

Dec 25, 21 12:03

rezo.lua

Page 1/8

```

1  --
2  --
3  --
4  --
5  --
6  --
7  --
8  --
9  -- (c)2021 Tim Menzies. Permission is hereby granted, free of charge,
10 -- to any person obtaining a copy of this software and associated
11 -- documentation files (the "Software"), to deal in the Software without
12 -- restriction, including without limitation the rights to use, copy,
13 -- modify, merge, publish, distribute, sublicense, and/or sell copies
14 -- of the Software, and to permit persons to whom the Software is
15 -- furnished to do so, subject to the following conditions:
16
17 -- The above copyright notice and this permission notice shall be included in all
18 -- copies or substantial portions of the Software.
19
20 -- THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
21 -- IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
22 -- FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
23 -- AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
24 -- LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
25 -- OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
26 -- SOFTWARE
27 local help = [[
28 lua rezo.lua [OPTIONS]
29 Tree learner (binary splits on numerics using Gaussian approximation)
30 (c)2021 Tim Menzies <timm@ieee.org> MIT license.
31
32 OPTIONS:
33 -best      X   Best examples are in 1..best*size(all)      = .2
34 -debug     X   run one test, show stackdumps on fail      = the
35 -epsilon   X   ignore differences under epsilon*stdev      = .35
36 -Far       X   How far to look for remove items           = .9
37 -file      X   Where to read data                          = ../data/auto93.csv
38 -h         X   Show help                                    = false
39 -little    X   size of subset of a list                    = 256
40 -p         X   distance calc coefficient                   = 2
41 -seed      X   Random number seed;                        = 10019
42 -Stop      X   Create subtrees while at least 2*stop eggs  = 4
43 -Tiny      X   Min range size = size(eggs)^tiny           = .5
44 -todo      X   Pass/fail tests to run at start time       = the
45               If "X=all", then run all.
46               If "X=s" then list all.
47
48 Data read from "-file" is a csv file whose first row contains column
49 names (and the other row contain data. If a name contains ":",
50 that column will get ignored. Otherwise, names starting with upper
51 case denote numerics (and the other columns are symbolic). Names
52 containing "!" are class columns and names containing "+" or "-"
53 are goals to be maximized or minimized. --]] --[[
54
55 Internally, columns names are read by a COLS object where numeric,
56 symbolic, and ignored columns generate NUM, SYM, and SKIP instances
57 (respectively). After row1, all the other rows are examples ('EG')
58 which are stored in a SAMPLE. As each example is added to a sample,
59 they are summarized in the COLS' objects.
60
61 Note that SAMPLEs can be created from disk data, or at runtimes from
62 lists of examples (see SAMPLE:clone()). --]]
63
64 local b4={}; for k,_ in pairs(_ENV) do b4[k]=k end
65 local THE = {} -- The THE global stores the global config for this software.
66 -- any line of help text starting with " -" has flag,default as first,last word
67 help:gsub("\n-([^\s%+])\n]*%s%([^\s%+])",
68 function(flag,x)
69   for n,word in ipairs(arg) do -- check for any updated to "flag" on command line
70     -- use any command line "word" that matches the start of "flag"
71     if flag:match("^"..word:sub(2)..".*") then
72       -- command line "word"s for booleans flip the default value
73       x=(x=="false" and "true") or (x=="true" and "false") or arg[n+1] end end
74   if x=="true" then x=true elseif x=="false" then x=false else x=tonumber(x) or x end
75   THE[flag] = x end)
76
77 THE.seed = THE.seed or 10019
78 if THE.h then return print(help) end

