Supplement for Survival Analysis Class

Jiung Huang

Define the cross-validation c-index:

$$\mathrm{CV}(\alpha)=\frac{\sum_{i<j}\left\{I(t_i<t_j)I\left(PI_i(\alpha)>PI_j(\alpha)\right)\delta_i+I(t_j<t_i)I\left(PI_j(\alpha)>PI_i(\alpha)\right)\delta_j\right\}}{\sum_{i<j}\{I(t_i<t_j)\delta_i+I(t_j<t_i)\delta_j\}}$$

t_i: survival times of the *i*th data.

δ_i: censoring indicators, 1=death, 0=censoring

$PI_i(\alpha)$: weight sum of the covariates

$$PI_i(\alpha)=\widehat{\boldsymbol{\beta}}'(\alpha)\boldsymbol{x}_i$$

$$=\hat{\beta}_1{}'(\alpha)x_{i1}+\cdots+\hat{\beta}_p{}'(\alpha)x_{ip}$$

We applied the copula-base univariate Cox regression to the 63 patients (training data from the data of Lung) bu using R package.

```
$beta_hat
 [1] -0.093375993 -0.408434049  0.130353250  0.098116445  0.241605405 -0.336581254
 [7]  0.261509722  0.398902263  0.266095768  0.268362732  0.184223873  0.263512860
[13] -0.488655251  0.197619906 -0.375110481  0.006129582  0.278121367  0.288987114
[19] -0.320723316 -0.082094997  0.257818708 -0.330515772  0.656239877 -0.624875463
[25]  0.540572116  0.271172066 -0.094673161  0.249274883  0.189003907  0.320792426
[31] -0.374360799  0.263925299  0.242777859 -0.331636901  0.237660825 -0.037019876
[37]  0.219903406 -0.201255558  0.264127602 -0.368379750  0.205859531  0.303007578
[43] -0.222823285  0.350905163  0.435007249  0.018825664  0.385610958  0.318174211
[49]  0.458662027 -0.017221212  0.241465851  0.466359369  0.915647516 -0.322410560
[55]  0.328023360  0.498317571  0.601208808  0.504701791  0.199566503  0.390451330
[61]  0.507833855  0.772102576  0.509293019  0.281418637  0.508153560  0.378453974
[67] -0.235335253  0.128602397  0.327407377  0.500834157  0.206542616  0.221622253
[73]  0.047233861 -0.371953261 -0.060417248  0.484060622  0.442705398  0.074746746
[79]  0.401623933  0.142455670  0.033660290  0.013279896 -0.094809515 -0.083531607
[85] -0.229396745 -0.270049288  0.238565294  0.500428628  0.399985747  0.436724494
[91] -0.247422735  0.566990435 -0.114654945  0.598094043  0.125716821  0.151322792
[97]  0.246196263
$SE
 [1] 0.1770702 0.1688348 0.1635717 0.1916058 0.3580466 0.1271631 0.1264374 0.2333966
 [9] 0.6685067 0.1705944 0.1920564 0.1430458 0.1510608 0.2771124 0.2034492 0.1349685
[17] 0.2584253 0.2047095 0.1670090 0.1799098 0.1723210 0.1641447 0.3007397 0.3588277
[25] 0.2043980 0.4691894 0.1777181 0.2968683 0.1492934 0.2709715 0.3157396 0.2277556
[33] 1.0634165 0.1480238 0.1848744 0.2686585 0.2042879 0.1768051 0.1781602 0.1788449
[41] 0.2452871 0.2981336 0.2317791 0.1917537 0.2100240 0.1644698 0.2311310 0.2887320
[49] 0.1453256 0.2435763 0.2829067 0.1474677 0.3294600 0.1614177 0.2101401 0.1422035
[57] 0.2768746 0.1539714 0.2022367 0.1842173 0.1430272 0.2640303 0.1816979 0.1500292
[65] 0.2017935 0.1885809 0.1856818 0.1733486 0.1967285 0.2701241 0.1714122 0.1865465
[73] 0.1700973 0.1512034 0.1740836 0.3582821 0.1913027 0.1825437 0.2404864 0.1394064
[81] 0.1421643 0.2177837 0.1228830 0.1257640 0.1700413 0.2050155 0.3913870 0.2424865
