

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 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  -t 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(s, u)
45     for _,v in ipairs(a) do u[l+u] = o(v) end; return cat(u) end
46   dict = function(d, u)
47     for k,v in pairs(d) do u[l+u] = fmt("%.5s", k, o(v)) end
48     return cat(sort(u)) end
49   return type(v) == "number" and fmt(v)~=0 and "%0.0f" or "%0.3f", v) or
50   type(v) == "table" and tostring(v) or (#v>0 and list or dict)(v, {}) end
51
52 -- s2a(s) --> a ;; Return array of words from string 's', split on " , ".
53 local function s2a(s, a)
54   a={} ; for sl in s:match("[^,]+") do a[l+1] = coerce(sl) end; return a end
55
56 -- csv(file) --> f ;; Iterator that returns rows from 'file'.
57 local function csv(file, src)
58   src = assert(io.open(file))
59   return function(s)
60     s = src:read(); if s then return s2a(s) else src:close() end end end
61
62 -- shuffle(t) --> t ;; Randomly shuffle the order of elements in 't'.
63 local shuffle = function(t, n)
64   for m=#t,2,-1 do n=math.random(m); t[m],t[n]=t[n],t[m] end; return t end
65
66 -- cut(a0,n,data) --> t,t ;; Split 'a0' at 'n' (if 'data' exists,split that too).
67 local function cut(a0,n, data)
68   local al,a2 = {},{}
69   for j,v in ipairs(a0) do if j <= n then al[l+1]=v else a2[l+1]=v end end
70   return data and clone(data,al),clone(data,a2) or al,a2 end
71
72 -- mode(d) --> v ;; Return the most frequent key in 'd'.
73 local function mode(d, v,n)
74   v,n = nil,0
75   for vl,nl in pairs(d) do if nl>n then v,n=vl,nl end end
76   return v end
77
78 -- box_muller(mu,sd) --> n ;; Return a random number from a Gaussian 'mu','sd'.
79 local function box_muller(mu,sd)
80   return mu + sd * sqrt(-2 * log(rand())) * cos(6.28 * rand()) end
81
82 --## Classes
83
84 -- DATA(src) --> DATA ;; Create a new DATA, populated with 'src'.
85 function DATA( src) return adds(src, {n=0,rows={},cols=nil}) end
86
87 -- clone(i,src) --> DATA ;; Return a new DATA with same structure as 'i'.
88 function clone(i, src) return adds(src, DATA(i.cols.names)) end
89
90 -- NUM(at,s) --> NUM ;; Create a NUM object to summarize numbers.
91 function NUM(at,s)
92   return {at=at or 0, of=s, n=0, mu=0, m2=0, sd=0, bins={},
93     best=(tostring(s) or ""):find"%S" and 1 or 0} end
94
95 -- SYM(at,s) --> SYM ;; Create a SYM object to summarize symbols.
96 function SYM(at,s) return {at=at, of=s, n=0, has={}, bins={}} end
97
98 -- COLS(row) --> COLS ;; Create a COLS object from a list of column names.
99 function COLS(row, t,x,y,all)
100   x,y,all = {}, {}, {}
101   for n,s in ipairs(row) do
102     all[n] = s:match("[A-Z]") and NUM or SYM (n,s)
103     if not s:match"%S" then
104       t = s:find"%[+]" and y or x
105       t[l+1] = all[n] end end
106   return {all=all, x=x, y=y, names=row} end

```

binr.lua

Page 2/3

