

## 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, timm@ieee.org, mit-license.org
5
6 Options:
7 -h          Show help.
8 -s ers=10   Number of rows in an ers
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
30 local say=io.write
31 local fmt = string.format
32
33 -- sort(t,f) --> t ;; Sort 't' using function 'f'.
34 local sort = function(t,f) table.sort(t,f); return t end
35
36 -- lt(f) --> f ;; Return a function that sorts 'a' and 'b' on 'f'.
37 local lt = function(f) return function(a,b) return f(a) < f(b) end end
38
39 -- cat(a) --> s ;; Return a string representation of array 'a'.
40 local cat = function(a) return "[".. table.concat(a,"") .."]" end
41
42 -- o(v) --> s ;; Return a string representation of 'v'.
43 local function o(v, list,dict)
44   list=function(a,u) for _,v in ipairs(a) do u[1+#u]=o(v) end; return u end
45   dict=function(u,v)
46     for k,v in pairs(d) do u[1+#u]=fmt("%.5s",k,o(v)) end; return sort(u) end
47   return type(v) == "number" and fmt(v%1==0 and "%.0f" or "%.3f", v) or
48     type(v) == "table" and tostring(v) or
49     cat({#v>0 and list or dict}(v,{})) end
50
51 -- s2a(s) --> a ;; Return array of words from string 's', split on " , ".
52 local function s2a(s, a)
53   a={} ; for sl in s:match("([^\s]+)") do a[1+#a] = coerce(sl) end; return a end
54
55 -- csv(file) --> f ;; Iterator that returns rows from 'file'.
56 local function csv(file, src)
57   src = assert(io.open(file))
58   return function( s )
59     s = src:read(); if s then return s2a(s) else src:close() end end end
60
61 -- shuffle(t) --> t ;; Randomly shuffle the order of elements in 't'.
62 local shuffle = function(t, n)
63   for m=#t,2,-1 do n=math.random(m); t[m],t[n]=t[n],t[m] end; return t end
64
65 -- cut(a0,n,data) --> t,t ;; Split 'a0' at 'n' (if 'data' exists,split that too).
66 local function cut(a0,n, data)
67   local a1,a2 = {},{}
68   for j,v in ipairs(a0) do if j <= n then a1[1+#a1]=v else a2[1+#a2]=v end end
69   return data and clone(data,a1),clone(data,a2) or a1,a2 end
70
71 -- mode(d) --> v ;; Return the most frequent key in 'd'.
72 local function mode(d, v,n)
73   v,n = nil,0
74   for v1,n1 in pairs(d) do if n1>n then v=v1,n1 end end
75   return v end
76
77 -- box_muller(mu,sd) --> n ;; Return a random number from a Gaussian 'mu','sd'.
78 local function box_muller(mu,sd)
79   return mu + sd * sqrt(-2 * log(rand())) * cos(6.28 * rand()) end
80
81 --## Classes
82
83 -- DATA(src) --> DATA ;; Create a new DATA, populated with 'src'.
84 function DATA( src) return adds(src, {n=0,rows={},cols=nil}) end
85
86 -- clone(i,src) --> DATA ;; Return a new DATA with same structure as 'i'.
87 function clone(i, src) return adds(src, DATA(i.cols.names)) end
88
89 -- NUM(at,s) --> NUM ;; Create a NUM object to summarize numbers.
90 function NUM(at,s)
91   return {at=at or 0, of=s, n=0, mu=0, m2=0, sd=0, bins={},
92     best=(tostring(s) or ""):find"%S" and 1 or 0} end
93
94 -- SYM(at,s) --> SYM ;; Create a SYM object to summarize symbols.
95 function SYM(at,s) return {at=at, of=s, n=0, has={}, bins={}} end
96
97 -- COLS(row) --> COLS ;; Create a COLS object from a list of column names.
98 function COLS(row, t,x,y,all)
99   x,y,all = {},{},{},{}
100   for n,s in ipairs(row) do
101     all[n] = (s:match"%[A-Z]" and NUM or SYM)(n,s)
102     if not s:match"%X%" then
103       t = s:find"%[a-z]" and y or x
104       t[1+#t] = all[n] end end
105   return {all=all, x=x, y=y, names=row} end

```

## binr.lua

Page 2/3

