Review on Nonparametric Estimation of Financial Risk

Nonparametric Estimation of Conditional VaR and Expected Shortfall

Young-geun Kim[2,1,*]

20 Dec, 2019

Abstract

Value at Risk (VaR) is one of many risk measures for financial assets. Expected shortfall is the other one. I investigate nonparametric estimation methods concerning these two. Mainly, I consider conditional information such as exogenous variables or past observed returns. Thus, our problem becomes estimating the conditional VaR (CVaR) and conditional expected shortfall (CES). It is widely known that local linear fitting involves in linear smoother. In this local linear scheme, we replace the smoother with weighted nadaraya watson kernel. This will estimate the conditional probability density. Given this estimator, we call the estimated CVaR and CES by weighted double kernel local linear estimator. It can also be shown that these estimators follow normal distribution asymptotically. *R package for this report is provided in links of this text line.*

[1] Sungkyunkwan University
[2] Department of Statistics

[*] Correspondence: Young-geun Kim <dudrms33@g.skku.edu>

1 Introduction

1.1 Concepts of Financial Risk

Consider any loss distribution. See Figure 1. This is the probability distribution of given time horizon. Value at risk (VaR) is the quantile for the right tail probability.

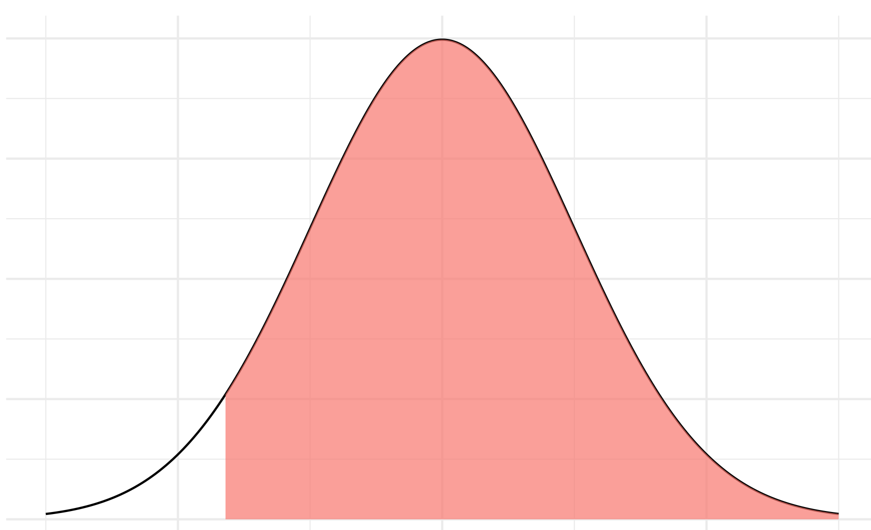

Figure 1: Loss Distribution - Can the financial institution still be in business after a catastrophic event?

Why do economists care about this kind of measure? Both financial institutions and the regulatory committee should analyze the risk of their impressive portfolio. Tsay (2010) interprets this in the two viewpoint. For financial institutions, VaR can be read as a maximal loss of a financial position during a given time horizon for a given probability. It leads to meaning the measure of loss under normal market conditions. For the regulatory committee, on the other hand, it can be read as a minimal loss under extraordinary market circumstances.

As we can see in Figure 1, VaR is defined using the probability distribution of loss. Let p be the right tail probability, the red area in the figure. Let l be the time horizon, let $L(l)$ be the loss function of the asset from t to $t+l$, and let F_l be the CDF of $L(l)$. Then

$$p = P\left[L(l) \geq VaR\right] \tag{1}$$

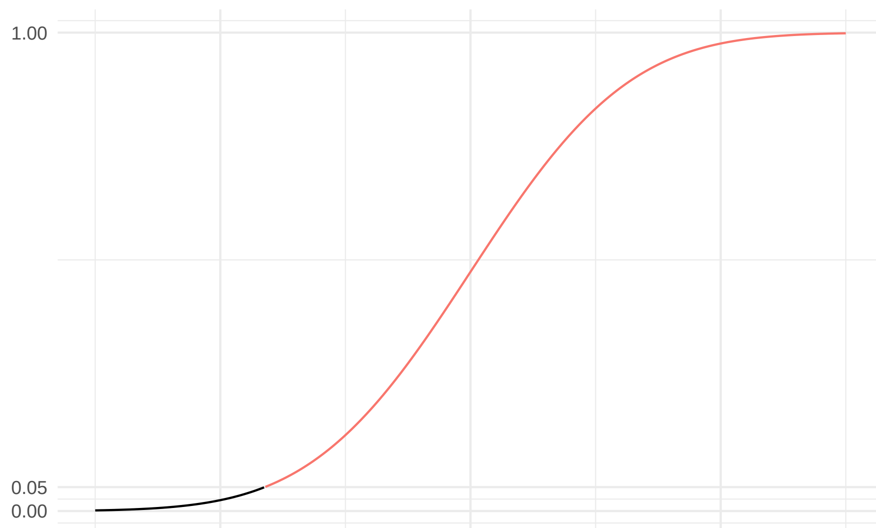

Figure 2: CDF of Loss

See Figure 2. VaR can be computed by finding the p-th quantile.

$$VaR = \inf\left\{x \colon F_l(x) \geq 1-p\right\} \tag{2}$$

1.2 Calculating VaR

There are many ways to get VaR in a parametric way. Equation (2) shows that VaR is the quantile in each time horizon. It is natural to employ quantile regression to get this quantile value.

