

```

1 #!/usr/bin/env lua
2 local help = [
3 binr.lua : build rules via stochastic incremental XAI
4 (c) 2023, Tim Menzies, timm@ieee.org, mit-license.org
5
6 Options:
7   -h           Show help.
8   -e era=10    Number of rows in an era
9   -b bins=7    Number of bins for discretization.
10  -B Budget=30 Max rows to eval.
11  -l lives=5   Number of lives.
12  -r repeats=20 Number of experimental repeats.
13  -s seed=42   Random number seed.
14  -f file=../data/auto93.csv ]
15
16 -- coerce(s) --> v ; Return int or float or bool or string from 's'.
17 local function coerce(s)
18   if s then return tonumber(s) or smatch("^s*(.-)is*$") end end
19
20 local the{}; for k,v in help:gmatch("(%)S+)(%S+)" do the[k] = coerce(v) end
21 math.randomseed(the.seed)
22
23 local DATA, NUM, SYM, COLS, clone, adds
24
25 --# Lib
26
27 local abs,exp,sqrt,log = math.abs, math.exp, math.sqrt, math.log
28 local floor,min,max,rand,cos = math.floor,math.min,math.max,math.random,math.cos
29 local say,fmt = io.write, string.format
30
31 -- sort(a,f) --> a ; Sort 'a' using function 'f'.
32 local sort = function(a,f) table.sort(a,f); return a end
33
34
35 -- C(v|t) --> s ; Return a string representation of 'v'.
36 local function C(v,t) return list(v) end
37 list=function(u,v) for _,v in pairs(u) do u[1+#u]=o(v) end; return sort(u) end
38 dict=function(d,u)
39   for k,v in pairs(d) do u[1+#u]=fmt("%s%%s",k,o(v)) end; return sort(u) end
40 return type(v) == "number" and fmt(v==0 and "%0." or "%."..v, v) or
41 type(v) == "table" and tostring(v) or
42 {"!",".",table.concat({#v>0 and list or dict}(v),{","," "})..""} end
43
44 -- s2a(s) --> a ; Return array of words from string 's', split on ",".
45 local function s2a(s, a)
46   a={}; for s1 in sigmatch"(%)+" do a[1+#a] = coerce(s1) end; return a end
47
48 -- csv(file) --> f ; Iterator that returns rows from 'file'.
49 local function csv(file, src)
50   src = assert(io.open(file))
51   return function() s = src:read(); if s then return s2a(s) else src:close() end end end
52
53 -- shuffle(t) --> t ; Randomly shuffle the order of elements in 't'.
54 local shuffle = function(t, n)
55   for m#=2,-1 do math.random(m); t[m],t[n]=t[n],t[m] end; return t end
56
57 -- box_muller(mu,sd) --> n ; Return a random number from a Gaussian 'mu','sd'.
58 local function box_muller(mu,sd)
59   return mu + sd * sqrt(-2 * log(rand())) * cos(2 * math.pi * rand()) end
60
61 --## Classes
62
63 -- DATA(src:s|t) --> DATA ; Create a new DATA, populated with 'src'.
64 function DATA(s|t) return adds(src, {n=0,rows={},cols=nil}) end
65
66 -- clone(data,src) --> DATA ; Return a new DATA with same structure as 'data'.
67 function clone(data, src) return adds(src, DATA(data.cols.names)) end
68
69 -- NUM(at=0,v="") --> NUM ; Create a NUM object to summarize numbers.
70 function NUM(at,v)
71   return {at=at, 0, of=v or "", n=0, mu=0, m2=0, sd=0, bins={},}
72   best=(tostring(v) or ""):find"%" and 1 or 0 end
73
74 -- SYM(at,v="") --> SYM ; Create a SYM object to summarize symbols.
75 function SYM(at,v)
76   return {at=at, of=v, n=0, has={}, bins={}} end
77
78 -- COLS(row) --> COLS ; Create a COLS object from a list of column names.
79 function COLS(row, t,x,y,all)
80   x,y,all = {},{},{}
81   for n,s in ipairs(row) do
82     all[n] = smatch"^(A-Z)*" and NUM or SYM(n,s)
83     if n:smatch"X$*" then
84       t = s:find"%" and y or x
85       t[1#+t] = all[n] end end
86   return {all=all, x=x, y=y, names=row} end

