January 03

MIDDLE – code

Opt.R

```
blob <- rawToChar(serialize(connection=NULL, uniRegModelRaw, ascii=TRUE))
```

serialize

It lets you take an object or group of objects, put them on a disk or send them through a wire or wireless transport mechanism, then later, perhaps on another computer, reverse the process: resurrect the original object(s). The basic mechanisms are to flatten object(s) into a one-dimensional stream of bits, and to turn that stream of bits back into the original object(s).

Like the Transporter on Star Trek, it's all about taking something complicated and turning it into a flat sequence of 1s and 0s, then taking that sequence of 1s and 0s (possibly at another place, possibly at another time) and reconstructing the original complicated "something."

object

In computer science, an object is a location in memory having a value and possibly referenced by an identifier. An object can be a variable, a data structure, or a function. In class-based object-oriented programming paradigm, "object" refers to a particular instance of a class where the object can be a combination of variables, functions, and data structures. In relational database management, an object can be a table or column, or an association between data and a database entity (such as relating a person's age to a specific person).

data structure

In computer science, a data structure is a particular way of organizing data in a computer so that it can be used efficiently. Data structures provide a means to manage large amounts of data efficiently for uses such as large databases and internet indexing services. Usually, efficient data structures are key to designing efficient algorithms. (linear data structures, Trees, Hashes, Graph, etc).

R function: serialize {base}

The function serialize serializes object to the specified connection. If connection is NULL then object is serialized to a raw vector, which is returned as the result of serialize.

Example:

```
> x <- serialize(list(1,2,3), NULL)
> x
[1] 58 0a 00 00 00 02 00 03 01 01 00 02 03 00 00 00 00 13 00 00 00 03 00 00 00 0e 00 00 00
[30] 01 3f f0 00 00 00 00 00 00 00 00 00 0e 00 00 00 01 40 00 00 00 00 00 00 00 00 00 00 0e
```

```
[59] 00 00 00 01 40 08 00 00 00 00 00 00
> unserialize(x)
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] 3
```

R function: rawToChar {base}

Conversion and manipulation of objects of type "raw". rawToChar converts raw bytes either to a single character string or a character vector of single bytes (with "" for 0).

Example:

```
> x <- "A test string"
> y <- charToRaw(x)
> y
 [1] 41 20 74 65 73 74 20 73 74 72 69 6e 67
> is.vector(y)
[1] TRUE
> rawToChar(y)
[1] "A test string"
> rawToChar(y, multiple = TRUE)
 [1] "A" " " "t" "e" "s" "t" " " "s" "t" "r" "i" "n" "g"
> xx <- c(y,  charToRaw("&"), charToRaw("more"))
> xx
 [1] 41 20 74 65 73 74 20 73 74 72 69 6e 67 26 6d 6f 72 65
> rawToChar(xx)
[1] "A test string&more"
```

```
r <- POST(paste0(apiurl, "/model/test"),
          content_type_json(),
          body=toJSON(list(model=blob, minRows=partitions), digits=8))
```

R function: paste0{base}

paste0(..., collapse) is equivalent to paste(..., sep = "", collapse), slightly more efficiently.

POST (HTTP)

In computing, POST is one of many request methods supported by the HTTP protocol used by the World Wide Web. The World Wide Web and HTTP are based on a number of request methods or 'verbs', including POST and GET as well as PUT, DELETE, and several others. The POST request method is designed to request that a web server accept the data enclosed in the request message's body for storage. It is often used when uploading a file or submitting a completed web form. In contrast, the HTTP GET request method is designed to retrieve information from the server.

R function: POST{httr}

POST file to a server.

PUT (HTTP)

Use PUT when you can update a resource completely through a specific resource. For instance, if you know that an article resides at http://example.org/article/1234, you can PUT a new resource representation of this article directly through a PUT on this URL.

If you do not know the actual resource location, for instance, when you add a new article, but do not have any idea where to store it, you can POST it to an URL, and let the server decide the actual URL. ??? need more explanation

GET (HTTP)

??? need more explanation

```
uniRegModelRaw <- mxModel(...
    mxComputeGradientDescent(),
    mxFitFunctionR(fitfun = function(model, state) {
    par <- omxGetParameters(model)
    r <- PUT(paste0(apiurl, "/model/test/param"),
             body=toJSON(list(param=par), digits=8),
             content_type_json())
    #content(r)

    fit <- NULL
    while (1) {
      Sys.sleep(1)
      r <- GET(paste0(apiurl, "/model/test/fit"))
      if (!is.null(content(r)$nan)) {
        fit <- NA
        break
      }
      fit <- content(r)$fit
      if (length(fit)) break
    }
    print(fit)
    return(fit)
  }))
```

What is the use of mxComputeGradient() inside a MX model?

All the parameters are set to default values.

```
mxComputeGradientDescent(freeSet = NA_character_, ..., engine = NULL,
  fitfunction = "fitfunction", verbose = 0L, tolerance = NA_real_,
  useGradient = NULL, warmStart = NULL)
```

fitfunction inside a MX model?

