

CvM Calculation for COVID-19 Forecasting Models - Demo

Estee Cramer, Nick Reich, Nutchcha Wattanachit

02/16/2021

Cramer von-Mises Criterion

Definition

The Cramer von-Mises criterion is defined as

$$\omega^2 = \int_{-\infty}^{\infty} [F_n(x) - F^*(x)]^2 dF^*(x),$$

where $F_n(x)$ and $F^*(x)$ are empirical cumulative distribution functions. The CvM is symmetric.

CvM calculation

The two-sample formulation of CvM criterion can be written in many forms. The equation below is currently used in the first part of this demo.

$$T = \frac{mn}{(m+n)} \left(\frac{2}{mn} \sum_{i=1..m, j=1..n}^{m,n} \phi(\|X_i - Y_j\|^2) - \frac{1}{m^2} \sum_{i=1..m, j=1..m}^{m,n} \phi(\|X_i - X_j\|^2) - \frac{1}{n^2} \sum_{i=1..n, j=1..n}^{m,n} \phi(\|Y_i - Y_j\|^2) \right) \phi_{\text{Cramer}}(z) = \sqrt{z}/2$$

Since we have sample quantiles from COVID-19 forecasting models, we do not have the whole empirical distributions for the CvM calculation. In the current implementation, we do the following:

- The monotonic spline function to interpolate is used to interpolate points between available sample quantiles from the forecasting models. Now we have samples (with points interpolated between sample quantiles and extrapolated at the tails).
- We can apply the `ecdf()` function to create and plot the ecdfs from these samples. For the calculation of CvM, samples created in the first step are used.

In this demo, the models are from the week of 02/08/2021.

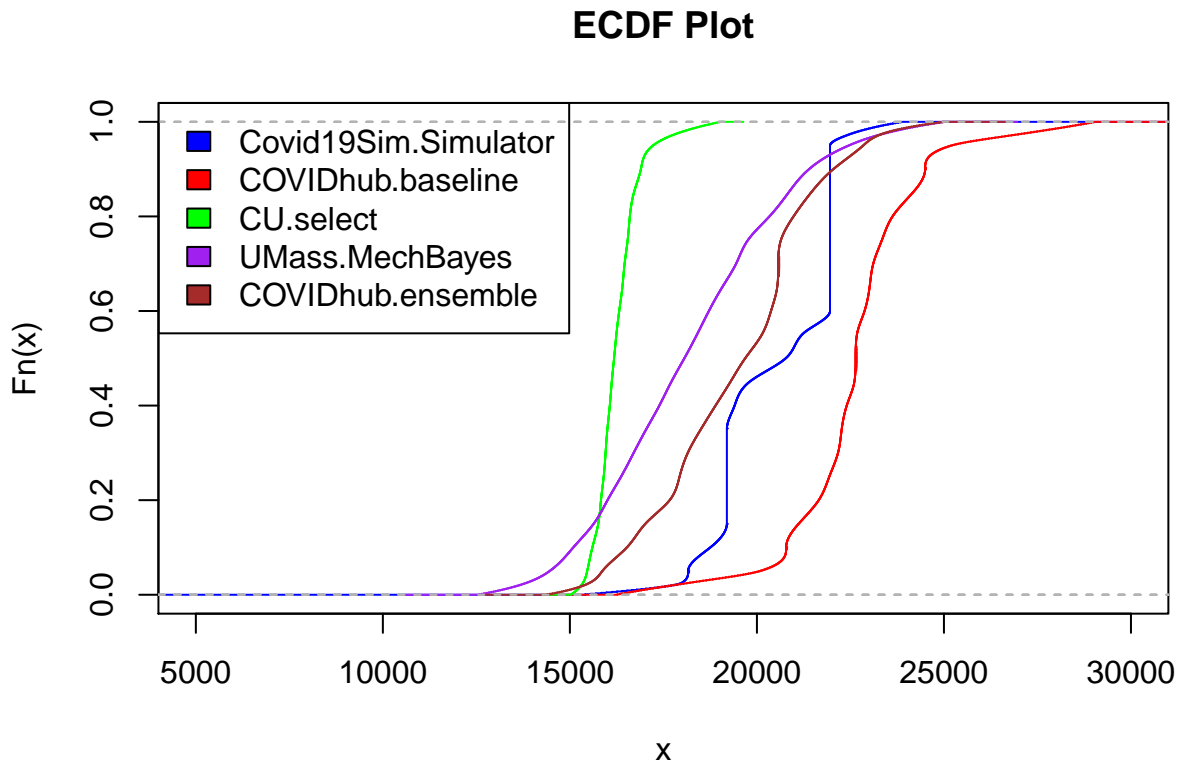
```
source("./cvm_script.R")
# create sample data for this demo
sample_frame <- load_latest_forecasts(models = c("CU-select", "UMass-MechBayes", "Covid19Sim-Simulator",
                                                "COVIDhub-ensemble", "COVIDhub-baseline"),
                                     last_forecast_date = "2021-02-08",
                                     forecast_date_window_size = 6,
                                     locations = "US",
                                     types = "quantile",
                                     targets = "1 wk ahead inc death",
                                     source = "zoltar")
```

```
FALSE polling for status change. job_url=https://zoltardata.com/api/job/43314/
FALSE QUEUED
FALSE QUEUED
FALSE SUCCESS
```

```
## make a single target data for demo run
small <- frame_format(sample_frame) %>%
  dplyr::filter(type=="quantile") %>%
  data.frame(.)
names <- colnames(small)[6:ncol(small)]
# interpolate points for these quantiles
point_to_interpolate <- seq(0, 1, by=0.001)
for(i in 1:5){
  assign(paste0("emp",i),
    ecdf(spline(x=small$quantile,
                y=small[,names[i]],
                method = "hyman",
                xout=point_to_interpolate)$y))
}
```

