ECS 256 Group Project

Saheel Godhane
Paari Kandappan
Jack Norman
Ivana Žetko

UC Davis

March 13, 2014

Problem 1

The asymptotic bias of $\hat{m}_{X;Y}(t)$ at $t = 0.5$ can be calculated as follows:

$$E(\hat{m}_{X;Y}(0.5) - m_{X;Y}(0.5)) = E(\hat{m}_{X;Y}(0.5)) - E(m_{X;Y}(0.5)) \quad (1)$$

$$= E(0.5\beta) - E(0.5^{0.75}) \quad (2)$$

$$\approx 0.5E(\beta) - 0.595 \quad (3)$$

Problem 1

In general, the mean squared error (MSE) associated with a particular choice of β estimated from points t_i, $i = 1, 2, \ldots, n$ is as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{m}_{X;Y}(t_i) - m_{X;Y}(t_i))^2 \tag{4}$$

$$= \frac{1}{n} \sum_{i=1}^{n} (\beta t_i - t_i^{0.75})^2 \tag{5}$$

Problem 1

$$Error = \lim_{n\to\infty}(\sum_{i=1}^{n}(\beta t_i - t_i^{0.75})^2) \tag{6}$$

$$= \int_0^1 (\beta t_i - t_i^{0.75})^2 dt \tag{7}$$

$$= \int_0^1 (\beta^2 t^2 - 2\beta t^{1.75} + t^{1.5}) dt \tag{8}$$

$$= \beta^2 \int_0^1 t^2 dt - 2\beta \int_0^1 t^{1.75} dt + \int_0^1 t^{1.5} dt \tag{9}$$

$$= \frac{1}{3}\beta^2 - \frac{2}{2.75}\beta + \frac{1}{2.5} \tag{10}$$

aiclogit(): AIC

```
aiclogit <- function(y, x) {
    y <- as.matrix(y)
    x <- as.matrix(x)
    fit <- glm(y ~ x,family=binomial())
    fitsum <- summary(fit)
    aic <- fitsum$aic
    return(aic)
}
```

ar2(): Adjusted R^2

```
ar2 <- function(y, x) {
    y <- as.matrix(y)
    x <- as.matrix(x)
    fit <- lm(y ~ x)
    fitsum <- summary(fit)
    adjr <- fitsum$adj.r.squared
    return(adjr)
}
```

prsm(): Input Validation

```
prsm <- function(y, x, k=0.01, predacc=ar2, crit=NULL,
    printdel=FALSE, cls=NULL) {
    require(parallel)
    # Convert y and x to matrix for the sake lm() and glm()
    y <- as.matrix(y)
    x <- as.matrix(x)

    minmax <- NULL
    # Determine whether to minimize of maximize the PAC
    if (identical(ar2, predacc)) {
        crit <- "max"
        minmax <- max
    } else if (identical(aiclogit, predacc)) {
        crit <- "min"
        minmax <- min
```

prsm(): Calculate Full Model

```
} else {
    if (is.null(crit)) {
        stop("Error: crit is NULL. Do you want to
            minimize or maximize the PAC?")
    }
            else if (crit == "min"){
                minmax <- min
            }
            else if (crit == "max"){
                minmax <- max
            }
}
# Calculate full model to begin
full <- predacc(y, x) # starting PAC
varsleft <- 1:ncol(x) # variable to keep track of
    current variables in the model
if (printdel) cat("full outcome =  ", full)
```

prsm(): Begin While Loop

```
# Loop: delete variables one at a time, a greedy
    approach
tmpbest <- full
flag <- TRUE
while(flag) {
    # Calculate PAC for each possible removal
    if (is.null(cls)) {
        tmp <- lapply(1:length(varsleft), function(i) {
            pac <- predacc(y, x[,varsleft[-i]])
            return(pac)
        })
    } else if (!is.null(cls)) {
        tmp <- clusterApply(cls, 1:length(varsleft),
            function(i) {
            pac <- predacc(y, x[,varsleft[-i]])
            return(pac)
        })
    }
```