```

107
108 --## Methods
109
110 -- add(i,v) --> v ;; Update 'i' with 'v'.
111 local function add(i,v)
112   if v == "?" then return v end
113   i.n = i.n + 1
114   if i.has then i.has[v] = 1 + (i.has[v] or 0)
115   elseif i.mu then
116     local d = v - i.mu
117     i.mu = i.mu + d / i.n
118     i.m2 = i.m2 + d * (v - i.mu)
119     i.sd = i.n<2 and 0 or sqrt((max(0,i.m2)/(i.n - 1)))
120   elseif i.rows then
121     if not i.cols then i.cols = COLS(v) else
122       for col in pairs(i.cols.all) do add(col, v[col.at]) end
123     i.rows[l + #i.rows] = v end end
124   return v end
125
126 -- adds(src,it) --> it ;; Update 'it' with all items from 'src'.
127 function adds(src, it)
128   it = it or NUM()
129   if type(src) == "string"
130   then for row in csv(src) do add(it,row) end
131   else for _,row in pairs(src or {}) do add(it,row) end end
132   return it end
133
134 -- norm(i,v) --> n ;; Normalize 'v' 0..1 using 'i'.
135 local function norm(i,v)
136   return (i.has or v=="?") and v or floor((the.bins * norm(i,v)
137     1 / (1 + math.exp(-1.7 * (v - i.mu)/(i.sd + 1e-32))) end
138
139 -- bin(i,v) --> n ;; Normalize 'v' 0..bins-1 using 'i'.
140 local function bin(i,v)
141   return (i.has or v=="?") and v or floor((the.bins * norm(i,v) end
142
143 -- disty(i,row) --> n ;; Return distance of 'row' to best goal (using Y cols).
144 local function disty(i,row, d)
145   d=0; for _,y in pairs(i.cols.y) do d= d + (norm(y, row[y.at]) - y.best)^2 end
146   return sqrt(d/#i.cols.y) end
147
148 local function distys(i,rows)
149   return sort(rows or i.rows,
150     function(a,b) return disty(i,a) < disty(i,b) end) end
151
152 --## Think
153
154 local function scoreGet(data,row, b,n)
155   n = 0
156   for _,col in pairs(data.cols.x) do
157     b = bin(col, row[col.at])
158     if b ~= "?" then
159       n = n + col.bins[b] then
160         return n end
161   end
162
163 local function scorePut(data,row, b,y)
164   y = disty(data,row)
165   for _,col in pairs(data.cols.x) do
166     b = bin(col, row[col.at])
167     if b ~= "?" then
168       col.bins[b] = col.bins[b] or NUM(col.at, b)
169       add(col.bins[b], y) end end end
170
171 local function scoreGuess(data,m,n,rows, t)
172   t = {}
173   --print((m or 1),min(#rows, n or #rows))
174   for n = (m or 1),min(#rows, n or #rows) do
175     if n <= #rows then
176       t[l+1] = {scoreGet(data, rows[n]), rows[n]} end end
177   return sort(t, function(a,b) return a[1] < b[1] end) end
178
179 local function scoresSeen(data, t,m,eps)
180   t={} ; for m,row in pairs(data.rows) do t[l+1] = disty(data,row) end
181   t=sort(t)
182   m=#t//10
183   eps = (t[9*m] - t[m])/2.56 * 0.35
184   print(fmt("%.2f,%.2f,%.2f,%.2f,%.2f,%.2f",
185     t[m], t[3*m], t[5*m], t[7*m], t[9*m], eps))
186   return data,eps end
187
188 local function score(data,eps, labelled,rows,bestRow,besty,loves,best,y,lives,n)
189   labelled = clone(data)
190   besty = le32
191   lives = lives or the.lives
192   seen = {}
193   n=0;
194   for m,row in pairs(data.rows) do
195     if lives < 0 or n > the.Budget then break end
196     add(labelled, row)
197     scorePut(labelled, row)
198     seen[row]=row, n=n+1
199     if m % the.era==0 then
200       best = scoreGuess(labelled, m+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
205       else lives = lives - 1 end end end
206   --for _,x in pairs(labelled.cols.x) do
207   --print(" "); for k,b in pairs(x.bins) do print(x.of,k,b,mu,b,n) end end
208   return bestRow, besty, n end
209
210 --## Demos
211
212 local eggs={}
213
214 eggs["-h"] = function(_) print("m"..help.."m") end
215 eggs["-s"] = function(n) math.randomseed(n or the.seed); the.seed = n end
216 eggs["-d"] = function(_) print(o(the)) end
217 eggs["-shuffle"] = function(_) print(o(shuffle(10,20,30,40,50))) end
218
219 eggs["-csv"] = function(_, n)
220   n=1; for row in csv(the.file) do
221     if n % 25 == 0 then print(o(row)) end
222     n = n + 1 end end
223
224 eggs["-num"] = function(_, num)
225   num=NUM()
226   for _,l,1000 do add(num, box_muller(10,5)) end

```

binr.lua

Page 3/3

```

227   print(fmt("%.3f,%.3f", num.mu, num.sd)) end
228
229 eggs["-data"] = function(_)
230   for n,col in pairs(DATA(the.file).cols.x) do
231     print(n,o(col)) end end
232
233 eggs["-disty"] = function(_, data,num,t)
234   data,t = DATA(the.file), {}
235   for n,row in pairs(data.rows) do
236     if n % 25 == 0 then t[l+1] = disty(data,row) end end
237   print(o(sort(t))) end
238
239 eggs["-score"] = function(_, t,data,eps,u,y)
240   data,eps = scoresSeen(DATA(the.file))
241   t,u={},{}
242   for n = 1,the.repeats do
243     data.rows = shuffle(data.rows)
244     _,y,seen = score(data,eps)
245     t[n] = 100*y//1 end
246   print(o(sort(u)).."m"..o(sort(t))) end
247
248 eggs["-all"] = function(_, n)
249   n = the.seed
250   for k,_ in pairs(eggs) do
251     math.randomseed(n)
252     if k=="-all" then print("k=-----",k); eggs[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]
258     then funs[s](coerce(arg[i+1]))
259     else for k,_ in pairs(d) do
260       if k:sub(1,1)==s:sub(2) then d[k]=coerce(arg[i+1]) end end end end end
261   if arg[0]:find"binr.lua" then cli(the,eggs) end

```