                         data = list(N=n_train, M=n_pred, K=n_ft,
                                     y=y_train, X=X_train, X_pred=X_pred),
                         pars = c("beta", "sigma", "y_pred"),
                         iter = n_train / 2,
                         algorithm = 'NUTS', seed=2017, verbose = FALSE)
  write_rds(x = istanbul_model, path = rds_path)
}
```

Report the posterior of the beta and sigma.

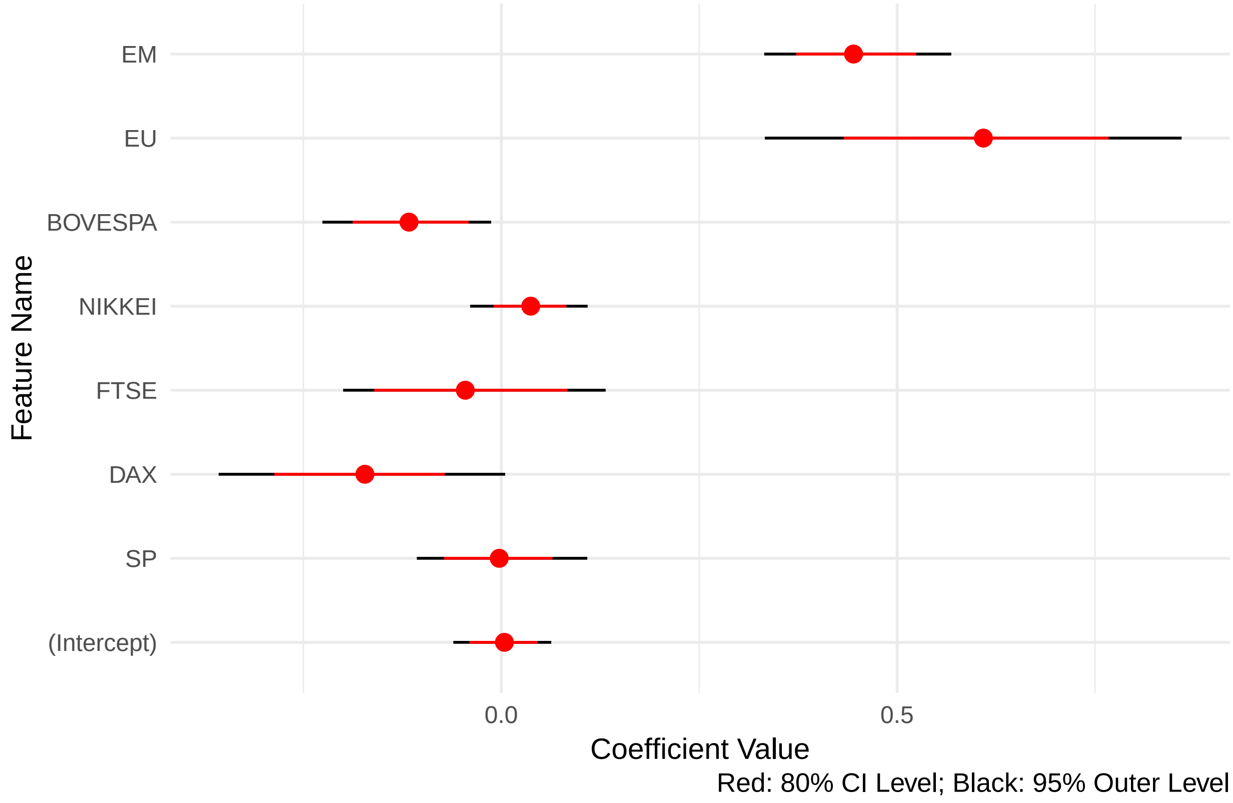
```
## Report the estimated beta and sigma
print(istanbul_model, c('beta', 'sigma'), prob=c(.1, .5, .9))
```

