Using prim to estimate highest density difference regions

Tarn Duong

3 July 2008

1 Introduction

The Patient Rule Induction Method (PRIM) was introduced by Friedman and Fisher (1999). It is a technique from data mining for finding 'interesting' regions in high-dimensional data. We start with regression-type data $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$ where $\boldsymbol{X}_i$ is d-dimensional and Y_i is a scalar response variable. We are interested in the conditional expectation function

$$m(\boldsymbol{x}) = \mathbb{E}(Y|\boldsymbol{x}).$$

In the case where we have 2 samples, we can label the response as

$$Y_i = \begin{cases} 1 & \text{if } \boldsymbol{X}_i \text{ is from sample 1} \\ -1 & \text{if } \boldsymbol{X}_i \text{ is from sample 2.} \end{cases}$$

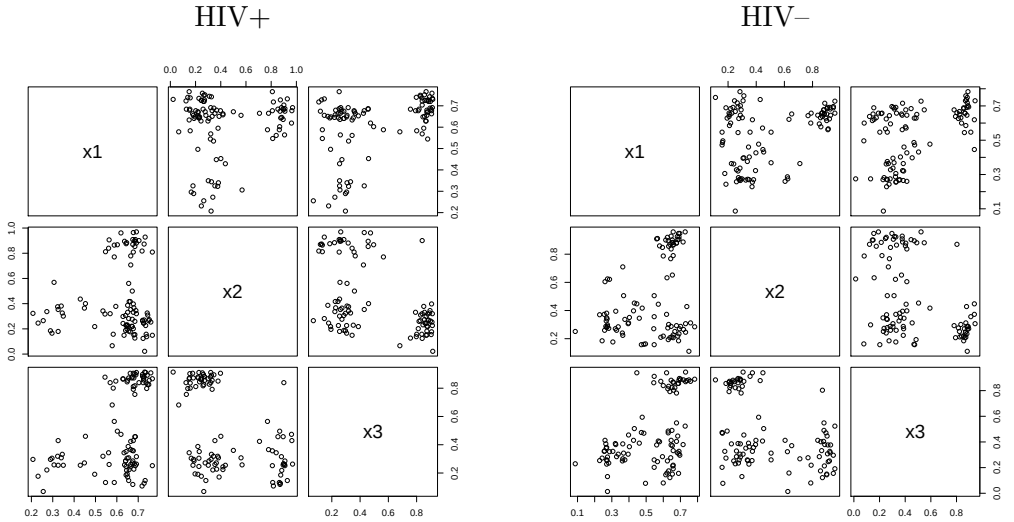
Then PRIM finds the regions where the samples are most different. Here we have a positive HDR (where sample 1 points dominate) and a negative HDR (where sample 2 points dominate).

We look at a 3-dimensional data set (`quasiflow`) included in the `prim` library. It is a randomly generated data set from two normal mixture distributions whose structure mimics some light scattering data, taken from a machine known as a flow cytometer.

```
> library(prim)
> data(quasiflow)
> yflow <- quasiflow[, 4]
> xflow <- quasiflow[, 1:3]
> xflowp <- quasiflow[yflow == 1, 1:3]
> xflown <- quasiflow[yflow == -1, 1:3]
```

We can think of `xflowp` as flow cytometric measurements from an HIV+ patient, and `xflown` from an HIV– patient.

```
> pairs(xflowp[1:100, ])
> pairs(xflown[1:100, ])
```



There are two ways of using `prim.box` to estimate where the two samples are most different (or equivalently to estimate the HDRs of the difference of the density functions). In the first way, we assume that we have suitable values for the thresholds. Then we can use

```
> qflow.thr <- c(0.38, -0.23)
> qflow.prim <- prim.box(x = xflow, y = yflow, threshold = qflow.thr,
+     threshold.type = 0)
```

An alternative is compute PRIM box sequences which cover the entire data range, and then use `prim.hdr` to experiment with different threshold values. This two-step process is more efficient and faster than calling `prim.box` for each different threshold. We're happy with the positive HDR threshold so we can compute the positive HDR directly:

```
> qflow.hdr.pos <- prim.box(x = xflow, y = yflow, threshold = 0.38,
+     threshold.type = 1)
```

On the other hand, we're not sure about the negative HDR thresholds.

```
> qflow.neg <- prim.box(x = xflow, y = yflow, threshold.type = -1)
> qflow.hdr.neg1 <- prim.hdr(qflow.neg, threshold = -0.23, threshold.type = -1)
> qflow.hdr.neg2 <- prim.hdr(qflow.neg, threshold = -0.43, threshold.type = -1)
> qflow.hdr.neg3 <- prim.hdr(qflow.neg, threshold = -0.63, threshold.type = -1)
```

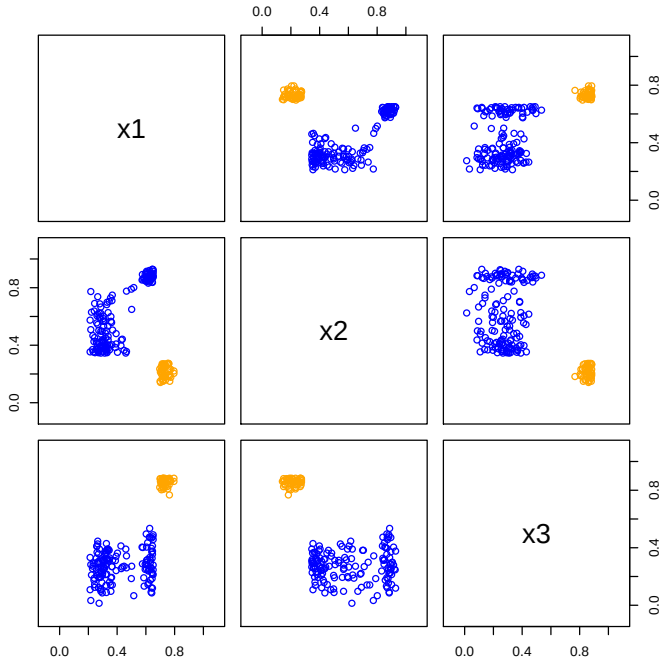
After examining the summaries and plots, we choose `qflow.hdr.neg1` to combine with `qflow.hdr.pos`.

```
> qflow.prim2 <- prim.combine(qflow.hdr.pos, qflow.hdr.neg1)
> summary(qflow.prim2)
```

```
           box-mean   box-mass threshold.type
box1      0.54003407 0.05127533              1
box2     -0.68237347 0.05005241             -1
box3     -0.39072848 0.05276031             -1
box4     -0.29465095 0.09634871             -1
box5*     0.11245776 0.74956324             NA
overall   0.02882600 1.00000000             NA
```

In the plot below, the positive HDR is coloured orange, and the negative HDR is coloured blue. The following plot is not exactly the output produced by the commands, but has been thinned for clarity.

```
> cols <- qflow.prim2$ind
> cols[cols == 1] <- "orange"
> cols[cols == -1] <- "blue"
> plot(qflow.prim2, col = cols)
```



References

Friedman, J. H. and Fisher, N. I. (1999). Bump-hunting for high dimensional data. *Statistics and Computing*, **9**, 123–143.