```

106 --## Methods
107
108 -- add(i,v) --> v ;; Update 'i' with 'v'.
109 local function add(i,v)
110   if v == "?" then return v end
111   i.n = i.n + 1
112   if i.has then i.has[v] = 1 + (i.has[v] or 0)
113   elseif i.mu then
114     local d = v - i.mu
115     i.mu = i.mu + d / i.n
116     i.m2 = i.m2 + d * (v - i.mu)
117     i.sd = i.n<2 and 0 or sqrt((max(0,i.m2)/(i.n - 1)))
118   elseif i.rows then
119     if not i.cols then i.cols = COLS(v) else
120       for _,col in pairs(i.cols.all) do add(col, v[col.at]) end
121     i.rows[1 + #i.rows] = v end end
122   return v end
123
124 -- adds(src,it) --> it ;; Update 'it' with all items from 'src'.
125 function adds(src, it)
126   it = it or NUM()
127   if type(src) == "string"
128     then for row in csv(src) do add(it,row) end
129     else for _,row in pairs(src or {}) do add(it,row) end end
130   return it end
131
132 -- norm(num,v) --> n ;; Normalize 'v' 0..1 using 'i'.
133 local function norm(num,v)
134   return 1 / (1 + math.exp(-1.7 * (v - num.sd) / (num.sd + 1e-32))) end
135
136 -- bin(i,v) --> n ;; Normalize 'v' 0..bins-1 using 'i'.
137 local function bin(i,v)
138   return (i.has or v=="?") and v or floor( (the.bins * norm(i,v)) end
139
140 -- disty(data,row) --> n ;; Return distance of 'row' to best goal (using Y cols).
141 local function disty(data,row, d)
142   d=0; for _,y in pairs(data.cols.y) do d= d + (norm(y, row[y.at]) - y.best)^2 end
143   return sqrt(d/#data.cols.y) end
144
145 local function distys(data,rows)
146   return sort(rows or data.rows,
147     function(a,b) return disty(data,a) < disty(data,b) end) end
148
149 --## Think
150
151 local function scoreGet(data,row, b,n)
152   n = 0
153   for _,col in pairs(data.cols.x) do
154     b = bin(col, row[col.at])
155     if b == "?" then
156       if col.bins[b]
157         then n = n + col.bins[b].mu end end end
158   return n end
159
160 local function scorePut(data,row, b,y)
161   y = disty(data,row)
162   for _,col in pairs(data.cols.x) do
163     b = bin(col, row[col.at])
164     if b == "?" then
165       col.bins[b] = col.bins[b] or NUM(col.at, b)
166       add(col.bins[b], y) end end end
167
168 local function scoreGuess(data,m,n,rows, t)
169   t = {}
170   --print((m or 1),min(#rows, n or #rows))
171   for n = (m or 1),min(#rows, n or #rows) do
172     if n <= #rows then
173       t[1+#t] = {scoreGet(data, rows[n]), rows[n]} end end
174   return sort(t, function(a,b) return a[1] < b[1] end) end
175
176 local function scoresSeen(data, t,m,eps)
177   t={} ; for m,row in pairs(data.rows) do t[1+#t] = disty(data,row) end
178   t=sort(t)
179   m=#t//10
180   eps = 0.35 * (t[9*m] - t[m])/2.56
181   print(fmt("%.2f,%.2f,%.2f,%.2f,%.2f,eps=%.2f",
182     t[m], t[3*m], t[5*m], t[7*m], t[9*m], eps))
183   return data,eps end
184
185 local function score(data,eps, labelled,rows,bestRow,besty,loves,best,y,lives,n)
186   print""
187   labelled = clone(data)
188   besty = le32
189   lives = lives or the.lives
190   seen = {}
191   n=0;
192   for m,row in pairs(data.rows) do
193     if lives < 0 or n = the.Budget then break end
194     add(labelled, row)
195     scorePut(labelled,row, n=1)
196     seen[row]=row, n=n+1
197     if m % the.ers==0 then
198       best = scoreGuess(labelled, 1, m+20, data.rows)[1][2]
199       if not seen[best] then seen[best]=rows; n=n+1 end
200       y = disty(data, best)
201       if y < besty - eps
202         then besty,bestRow = y,best ; say""
203         else lives = lives - 1 ; say""
204       end end end
205   return bestRow, besty,n end

```

## binr.lua

Page 3/3

```

207 --## Demos
208
209 local eggs={}
210
211 eggs["-h"] = function(_) print("\n".help.."\n") end
212 eggs["-s"] = function(n) math.randomseed(n or the.seed); the.seed =n end
213 eggs["-the"] = function(_) print(o(the)) end
214 eggs["--shuffle"] = function() print(o(shuffle(10,20,30,40,50))) end
215
216 eggs["--csv"] = function(_, n)
217   n=1; for row in csv(the.file) do
218     if n % 25 == 0 then print(o(row)) end
219     n = n + 1 end end
220
221 eggs["--num"] = function(_,num)
222   num=NUM()
223   for _,1,1000 do add(num, box_muller(10,5)) end
224   print(fmt("%.3f%.3f", num.mu, num.sd)) end
225
226 eggs["--data"] = function(_)
227   for n,col in pairs(DATA(the.file).cols.x) do
228     print(n,o(col)) end end
229
230 eggs["--disty"] = function(_, data,num,t)
231   data,t = DATA(the.file), {}
232   for n,row in pairs(data.rows) do
233     if n % 25 == 0 then t[1+#t] = disty(data,row) end end
234   print(o(sort(t))) end
235
236 eggs["--score"] = function(_, t,data,eps,u,y)
237   data,eps = scoresSeen(DATA(the.file))
238   t,u={},{}
239   for n = 1,the.repeats do
240     data,rows = shuffle(data.rows)
241     _,y,seen = score(data,eps)
242     [u]n] = seen
243     t[n] = 100*y//1 end
244   print(o(sort(u)).."\n"..o(sort(t))) end
245
246 eggs["--all"] = function(_, n)
247   n = the.seed
248   for k,_, in pairs(eggs) do
249     math.randomseed(n)
250     if k=="--all" then print("\n-----",k); eggs[k]() end end end
251
252 -- cli(d,funs) --> nil ;; Update 'd' with flags from command-line; run 'funs'.
253 local function cli(d,funs)
254   for i,s in pairs(arg) do
255     if funs[s]
256       then funs[s](coerce(arg[i+1]))
257       else for k,1 in pairs(sub(2) then d[k]=coerce(arg[i+1]) end end end end end
258   if arg[0]:find"binr.lua" then cli(the,eggs) end

```