```

```

87 --## Methods
88
89 -- add(i:DATA|NUM|SYM, z:v|t) --> z ; Update 'i' with 'z'.
90 local function add(i,z)
91   if z == "?" then return z end
92   i.n = i.n + 1
93   if i.mu then i.mu[i].has[z] = 1 + (i.has[z] or 0)
94   elseif i.mu then
95     local d = z - i.mu
96     i.mu = i.mu + d / i.n
97     i.m2 = i.m2 + d * (d - i.mu)
98     i.s = sqrt(d^2 / n or sqrt((max(0,i.m2)/(i.n - 1)))
99   end
100  elseif i.rows then
101    if not i.cols then i.cols = COLS(z) else
102      for _,col in pairs(i.cols.all) do add(col, z[col.at]) end
103      i.rows[1 + #i.rows] = z end end
104  return z end
105
106 -- adds(srcs:|,it=NUM()) --> it ; Update 'it' with all items from 'src'.
107 function adds(srcs, it)
108   it = it or NUM()
109   if type(srcs) == "string"
110     then for row in csv(srcs) do add(it,row) end
111   else for _,row in pairs(srcs or {}) do add(it,row) end end
112   return it end
113
114 -- norm(num,v) --> n ; Normalize 'v' 0..1 using 'i'.
115 local function norm(num,v)
116   return 1 / (1 + math.exp(-1.702 * (v - num.mu)/(num.sd + 1e-32))) end
117
118 -- bin(col,v) --> n ; Normalize 'v' 0..bins-1 using 'i'.
119 local function bin(col,v)
120   return (col.has[v] or v=="?") and v or floor( the.bins * norm(col,v) ) end
121
122 -- disty(data,row) --> n ; Return distance of 'row' to best goal (using Y cols).
123 local function disty(data,row)
124   d=0; for y in pairs(data.cols.y) do d=d+(norm(y,row[y].at) - y.best)^2 end
125   return sqrt(d/#data.cols.y) end
126
127 --## Think
128
129 -- scoreGet(data,n) --> n ; Score row by sum score of the bins it uses.
130 local function scoreGet(data, n, b)
131   r = 0
132   for _,col in pairs(data.cols.x) do
133     b = bin(data, row[col.at])
134     if b == "?" then
135       if col.bins[b] then
136         r = r + col.bins[b].mu end end end
137   return r end
138
139 -- scoreGet(data,row,n) --> nil ; Add a score 'n' to each bin used by this row.
140 local function scorePut(data, row,n, b,y)
141   for _,col in pairs(data.cols.x) do
142     b = bin(data, row[col.at])
143     if b == "?" then
144       col.bins[b] = col.bins[b] or NUM(col.at, b)
145       add(col.bins[b], n) end end end
146
147 -- scoreGuess(data,m,n,rows)-->t ; sort rows[m] to rows[n] by their guesses
148 local function scoreGuess(data,m,n,rows, t)
149   t = {}
150   for i=m,(m or 1).min#rows, n or #rows) do
151     if i < n then
152       t[1+i] = (scoreGet(data, rows[n]), rows[n]) end end
153   return sort(t, function(a,b) return a[1] < b[1] end) end
154
155 -- scoreSeen(data) -->data,n ; collect and print stats for this data
156 local function scoreSeen(data, t,m,epsi)
157   t={}; for m, row in pairs(data.rows) do t[1+i#t] = disty(data, row) end
158   t=sort(t)
159   m=t[1]
160   epsi = 0.35 * (t[1*m] - t[m])/2.56
161   print(fmt("%%.2f%.2f%.2f%.2f",eps=epsi))
162   print(fmt("%s",t[1*m], t[3*m], t[5*m], t[7*m], eps))
163   return data,epsi end
164
165 -- score(data,epsi) --> row,n,n ; Guess what are good rows in data.
166 local function score(data,epsi)
167   seen, labelled, row, bestRow, bestY, loves, bestY, lives, n
168   print""
169   labelled = clone(data)
170   bestY = 1e32
171   lives = lives or the.lives
172   seen = {}
173   seen[""] = true
174   for m, row in pairs(data.rows) do
175     if lives < 0 or n >= the.Budget then break end
176     add(labelled, row)
177     scorePut(labelled, row,disty(labelled, row))
178     seen[row] = true
179     if m < the.era=0 then
180       best = scoreGuess(labelled, 1, m+20, data.rows)[1][2]
181       if not seen[best] then seen[best]=best; n=n+1 end
182       y = disty(data, best)
183       if y < bestY then
184         bestY, bestRow = y,best ; say"! "
185         else lives = lives - 1 ; say". "
186       end end end
187   return bestRow, bestY, n end

```

```

188 --## Demos
189
190 local egs={}
191
192 egs["+"] = function(_) print("un .help..un") end
193 egs["-he"] = function(n) math.randomseed(n or the.seed); the.seed = n end
194 egs["-shuffle"] = function(_) print(o(shuffle(10,20,30,40,50))) end
195
196 egs["-csv"] = function(_, n)
197   for row in csv(the.file) do
198     if n % 25 == 0 then print(o(row)) end
199   n = n + 1 end end
200
201 egs["-num"] = function(_, num)
202   num=NUM()
203   for _=1,1000 do add(num, box_muller(10,5)) end
204   print(fmt("%.3f.M",num.mu,num.sd)) end
205
206 egs["-data"] = function_()
207   for n,col in pairs(DATA(the.file).cols.x) do
208     print(n,o(col)) end end
209
210 egs["-disty"] = function(_, data,num,t)
211   data,t = DATA(the.file), {}
212   for n, row in pairs(data.rows) do
213     if n % 25 == 0 then t[l#+t] = disty(data, row) end end
214   print(o(sort(t))) end
215
216 egs["-score"] = function(_, t,data,epsi,y)
217   data,eps = scoreSeen(DATA(the.file))
218   t={}
219   for n = 1, the.repeats do
220     data.rows = shuffle(data.rows)
221     y,sees = scoreSeen(epsi)
222     t[(n-1)*100+y/1] = sees
223   print("un...o(sort(t))) end")
224
225 egs["-all"] = function(_, n)
226   n = the.seed
227   for k,_ in pairs(egs) do
228     math.randomseed(n)
229     if k=="-all" then print("un----",k); egs[k]() end end end
230
231 -- cli(d,funs) --> nil ; Update 'd' with flags from command-line; run 'funs'.
232 local cli = cli(d,funs)
233 for k,_ in pairs(egs) do
234   if funs[s](coerce(arg[i+1])) then
235     else for k_ in pairs(d) do
236       if k_:sub(1,1)==s:sub(2) then d[k]=coerce(arg[i+1]) end end end end
237
238 if arg[0]:find"binr.lua" then cli(the,egs) end

```