

## binr.lua

Page 1/3

```

1 #!/usr/bin/env lua
2 local help = {}
3 binr.lua : build rules via stochastic incremental XAI
4 (c) 2025, Tim Menzies, tim@ieee.org, mit-license.org
5
6 Options:
7 -h          Show help.
8 -s 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 s:match`^%s*(.)%s*$' end end
19
20 local the={}; for k,v in help:match`(%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 -- o(v,t) --> s ;; Return a string representation of 'v'.
36 local function o(v, list,dict)
37   list=function(a,u) for _,v in ipairs(a) do u[1+#u]=o(v) end; return 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%1==0 and "%0" or "%3", 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 sl in s:match`([^\s,])` do a[1+#a] = coerce(sl) end; return a end
47
48 -- csv(file:s) --> t ;; Iterator that returns rows from 'file'.
49 local function csv(file, src)
50   src = assert(io.open(file))
51   return function()
52     s = src:read(); if s then return s2a(s) else src:close() end end end
53
54 -- shuffle(t) --> t ;; Randomly shuffle the order of elements in 't'.
55 local shuffle = function(t, n)
56   for m=#t,2,-1 do n=math.random(m); t[m],t[n]=t[n],t[m] end; return t end
57
58 -- box_muller(mu, sd:n) --> n ;; Return a random number from a Gaussian 'mu', 'sd'.
59 local function box_muller(mu, sd)
60   return mu + sd * sqrt(-2 * log(rand())) * cos(2 * math.pi * rand()) end
61
62 --## Classes
63
64 -- DATA(src:s|t) --> DATA ;; Create a new DATA, populated with 'src'.
65 function DATA( src) return adds(src, {n=0, rows={}, cols=nil}) end
66
67 -- clone(data,src) --> DATA ;; Return a new DATA with same structure as 'data'.
68 function clone(data, src) return adds(src, DATA(data.cols.names)) end
69
70 -- NUM(at=0,v="") --> NUM ;; Create a NUM object to summarize numbers.
71 function NUM(at,v)
72   return {at=at or 0, of=v or "", n=0, mu=0, m2=0, sd=0, bins={},
73     best=(tostring(v) or ""):find"%S" and 1 or 0} end
74
75 -- SYM(at=0,v="") --> SYM ;; Create a SYM object to summarize symbols.
76 function SYM(at,v) 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] = (s:match`^[A-Z]` and NUM or SYM)(n,s)
83     if not s:match"%S" then
84       t = s:find"%[i-j]" and y or x
85       t[1+#t] = all[n] end end
86   return {all=all, x=x, y=y, names=row} end

```

## binr.lua

Page 2/3

```

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.has then 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 * (z - i.mu)
98     i.sd = i.n<2 and 0 or sqrt((max(0,i.m2)/(i.n - 1)))
99   elseif i.rows then
100     if not i.cols then i.cols = COLS(z) else
101       for _,col in pairs(i.cols,all) do add(col, z[col.at]) end
102       i.rows[1 + #i.rows] = z end end
103   return z end
104
105 -- adds(src:s|t,it=NUM()) --> it ;; Update 'it' with all items from 'src'.
106 function adds(src, it)
107   it = it or NUM()
108   if type(src) == "string"
109   then for row in csv(src) do add(it,row) end
110   else for _,row in pairs(src or {}) do add(it,row) end end
111   return it end
112
113 -- norm(num,v) --> n ;; Normalize 'v' 0..1 using 'i'.
114 local function norm(num,v)
115   return 1 / (1 + math.exp(-1.702 * (v - num.mu)/(num.sd + 1e-32))) end
116
117 -- bin(col,v) --> n ;; Normalize 'v' 0..bins-1 using 'i'.
118 local function bin(col,v)
119   return (col.has or v=="?") and v or floor((the.bins * norm(col,v)) end
120
121 -- disty(data,row) --> n ;; Return distance of 'row' to best goal (using Y cols).
122 local function disty(data,row, d)
123   d=d; for _,y in pairs(data.cols,y) do d=d + (norm(y,row[y.at]) - y.best)^2 end
124   return sqrt(d/#data.cols.y) end
125
126 --## Think
127
128 -- scoreGet(data,row) --> n ;; Score row by sum score of the bins it uses.
129 local function scoreGet(data,row, b,n)
130   n = 0
131   for _,col in pairs(data.cols.x) do
132     b = bin(col, row[col.at])
133     if b ~= "" then
134       if col.bins[b] then
135         n = n + col.bins[b].mu end end end
136   return n end
137
138 -- scoreGet(data,row,n) --> nil ;; Add a score 'n' to each bin used by this row.
139 local function scorePut(data,row, b,y)
140   for _,col in pairs(data.cols.x) do
141     b = bin(col, row[col.at])
142     if b ~= "" then
143       col.bins[b] = col.bins[b] or NUM(col.at, b)
144       add(col.bins[b], n) end end end
145
146 -- scoreGuess(data,m,n,rows)--> t ;; sort rows[m] to rows[n] by their guesses
147 local function scoreGuess(data,m,n,rows, t)
148   t = {}
149   --print((m or 1),min(#rows, n or #rows))
150   for n = (m or 1),min(#rows, n or #rows) do
151     if n <= #rows then
152       t[1+#t] = (scoreGet(data, rows[n], rows[n]) end end
153     return sort(t, function(a,b) return a[1] < b[1] end) end end
154
155 -- scoreSeen(data)-->data,n ;; collect and print stats for this data
156 local function scoresSeen(data, t,m,eps)
157   t={}; for m,row in pairs(data.rows) do t[1+#t] = disty(data,row) end
158   t=sort(t)
159   m=t/10
160   eps = 0.35 * (t[9*m] - t[m])/2.56
161   print(fmt("(%2f,%2f,%2f,%2f,%2f,eps=%2f",
162     t[m], t[3*m], t[5*m], t[7*m], t[9*m], eps))
163   return data,eps end
164
165 -- score(data,eps)--> row,n,n ;; Guess what are good rows in data.
166 local function score(data,eps)
167   labelled = clone(data)
168   labelled = labelled
169   besty = 1e32
170   lives = lives or the.lives
171   seen = {}
172   n=0;
173   for n,row in pairs(data.rows) do
174     if lives < 0 or n >= the.Budget then break end
175     add(labelled, row)
176     scorePut(labelled, row,disty(labelled,row))
177     seen[row]=row; n=n+1
178     if m & the.era==0 then
179       best = scoreGuess(labelled, 1, m*20, data.rows)[1][2]
180       if not seen[best] then seen[best]=best; n=n+1 end
181       y = disty(data, best)
182       if y < besty - eps
183       then besty,bestRow = y,best ; say"!"
184       else lives = lives - 1 ; say"."
185       end end end
186   return bestRow, besty,n end

```

## binr.lua

Page 3/3

```

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

```