prsm(): Find Best PAC

```
bestpac <- minmax(unlist(tmp))
# Is the ratio "almost" enough (parsimoniously) to
    justify deleting the variable?
if (crit == "min") {
    flag <- (bestpac / tmpbest) < 1 + k
} else if (crit == "max") {
    flag <- (bestpac / tmpbest) > 1 - k
}
```

prsm(): Find Variable to Remove

```
        # If flag is still true, remove the variable and
            update varsleft
        if (flag) {
            var2rem <- which(tmp == bestpac)[1]
            nameOfvar2rem <- colnames(x)[varsleft[var2rem]]
            varsleft <- varsleft[-var2rem]
            if (printdel) cat("\ndeleted   ", nameOfvar2rem,
                "\nnew outcome =  ", bestpac)
                tmpbest <- bestpac
        }
            if(length(varsleft) == 1)
                break;
    } # end while()
    cat("\n")
    print(varsleft)
    return(varsleft)
}
```

prsm(): Pima Data Example

```
# Compare the answers and runtimes of the serial method
    versus parallel method
system.time(prsm(pima[,9], pima[,1:8], predacc = aiclogit,
    printdel = TRUE))
```

```
full outcome = 741.4454
deleted Thick
new outcome = 739.4534
deleted Insul
new outcome = 739.4617
deleted Age
new outcome = 740.5596
deleted BP
new outcome = 744.3059
[1] 1 2 6 7
user system elapsed
0.393 0.034 0.470
```

prsm(): Pima Data Example In Parallel

```
# make cluster for parallel method
cls <- makeCluster(rep('localhost', 4))

system.time(prsm(pima[,9], pima[,1:8], predacc = aiclogit,
    printdel = TRUE, cls = cls))
```

```
full outcome = 741.4454
deleted Thick
new outcome = 739.4534
deleted Insul
new outcome = 739.4617
deleted Age
new outcome = 740.5596
deleted BP
new outcome = 744.3059
[1] 1 2 6 7
user system elapsed
0.038 0.006 0.387
```