```

Dec 25, 21 12:03

rezo.lua

Page 2/8

```

79 --
80 --
81 --
82 --
83 -- meta
84 local same
85 function same(x,...) return x end
86
87 -- sorting
88 local push,sort,ones
89 function push(t,x) table.insert(t,x); return x end
90 function sort(t,f) table.sort(t,f); return t end
91 function ones(a,b) return a[1] < b[1] end
92
93 -- tables
94 local copy,keys,map,sum
95 function copy(t, u) u={};for k,v in pairs(t) do u[k]=v end; return u end
96 function keys(t, u) u={};for k,_ in pairs(t) do u[1+#u]=k end; return sort(u) end
97 function map(t,f, u) u={};for k,v in pairs(t) do u[1+#u]=f(k,v) end; return u end
98 function sum(t,f, n) n=0;for _,v in pairs(t) do n=n+(f or same)(v) end;return n end
99
100 -- printing utils
101 local hue,shout,out,say,fmt
102 fmt = string.format
103 function say(...) print(string.format(...)) end
104 function hue(n,s) return string.format("%27[1m%27[%sm%27[0m",n,s) end
105 function shout(x) print(out(x)) end
106 function out(t, u,key,val) -- convert nested tables to a string
107   function key(_,k) return string.format("%s %s", k, out(t[k])) end
108   function val(_,v) return out(v) end
109   if type(t) ~= "table" then return tostring(t) end
110   u = #t>0 and map(t, val) or map(keys(t), key)
111   return {"..table.concat(u, " ").."}" end
112
113 -- reading from file
114 local coerce,csv
115 function coerce(x)
116   if x=="true" then return true elseif x=="false" then return false end
117   return tonumber(x) or x end
118
119 function csv(file, x)
120   file = io.input(file)
121   return function() t,tmp)
122     x = io.read()
123     if x then
124       t={};for y in x:gsub("[\n]*",""):gmatch("[^\n]+") do push(t,coerce(y)) end
125       if #t>0 then return t end
126     else io.close(file) end end end
127
128 -- maths
129 local log,sqrt,rnd,rnds
130 log = math.log
131 sqrt= math.sqrt
132 function rnd(x,d, n) n=10^(d or 0); return math.floor(x*n+0.5) / n end
133 function rnds(t,d) return map(t, function(_,x) return rnd(x,d or 2) end) end
134
135 -- random stuff (LUA's built-in randoms give different results on different platfors)
136 local randi,rand,any,some
137 function randi(lo,hi) return math.floor(0.5 + rand(lo,hi)) end
138 function rand(lo,hi)
139   lo, hi = lo or 0, hi or 1
140   THE.seed = (16807 * THE.seed) % 2147483647
141   return lo + (hi-lo) * THE.seed / 2147483647 end
142
143 function any(t) return t[randi(1,#t)] end
144 function some(t,n, u)
145   if n >= #t then return copy(t) end
146   u={}; for i=1,n do push(u,any(t)) end; return u end
147
148 -- objects
149 local ako,has,obj
150 ako= getmetatable
151 function has(mt,x) return setmetatable(x,mt) end
152 function obj(s, o,new)
153   o = {_is=s, __tostring=lib.out}
154   o.__index = o
155   return setmetatable(o, {_call=function(_,...) return o.new(...) end}) end

