

binr.lua

Page 1/3

```

1  #!/usr/bin/env lua
2  local help = [
3  binr.lua will 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  -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=1000 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:gmatch("(%)=(%)") 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 sayio,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 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(d,u)
46     for k,v in pairs(d) do u[1+#u]=fmt("%s%#s",k,o(v)) end; return sort(u) end
47   return type(v) == "table" and fmt(v.."-") or type(v) == "nil" or "" 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:gmatch"(%)+" do a[l+#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 = src:read(); if s then return s2a(s) else src:close() end end
59
60 -- shuffle(t) --> t ; Randomly shuffle the order of elements in `t`.
61 local shuffle=function(t, n)
62   for m=#t,1 do m=math.random(m); t[m],t[n]=t[n],t[m] end; return t end
63
64 -- cut(a,n,data) --> t ; Split 'a0' at `n` (if 'data' exists,split that too).
65 local function cut(a, n, data)
66   local al,a2={},{}
67   for j,v in ipairs(a) do if j <= n then al[#al]=v else a2[#a2]=v end end
68   return data and clone(data,al),clone(data,a2) or al,a2 end
69
70 -- mode(d) --> v ; Return the most frequent key in `d`.
71 local function mode(d, v,n)
72   v,n = nil,0
73   for v1,l1 in pairs(d) do if nl>n then v,n=v1,l1 end end
74   return v end
75
76 -- box_muller(mu, sd) --> n ; Return a random number from a Gaussian `mu`,`sd`.
77 local function box_muller(mu, sd)
78   return mu + sd * sqrt(-2 * log(rand())) * cos(6.28 * rand()) end
79
80 --## Classes
81
82 -- DATA(src) --> DATA ; Create a new DATA, populated with `src`.
83 function DATA( src ) return adds(src, {n=rows(),cols=cols()}) end
84
85 -- clone(i,src) --> DATA ; Return a new DATA with same structure as `i`.
86 function clone(i, src) return adds(src, DATA(i.cols.names)) end
87
88 -- NUM(s) --> NUM ; Create a NUM object to summarize numbers.
89 function NUM(at,s)
90   return {at=at, o=ofs, n=0, mu=0, m2=0, sd=0, bins={}, best=(tostring(s) or ""):find"%" and 1 or 0} end
91
92 -- SYM(at,s) --> SYM ; Create a SYM object to summarize symbols.
93 function SYM(at,s) return {at=at, of=ofs, n=0, has={}, bins={}} end
94
95 -- COLS(row) --> COLS ; Create a COLS object from a list of column names.
96 function COLS(row, t,x,y,all)
97   x,y,all = {},{},{}
98   for n,s in ipairs(row) do
99     all[n] = s:match'^%s-%s$' and NUM or SYM(n,s)
100    if not s:match"^-%s$" and y or x
101      then t[1+#t]=all[n] end end
102   return {all=all, x=x, y=y, names=row} end

```

binr.lua

Page 2/

```

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

```

binr.lua

Page 3/3

```

208 --## Demos
209
210 local egs=()
211
212 egs["-h"] = function(_) print("m..help..m") end
213 egs["-n"] = function(n) math.randomseed(n or the.seed); the.seed = n end
214 egs["-othe"] = function(_) print(o(the)) end
215 egs["-shuffle"] = function(_) print(o(shuffle(10,20,30,40,50))) end
216
217 egs["-cav"] = function(_, n)
218   n=1; for row in csv(the.file) do
219     if n % 25 == 0 then print(o(row)) end
220     n = n + 1 end end
221
222 egs["-num"] = function(_, num)
223   num=NUM
224   for i=1,1000 do add(num, box_muller(10,5)) end
225   print(fmt("%.3f%.3f", num.mu, num.sd)) end
226
227 egs["-data"] = function_()
228   for n,col in pairs(DATA(the.file).cols.x) do
229     print(n,o(col)) end end
230
231 egs["-disty"] = function(_, data,num,t)
232   data,t = DATA(the.file), {}
233   for n, row in pairs(data.rows) do
234     if n % 25 == 0 then t[1#t] = disty(data,row) end end
235   print(o(sort(t))) end
236
237 egs["-score"] = function(_,
238   t,data,eps,u,y)
239   data,eps = scoresSeen(DATA(the.file))
240   t,u={},{}
241   for n = 1, the.repeats do
242     data.rows = shuffle(data.rows)
243     y,seen = score(data,eps)
244     u[n] = seen
245     t[n] = 100*y/1 end
246   print(o(sort(u).."\n"..o(sort(t))) end
247
248 egs["-all"] = function(_,
249   p)
250   for k,_ in pairs(egs) do
251     math.randomseed(n)
252     if k=="-all" then print("m-----",k); egs[k]() end end end
253
254 -- cli(d,funs) => nil; Update 'd' with flags from command-line; run 'funs'.
255 local function cli(d,funs)
256   for i,s in pairs(arg) do
257     if funs[s] then funs[s](coerce(arg[i+1])) end
258     else for k,_ in pairs(d) do
259       if k:sub(1,1)==s:sub(2) then d[k]=coerce(arg[i+1]) end end end end
260
261 if arg[0]:find("huruua") then cli(the.egs) end

```