This report consists of three parts. In Section 2, we covered basic concepts about CVaR and CES. In Section 3, I reviewed the paper (Cai and Wang 2008) briefly. Theoretical parts were mostly skipped. In Section 4, we applied the method in both simulation settings and real data.

2 Background

2.1 Expected Shortfall

Many authors only cover the value-at-risk or expected shortfall in their papers. Value at risk is famous, and its concept is simple. In the market, two portfolios can be merged, sometimes. When this happens, the risk measure should not be higher than the sum of each, which is called the subadditivity. VaR does not satisfy this property, so it can underestimate the actual loss. The expected shortfall as the other risk measure, however, satisfies the condition. The expected shortfall is defined by the expected value of loss function if the VaR is exceeded.

$$ES := E\left[L(l) \mid L(l) \geq VaR\right] \tag{3}$$

In other words, while VaR cares about the maximal loss in the right tail probability part, saying 0.95, ES cares average loss in the remaining part, 0.05.

2.2 Return

We prefer to use log return data. Let $\{P_t\}$ be the price series. Then the log return is defined by $\{Y_t := \ln \frac{P_t}{P_{t-1}}\}$. Loss occurs when the return $\{P_t - P_{t-1}\}$ are negative, so we should use the negative returns or negative log returns.

Consider Taylor expansion for this log function. For any $x_0 > 0$,

$$\ln x \approx \ln x_0 + \frac{1}{x_0}(x - x_0) \tag{4}$$

Write $x = x_2$, $x_0 = x_1$. Then

$$\ln \frac{x_2}{x_1} \approx \frac{x_2}{x_1} - 1 = \frac{x_2 - x_1}{x_1} \tag{5}$$

Observe that the log return approximates to the change rate. Cai and Wang (2008) used the following value in a real example part.

$$-100Y_{t+1} = -100 \ln \frac{P_{t+1}}{P_t}$$

which approximates to percentage loss.

2.3 Conditional Information

In econometrics, conditional information is always the researchers' interests. For example, we are interested in the exogenous variables like economic or market variables. When we look at return data, we can condition past observed returns.

Let $\{Y_t\}$ be stationary log returns and let $\{X_t\}$ be conditional information series.

Definition 2.1 (Conditional Value-at-Risk)**.** Let $F(y \mid x)$ be the conditional cdf of Y_t given $X_t = x$ and let $S(y \mid x) := 1 - F(y \mid x)$. Then Conditional VaR is

$$\nu_p(x) := S^{-1}(p \mid x)$$

When formulating the conditional expected shortfall, we just add the term $X_t = x$ in Equation (3). Let $B \equiv \{\omega : Y_t(\omega) \mid X_t = x \geq \nu_p(x)\} \in \mathcal{B}$. Then

$$
\begin{aligned}
\mu_p(x) &= E\left[Y_t \mid Y_t \geq \nu_p(x), X_t = x\right] \\
&= \frac{1}{P(B)} \int_B Y_t dP \\
&= \frac{1}{P\left(Y_t \geq \nu_p(x) \mid X_t = x\right)} \int_{\nu_p(x)}^{\infty} y f(y \mid x) dy \\
&= \frac{1}{p} \int_{\nu_p(x)}^{\infty} y f(y \mid x) dy
\end{aligned}
\tag{6}
$$

3 Nonparametric Estimation

The workflow of estimating is very simple. It uses the formulation of each CVaR and CES. Just put estimating notation in Equation (6).

$$
\hat{\mu}_p(x) = \frac{1}{p} \int_{\hat{\nu}_p(x)}^{\infty} y \hat{f}(y \mid x) dy \tag{7}
$$

Observe $\hat{\mu}_p(x)$ and $\hat{f}(y \mid x)$. After estimating these two, we will plug the above integration and get the result. THen we can summarize the prcess as follows.

1. Estimate the conditional pdf $\hat{f}(y \mid x)$.
2. Estimate the conditional cdf $\hat{f}(y \mid x)$.
3. Invert the conditional cdf, and get $\hat{\nu}_p(x)$.
4. Plugging, and get $\hat{\mu}_p(x)$.

3.1 Double Kernel Local Linear

Consider any symmetric kernel $K_h(\cdot)$. Then by Taylor expansion,

$$
\begin{aligned}
E[K_{h_0}(y - Y_t) \mid X_t = x] &= K_{h_0} * f_{y|x}(y) \\
&= f(y \mid x) + \frac{h_0^2}{2} \mu_2(K) f^{(2)}(y \mid x) + o(h_0^2)
\end{aligned}
\tag{8}
$$

where $\mu_j(K) = \int_{\mathbb{R}} u^j K(u) du$. Thus, we now have a smoothing problem

$$
f(y \mid x) \approx E\left[K_{h_0}(y - Y_t) \mid X_t = x\right] \tag{9}
$$

Different with the other usual smoothing setting, the response becomes $Y_t^*(y) \equiv K_{h_0}(y - Y_t)$. Given this, implement local linear least squares.

$$
\hat{f}(y \mid x) = \underset{\alpha(x), \beta(x)}{\operatorname{argmin}} \sum_{t=1}^{n} W_h(x - X_t) \left[Y_t^*(y) - \alpha(x) - \beta(x)(X_t - x)\right]^2 \tag{10}
$$