[89] 0.1595170 0.1979140 0.1734282 0.1911465 0.2088276 0.2473197 0.2727306 0.3000505
[97] 0.2209265
```

```
$Z
 [1] -0.52733890 -2.41913398  0.79691826  0.51207457  0.67478754 -2.64684655  2.06829343
 [8]  1.70911767  0.39804502  1.57310416  0.95921770  1.84215779 -3.23482533  0.71313986
[15] -1.84375502  0.04541492  1.07621576  1.41169404 -1.92039504 -0.45631187  1.49615381
[22] -2.01356340  2.18208578 -1.74143614  2.64470336  0.57795859 -0.53271535  0.83968179
[29]  1.26598969  1.18386029 -1.18566298  1.15880936  0.22829988 -2.24042921  1.28552555
[36] -0.13779528  1.07643883 -1.13829055  1.48252901 -2.05977222  0.83925939  1.01634817
[43] -0.96136071  1.82997841  2.07122613  0.11446273  1.66836527  1.10197059  3.15609909
[50] -0.07070151  0.85351751  3.16245011  2.77923725 -1.99736804  1.56097444  3.50425649
[57]  2.17141213  3.27789392  0.98679653  2.11951554  3.55060986  2.92429568  2.80296667
[64]  1.87575926  2.51818604  2.00685189 -1.26741153  0.74187165  1.66425970  1.85408921
[71]  1.20494693  1.18802686  0.27768735 -2.45995321 -0.34705874  1.35105996  2.31416174
[78]  0.40947309  1.67004867  1.02187306  0.23677030  0.06097745 -0.77154301 -0.66419316
[85] -1.34906473 -1.31721369  0.60953808  2.06373806  2.50748037  2.20663808 -1.42665775
[92]  2.96626059 -0.54904112  2.41830284  0.46095604  0.50432439  1.11438066

$P
 [1] 0.5979582683 0.0155575088 0.4254985048 0.6085988212 0.4998107573 0.0081246211
 [7] 0.0386124391 0.0874291554 0.6905969991 0.1156947029 0.3374490861 0.0654520727
[13] 0.0012171717 0.4757592171 0.0652188513 0.9637765902 0.2818307714 0.1580400676
[19] 0.0548080201 0.6481657318 0.1346135770 0.0440553927 0.0291032028 0.0816071584
[25] 0.0081762560 0.5632920791 0.5942306396 0.4010868276 0.2055167744 0.2364683702
[31] 0.2357554204 0.2465339033 0.8194131195 0.0250630715 0.1986086942 0.8904022115
[37] 0.2817310421 0.2549991767 0.1381995909 0.0394203216 0.4013237643 0.3094636186
[43] 0.3363708347 0.0672531679 0.0383376691 0.9088709927 0.0952432376 0.2704744564
[49] 0.0015989453 0.9436353187 0.3933723842 0.0015644754 0.0054486713 0.0457852167
[55] 0.1185297828 0.0004578841 0.0299000330 0.0010458469 0.3237423957 0.0340469221
[61] 0.0003843397 0.0034523664 0.0050634902 0.0606883449 0.0117960995 0.0447654357
[67] 0.2050081793 0.4581651011 0.0960605414 0.0637263984 0.2282237891 0.2348228270
[73] 0.7812523683 0.0138955129 0.7285471898 0.1766762263 0.0206588516 0.6821925126
[79] 0.0949097354 0.3068409847 0.8128349900 0.9513771670 0.4403851376 0.5065666913
[85] 0.1773161756 0.1877670062 0.5421678406 0.0390425590 0.0121595336 0.0273393544
[91] 0.1536785699 0.0030144495 0.5829772302 0.0155930932 0.6448301459 0.6140334403
[97] 0.2651159165

$alpha
[1] 18

$c_index
[1] 0.6312719
```

data_train : the training data for 63 patients with 97 genes.

data_test : the testing data for 62 patients with 97 genes.

data_train_TOP16 : the training data for 63 patients with the 16 genes which are selected by P-values..

data_test_TOP16 : the testing data for 63 patients with the 16 genes which are selected by P-values..