```

Dec 25, 21 12:03

rezon.lua

Page 3/8

```

156 --
157 -- NUM
158 --
159 --
160 local NUM=obj"NUM"
161 function NUM.new(inits,at,txt, self)
162     self = has(NUM,{n=0, at=at or 0, txt=txt or "",
163         w=(txt or ""):find"- " and -1 or 1,
164         mu=0, m2=0, lo=math.huge, hi=-math.huge})
165     for _,x in pairs(inits or {}) do self:add(x) end
166     return self end
167
168 -- summarizing
169 function NUM:mid() return self.mu end
170 function NUM:spread() return (self.m2/(self.n-1))^0.5 end
171
172 -- updating
173 function NUM:add(x, d)
174     if x ~= "?" then
175         self.n=self.n+1
176         d=x-self.mu
177         self.mu= self.mu+d/self.n
178         self.m2= self.m2+d*(x-self.mu)
179         self.lo = math.min(x, self.lo)
180         self.hi = math.max(x, self.hi) end
181     return x end
182
183 -- querying
184 function NUM:norm(x)
185     local lo,hi = self.lo,self.hi
186     return math.abs(hi - lo) < 1E-9 and 0 or (x-lo)/(hi-lo) end
187
188 function NUM:dist(x,y)
189     if x=="?" then y=self:norm(y); x=y>0.5 and 0 or 1
190     elseif y=="?" then x=self:norm(x); y=x>0.5 and 0 or 1
191     else x, y = self:norm(x), self:norm(y) end
192     return (x-y) end
193
194 -- discretization
195 function NUM:splits(other)
196     function cuts(x,s,at) return {
197         {val=x, at=at, txt=fmt("%s<=$s",s,x), when=function(z) return z<=x end},
198         {val=x, at=at, txt=fmt("%s>$s",s,x), when=function(z) return z>x end}}
199     end
200     local i, j, e, a, b, c, x1, x2 = self, other, 2.71828
201     a = 1/(2*sd(i)^2) - 1/(2*sd(j)^2)
202     b = j.mu/(sd(j)^2) - i.mu/(sd(i)^2)
203     c = i.mu^2/(2*sd(i)^2) - j.mu^2/(2*sd(j)^2) - mat
204     x1 = (-b - sqrt(b*b - 4*a*c))/2*a
205     x2 = (-b + sqrt(b*b - 4*a*c))/2*a
206     if i.mu<=x1 and x1<=j.mu
207     then return cuts(x1,self.txt,self.at)
208     else return cuts(x2,self.txt,self.at) end end
209
210 --
211 -- SYM
212 --
213 local SYM=obj"SYM"
214 function SYM.new(inits,at,txt,sample, self)
215     self= has(SYM,{n=0, at=at or 0, txt=txt or "", sample=sample,
216         seen={}, mode=nil, most=0})
217     for _,x in pairs(inits or {}) do self:add(x) end
218     return self end
219
220 -- Summarizing
221 function SYM:mid() return self.mode end
222 function SYM:spread()
223     return sum(self.seen, function(n) return -n/self.n*log(n/self.n,2) end) end
224
225 -- update
226 function SYM:add(x)
227     self.seen[x] = (self.seen[x] or 0) + 1
228     if self.seen[x] > self.most then self.mode, self.most = x, self.seen[x] end
229     return x end
230
231 -- querying
232 function SYM:dist(x,y) return x==y and 0 or 1 end
233
234 -- discretization
235 function SYM:splits(other)
236     function cut(_,x) return
237         {val=x, at=self.at, txt=fmt("%s==$s",self.txt,x),
238         when = function(z) return z==x end} end
239     local out={}
240     for k,_ in pairs(self.seen) do push(out,k) end
241     for k,_ in pairs(other.seen) do push(out,k) end
242     return map(sort(out),cut) end
243

```

Dec 25, 21 12:03

rezon.lua

Page 4/8

```

244 --
245 -- SKIP
246 --
247 --
248 -- Columns for values we want to ignore.
249 local SKIP=obj"SKIP"
250 function SKIP.new(inits,at,txt)
251     return has(SKIP,{n=0, at=at or 0, txt=txt or ""}) end
252
253 function SKIP:mid() return "?" end
254 function SKIP:spread() return 0 end
255 function SKIP:add(x) return x end
256 function SKIP:splits(_) return {} end
257
258 -- EG
259 --
260 -- One example
261 local EG=obj"EG"
262
263 function EG.new(cells) self.cells = cells end
264
265 -- Sumamrizing
266 function EG:mid(cols) return map(cols, function(_,c) return c:mid() end) end
267 function EG:spread(cols) return map(cols, function(_,c) return c:spread() end) end
268
269 -- Queries
270 function EG:dist(other,cols, a,b,d,n,inc)
271     d,n = 0,0
272     for _,col in pairs(cols) do
273         a,b = self.cells[col.at], other.cells[col.at]
274         inc = a=="?" and b=="?" and 1 or col:dist(a,b)
275         d = d + inc^THE.p
276         n = n + 1 end
277     return (d/n)^(1/THE.p) end
278
279 -- Sorting
280 function EG:better(other,cols, e,n,a,b,s1,s2)
281     n,s1,s2,e = #cols, 0, 0, 2.71828
282     for _,num in pairs(cols) do
283         a = num:norm(self.cells[ num.at])
284         b = num:norm(other.cells[num.at])
285         s1 = s1 - e^(num.w * (a-b)/n)
286         s2 = s2 - e^(num.w * (b-a)/n) end
287     return s1/n < s2/n end
288
289 --
290 -- COLS
291 --
292 -- Convert column headers into NUMs and SYMs, etc.
293 local COLS=obj"COLS"
294 function COLS.new(names, self, new,what)
295     self = has(COLS, {names=names, xs={}, all={}, ys={}})
296     for n,x in pairs(names) do
297         new = (x:find"." and SKIP or x:match"^[A-Z]" and NUM or SYM) ({},n,x)
298         push(self.all, new)
299         if not x:find"." then
300             if x:find"!" then self.klass = new
301             what = (x:find"-" or x:find"+") and self.ys or self.xs
302             push(what, new) end end end
303     return self end
304
305 -- Updates
306 function COLS:add(eg)
307     return map(eg, function(n,x) self.all[n]:add(x); return x end) end
308
309