This is called double local linear in that the problem involves in the two kernel, K_{h_0} and W_h. Recall that the local linear estimate is equivalent to the weighted least squares. Let $\mathbf{Y}_y^* = (Y_1(y), \ldots, Y_n(y))^T \in \mathbb{R}^n$, let $\mathbf{b}_x(x_t) := (1, x_t - x)^T \in \mathbb{R}^2$, let $\mathbf{b}_x(x) = \mathbf{e}_1 := (1, 0)^T$, let $X_x := \left(\mathbf{b}_x(x_i)^T\right) \in \mathbb{R}^{n \times 2}$, and let $W_x := diag(W_h(x - X_j)) \in \mathbb{R}^{n \times n}$. Then the local linear solution is given by $\hat{f}_{ll} = \hat{\alpha}$,

$$
\begin{aligned}
\hat{f}_{ll}(y \mid x) &= \mathbf{e}_1^T (X_x^T W_x X_x)^{-1} X_x^T W_x \mathbf{Y}_y^* \\
&= \mathbf{l}(x)^T \mathbf{Y}_y^* \\
&\equiv \sum_{t=1}^{n} l_t(x) Y_t^*(y)
\end{aligned}
\tag{11}
$$

Cai and Wang (2008) provides the exact form of each element by matrix calculation in the paper. This linear form of pdf easily gives its CDF. This LL estimator has some good properties, such as differentiability. The conditional CDF can be calculated by

$$
\begin{aligned}
\hat{F}_{ll}(y \mid x) &= \int_{\infty}^{y} \hat{f}_{ll}(y \mid x) dy \\
&= \sum_{t=1}^{n} l_t(x) G_{h_0}(y - Y_t)
\end{aligned}
\tag{12}
$$

where $G(\cdot)$ is the CDF of $K(\cdot)$. Since it is CDF, so it must be $\hat{F}_{ll} \in [0, 1]$ and monotone increasing. Double local linear, however, does not guarantee these properties.

3.2 Weighted Nadaraya Watson

Since the above estimator cannot give a desirable cdf to us, we consider another method, weighted nadaraya watson suggested by Cai (2001). The form is similar to the nadaraya watson estimator.

$$
\hat{F}_{NW}(y \mid x) = \sum_{t=1}^{n} W_{c,t}(x, \lambda) I(Y_t \leq y) \tag{13}
$$

where the WNW weights given by

$$
W_{c,t}(x, h) = \frac{p_t(x) W_h(x - X_t)}{\sum\limits_{i=1}^{n} p_i(x) W_h(x - X_i)} \tag{14}
$$

The term “weighted” is due to the weight for each kernel. $\{p_t(x)\}$ is chosen by

$$
p_t(x) = \frac{1}{n\left[1 + \lambda(X_t - x) W_h(x - X_i)\right]} \geq 0 \tag{15}
$$

We should find λ maximizing the empirical log likelihood function

$$
L_n(\lambda) = -\sum_{t=1}^{n} \ln\left[1 + \lambda(X_t - x) W_h(x - X_i)\right] \tag{16}
$$

Cai (2001) recommended employing Newton-Raphson to find it.

3.3 Weighted Double Kerenl Local Linear

Weighted Nadaraya Watson estimator for conditional cdf (Equation (13)) can fill the defect of local linear (Equation (12)). In Equation (11), Cai and Wang (2008) can replace the linear smoother with WNW weights. This is called the weighted double kernel local linear estimator.

$$\hat{f}_{cai}(y \mid x) = \sum_{t=1}^{n} W_{c,t}(x, h) Y_t^*(y) \tag{17}$$

It proceeds in a same way that the conditional cdf can be estimated by

$$\hat{F}_{cai}(y \mid x) = \sum_{t=1}^{n} W_{c,t}(x, h) G_{h_0}(y - Y_t) \tag{18}$$

The next step is estimating $\hat{\nu}_p(x)$ and $\hat{\mu}_p(x)$ one by one. Let $\hat{S}_{cai}(y \mid x) = 1 - \hat{F}_{cai}(y \mid x)$. By inverting this function, we can get the CVaR function.

$$\hat{\nu}_p^{(cai)}(x) = \hat{S}_{cai}^{-1}(p \mid x) \tag{19}$$

When inverting it, I used Equation (2). Finally, we plug $\hat{\nu}_p(x)$ and $\hat{f}_{cai}(y \mid x)$ into Equation (7). Then

$$\hat{\mu}_p(x) = \frac{1}{p} \sum_{t=1}^{n} W_{c,t}(x, h) \left[Y_t \bar{G}_{h_0}(\hat{\nu}_p(x) - Y_t) + h G_{1,h_0}(\hat{\nu}_p(x) - Y_t) \right] \tag{20}$$

where $\bar{G}(u) = 1 - G(u)$ and $G_1(u) = \int_u^\infty v K(v) dv$.

