Distance Calculation for COVID-19 Forecasting Models - Demo

Estee Cramer, Nick Reich, Nutcha Wattanachit

02/23/2021

Definitions

Cramer von-Mises criterion

The Cramer von-Mises criterion is defined as

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

where $F_n(x)$ is an empirical cumulative distribution function and $F^*(x)$ is a theoretical cumulative distribution function for a one-sample case.

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 (from `cramer::cramer.test()`).

$$T = \frac{mn}{(m+n)}\Big(\frac{2}{mn}\sum_{i=1..m,j=1..n}^{m,n}\phi(||X_i-Y_j||^2) - \frac{1}{m^2}\sum_{i=1..m,j=1..m}^{m,n}\phi(||X_i-X_j||^2) - \frac{1}{n^2}\sum_{i=1..n,j=1..n}^{m,n}\phi(||Y_i-Y_j||^2)\Big)$$

with $\phi_{\text{Cramer}}(z) = \sqrt{z}/2$. We can factor out the fraction at the front to get the distance. The formula that `twosample` use is $\sum |F_1(x) - F_2(x)|^p$ with $p = 2$. The CvM values differ greatly in scale.

Cramer's Distance

Cramer's Distance is defined as

$$CD(F, G) = \int_{-\infty}^{\infty} [F(x) - G(x)]^2 dx.$$

For univariate distributions, the Cramer's distance is exactly twice the energy distance. The current implementation in this demo uses the function `eqdist.e()` from the `energy` package to calculate the energy distance or statistic and then divide the number by 2 and by the sample size factor $n^2/2n$ to get Cramer's distance.

Setup Process

Since we have sample quantiles from COVID-19 forecasting models, we do not have the whole empirical distributions for the distance 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.

Simpler toy example of known distributions

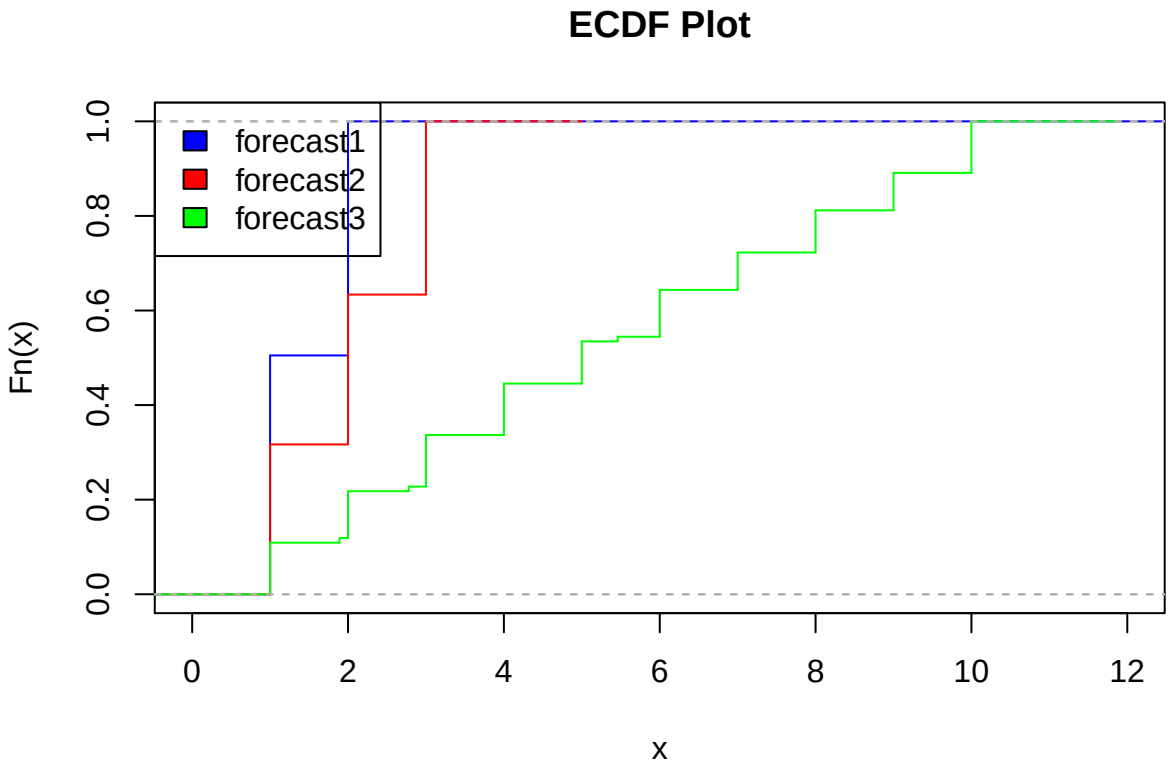
Due to large discrepancies in what we see above. We want to see how those functions compare in this toy example. We simulated 3 discrete uniform distributions:

