Деревья решений и случайные леса (Binary decision tree & Random forest)

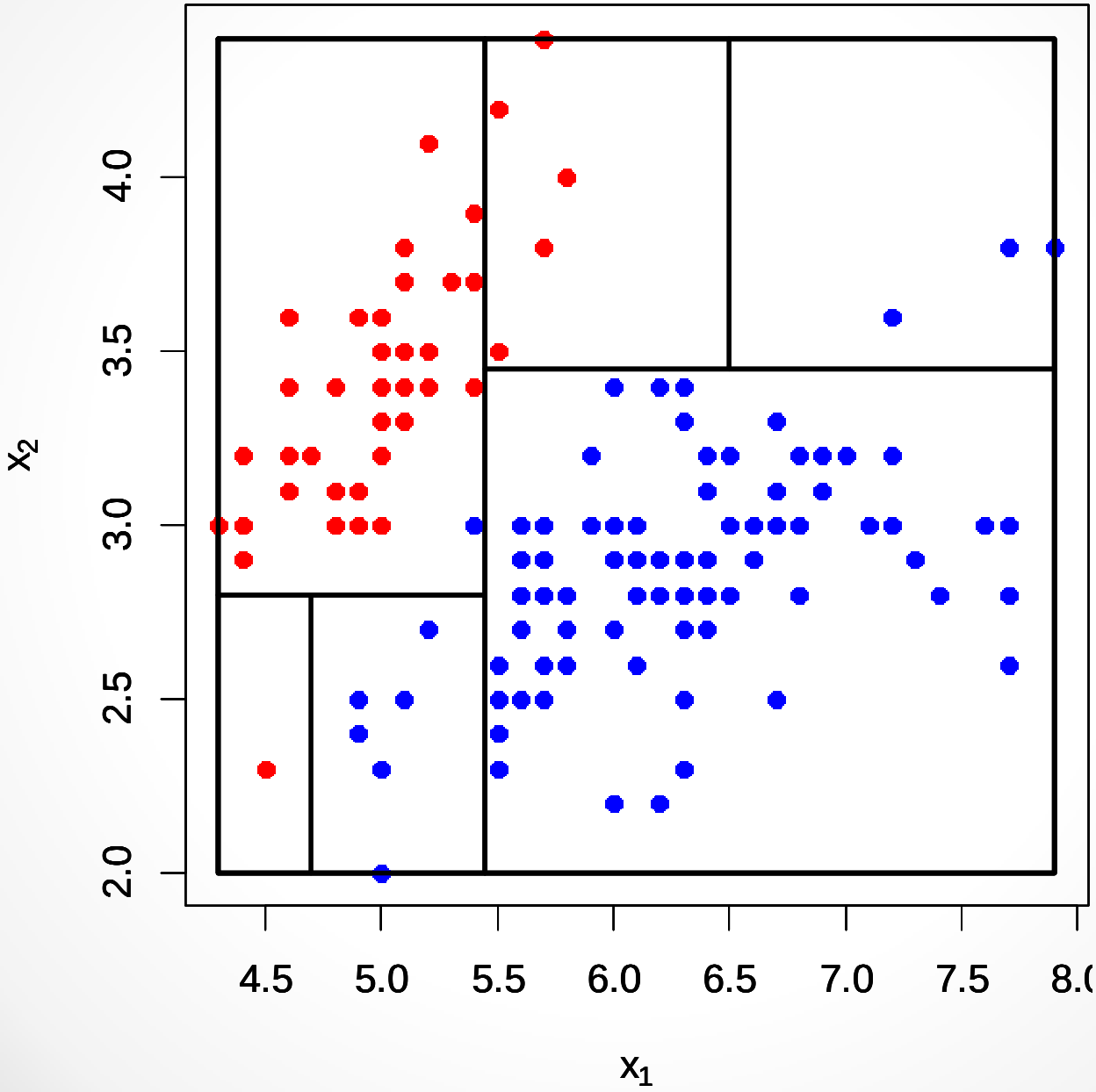
Количественная аналитика

Основная идея

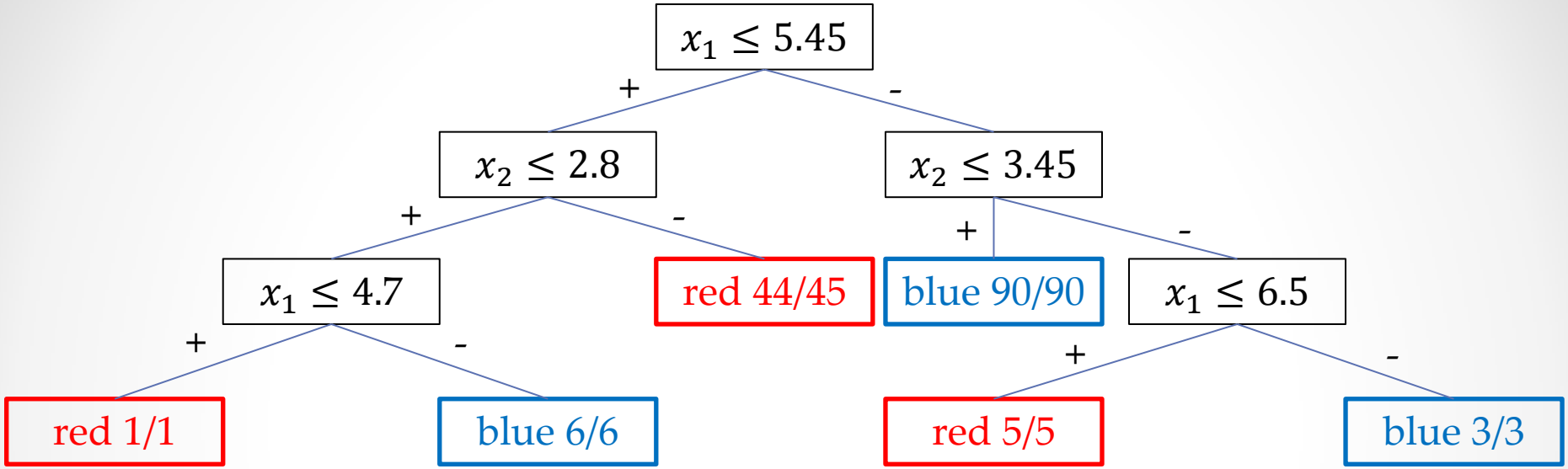
Классификация наблюдений на основе последовательного применения критериев (>, <, ∈) к тому или иному признаку

При этом пространство $R \supset X$ рекурсивно разделяется гиперплоскостями, параллельными оси одного из признаков, до тех пор, пока в каждой из получившихся областей не образуется значительное большинство наблюдений одного класса

Классификация бинарными деревьями



Дерево решений



Основные понятия

Пусть условие $x_j \leq v$ разделяет пространство R на 2 части: R_Y и R_N, тогда множество наблюдений $D = \{\vec{x}^{(i)}, i \in \{1; \dots; m\}\}$ также разделяется на $D_Y = \{\vec{x}^{(i)} \in D : x_{i,j} \leq v\}$ и $D_N = \{\vec{x}^{(i)} \in D : x_{i,j} > v\}$