3.4 Asymptotic Normality

Cai and Wang (2008) has shown that each estimator follows Normal distribution at both interior and boundary point. Since the formulation is quite complicated, I provide the Monte Carlo simulation setting here. Consider AR(1)-GARCH(0,1) model as in the paper.[1] This setting continues to the next simulation section.

```r
ar0 <- .01
ar1 <- .62
omega <- .15
alp <- 0
bet <- .65
```

Example 3.1 (AR(1)-GARCH(0,1) Model)**.** With $X_t = Y_{t-1}$,

$$\begin{cases} Y_t = 0.01 + 0.62 X_t + \sigma_t \epsilon_t \\ \sigma_t^2 = 0.15 + 0.65 \sigma_{t-1}^2 \\ \epsilon_t \sim N(0, 1) \end{cases}$$

```r
garch_sim <- function(n, cond, ar_mu = .01, ar = .62) {
  garch_spec <-
    garchSpec(
      cond.dist = "norm",
      model = list(
        omega = .15, alpha = 0, beta = .65
```

[1]Authors wrote that the model was ARCH(1), but many textbooks including Tsay (2010) specify that model as GARCH(0,1).

```r
      )
    )
  tibble(garch = garchSim(garch_spec, n = n) %>% as.numeric()) %>%
    mutate(
      x = cond,
      y = ar_mu + ar * x + garch
    ) %>%
    select(y) %>% # to use youngtool (experimental stage)
    pull()
}
#-------------------------
cond_sim <- function(n, m, xcond) {
  mc_data(garch_sim, N = n, M = m, cond = xcond) %>%
    .[,
      xcond := xcond] %>%
    .[]
}
#--------------------------
x <- runif(1, -1, 0)
mc <- cond_sim(200, 500, x)
```

Monte Carlo Samples were generate by the following procedure.

1. For fixed x,
2. Generate GARCH(0, 1): (σ_t, ϵ_t)
3. Generate Y_t using AR(1) for each $X_t = Y_{t-1}$
4. AR(1): $Y_t = 0.01 + 0.62 Y_{t-1} + \sigma_t \epsilon_t$

3.4.1 Interior

```r
CES <-
  mc[,
     .(mu =
         wdkll_ces(x ~ xcond, .SD) %>%
         predict(newx = unique(xcond))),
     by = mc]
```

```r
CES %>%
  ggplot(aes(x = mu)) +
  geom_histogram(alpha = .7, binwidth = .01) +
  xlab("Conditional Expected Shortfall")
```

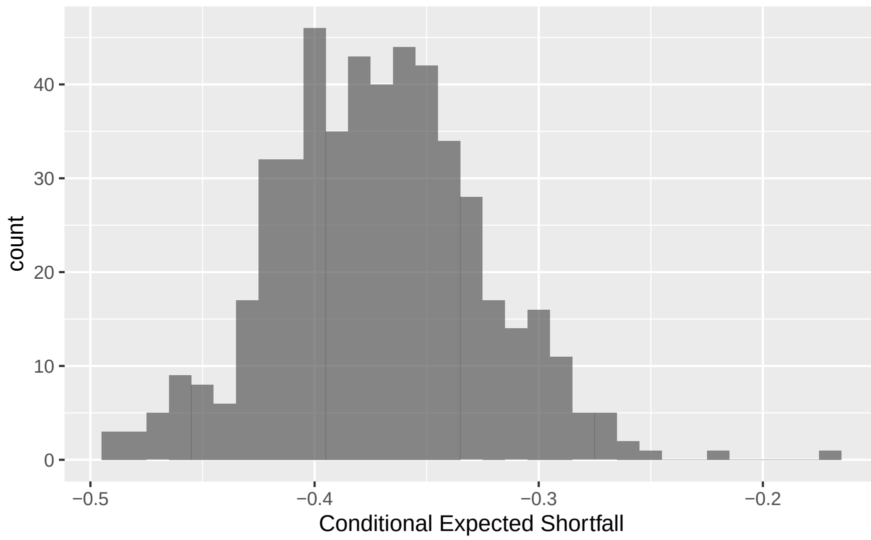

Figure 3: Asymptotic normality for $\hat{\mu}_p(x)$ in interior point

In Figure 3, we can see that $\hat{\mu}_p(x)$ follows the normal distribution approximately at $x = -0.734$.

3.4.2 Boundary

Cai and Wang (2008) proved at the left boundary point $x = ch$, $0 < c < 1$.

```
bound_c <- runif(1) * 200^(-4/5)
mc <- cond_sim(200, 500, bound_c)
```

```
CES2 <-
  mc[,
     .(mu =
         wdkll_ces(x ~ xcond, .SD) %>%
         predict(newx = unique(xcond))),
     by = mc]
```

```
CES2 %>%
  ggplot(aes(x = mu)) +
  geom_histogram(alpha = .7, binwidth = .01) +
  xlab("Conditional Expected Shortfall")
```

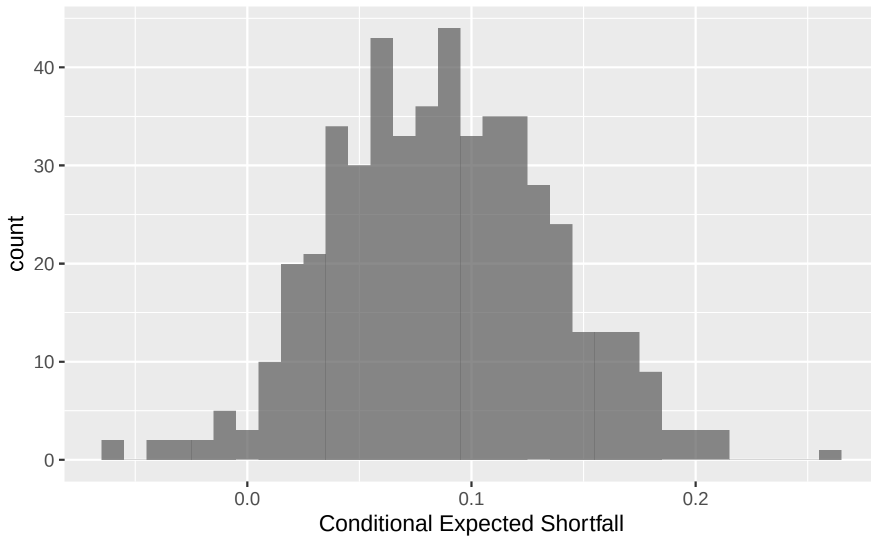

Figure 4: Asymptotic normality for $\hat{\mu}_p(x)$ at left boundary

Similarly, Figure 4 presents asymptotic normality of WDKLL estimator of CES at the left boundary point.