$$f_1 \sim \mathrm{U}(1,2), f_2 \sim \mathrm{U}(1,3), f_3 \sim \mathrm{U}(1,10)$$

```
source("./distance_func_script.R")
# create toy data
n <- 1000
b <- c(2,3,10)
a <- c(1,1,1)
q <- seq(0, 1, by=0.01)
#  fill in bin probs
toy_forecasts <- data.frame(q=q) %>%
   dplyr::mutate(forecast1=sample_quantile(rdunif(n, b[1] , a[1]),q),
                 forecast2=sample_quantile(rdunif(n, b[2] , a[2]),q),
                 forecast3=sample_quantile(rdunif(n, b[3] , a[3]),q))

tnames <- c("forecast1","forecast2","forecast3")
```

We can see the clear step functions here since the distributions are discrete and and the ranges of values are relatively narrow :



Looking at these value make me wonder if some of these functions accommodate discrete distributions. I can try continuous distributions later.

Table 1: From cramer package

	forecast1	forecast2	forecast3
forecast1	0.0000000	0.1693564	1.979769
forecast2	0.1693564	0.0000000	1.426381
forecast3	1.9797693	1.4263813	0.000000

Table 2: From twosamples package

	forecast1	forecast2	forecast3
forecast1	0.000000	1.693596	30.93570
forecast2	1.693596	0.000000	20.47307
forecast3	30.935698	20.473069	0.00000

Table 3: From CDFt package

	forecast1	forecast2	forecast3
forecast1	40.61	15.03	89.09
forecast2	72.47	17.83	59.92
forecast3	125.50	91.61	1.23

Table 4: From http://estatcomp.github.io/henrique/exer_chap8.html

	forecast1	forecast2	forecast3
forecast1	-1.243737	-1.619129	-3.823166
forecast2	-1.619129	-1.994520	-4.198557
forecast3	-3.823166	-4.198557	-6.402595

Table 5: cramer's distance based on energy distance from energy package

	forecast1	forecast2	forecast3
forecast1	0.000	0.169	1.980
forecast2	0.169	0.000	1.426
forecast3	1.980	1.426	0.000

Table 6: Approx. cramer's distance based on quantile from Johannes

	forecast1	forecast2	forecast3
forecast1	0.000	0.173	1.992
forecast2	0.173	0.000	1.442
forecast3	1.992	1.442	0.000