```
## Inference for Stan model: s01_lm.
## 4 chains, each with iter=214.5; warmup=107; thin=1;
## post-warmup draws per chain=107, total post-warmup draws=428.
##
##          mean se_mean   sd   10%   50%   90% n_eff Rhat
## beta[1]  0.00    0.00 0.03 -0.04  0.00  0.05   428 0.99
## beta[2]  0.00    0.00 0.05 -0.07  0.00  0.06   428 0.99
## beta[3] -0.18    0.01 0.09 -0.29 -0.17 -0.07   171 1.01
## beta[4] -0.04    0.01 0.09 -0.16 -0.05  0.08   139 1.04
## beta[5]  0.04    0.00 0.04 -0.01  0.04  0.08   428 1.00
## beta[6] -0.12    0.00 0.06 -0.19 -0.12 -0.04   322 1.00
## beta[7]  0.61    0.01 0.14  0.43  0.61  0.77   109 1.04
## beta[8]  0.45    0.00 0.06  0.37  0.45  0.52   256 1.02
## sigma    0.62    0.00 0.02  0.60  0.62  0.65   428 0.99
##
## Samples were drawn using NUTS(diag_e) at Sun Apr  2 21:41:50 2017.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Better way is to visualize the posteriors of parameters.

```
istanbul_res <- extract(istanbul_model)
```

```
## Visualize the beta
# default: plot(istanbul_model, pars=c('beta'))
beta_mat <- istanbul_res$beta
colnames(beta_mat) <- colnames(X_train)
beta_report <- apply(beta_mat, 2, function(c) quantile(c, prob=c(0.025, 0.1,
                                                                 0.5,
                                                                 0.9, 0.975)))
# [0.1, 0.9]: 80% CI_level; [0.025, 0.975]: 95% outer_level
rownames(beta_report) <- c('ll', 'l', 'm', 'h', 'hh')
figdf <- t(beta_report) %>% as.data.frame()
figdf$VAR <- factor(rownames(figdf), levels = rownames(figdf))
p <- ggplot(figdf, aes(x=VAR, y=m)) +
  geom_linerange(aes(ymin = ll, ymax = hh)) +
  geom_pointrange(aes(ymin=l, ymax=h), colour='red') +
  labs(x="Feature Name", y="Coefficient Value", title = '',
       caption="Red: 80% CI Level; Black: 95% Outer Level") +
  coord_flip()
ggsave(filename = file.path(DIRFIG, 's01_beta.pdf'), p, width=8, height=6)
print(p)
```