Plot from the ecdf created from the samples:

```
plot(emp1,verticals=TRUE, do.points=FALSE, col='blue',xlim=c(5000,30000), main="ECDF Plot")
plot(emp2,verticals=TRUE, do.points=FALSE, add=TRUE, col='red')
plot(emp3,verticals=TRUE, do.points=FALSE, add=TRUE, col='green')
plot(emp4,verticals=TRUE, do.points=FALSE, add=TRUE, col='purple')
plot(emp5,verticals=TRUE, do.points=FALSE, add=TRUE, col='brown')
legend("topleft",names,fill=c("blue","red","green","purple","brown"))
```

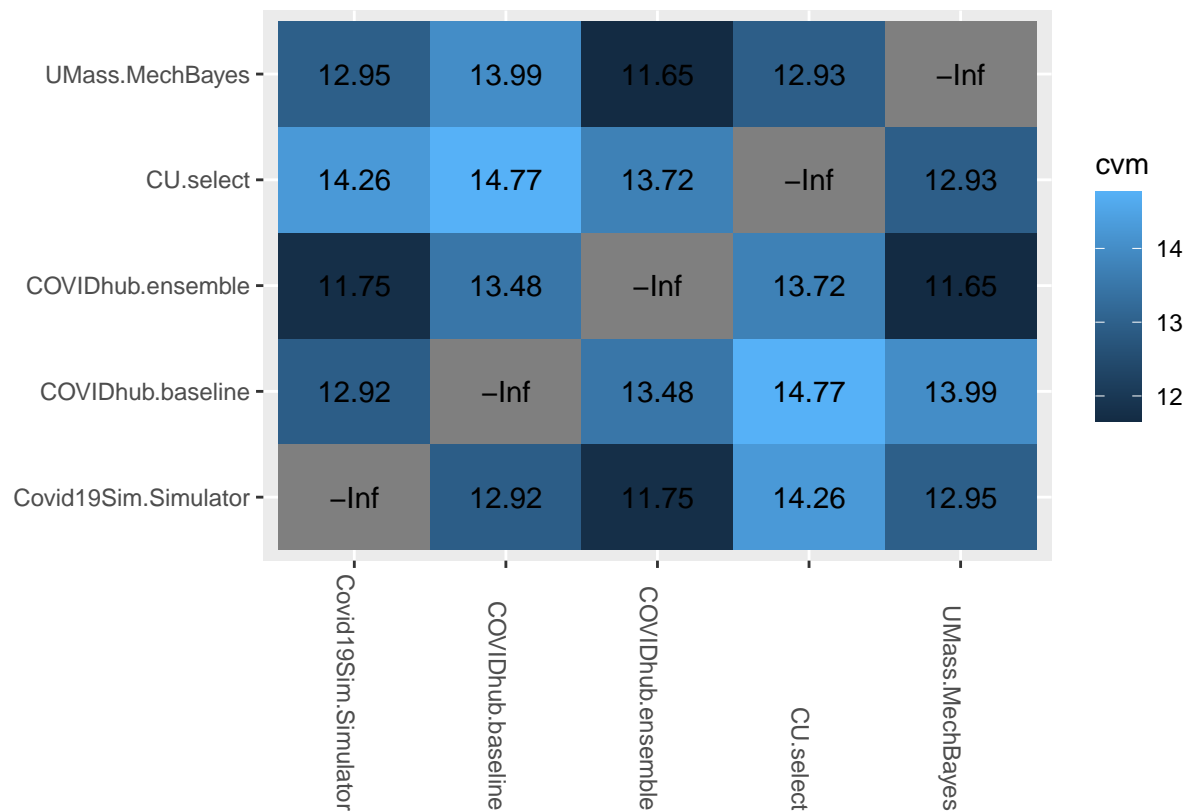


Now we can create a matrix of CvM for the models as shown in the table. We do a log transformation on these values for the heatmap because they are huge:

```
cvm_mat <- build_CvM_frame(small,
                           spline_method="hyman",
                           target_list=unique(small$target_variable),
                           point_to_interpolate)
kable(cvm_mat[[1]])
```

	Covid19Sim.Simulator	COVIDhub.baseline	CU.select	UMass.MechBayes	COVIDhub.ensemble
Covid19Sim.Simulator	0.0	406507.1	1553381.9	419701.4	127087.3
COVIDhub.baseline	406507.1	0.0	2588370.7	1195018.8	717939.8
CU.select	1553381.9	2588370.7	0.0	410456.0	911061.4
UMass.MechBayes	419701.4	1195018.8	410456.0	0.0	115126.9
COVIDhub.ensemble	127087.3	717939.8	911061.4	115126.9	0.0

```
transf_mat <-log(cvm_mat[[1]])
CvM_heatmap(transf_mat)
```



Comparison of CvM values from various sources

```
for(i in 1:4){
  assign(paste0("cvm_vals",i),build_CvM_frame_compare(small,
                                                         spline_method="hyman",
                                                         target_list=unique(small$target_variable),
                                                         point_to_interpolate,
                                                         est_method=i))
}
```

```
}
kable(cvm_vals1[[1]],caption="From cramer package")
