

Dec 04, 21 19:57

I5.lua

Page 3/5

```

301 end
302 if not self.names
303 then self.names = eg
304   map(eg, function(col,x) name(col,x) end)
305 else push(self.egs, eg)
306   map(eg, function(col,x) datum(col,x) end) end
307 return self end
308
309 -- bins his
310 -- bins sorts
311
312 function Sample:tree(min, node,min,sub)
313   node = (node=self, kids={})
314   min = min or (#self.egs)^the.small
315   if #self.egs >= 2*min then
316     here
317     for _,span in pairs(splits.best(sample)) do
318       sub = self:clone()
319       for _,at in pairs(span.has) do sub:add(self.egs[at]) end
320       push(node.kids, span)
321       span.has = sub:tree(min) end end
322   return node end
323
324 -- at node
325 function Sample:where(tree,eg, max,x,default)
326   if #kid.has==0 then return tree end
327   max = 0
328   for _,kid in pairs(tree.node) do
329     if #kid.has > max then default,max = kid,#kid.has end
330     x = eg[kid.col]
331     if x == "?" then
332       if x <= kid.hi and x >= kid.lo then
333         return self:where(kid.has.eg) end end end
334   return self:where(default, eg) end
335
336 -- ordered object
337 -- per sd add sort here. mergabe
338
339 -----
340 -- geometry tricks
341 -- y column rankings
342 local dist, better,betters
343 function dist(eg1,eg2,sample, a,b,d,n,inc,dist1)
344   function dist1(num,a,b)
345     if not num then return a==b and 0 or 1 end
346     if a=="?" then b=norm(b, num.lo, num.hi); a = b>.5 and 0 or 1
347     elseif b=="?" then a=norm(a, num.lo, num.hi); b = a>.5 and 0 or 1
348     else a,b = norm(a, num.lo, num.hi), norm(b, num.lo, num.hi)
349     end
350     return abs(a-b)
351   end
352   d,n=0,0
353   for col,_ in pairs(sample.xls) do
354     a,b = eg1[col], eg2[col]
355     inc = a=="?" and b=="?" and 1 or dist1(sample.nums[col],a,b)
356     d = d + inc^the.p
357     n = n + 1 end
358   return (d/n)^(1/the.p) end
359
360 function better(egs,sample)
361   return sort(egs,function(a,b) return better(a,b,sample) end) end
362
363 function better(eg1,eg2,sample, e,n,a,b,s1,s2)
364   n,s1,s2,e = #sample.xls, 0, 0, 2.71828
365   for _,num in pairs(sample.xls) do
366     a = norm(eg1[num.col], num.lo, num.hi)
367     b = norm(eg2[num.col], num.lo, num.hi)
368     s1 = s1 - e^(num.w * (a-b)/n)
369     s2 = s2 - e^(num.w * (b-a)/n) end
370   return s1/n < s2/n end
371
372 -----
373 -- sample sample sorting
374 local hints={}
375 function hints.default(eg) return eg end
376
377 function hints.sort(sample,score, test,train,evals)
378   sample = Sample.new(the.file)
379   train,test = {}, {}
380   for i,eg in pairs(shuffle(sample.egs)) do
381     push(i<= the.train*#sample.egs and train or test, eg) end
382   evals,train = hints.recurse(sample, train,0,
383     score or hints.default, {}, ({train}^the.small)
384   return evals,sample:clone(train), sample:clone(test) end
385
386 function hints.recurse(sample, egs, evals, scorefun, out, small, worker)
387   if #egs < small then
388     for i=1, #egs do push(out, pop(egs)) end
389     return evals,out
390   end
391   local scored = {}
392   function worker(eg) return hints.locate(scoreds,eg,sample) end
393   for j=1,the.hints do push(scoreds, scorefun(pop(egs))) end
394   scored = better(scoreds, sample)
395   egs = lap(sort(lap(egs, worker),firsts),second)
396   for i=1,#egs/2 do push(out, pop(egs)) end
397   return hints.recurse(sample, egs,evals, scorefun, out, small)
398 end
399
400 function hints.locate(scoreds,eg,sample, closest,rank,tmp)
401   closest,rank,tmp = 1E32, 1E32, nil
402   for rank0, scored in pairs(scoreds) do
403     tmp = dist(eg, scored, sample)
404     if tmp < closest then closest,rank = tmp,rank0 end end
405   return (rank+closest/10^6, eg) end
406
407 -----
408
409 local eg,fail,example={},0
410 function example(k, f,ok,msg)
411   f = eg[k]; assert(f,"unknown action"..k)
412   the=eli(options)
413   if the.wild then return f() end
414   ok,msg = pcall(f)
415   if ok then print(green("PASS"),k)
416   else print(red("FAIL"), k,msg); fail=fail+1 end end
417
418 function eg.norm()
419   assert(norm(5,0,10)==.5,"small") end
420
421 function eg.map()
422   assert(3==map({1,2},function(_,x) return x+1 end)[2]) end
423
424 function eg.tables()
425   assert(20==sort(shuffle({{10,20},{30,40},{40,50}}),firsts)[1][2]) end
426
427 function eg.csv( n,z)
428   n=0
429   for eg in csv(the.file) do n=n+1; z=eg end
430   assert(n==399 and z[#z]==50) end
431
432 function eg.nums( n)
433   n=Nums(10,20,30,40,50,10,20,30,40,50,10,20,30,40,50)
434   assert(15.625 == n:sd()) end
435
436 function eg.nums( n1,n2,n3,n4)
437   n1=Nums(10,20,30,40,50,10,20,30,40,50,10,20,30,40,50)
438   n2=Nums(10,20,30,40,50,10,20,30,40,50,10,20,30,40,50)
439   assert(n1:mergeable(n2)==nil)
440   n3=Nums(10,20,30,40,50,10,20,30,40,50,10,20,30,40,50)
441   n4=Nums(100,200,300,400,500,100,200,300,400,500,100,200,300,400,500)
442   assert(n3:mergeable(n4)==nil) end
443
444 function eg.sample( s,tmp,d1,d2)
445   s=Sample(the.file)
446   assert(s.xls[4].lo==1613)
447   tmp = sort(map(shuffle(s.egs),
448     function(_,eg2) return (dist(eg2,s.egs[1],s), eg2) end),
449     firsts)
450   d1=dist(tmp[1][2], tmp[10][2], s)

```

Dec 04, 21 19:57

I5.lua

Page 4/5

```

451   d2=dist(tmp[1][2], tmp[#tmp][2], s)
452   assert(d1<10<d2)
453 end
454
455 function eg.hints( s,_,_,evals)
456   s=Sample(the.file)
457   sort1= better(s.egs,s)
458   for _,eg in pairs(sort1) do shout(lap(s.xls, function(col) return eg[col.col] end )) end
459   -- assert(s.xls[4].lo==1613)
460   -- evals, train,_,_ = hints.sort(s)
461   -- print("=",evals)
462   -- for m,eg in pairs(sort1) do
463     -- n = bchop(sort1, eg,function(a,b) return better(a,b,s) end)
464     -- print(m,n) end
465   end
466
467 if the.todo=="all" then lap(keys(eg),example) else example(the.todo) end
468
469 -----
470 -- trick for checking for rogues.
471 for k,v in pairs(_ENV) do if not b4[k] then print("?rogue:",k,type(v)) end end
472 os.exit(fail)
473
474
475

```

Dec 04, 21 19:57

l5.lua

Page 5/5

```
476 --[[  
477 needs stats on samples  
478  
479 teaching:  
480 - sample is v.useful  
481  
482  
483 --]]
```