SMS Spam Dataset

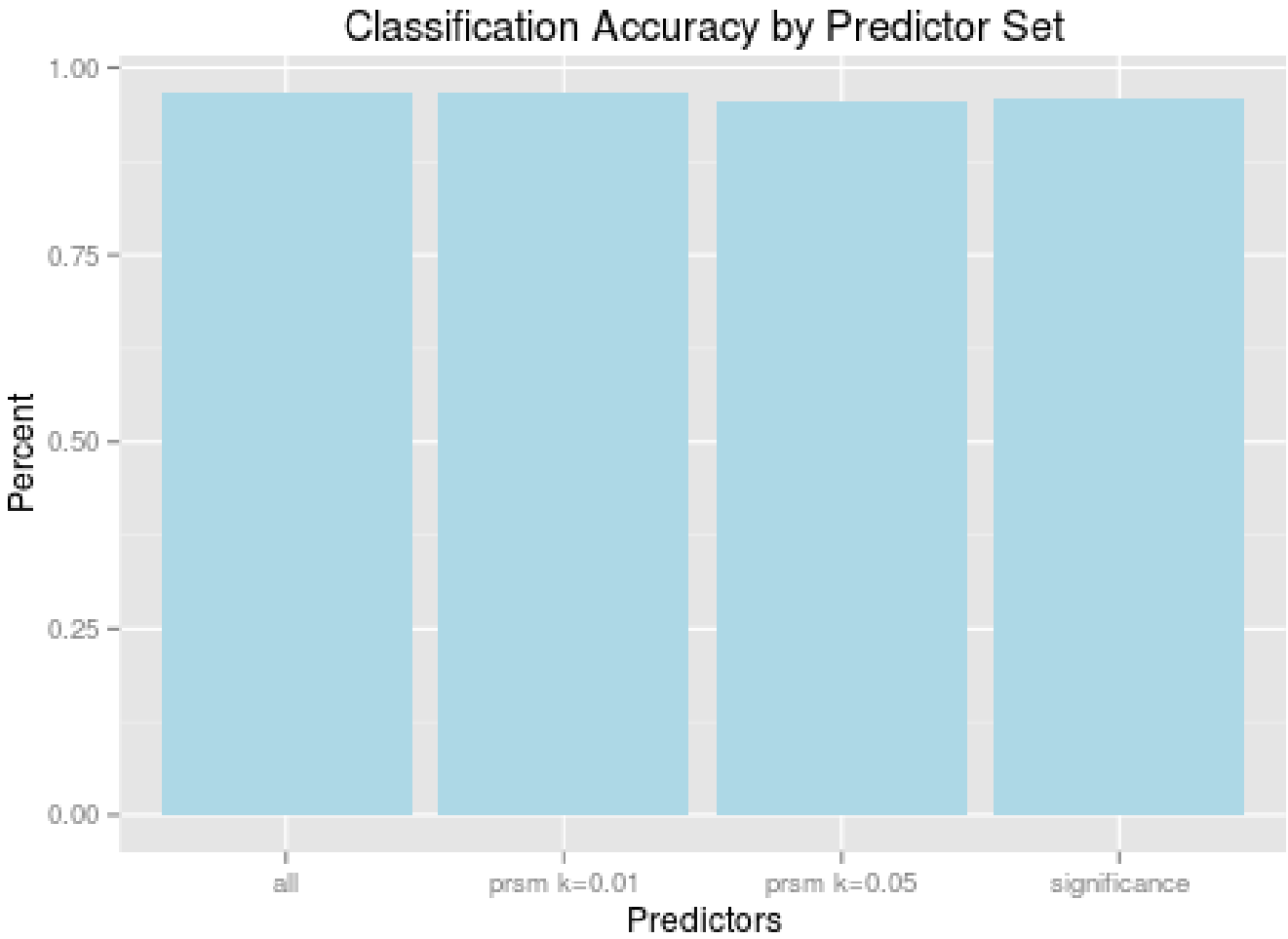
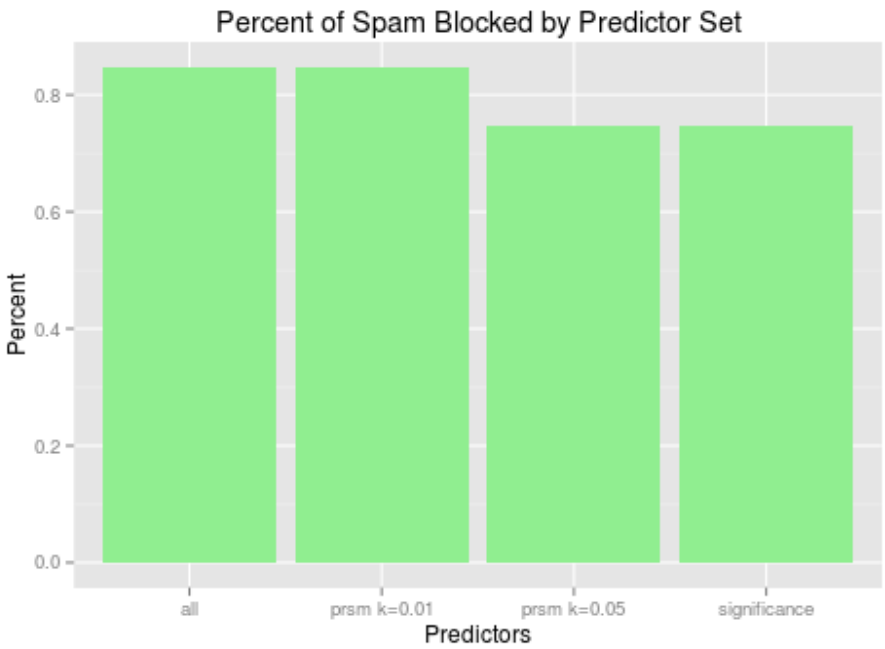