```

Table 2: From cramer package

	Covid19Sim.Simulator	COVIDhub.baseline	CU.select	UMass.MechBayes	COVIDhub.ensemble
Covid19Sim.Simulator	0.0	406507.1	1553381.9	419701.4	127087.3
COVIDhub.baseline	406507.1	0.0	2588370.7	1195018.8	717939.8
CU.select	1553381.9	2588370.7	0.0	410456.0	911061.4
UMass.MechBayes	419701.4	1195018.8	410456.0	0.0	115126.9
COVIDhub.ensemble	127087.3	717939.8	911061.4	115126.9	0.0

```
kable(cvm_vals2[[1]],caption="From twosamples package")
```

Table 3: From twosamples package

	Covid19Sim.Simulator	COVIDhub.baseline	CU.select	UMass.MechBayes	COVIDhub.ensemble
Covid19Sim.Simulator	0.00000	209.0899	510.9355	165.28760	55.61059
COVIDhub.baseline	209.08991	0.0000	656.6394	471.68663	397.67584
CU.select	510.93550	656.6394	0.0000	233.43204	467.07893
UMass.MechBayes	165.28760	471.6866	233.4320	0.00000	60.88175
COVIDhub.ensemble	55.61059	397.6758	467.0789	60.88175	0.00000

```
kable(round(cvm_vals3[[1]],2),caption="From package")
```

Table 4: From package

	Covid19Sim.Simulator	COVIDhub.baseline	CU.select	UMass.MechBayes	COVIDhub.ensemble
Covid19Sim.Simulator	8.54	79.35	161.34	53.15	17.96
COVIDhub.baseline	79.35	0.00	164.16	117.92	99.42
CU.select	161.34	164.16	0.00	58.36	116.77
UMass.MechBayes	53.15	117.92	58.36	0.00	15.22
COVIDhub.ensemble	17.96	99.42	116.77	15.22	0.00

```
kable(cvm_vals4[[1]],caption="From http://estatcomp.github.io/henrique/exer\_chap8.html")
```

Table 5: From http://estatcomp.github.io/henrique/exer_chap8.html

	Covid19Sim.Simulator	COVIDhub.baseline	CU.select	UMass.MechBayes	COVIDhub.ensemble
Covid19Sim.Simulator	400440.4	446005.2	324111.5	358494.4	380704.7
COVIDhub.baseline	446005.2	491569.9	369676.3	404059.2	426269.5
CU.select	324111.5	369676.3	247782.7	282165.5	304375.8
UMass.MechBayes	358494.4	404059.2	282165.5	316548.4	338758.7
COVIDhub.ensemble	380704.7	426269.5	304375.8	338758.7	360969.0