Distance metrics of Some COVID-19 Forecasting Models

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

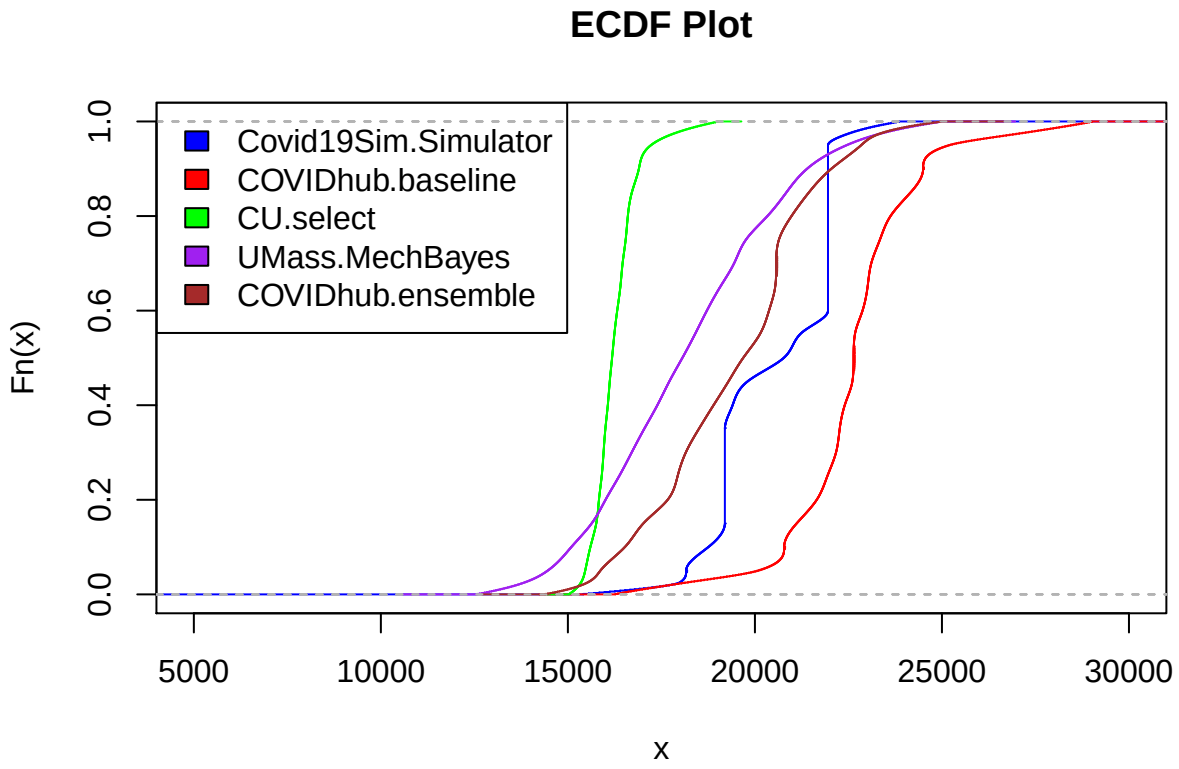
```
# create sample data for this demo
sample_frame <- load_latest_forecasts(models = c("CU-select", "UMass-MechBayes", "Covid19Sim-Simulator"
                                                 "COVIDhub-ensemble", "COVIDhub-baseline"),
                                      last_forecast_date = "2021-02-08",
                                      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/44238/
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:




```
cvm_mat <- build_distance_frame(small,
                            spline_method="hyman",
                            target_list=unique(small$target_variable),
                            point_to_interpolate,
                            distance="CvM")
cvm2_mat <- build_distance_frame(small,
                            spline_method="hyman",
                            target_list=unique(small$target_variable),
                            point_to_interpolate,
                            distance="CvM2")
cramer_mat <- build_distance_frame(small,
                            spline_method="hyman",
                            target_list=unique(small$target_variable),
                            point_to_interpolate,
                            distance="cramer")
approx_cd_mat <- build_distance_frame(small,
                            target_list=unique(small$target_variable),
                            distance="approx_cramer")
```

Table 7: From cramer package

	Covid19Sim.Simulator	COVIDhub.baseline	CU.select	UMass.MechBayes	COVIDhub.ensemble
Covid19Sim.Simulator	0.0000	812.202	3103.660	838.5643	253.9207
COVIDhub.baseline	812.2020	0.000	5171.570	2387.6500	1434.4452
CU.select	3103.6602	5171.570	0.000	820.0920	1820.3026
UMass.MechBayes	838.5643	2387.650	820.092	0.0000	230.0238
COVIDhub.ensemble	253.9207	1434.445	1820.303	230.0238	0.0000

Table 8: 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

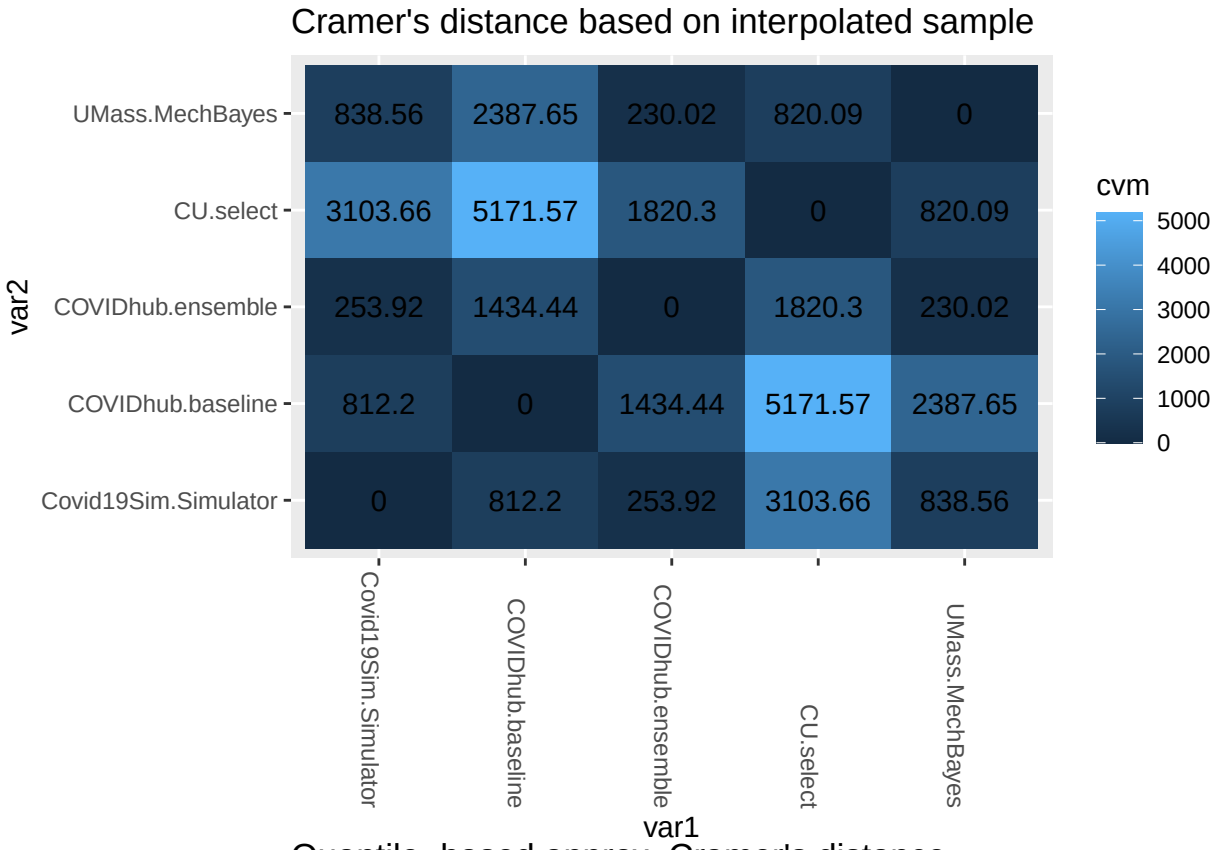
Table 9: cramer's distance based on energy distance from energy package