```

Dec 25, 21 12:03

rezon.lua

Page 5/8

```

310 --
311 -- SAMPLE
312 --
313 --
314 -- SAMPLEs hold many examples
315 local SAMPLE=obj"SAMPLE"
316 function SAMPLE:new(inits, self)
317   self = has(SAMPLE, {cols=nil, eggs={}})
318   if type(inits)=="string" then for eg in csv(inits) do self:add(eg) end end
319   if type(inits)=="table" then for eg in pairs(inits) do self:add(eg) end end
320   return self end
321
322 -- Create a new sample with the same structure as this one
323 function SAMPLE:clone(inits, out)
324   out = SAMPLE:new(self.cols.names)
325   for _,eg in pairs(inits or {}) do out:add(eg) end
326   return out end
327
328 -- Updates
329 function SAMPLE:add(eg)
330   eg = eg.cells and eg.cells or eg
331   if self.cols
332   then push(self.egs,eg); self.cols:add(eg)
333   else self.cols = COLS(eg) end end
334
335 -- Distance queries
336 function SAMPLE:neighbors(eg1,egs,cols)
337   local dist_eg2 = function(_,eg2) return {eg1:dist(eg2,cols or self.xls),eg2} end
338   return sort(map(egs or self.egs,dist_eg2),firsts) end
339
340 function SAMPLE:distance_farExample(eg1,egs,cols, tmp)
341   tmp = self:neighbors(eg1, egs, cols)
342   return table.unpack(tmp[#tmp*self.Far//1]) end
343
344 -- Discretization
345 function SAMPLE:twain(egs,cols)
346   local egs, north, south, a,b,c, lo,hi
347   egs = many(egs or self.egs, self.little)
348   _,north = self:distance_farExample(any(self.egs), egs, cols)
349   c,south = self:distance_farExample(north, egs, cols)
350   for _,eg in pairs(self.egs) do
351     a = eg:dist(north, cols)
352     b = eg:dist(south, cols)
353     eg.x = (a^2 + c^2 - b^2)/(2*c) end
354   lo, ho = self:clone(), self:clone()
355   for n,eg in pairs(sort(self.egs, function(a,b) return a.x < b.x end)) do
356     if n < .5*#eg then lo:add(eg) else hi:add(eg) end end
357   return lo, hi end
358
359 function SAMPLE:mid(cols)
360   return map(cols or self.cols.all,function(_,col) return col:mid() end) end
361 function SAMPLE:spread(cols)
362   return map(cols or self.cols.all,function(_,col) return col:spread() end) end
363

