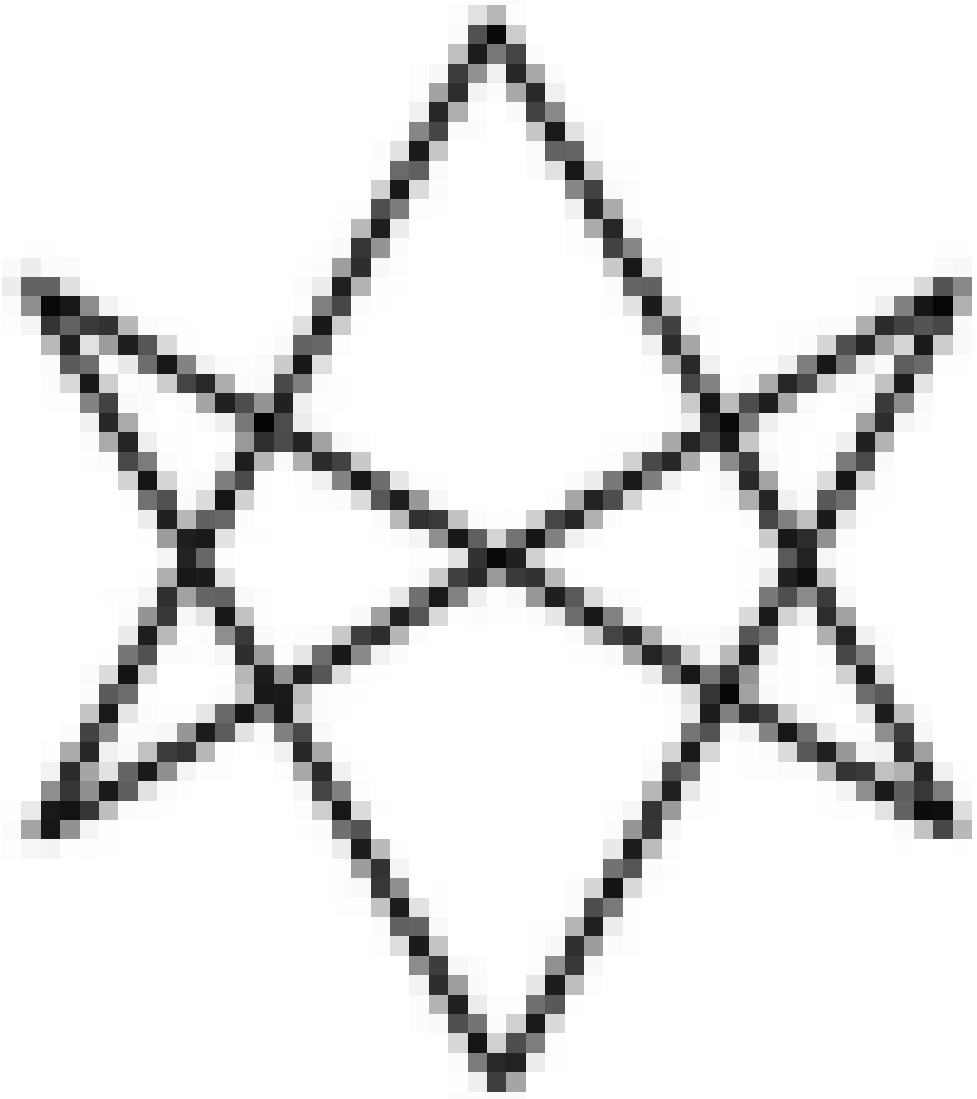
Guide to the pbdML Package

August 11, 2020

Drew Schmidt
wrathematics@gmail.com



Version 0.1-1

Acknowledgements and Disclaimer

Work for the **remoter** package is supported in part by the project *Harnessing Scalable Libraries for Statistical Computing on Modern Architectures and Bringing Statistics to Large Scale Computing* funded by the National Science Foundation Division of Mathematical Sciences under Grant No. 1418195.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The findings and conclusions in this article have not been formally disseminated by the U.S. Department of Health & Human Services nor by the U.S. Department of Energy, and should not be construed to represent any determination or policy of University, Agency, Administration and National Laboratory.