Code

```
data_train = Lung[which(Lung$train == "TRUE"),]

data_test  = Lung[which(Lung$train == "FALSE"),]


gene_TOP16 = colnames(Lung)[-c(1,2,3)][order(res$P)][1:16]

data_TOP16 = Lung[,c("t.vec","d.vec","train",gene_TOP16)]

data_train_TOP16 = data_TOP16[which(Lung$train == "TRUE"),]

data_test_TOP16  = data_TOP16[which(Lung$train == "FALSE"),]


# this is calculate PI_i with 97 genes.

PI = function(datax, i){

  gene = as.numeric(datax[i,-c(1,2,3)])

  return(sum(res$beta_hat * gene))

}


# this is calculate PI_i with Top 16 genes.

PI_16 = function(datax, i){

  gene_16     = as.numeric(datax[i,-c(1,2,3)])

  beta_hat_16 = res$beta_hat[order(res$P)][1:16]

  return(sum(beta_hat_16 * gene_16))

}
```

```
#calculate the c_index with beta_hat and formula

c_index = function(data1, selectPI){

  PI = selectPI

  n = nrow(data1)

  numerator_1 = 0

  numerator_2 = 0

  denominator = 0

  for (i in c(2 : n)) {

    for (j in c(1 : (i-1))) {

   # this is the count for I(t_i < t_j)I(PI_i(α) > PI_j(α))δ_i

     numerator_1 = numerator_1 +

                   (((data1$t.vec[i] < data1$t.vec[j]) * 1) *

                   (PI(data1,i) > PI(data1,j)) * 1 * data1$d.vec[i])

     # this is the count for I(t_j < t_i)I(PI_j(α) > PI_i(α))δ_j

     numerator_2 = numerator_2 +

                   (((data1$t.vec[i] > data1$t.vec[j]) * 1) *

                   (PI(data1,i) < PI(data1,j)) * 1 * data1$d.vec[j])

      # this is the term for I(t_i < t_j)δ_i + I(t_j < t_i)δ_j

     denominator = denominator +

          ((data1$t.vec[i] < data1$t.vec[j]) * 1 * data1$d.vec[i] +

           (data1$t.vec[i] > data1$t.vec[j]) * 1 * data1$d.vec[j] )

    }

  }

  cat(" Numerator number : ", numerator_1 + numerator_2, "\n",

     "Denominator number : ", denominator, "\n",
```

	concordant	discordant	sum	C_index

```
      "c_index : ", (numerator_1 + numerator_2)/denominator, "\n")

}


c_index(data_train, PI)

c_index(data_train_TOP16, PI_16)

c_index(data_test, PI)

c_index(data_test_TOP16, PI_16)
```

Outcome

```
> c_index(data_train, PI)
 Numerator number :  652
 Denominator number :  856
 c_index :  0.7616822
> c_index(data_train_TOP16, PI_16)
 Numerator number :  590
 Denominator number :  856
 c_index :  0.6892523
> c_index(data_test, PI)
 Numerator number :  473
 Denominator number :  786
 c_index :  0.6017812
> c_index(data_test_TOP16, PI_16)
 Numerator number :  425
 Denominator number :  786
 c_index :  0.5407125
```

Because the c-index is very strange in the previous outcome, so I check my code again and do some modification. The new outcomes seem more reasonable. And then I use the code survConcordance from the *survival* R package to count the concordant and compare the values with my outcomes.

survConcordance	Training data (97 genes)	653	204	857	0.7619603
	Testing data (97 genes)	473	314	787	0.6010165
	Training data (16 genes)	590	267	857	0.6887781
	Testing data (16 genes)	426	361	787	0.5412961
My function	Training data (97 genes)	652	204	856	0.6892523
	Testing data (97 genes)	473	313	786	0.6017812
	Training data (16 genes)	590	266	856	0.6892523
	Testing data (16 genes)	425	361	786	0.5407125

After comparing, I find they're a little different between survConcordance and my function, but the c-index is almost the same. So I think it's a good result which is using my code.