Однородность j-й области: $purity(D_j) = \max\limits_k \frac{m_{j,k}}{m_j}$, где

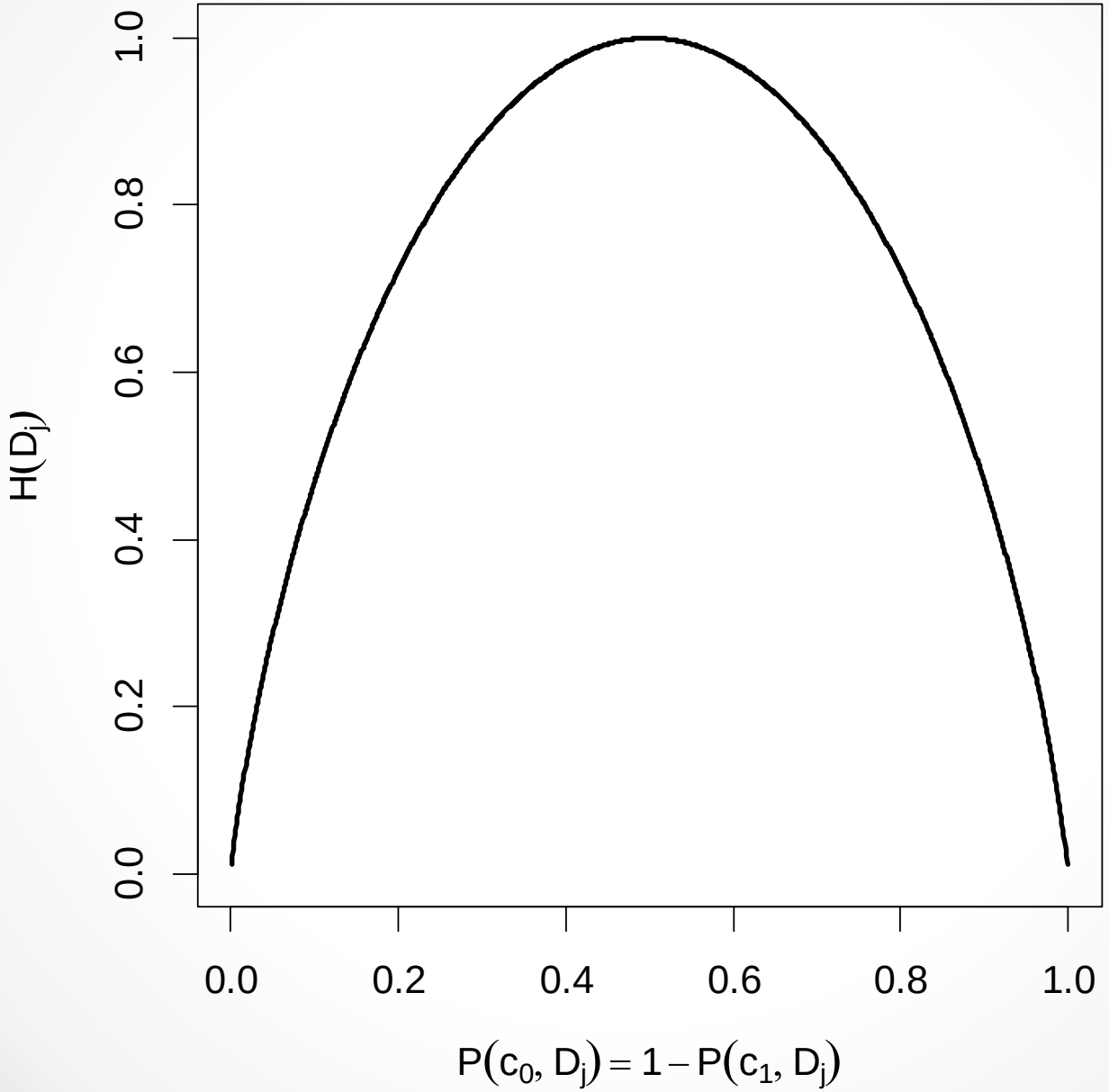
$m_{j,k} = \sum_{i=1}^{m} \boldsymbol{I}(\vec{x}^{(i)} \in D_j, y_i = c_k)$, $m_j = \sum_{i=1}^{m} \boldsymbol{I}(\vec{x}^{(i)} \in D_j) = \sum_{k=1}^{K} m_{j,k}$