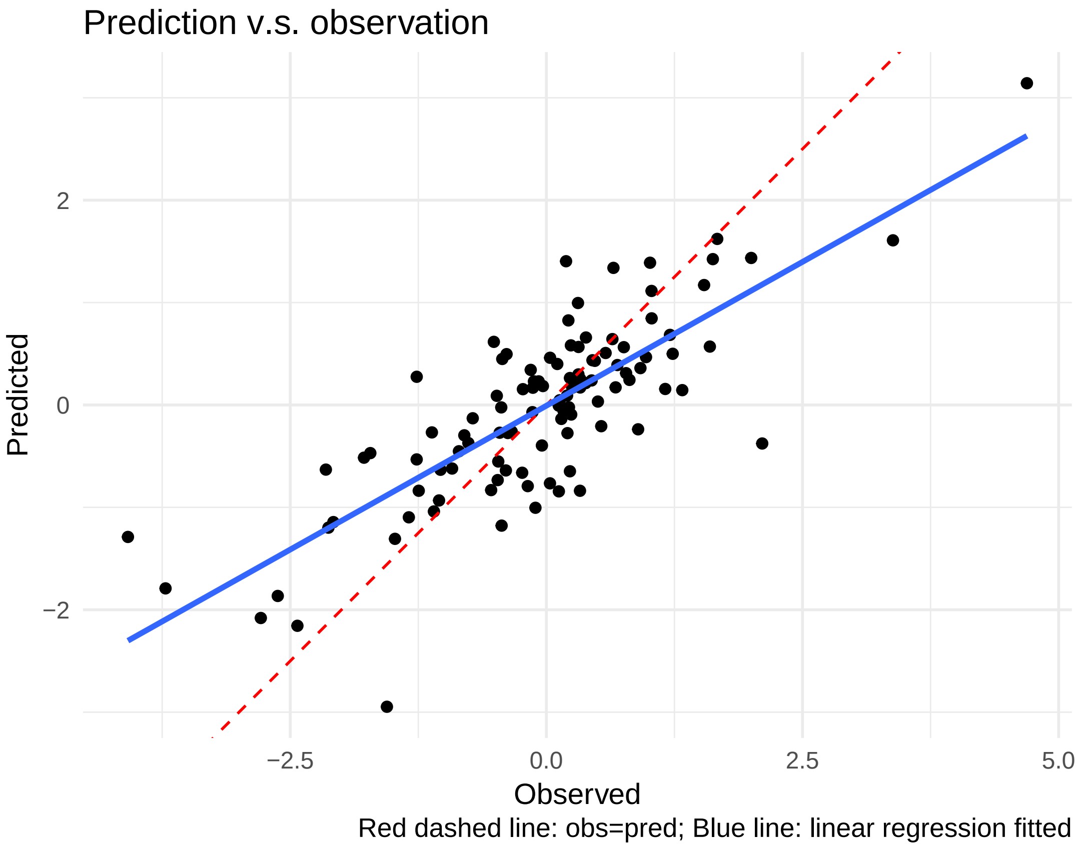
For the 20% data-set that are not used for building model, compare their predicted values with their observation in actual world.

```
## For the unseen dataset, compare the predited v.s. observed values
y_pred_mat <- istanbul_res$y_pred
y_pred_val <- apply(y_pred_mat, 2, median)
```

```r
c <- cor.test(y_true, y_pred_val)
print(c)
```

```
##
##  Pearson's product-moment correlation
##
## data:  y_true and y_pred_val
## t = 13.448, df = 105, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7132731 0.8560035
## sample estimates:
##       cor
## 0.7954156
```

```r
p <- qplot(y_true, y_pred_val, xlab='Observed', ylab='Predicted') +
  coord_fixed() +
  geom_abline(slope = 1, intercept = 0, color = 'red', linetype = 'dashed') +
  geom_smooth(method='lm', se=FALSE) +
  labs(title="Prediction v.s. observation",
       caption="Red dashed line: obs=pred; Blue line: linear regression fitted")
print(p)
```



```r
ggsave(filename = file.path(DIRFIG, 's01_pred_obs.pdf'), p,
       width = 5, height = 5)
```

9 Session Info

```r
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.11.6 (El Capitan)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] scales_0.4.1         readxl_0.1.1         readr_1.0.0
## [4] dplyr_0.5.0          rstan_2.14.2         StanHeaders_2.14.0-1
## [7] ggplot2_2.2.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.9      knitr_1.15.1     magrittr_1.5     munsell_0.4.3
##  [5] colorspace_1.3-2 R6_2.2.0         stringr_1.1.0    plyr_1.8.4
##  [9] tools_3.3.1      parallel_3.3.1   grid_3.3.1       gtable_0.2.0
## [13] DBI_0.5-1        htmltools_0.3.5  yaml_2.1.14      lazyeval_0.2.0
## [17] rprojroot_1.1    digest_0.6.12    assertthat_0.1   tibble_1.2
## [21] gridExtra_2.2.1  inline_0.3.14    evaluate_0.10    rmarkdown_1.3
## [25] labeling_0.3     stringi_1.1.2    backports_1.0.4  stats4_3.3.1
```

Avramov, Doron, and Guofu Zhou. 2010. “Bayesian Portfolio Analysis.” *Annual Review of Financial Economics* 2 (1): 25–47. doi:10.1146/annurev-financial-120209-133947.

Greyserman, Alex, Douglas H. Jones, and William E. Strawderman. 2006. “Portfolio selection using hierarchical Bayesian analysis and MCMC methods.” *Journal of Banking & Finance* 30 (2): 669–78. doi:10.1016/j.jbankfin.2005.04.008.