Figure 1 : Percent of spam (left) and ham (right) messages blocked in

SMS Spam Dataset



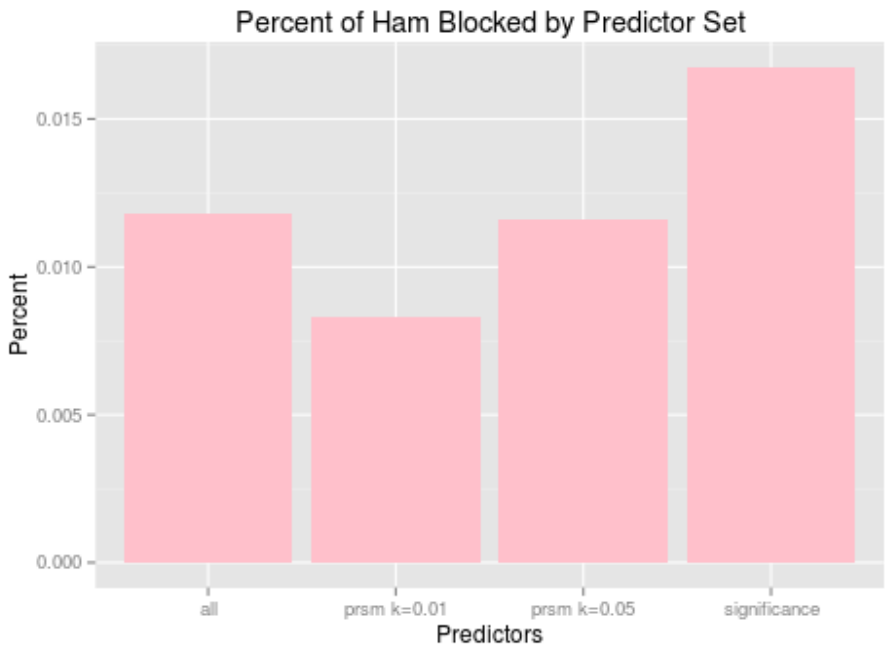


Figure 2 : Percent of spam (left) and ham (right) messages blocked in 5-fold cross validation

Istanbul Stock Exchange Dataset

(small n, small p, regression)

	$k = 0.05$	$k = 0.01$	$p < 0.05$
Predictors chosen	6 7	5 6 7	5 6 7
Adjusted R^2	0.564	0.578	0.578

Figure 3 : Predictors (X_i) chosen by the various parsimony inducing methods, adjusted R^2 using each of those sets of predictors

Automobile Prices Dataset

(small n, large p, regression)

	$k = 0.05$	$k = 0.01$	$p < 0.05$
Predictors chosen	2 14 16	2 3 4 14 16 17 18 21 23	3 14 16 17
Adjusted R^2	0.2873	0.3271	0.578

Figure 4 : Model fitting methods with the predictors chosen and adjusted R^2

Custom PAC: *leave1out01()*

- Jackknife analysis: train $n - i$ samples and test on i^{th} sample
- Only considered the classification case

Custom PAC: *leave1out01()*

- Jackknife analysis: train $n - i$ samples and test on i^{th} sample
- Only considered the classification case
- Basic idea:
  1. $model = lm(y[-i,] \sim x[-i,])$
  2. $prediction = (model\$weights \cdot x_i) + model\$intercept$

*leave1out*01() Pima results

```
[1] ``Testing leave1out01() on Pima dataset''
[1] ``PAC value:''
[1] 0.77474
```

leave1out01() results with prsm()

```
[1] ``Testing leave1out01 as PAC for prsm() on Pima''
full outcome =   0.77474
deleted    Thick
new outcome =   0.77474
deleted    NPreg
new outcome =   0.77344
deleted    Insul
new outcome =   0.77083
deleted    BP
new outcome =   0.77604
deleted    Age
new outcome =   0.76953
[1] 2 6 7
```