Энтропия области: $H(D_j) = -\sum_{k=1}^{K} P(c_k|D_j) \log_2 P(c_k|D_j)$, где

$P(c_k|D_j) = \frac{m_{j,k}}{m_j}$ — вероятность нахождения наблюдения k-го класса в области D_j

Энтропия разделения: $H(D_Y, D_N) = \frac{m_Y}{m} H(D_Y) + \frac{m_N}{m} H(D_N)$

Энтропия



Оценка эффективности разделения

Информативность (сокращение энтропии):

$$gain(D, D_Y, D_N) = H(D) - H(D_Y, D_N)$$

Вместо энтропии можно использовать коэффициент Джини:

$$G(D) = 1 - \sum_{k=1}^{K} P^2(c_k|D), \;\; G(D_Y, D_N) = \frac{m_Y}{m} G(D_Y) + \frac{m_N}{m} G(D_N)$$

Если величины $x_{i,j}$ непрерывны, то в качестве кандидатов на точки разделения рассматриваются середины интервалов между последовательными уникальными значениями

Если $x_{i,j}$ дискретны, то каждое их уникальное значение рассматривается как возможная точка разделения

Деревья решений в R

исходные данные

```
data(iris)
m <- nrow(iris); n <- ncol(iris) - 1
head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
```

обучающая и тестовая выборки

```
train.obs <- sample(1:m, size = trunc(0.8*m))
test.obs <- (1:m)[-train.obs]
X.tr <- iris[train.obs,-(n+1)]; y.tr <- as.numeric(iris[train.obs,n+1])
X.ts <- iris[test.obs,-(n+1)]; y.ts <- as.numeric(iris[train.obs,n+1])
head(X.tr)
    Sepal.Length Sepal.Width Petal.Length Petal.Width
107          4.9         2.5          4.5         1.7
117          6.5         3.0          5.5         1.8
89           5.6         3.0          4.1         1.3
head(y.tr)
[1] 3 3 2
```

Построение модели

форматы признаков

```
X.tr <- as.matrix(X.tr); X.ts <- as.matrix(X.ts)
```

определение границ области классификации

```
apply(iris[,-(n+1)], 2, range)
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]          4.3         2.0          1.0         0.1
[2,]          7.9         4.4          6.9         2.5
bound <- cbind(c(3,1,0,0), c(9, 6, 8, 4))
```

формализация области классификации

```
area <- list(bound = bound, class = y.tr, X = X.tr, majorClass = NULL)
```

классификация

```
source("rf_func.r")
dt <- decisionTree(area)
  Areas: 1
  Areas: 2
  Areas: 3
  Areas: 5
Compiling return...
```

```
dt[[1]]
$bound
     [,1] [,2]
[1,]    3 9.00
[2,]    1 6.00
[3,]    0 2.45
[4,]    0 4.00

$majorClass
[1] 1
```

Построение прогноза

```
pred <- predictDT(X.ts, dt)
head(pred)
 1  2 11 15 21 24
 1  1  1  1  1  1

acc <- function(y, y.hat) {
  if (length(y) != length(y.hat)) stop("y & y.hat must have equal length")
  sum(y == y.hat) / length(y)
}

acc(pred, y.ts)
[1] 0.9666667
```

Случайный лес

Случайный лес представляет собой совокупность моделей — бинарных деревьев решений, — отличающихся случайным выбором экзогенных параметров

Такими параметрами могут быть: выбор точек разделения областей, выбор набора обучающих наблюдений из тренировочной совокупности и др.

Прогнозным значением в задачах классификации может являться наиболее часто встречающийся номер класса среди прогнозов по деревьям, составляющим лес

Случайный лес в R

```
ntree <- 20; frac <- 1/2
rf <- randomForest(area, ntree = ntree, frac = frac)

pred <- predictRF(X.ts, rf)
acc(pred, y.ts)
[1] 1
```

Домашнее задание

- классифицировать рукописные цифры из файла «tree_digits_test.csv»

Для выполнения задания можно использовать пакеты `rpart`, `randomForest`