4 Experiments

4.1 Simulation

Now we try another simulation under Example 3.1. This model implies the true CVaR and CES. Since $\epsilon_t \sim N(0, 1)$,

$$\sigma_t \epsilon_t \sim N(0, \sigma_t^2) \tag{21}$$

By construction,

$$Y_t \mid X_t \sim N\left(0.01 + 0.62X_t, \sigma_t^2\right) \tag{22}$$

Given this conditional pdf, we can compute cdf using statistical packages such as `R`. The procedure after this is the same as before. In this Monte Carlo simulation setting, we want to simulate the expected prediction error between the true $\mu_p(x)$ and $\hat{\mu}_p(x)$. Absolute loss was used following Cai and Wang (2008). Figure 5 shows the structure of MC simulation. After training each model using each MC sample, we test it in the independent test set. Individual samples and a test set are generated as follows.

1. For fixed x_t (pre-determined grid points)
2. Generate GARCH(0, 1): (σ_t, ϵ_t)
3. Generate Y_t using AR(1) for each $X_t = Y_{t-1}$
4. AR(1): $Y_t = 0.01 + 0.62Y_{t-1} + \sigma_t \epsilon_t$

```r
arch <-
  ugarchspec(
    fixed.pars = c("omega" = omega, "alpha1" = alp, "beta1" = bet),
    mean.model = list(armaOrder = c(0, 0), include.mean = FALSE)
  )
```

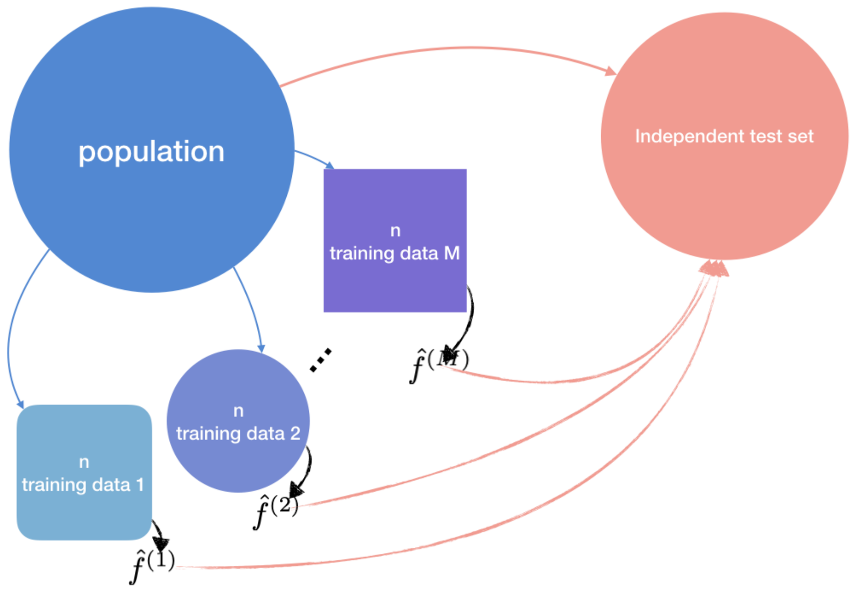

Figure 5: Simulating Expected prediction error

```
N <- 100
M <- 20
xcond <- seq_len(N) / (2 * N)
mc <-
  lapply(
    1:M,
    function(i) {
      X <- ugarchpath(arch, n.sim = N)@path
      X <- do.call("cbind", X)[,-3]
      colnames(X) <- c("sigma", "garch")
      X %>%
        data.table() %>%
        .[,
          mc := paste0("s", i)]
    }
  ) %>%
  rbindlist() %>%
  .[,
    xt := xcond,
    by = mc] %>%
  .[,
    yt := ar0 + ar1 * xt + garch]
# true ces-------------------------------------------
sig <- unique(mc$sigma)
mc %>%
  .[,
    true_ces := plugin_ces(
      pdf = function(y, x) dnorm(y, mean = ar0 + ar1 * x, sd = sig),
      cdf = function(y, x) pnorm(y, mean = ar0 + ar1 * x, sd = sig),
      x = xt,
```

```
      lower = -1,
      upper = 1
    )] %>%
  .[,
    true_cvar := invert_cvar(
      cdf = function(y, x) pnorm(y, mean = ar0 + ar1 * x, sd = sig),
      x = xt
    )]
# test set-----------------------------------------
N0 <- 150
x <- ugarchpath(arch, n.sim = N0)@path
x <- do.call("cbind", x)[, -3]
colnames(x) <- c("sigma", "garch")
mc_test <-
  x %>%
  data.table() %>%
  .[,
    xt := seq_len(N0) / (2 * N0) + .5] %>%
  .[,
    yt := ar0 + ar1 * xt + garch] %>%
  .[,
    true_ces := plugin_ces(
      pdf = function(y, x) dnorm(y, mean = ar0 + ar1 * x, sd = sig),
      cdf = function(y, x) pnorm(y, mean = ar0 + ar1 * x, sd = sig),
      x = xt,
      lower = -1,
      upper = 1
    )] %>%
  .[,
    true_cvar := invert_cvar(
      cdf = function(y, x) pnorm(y, mean = ar0 + ar1 * x, sd = sig),
      x = xt
    )]
```

```
test_err <- function(band) {
  mc %>%
    .[,
      .(pred_ces = wdkll_ces(yt ~ xt, data = .SD, nw_h = band, h0 = band / 10) %>%
          predict(mc_test$xt),
        pred_wnw = wnw_ces(yt ~ xt, data = .SD, nw_h = band, h0 = band / 10) %>%
          predict(mc_test$xt, nw = TRUE),
        ces = mc_test$true_ces),
      by = mc] %>%
    .[,
      `:=`(
        abs_err_ces = abs(ces - pred_ces),
        abs_err_wnw = abs(ces - pred_wnw)
      )] %>%
    .[,
      .(
        abs_err_ces = mean(abs_err_ces),
        abs_err_wnw = mean(abs_err_wnw)
      ),
      by = mc]
```

```
}
```

```
err_tab <-
  parallel::mclapply(
    c(1, 3, 10),
    function(h) {
      test_err(h)[,
                  h := h]
    },
    mc.cores = 3
  ) %>%
  rbindlist()
```