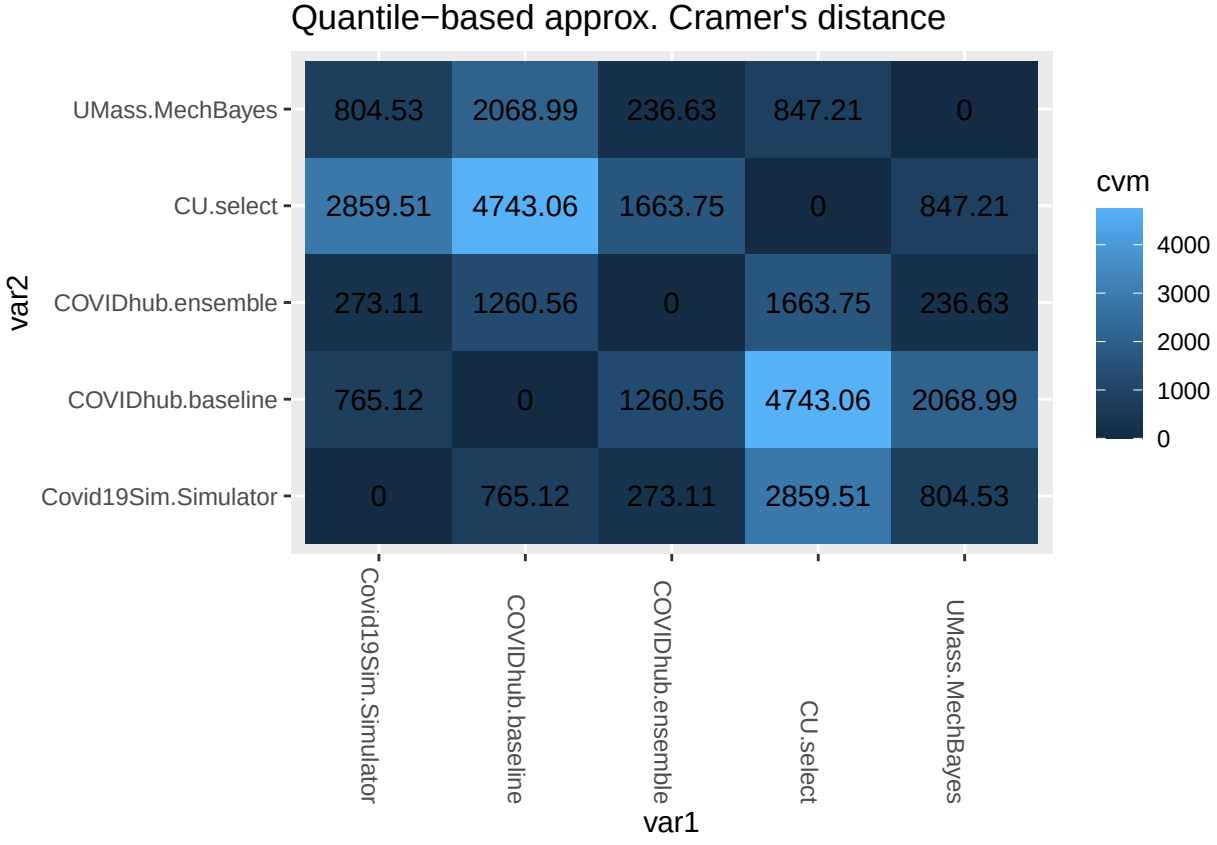
	Covid19Sim.Simulator	COVIDhub.baseline	CU.select	UMass.MechBayes	COVIDhub.ensemble
Covid19Sim.Simulator	0.000	812.202	3103.660	838.564	253.921
COVIDhub.baseline	812.202	0.000	5171.570	2387.650	1434.445
CU.select	3103.660	5171.570	0.000	820.092	1820.303
UMass.MechBayes	838.564	2387.650	820.092	0.000	230.024
COVIDhub.ensemble	253.921	1434.445	1820.303	230.024	0.000

Table 10: Approx. cramer's distance based on quantile from Johannes

	Covid19Sim.Simulator	COVIDhub.baseline	CU.select	UMass.MechBayes	COVIDhub.ensemble
Covid19Sim.Simulator	0.000	765.125	2859.507	804.531	273.110
COVIDhub.baseline	765.125	0.000	4743.063	2068.992	1260.557
CU.select	2859.507	4743.063	0.000	847.207	1663.754
UMass.MechBayes	804.531	2068.992	847.207	0.000	236.627
COVIDhub.ensemble	273.110	1260.557	1663.754	236.627	0.000

We can visualize the matrices above in heat maps shown below.


Cramer's distance based on interpolated sample
UMass.MechBayes
CU.select
COVIDhub.ensemble
COVIDhub.baseline
Covid19Sim.Simulator
838.56
2387.65
230.02
820.09
0
3103.66
5171.57
1820.3
0
820.09
253.92
1434.44
0
1820.3
230.02
812.2
0
1434.44
5171.57
2387.65
0
812.2
253.92
3103.66
838.56
var2
var1
cvm
5000
4000
3000
2000
1000
0


Quantile−based approx. Cramer's distance
UMass.MechBayes
CU.select
COVIDhub.ensemble
COVIDhub.baseline
Covid19Sim.Simulator
804.53
2068.99
236.63
847.21
0
2859.51
4743.06
1663.75
0
847.21
273.11
1260.56
0
1663.75
236.63
765.12
0
1260.56
4743.06
2068.99
0
765.12
273.11
2859.51
804.53
var2
var1
cvm
4000
3000
2000
1000
0