The **remoter** logo comes from the image "Tradtelefon-illustration". Licensed under Public Domain via Commons.

This manual may be incorrect or out-of-date. The author(s) assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

This publication was typeset using LaTeX.

Contents

1 Introduction

[3] [5] [1]

1.1 Installation

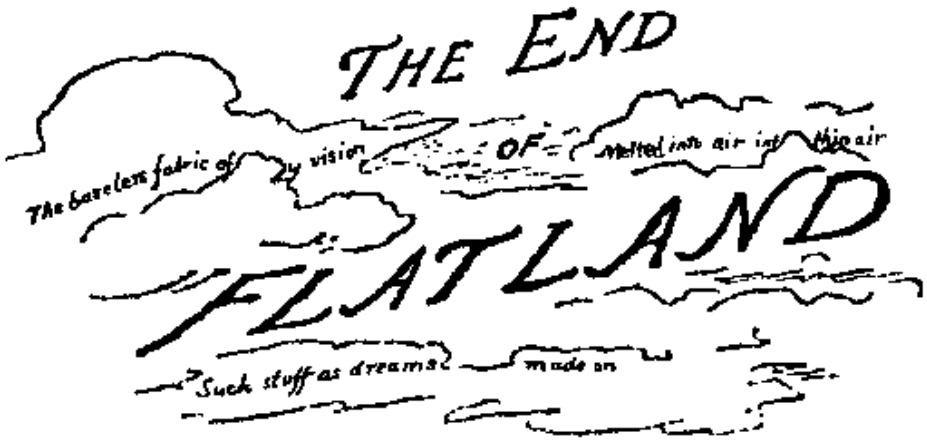
You can install the stable version from CRAN using the usual `install.packages()`:

```
install.packages("pbdML")
```

The development version is maintained on GitHub, and can easily be installed by any of the packages that offer installations from GitHub:

```
### Pick your preference
devtools::install_github("RBigData/pbdML")
ghit::install_github("RBigData/pbdML")
remotes::install_github("RBigData/pbdML")
```

2 Dimension Reduction



"Be patient, for the world is broad and wide." Image from *Flatland*

2.1 FLD

TODO

2.2 Randomized SVD/PCA

The singular value decomposition (SVD) is a matrix factorization, with numerous applications.

$$A = U\Sigma V^T$$

An application of the SVD well-known to statisticians is principal components analysis (PCA) [4].

A common technique is to compute the first 2 or 3 principal components in order to visualize high-dimensional data.

Estimation [2]

To show how this works, we generate a 30000×5000 matrix with 3 different separate clusters:

```
gen <- function(m, n, mean, sd) matrix(rnorm(m*n, mean, sd), m, n)

m <- 10000
n <- 5000
sd <- 10

x1 <- gen(m, n, 0, sd)
x2 <- gen(m, n, 4, sd)
x3 <- gen(m, n, -2, sd)

x <- rbind(x1, x2, x3)

library(pbdML)
system.time({
  pc <- rpca(x, k=2)
})
##  user  system elapsed
## 8.908   4.028  11.674
```

Compare this to the full PCA computation:

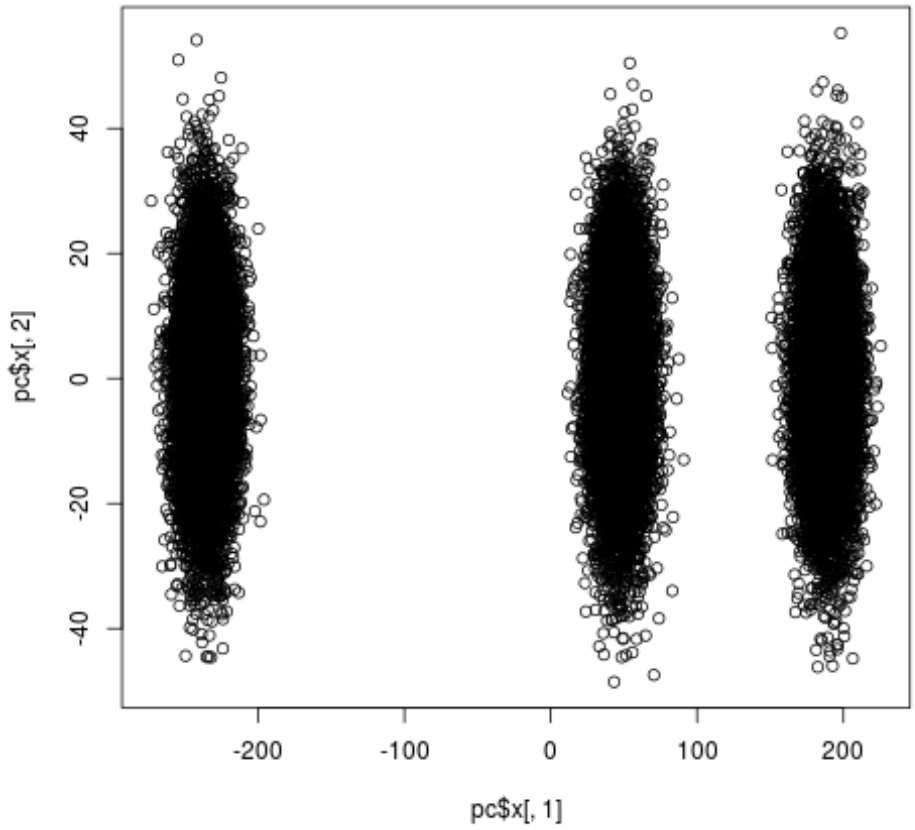
```
system.time({
  pc.full <- prcomp(x)
})
##     user   system  elapsed
## 644.632  159.220  240.094
```

The size comparisons are even more striking:

```
library(memuse)

memuse(x)
## 1.118 GiB
memuse(pc)
## 548.773 KiB
memuse(pc.full)
## 1.304 GiB

plot(pc$x[, 1], pc$x[, 2])
```



2.3 Decomp/Recomp

One day on Twitter, someone asked a very interesting question:

Ahmed Moustafa
@AhmedMoustafa

Help please on how to rebuild the matrix after excluding some principal components from a PCA analysis? #statistics #rstats #bioinformatics

6:10 AM - 22 Sep 2015

After some requests for clarification, the problem was stated as follows:

Ahmed Moustafa
@AhmedMoustafa

@wrathematics I mean, is it possible to exclude the variance contributed by certain PCs from the original matrix?

7:36 AM - 22 Sep 2015

3 Legal

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The findings and conclusions in this article have not been formally disseminated by the U.S. Department of Health & Human Services nor by the U.S. Department of Energy, and should not be construed to represent any determination or policy of University, Agency, Administration and National Laboratory.

This manual may be incorrect or out-of-date. The authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

4 References

References

[1] Wei-Chen Chen, George Ostrouchov, Drew Schmidt, Pragneshkumar Patel, and Hao Yu. pbdMPI: Programming with big data – interface to MPI, 2012. R Package, URL http://cran.r-project.org/package=pbdMPI.

[2] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[3] G. Ostrouchov, W.-C. Chen, D. Schmidt, and P. Patel. Programming with Big Data in R, 2012.

[4] A.C. Rencher. *Methods of Multivariate Analysis*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. John Wiley & Sons, 2002.

[5] Drew Schmidt, Wei-Chen Chen, George Ostrouchov, and Pragneshkumar Patel. pbdDMAT: Programming with big data – distributed matrix algebra computation, 2012. R Package.