See Figure 6. WDKLL estimator produces much less expected prediction error in each bandwidth than usual Nadaraya Watson estimator.

```
err_tab %>%
  melt(id.vars = c("mc", "h")) %>%
  ggplot() +
  geom_boxplot(aes(x = factor(h), y = value, fill = variable)) +
  labs(
    x = "Bandwidth",
    y = "MADE"
  ) +
  scale_fill_discrete(
    name = "Methods",
    label = c("WDKLL", "NW")
  ) +
  theme(
    legend.position = "bottom"
  )
```

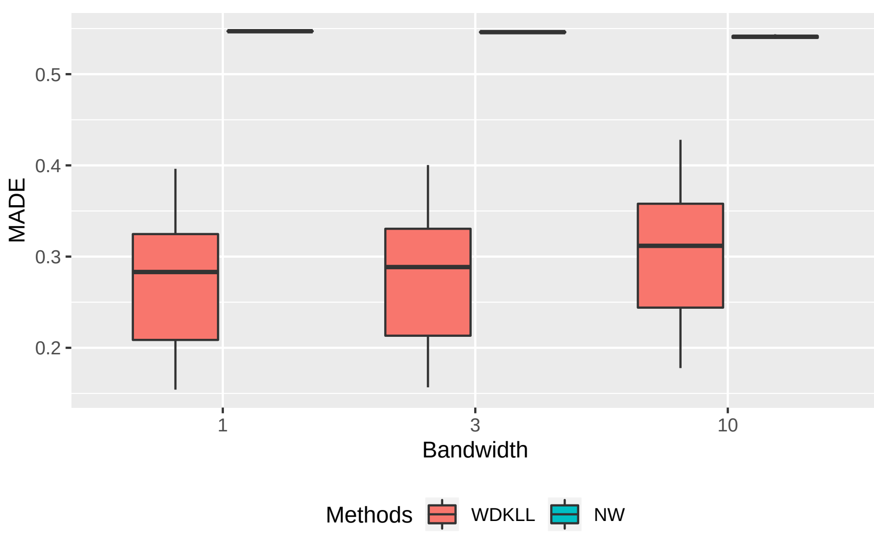

Figure 6: WDKLL versus Nadaraya-Watson

4.2 Data Analysis

4.2.1 Data Description

```r
coin <-
  read_csv("../data/btc.csv") %>%
  select(Date, price = `24h Open (USD)`)
```

We can implement CVaR and CES to measure risk of Bitcoin. We analyze 607 daily prices of bitcoin from $2018-04-05$ to $2019-12-12$.

Figure 7: Bitcoin price in USD

We use $y_t := -100 \ln \frac{P_t}{P_{t-1}}$ as daily return to approximate percentage loss.

```r
coin_return <-
  coin %>%
  mutate(
    yt = -log(price / dplyr::lag(price)) * 100,
    xt = dplyr::lag(yt)
  ) %>%
  dplyr::filter(
    Date >= "2018-06-01",
    Date < "2019-12-01"
  )
```

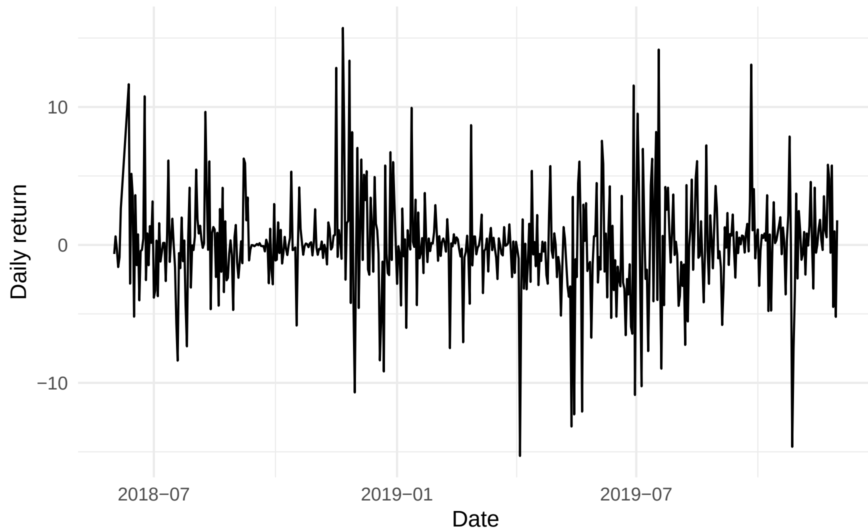

Figure 8: Daily return of bitcoin price

Figure 8 is the daily return $\{y_t\}$. We now try to estimate using this y_t given the first lagged variable $x_t = y_{t-1}$.

