

Example: Pre-Computed Kernel Matrices with SVM Optimization via KeBABS

Ulrich Bodenhofer*

December 21, 2017

The results below are generated from an R script.

```
## to run this file as a whole and produce a PDF report, enter the following:
## > install.packages("knitr") ## if not already installed
## > library(knitr)
## > stitch("KeBABS_MWE.R")

library(kebabs)
## if the package is not installed, enter the following:
## > source("https://bioconductor.org/biocLite.R")
## > biocLite("kebabs")

## load data
data(iris)

## transform into binary -1/+1 classification problem
iris$Species <- factor(ifelse(iris$Species == "setosa", +1, -1))

train <- sort(sample(1:nrow(iris), 0.6 * nrow(iris)))
test <- (1:nrow(iris))[-train]

## compute simple kernel matrix using linear kernel
Kmat <- tcrossprod(as.matrix(iris[train, 1:4]))

## normalize kernel matrix
selfSim <- sqrt(diag(Kmat))
Kmat <- sweep(Kmat, 1, selfSim, FUN="/") # scale rows
Kmat <- sweep(Kmat, 2, selfSim, FUN="/") # scale columns
Kmat <- as(Kmat, "KernelMatrix")

## train SVM
model <- kebabs:::svmd.default(Kmat, iris$Species[train], cost=1)

## compute "kernel matrix" of test vs. support vectors
## (model$index contains indices of support vectors)
KmatTest <- tcrossprod(as.matrix(iris[test, 1:4]),
                      as.matrix(iris[train[model$index], 1:4]))
selfSimTest <- sqrt(apply(as.matrix(iris[test, 1:4]), 1,
                             function(x) sum(x^2)))
```

*This report is automatically generated with the R package **knitr** (version 1.17).

```

KmatTest <- sweep(KmatTest, 1, selfSimTest, FUN="/") # scale rows
KmatTest <- sweep(KmatTest, 2, selfSim[model$index], FUN="/") # scale columns
KmatTest <- as(KmatTest, "KernelMatrix")

## make predictions for test samples
## (model$coefs ... alpha_i * y_i for support vectors,
## model$rho ... -b,
## model$levels ... labels used by SVM,
## model$label ... assignment of labels)
yTest <- factor(model$levels[ifelse(KmatTest %*% model$coefs - model$rho >= 0,
                                   model$label[1], model$label[2])])

## compute confusion table
table(y=iris$Species[test], g=yTest)

##      g
## y    -1  1
##   -1 42  0
##    1  0 18

```

The R session information (including the OS info, R version and all packages used):

```

sessionInfo()

## R version 3.4.3 (2017-11-30)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 17.10
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C              LC_TIME=en_US.UTF-8
##  [4] LC_COLLATE=en_US.UTF-8   LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C                LC_ADDRESS=C
## [10] LC_TELEPHONE=C          LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4      parallel    stats       graphics    grDevices   utils       datasets    methods
## [9] base
##
## other attached packages:
## [1] kebabs_1.12.0      kernlab_0.9-25      Biostrings_2.46.0   XVector_0.18.0
## [5] IRanges_2.12.0     S4Vectors_0.16.0    BiocGenerics_0.24.0 knitr_1.17
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.13      lattice_0.20-35     class_7.3-14        grid_3.4.3          magrittr_1.5
## [6] evaluate_0.10.1   e1071_1.6-8         highr_0.6           stringi_1.1.6       zlibbioc_1.24.0
## [11] Matrix_1.2-11     apcluster_1.4.4     tools_3.4.3         stringr_1.2.0       LiblineaR_2.10-8
## [16] compiler_3.4.3
##
Sys.time()

## [1] "2017-12-21 09:58:42 CET"

```