Functional programming (quora question)

What is functional programming?

http://www.quora.com/What-is-functional-programming

http://blog.8thlight.com/uncle-bob/2012/12/22/FPBE1-Whats-it-all-about.html (very insightful)

Functional programming is a paradigm which concentrates on computing results rather than on performing actions. That is, when you call a function, the only significant effect that the function has is usually to compute a value and return it. Of course, behind the scenes the function is using CPU time, allocating and writing memory, but from the programmer's point of view, the primary effect is the return value. Objects in a functional programming language are often immutable (a.k.a. const or final); instead of changing an object, you allocate a new object which looks like the old one except for the change. Compare this with an imperative programming language like Java, where progress is made by changing objects' fields, inserting them into sets, etc.

In a pure functional language, like Haskell, most functions are guaranteed by the type system not to perform any other actions. In an impure functional language, like ML, a function may have other side effects, such as querying a database or server, generating random numbers, reading or writing the disk, etc.

Example: Lisp, Haskell, Scala, F# and Clojure

In contrast, imperative programming changes state with commands in the source language, the most simple example being assignment. Imperative programming does have functions, not in the mathematical sense, but in the sense of subroutines. They can have side effects that may change the value of program state. Functions without return values therefore make sense. Because of this, they lack referential transparency, i.e. the same language expression can result in different values at different times depending on the state of the executing program.

Command: df

df displays the amount of disk space available on the file system containing each file name argument.

```
yang@Yang:~/OpenMx2/openmx$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda4        25G   15G  8.3G  65% /
udev             10M     0   10M   0% /dev
tmpfs           1.2G  1.1M  1.2G   1% /run
tmpfs           2.9G   50M  2.9G   2% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           2.9G     0  2.9G   0% /sys/fs/cgroup
tmpfs           588M   44K  588M   1% /run/user/1000
/dev/sdb2       201G 1004M  200G   1% /media/yang/Data
/dev/sda2        71G   63G  7.9G  89% /media/yang/Windows7_OS
```

Daily task

Oral english

done.

Recitation paragraph

Oil workers to pay for near 50% price fall (Financial Times)

Oil prices are on course for their largest annual slide since 2008, capping another dire year for commodities, as crude fell again yesterday to hover at close to half its level of six months ago. Brent crude's 49 percent plummet since June – alongside a near halving of iron ore prices and sharp drops in coal and copper – has helped drag the Bloomberg Commodity index down 15.6 percent in 2014 to a five-year low. While the international benchmark's price plunge could prove a mojor boon for the global economy, it has thrown big oil exporters such as Russia and Venezuela into disarray and forced oil companies to re-examine their investment plans and look for ways to reduce costs.

油价将录得自2008年以来最大幅度的年度下滑，为大宗商品经历的又一个惨淡年份画上句号。原油价格昨日再度下跌，六个月前的价位几乎被腰斩。布伦特原油自6月以来暴跌49%, 铁矿石价格也接近减半，煤炭和铜的价格大幅下滑，这一切促使彭博大宗商品指数2014年下降15.6%，跌至五年新低。虽然国际基准油价暴跌可能被证明是全球经济的一大福音，但这一局面使俄罗斯和委内瑞拉等大型石油出口国陷入混乱，也迫使石油公司重新审视各自的投资计划，并想方设法降低成本。

In a sign of how oil majors are scrambling to make savings, BP, Royal Dutch Shell, Total and Chevron have all ordered sharp cuts in the rates paid to contractors on North Sea projects. The groups are cutting up to 15 percent of the pay of thousands of self-employed oil and gas workers in the region. US-based Chevron told employment agencies that it would reduce rates from January 11 to "better align with industry benchmarks and manage cost pressures", while BP has decided to cut the wages of 450 workers by up to 15 percent from the new year. A senior oil executive said that the drop in crude was "an opportunity" to lower exploration costs globally by renegotiating contracts with providers such as drilling companies. "Exploration is the easiest activity to reduce, " he said. Industry analysts believe that the North Sea will be targeted as energy groups curb spending on development. Although investment in new projects recently hit record levels, leading to several years of inflation-beating pay increases, UK oil and gas output has declined and exploration has been poor. Soaring costs and a complicated tax system have made smaller, more mature fields a riskier bet. 突显大型石油企业争先恐后消减成本的一个迹象是，英国石油（BP），荷兰皇家壳牌，道达尔和雪佛龙均已下令大幅降低支付给北海项目合同工的薪酬标准。这些集团正将该地区数千自雇石油和天然气工人的薪酬削减至多15%。总部位于美国的雪佛龙通知职业介绍机构，从1月1号起将降低支付标准，以更好与行业基准保持一致，更好地管理成本压力。BP已决定从新年起将450名工人的工资削减至多15%。一位石油行业高管表示，原油价格下跌是一个机遇，可通过也提供商（如钻井公司）重新谈判合约，在全球范围降低勘探成本。"勘探是最容易削减成本的活动"，他说。行业分析师相信，随着能源集团削减开发支出，北海将首当其冲。虽然近年对新项目的投资达到创纪录水平，导致连续数年高于通胀的薪酬上调，但英国的石油和天然气产量有所下降，勘探成果也不好。飙升的成本和复杂的税收体系加大了较小较成熟的油气田的风险。