4.2.2 Conditional Value-at-Risk

```
coin_cvar <- wdkll_cvar(yt ~ xt, data = coin_return, nw_h = 1, h0 = .1, lower_invert = -5, upper_invert
```

```
cvar_pred <-
  tibble(x = seq(-1, 1, by = .1)) %>%
  mutate(CVaR = predict(coin_cvar, x))
```

Look at Figure 9. It is the plot of CVaR for each conditional information $x_t = y_{t-1}$. CVaR which is the maximal loss under 0.95 probability is increasing given the last day return. How about the other 0.05 case?

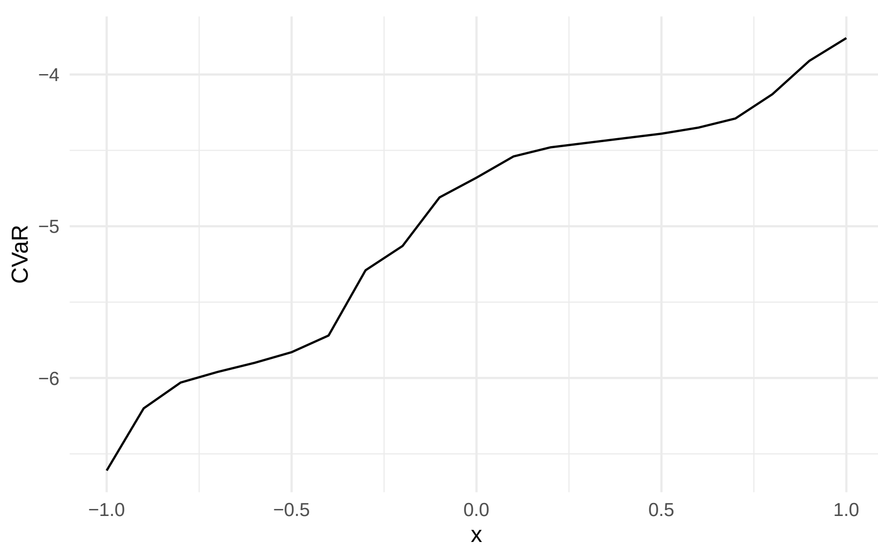

Figure 9: Conditional Value-at-Risk given each lagged variable value

As in Cai and Wang (2008), we can state in the viewpoint of financial institution or the regulatory committee.

Figure 9 presents that a maximal loss rate in usual case is increasing as loss of the last day increases. Investors can think this asset as very risky. See Figure 10. It is CVaR of Dow Jones Industrials (DJI) index in the same period. One of stock is not always increasing.

```
dji <-
  read_csv("../data/dji.csv") %>%
  select(Date, price = Open)
#---------------------------------------
dji_return <-
  dji %>%
  mutate(
    yt = -log(price / dplyr::lag(price)) * 100,
    xt = dplyr::lag(yt)
  ) %>%
  dplyr::filter(
    Date >= "2018-06-01",
    Date < "2019-12-01"
  )
```

```
dji_fit <- wdkll_cvar(yt ~ xt, data = dji_return, nw_h = 1, h0 = .1, lower_invert = -5, upper_invert = 5
```

```
dji_cvar <-
  tibble(x = seq(-1.5, 1.5, by = .1)) %>%
  mutate(CVaR = predict(dji_fit, x))
```

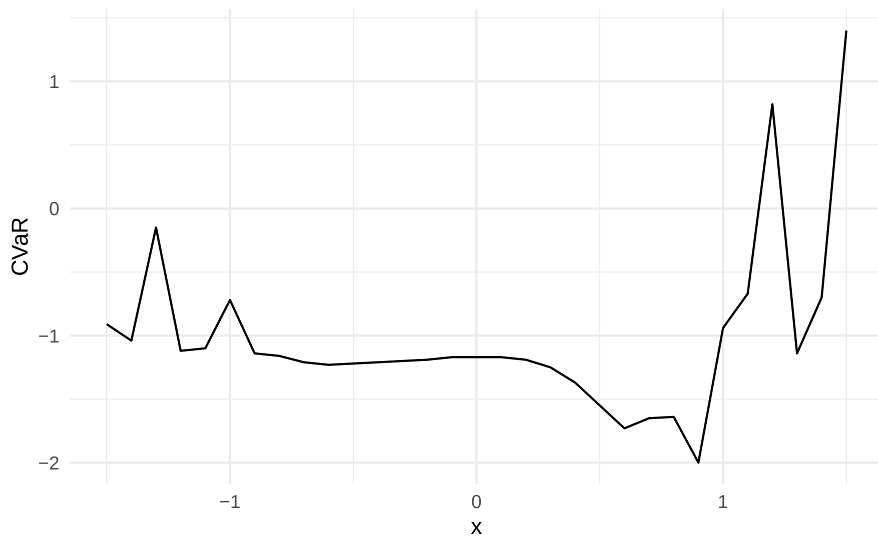

Figure 10: Conditional Value-at-Risk of Dow Jones Industrial Index

4.2.3 Conditional Expected Shortfall

How about CES? See Figure 11. This is the same plot for CES. The pattern is quite different with before. It shows a U-shape. Recall that we are dealing with negative returns. Thus, positive x_t means the loss. We can expect that its price is more likely to fall if there were a loss within the last day than if there was the same amount of positive return.[2]

As mentioned in Section 2, the expected shortfall deals with the extreme part that VaR does not - when VaR is exceeded. As the average loss when VaR is surpassed given conditional information, CES can show the difference of actual loss more precisely.