Code

```
#PI, survival times, censored indicator for training data with 97 genes.

PI_T = c(as.matrix(data_train[,-c(1,2,3)]) %*% as.matrix(res$beta_hat))

t.vec_T = Lung$t.vec[Lung$train == TRUE]

d.vec_T = Lung$d.vec[Lung$train == TRUE]



#PI, survival times, censored indicator for testing data with 97 genes.

PI_F = c(as.matrix(data_test[,-c(1,2,3)]) %*% as.matrix(res$beta_hat))

t.vec_F = Lung$t.vec[Lung$train == FALSE]

d.vec_F = Lung$d.vec[Lung$train == FALSE]



#PI, survival times, censored indicator for training data with Top 16 genes.

PI_16_T = c(as.matrix(data_train_TOP16[,-c(1,2,3)]) %*%

as.matrix(beta_hat_16))

t.vec_16_T = data_TOP16$t.vec[data_TOP16$train == TRUE]

d.vec_16_T = data_TOP16$d.vec[data_TOP16$train == TRUE]



#PI, survival times, censored indicator for testing data with Top 16 genes.

PI_16_F = c(as.matrix(data_test_TOP16[,-c(1,2,3)]) %*%

as.matrix(beta_hat_16))

t.vec_16_F = data_TOP16$t.vec[data_TOP16$train == FALSE]

d.vec_16_F = data_TOP16$d.vec[data_TOP16$train == FALSE]
```

```
survConcordance(Surv(t.vec_T, d.vec_T) ~ PI_T)
```

```
survConcordance(Surv(t.vec_F, d.vec_F) ~ PI_F)

survConcordance(Surv(t.vec_16_T, d.vec_16_T) ~ PI_16_T)

survConcordance(Surv(t.vec_16_F, d.vec_16_F) ~ PI_16_F)
```

Outcome

```
> survConcordance(Surv(t.vec_T, d.vec_T) ~ PI_T)
Call:
survConcordance(formula = Surv(t.vec_T, d.vec_T) ~ PI_T)

  n= 63
Concordance= 0.7619603 se= 0.06990057
concordant discordant  tied.risk  tied.time   std(c-d)
  653.0000   204.0000     0.0000     0.0000   119.8096
> survConcordance(Surv(t.vec_F, d.vec_F) ~ PI_F)
Call:
survConcordance(formula = Surv(t.vec_F, d.vec_F) ~ PI_F)

  n= 62
Concordance= 0.6010165 se= 0.07226607
concordant discordant  tied.risk  tied.time   std(c-d)
  473.0000   314.0000     0.0000     1.0000   113.7468
> survConcordance(Surv(t.vec_16_T, d.vec_16_T) ~ PI_16_T)
Call:
survConcordance(formula = Surv(t.vec_16_T, d.vec_16_T) ~ PI_16_T)

  n= 63
Concordance= 0.6884481 se= 0.06989828
concordant discordant  tied.risk  tied.time   std(c-d)
  590.0000   267.0000     0.0000     0.0000   119.8056
> survConcordance(Surv(t.vec_16_F, d.vec_16_F) ~ PI_16_F)
Call:
survConcordance(formula = Surv(t.vec_16_F, d.vec_16_F) ~ PI_16_F)

  n= 62
Concordance= 0.5412961 se= 0.07226607
concordant discordant  tied.risk  tied.time   std(c-d)
  426.0000   361.0000     0.0000     1.0000   113.7468
```