```

Dec 25, 21 12:03

rezon.lua

Page 6/8

```

364 --
365 -- SAMPLE TREE
366 --
367 --
368 -- need to sort first
369
370 -- how to score
371 function SAMPLE:splits(other,both, cuts,unplaced,place,score)
372   function guess(todos,cuts)
373     for _,todo in pairs(todos) do
374       local f=function(_,cut)
375         return {Row(cut.has:mid()):dist(todo, both.cols.xls),cut} end
376       sort(map(cuts,f),firsts)[1][2].has:add(todo) end
377     return cuts end
378   function divide(cuts, todos,placed)
379     todos = {}
380     for _,eg in pairs(both.egs) do
381       placed = false
382       for _,cut in pairs(cuts) do
383         if cut.what(eg.cells[cut.at])
384         then cut.has = cut.has or self:clone()
385             cut.has:add(eg)
386             placed = true
387             break end end
388       if not placed then push(todos, eg) end end
389     return guess(todos,cuts) end
390   function score(cut, m,n)
391     m,n = #cut.has.egs,both.egs; return -m/n*log(m/n,2) end
392   local best, cutsx, tmp = math.huge
393   for pos,col in pairs(both.cols.xls) do
394     cutsx = col:splits(other.cols.xls[pos])
395     tmp = sum(divide(cutsx),score)
396     if tmp < best then best,cuts = tmp,cutsx end end
397   return cuts end
398
399 function SAMPLE:tree(top)
400   top = top or self
401   one,two = self:twain(self.egs, top.cols.xls)
402   for _,cut in pairs(one:splits(two,self)) do
403     if cut.stats.n > (#top.egs)^THE.Tiny then
404       cut.sub= cut.has:tree(top) end end end
405
406 function SAMPLE:show(tree)
407   local vals=function(a,b) return a.val < b.val end
408   local function show1(tree,pre)
409     if #tree.kids==0 then io.write(fmt("==> %s[%s]",tree.mode, tree.n)) end
410     for _,kid in pairs(sort(tree.kids,vals)) do
411       io.write("\n"..fmt("%s%s",pre, showDiv(i, kid.at, kid.val)))
412       show1(kid.sub, pre.."|. ") end
413   end -----
414   show1(tree,""); print("") end
415

```

Dec 25, 21 12:03

rezon.lua

Page 7/8

```

416 -----
417 --
418 --  E X A M P L E S
419 --
420 --
421 local go={}
422 function go.ls()
423   print("\nlua ".arg[0].." -todo ACTION\n\nACTIONS:")
424   for _,k in pairs(keys(go)) do print(" -todo",k) end end
425 function go.the() shout(THE) end
426 function go.bad( s) assert(false) end
427 function go.ordered( s,n)
428   s = ordered(slurp())
429   n = #s.egs
430   shout(s.heads)
431   for i=1,15 do shout(s.egs[i].cells) end
432   print("#")
433   for i=n,n-15,-1 do shout(s.egs[i].cells) end
434 end
435
436 function go.num( cut,min)
437   local xy, xnum, ynum = {}, NUM(), NUM()
438   for i=1,400 do push(xy, {add(xnum,i), add(ynum, rand()^3 )}) end
439   for i=401,500 do push(xy, {add(xnum,i), add(ynum, rand()^.25)}) end
440   cut,min= minXpect(xy, ynum, .35*sd(xnum), (#xy)^the.Tiny)
441   shout{cut=cut, min=min} end
442
443 function go.symcuts( s,xpect,cuts)
444   s=ordered(slurp())
445   print(out(s.xs),out(s.ys))
446   xpect,cuts = symcuts(7,s.egs, "origin")
447   for _,cut in pairs(cuts) do print(xpect, out(cut)) end end
448
449 function go.numcuts( s,xpect,cuts)
450   s=ordered(slurp())
451   xpect,cuts = numcuts(s,2,s.egs,"Dsplacement")
452   if xpect then
453     for _,cut in pairs(cuts) do print(xpect, out(cut)) end end end
454
455 function go.atcuts(s,cuts,at,ynum)
456   s=ordered(slurp())
457   ynum=NUM(a); map(s.egs,function(_,eg) add(ynum, eg.klass) end)
458   at,cuts = at_cuts(s,egs,sd(ynum)*THE.epsilon, (#s.egs)^THE.Tiny)
459   for _,cut in pairs(cuts) do print(at, out(cut)) end end
460

```

Dec 25, 21 12:03

rezon.lua

Page 8/8

```

461 --
462 --  S T A R T - U P
463 --
464 --
465 local fails, defaults = 0, copy(THE)
466 go[ THE.debug ]()
467 local todos = THE.todo == "all" and keys(go) or {THE.todo}
468 for _,todo in pairs(todos) do
469   THE = copy(defaults)
470   local ok,msg = pcall( go[todo] )
471   if ok then print(hue(32,"PASS")..todo)
472     else print(hue(31,"FAIL")..todo,msg)
473     fails=fails+1 end end
474
475 for k,v in pairs(_ENV) do if not b4[k] then print("?:",k,type(v)) end end
476 os.exit(fails)

```