[2]This interpretation have followed the one by Cai and Wang (2008)

```
coin_ces <- wdkll_ces(yt ~ xt, data = coin_return, nw_h = 1, h0 = .1, lower_invert = -5, upper_invert =
```

```
dji_ces <- wdkll_ces(yt ~ xt, data = dji_return, nw_h = 1, h0 = .1, lower_invert = -5, upper_invert = 5)
```

```
ces_pred <-
  tibble(x = seq(-1.5, 1.5, by = .1)) %>%
  mutate(
    coin_pred = predict(coin_ces, x),
    dji_pred = predict(dji_ces, x)
  )
```

```
ces_pred %>%
  pivot_longer(ends_with("pred"), names_to = "pred", values_to = "CES") %>%
  ggplot(aes(x, CES, colour = pred)) +
  geom_path() +
  theme_minimal() +
  scale_colour_discrete(
    name = "Data",
    labels = c("Bitcoin", "DJI")
  ) +
  theme(legend.position = "bottom")
```

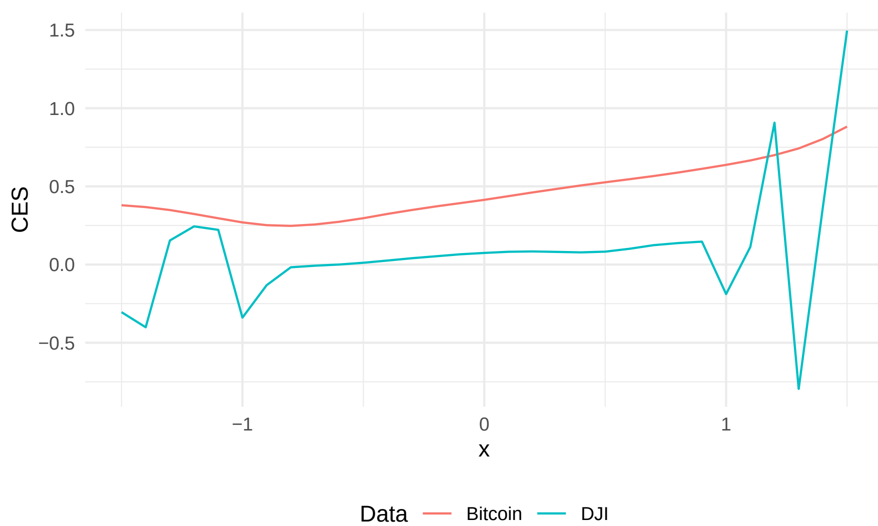

Figure 11: Conditional Expected Shortfall of Bitcoin and DJI

As we can see in Figure 11, the risk of Bitcoin is larger than of DJI overall. This suggests the similar results to of CVaR that Bitcoin is danger asset to invest, and should be regulated.

5 Conclusion

This report have reviewd WDKLL estimator designed by Cai and Wang (2008). Its final goal is conditional expected shortfall. To estimate it, we have to estimate conditional PDF, conditional CDF, and conditional value-at-risk. To make the estimator reasonable - for instance, PDF is differentiable, CDF is in its range and monotone increasing - authors combine double kernel local linear and weighted nadaraya watson estimate methods.

Nonparametric method has its own advantages like low bias. Moreover, the estimator seems to improve boundary problem in nonparametric literature using asymptotic normality. It gives superiority in test error to the other estimators such as nadaraya watson.

5.1 Discussion

Bandwidth selection in this kind of kernel smoothing is important factor. Cai and Wang (2008), however, only provided an ad-hoc process using nonparametric AIC (Cai and Tiwari 2000), defined by

$$AIC_C(h,p) := \ln(RSS) + \frac{n + tr(S)}{n - [tr(S) + 2]} \tag{23}$$

where $RSS = \mathbf{Y}^T(I-S)^T(I-S)\mathbf{Y}$ and S is the linear smoother.

Since this is linear smoother, it might be possible to think about various criteria for selection.

5.2 Supplement

For this report, I have built an `R` package due to the lack of the original code. It is in github.com/ygeunkim/ceshat. This report also has its own code, and it can be shown in the Github repository: github.com/ygeunkim/nonparam-cvar.

References

Cai, Zongwu. 2001. "Weighted Nadaraya–Watson Regression Estimation." *Statistics & Probability Letters* 51 (3): 307–18.

Cai, Zongwu, and Ram C Tiwari. 2000. "Application of a Local Linear Autoregressive Model to Bod Time Series." *Environmetrics: The Official Journal of the International Environmetrics Society* 11 (3): 341–50.

Cai, Zongwu, and Xian Wang. 2008. "Nonparametric estimation of conditional VaR and expected shortfall." *Journal of Econometrics* 147 (1): 120–30. https://doi.org/10.1016/j.jeconom.2008.09.005.

Tsay, Ruey S. 2010. *Analysis of Financial Time Series*. John Wiley & Sons.