



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного автономного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИУК «Информатика и управление»**

КАФЕДРА **ИУК4 «Программная инженерия»**

Лабораторная работа №6

**Методы классификации многомерных объектов
пересекающихся классов с использованием карт Кохонена**

ДИСЦИПЛИНА: «Методы машинного обучения»

Выполнил: студент гр. ИУК5-72Б

(Подпись)

Ли Р. В.
(Ф.И.О.)

Проверил:

(Подпись)

(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Вариант 14

Больные увеличением щитовидной железы общим числом 70 человек были разделены на две группы.

Группа 1. Лечение оказалось успешным; пациент здоров.

Группа 2. Лечение успешно на 75 %, возможен рецидив.

По результатам обследования 70 пациентов имеются следующие измерения:

Y3 – йод, регистрируемый через 3 часа после принятия испытательной дозы;

Y5 – йод, регистрируемый через 12 часов после принятия испытательной дозы;

Y7 – содержание в крови белковосвязанного йода (РВ¹³¹J) через 20 часов;

Y9 – содержание в крови белковосвязанного йода (РВ¹³¹J) через 33 часа;

kl – номер группы.

Таблица

№	K l	Y3	Y5	Y7	Y9
1	1	14. 4	25. 1	0.2 5	1.7 7
2	1	20. 1	40. 1	0.1 7	2.8 6
3	1	24. 1	32. 1	0.1 9	3.1 1
4	1	13. 1	16. 9	0.1 4	1.4 8
5	1	17. 3	32. 1	0.3 9	12
6	1	40. 5	64. 4	0.2 2	40. 3

7	1	52. 7	50. 0	0.5 6	1.9 7
8	1	20. 8	22. 3	0.1 7	13. 86
9	1	14. 0	3.1	0.1 9	15. 23
1 0	1	27. 0	41. 7	0.1 3	24. 08
1 1	1	44. 3	63. 8	0.2 6	...
1 2	1	47. 5	50. 1	0.2 1	
1 3	1	54. 0	57. 0	0.1 7	...
1 4	1	16. 1	20. 6	0.2 9	
1 5	1	87. 5	74. 5	0.4 8	
1 6	1	37. 8	63. 4	0.3 1	
1 7	2	35. 8	48. 0	2.7 4	
1 8	2	65. 0	60. 0	1.3 7	
1 9	2	62. 0	65. 0	0.7 0	
2 0	2	71. 6	45. 0	1.4 0	
2 1	2	24. 1	45. 0	0.2 2	
2 2	2	37. 2	55. 0	0.0 1	

2 3	2	35. 4	44. 6	0.0 9	
			
7 0	2		61. 1		0.9 8

Необходимо разработать метрический классификатор, который функционирует на предметной области исследования медицинских анализов. Четыре характеристических показателя надо взять из таблицы. Используем метод К-ближайших соседей и метод Парзена. Сформировать обучающие и тестовые выборки. Полученные результаты визуализировать и сравнить. Представить значения параметров с минимальным уровнем ошибки. Для метода К соседей параметр $K = 37$, для метода Парзена тип ядра выбрать , "epanechnikov", "eddy" , а параметр `optim. method = "Nelder-Mead"`. Проверить точность прогнозов.

task.R

```
# -----
```

```
# 0) Библиотеки
```

```
# -----
```

```
library(kohonen)
```

```
library(RSNNS)
```

```
set.seed(123)
```

```
# -----
```

```
# 1) Данные
```

```
# -----
```

```
n <- 33
```

```
speed <- c(runif(12, 0.01, 0.65), runif(12, 0.66, 2.0), runif(9, 2.1, 4.0))
```

```
height <- c(runif(12, 0.01, 0.55), runif(12, 0.56, 1.2), runif(9, 1.21, 2.5))
```

```
data_df <- data.frame(speed = speed, flame_height = height)
```

```

rownames(data_df) <- paste0("obs", 1:n)

data_df$true_class <- factor(
  ifelse(data_df$speed <= 0.65 & data_df$flame_height <= 0.55, "weak",
    ifelse(data_df$speed > 2 | data_df$flame_height > 1.2, "strong", "medium"))
)

# -----
# 2) Стандартизация для SOM
# -----
data_matrix <- scale(data_df[, c("speed", "flame_height")])

# -----
# 3) SOM с supersom
# -----
som_grid <- somgrid(xdim = 8, ydim = 4, topo = "rectangular")

# supersom без neigh
som_model <- supersom(data_matrix, grid = som_grid, rlen = 200,
  alpha = c(0.05, 0.01), keep.data = TRUE)

# -----
# 4) Визуализация
# -----
# Learning progress
plot(som_model$changes, type = "o", main = "SOM: learning progress",
  xlab = "Iteration", ylab = "Mean distance change")

# Mapping объектов
class_cols <- c("weak"="blue", "medium"="orange", "strong"="red")
plot(som_model, type = "mapping", main = "SOM: mapping (objects)",
  pchs = 19, col = class_cols[data_df$true_class])
legend("topright", legend = levels(data_df$true_class),
  col = class_cols, pch = 19, bty = "n")

```

```

# U-Matrix
plot(som_model, type = "dist.neighbours", main = "SOM: U-Matrix")

# Node prototypes (коды)
plot(som_model, type = "codes", main = "SOM: node prototypes")
legend("topright", legend = levels(data_df$true_class),
      col = class_cols, pch = 19, bty = "n")

# -----
# 5) Кластеризация узлов SOM
# -----
codes <- som_model$codes[[1]]
hc <- hclust(dist(codes), method = "complete")
k <- 10
som_cluster <- cutree(hc, k)

pretty_palette <- c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd",
                  "#8c564b", "#e377c2", "#7f7f7f", "#bcbd22", "#17becf")

plot(som_model, type = "codes", bgcol = pretty_palette[som_cluster],
      main = paste("SOM codes with", k, "clusters"))
add.cluster.boundaries(som_model, som_cluster)

# таблица соответствия
obj_cluster <- som_cluster[som_model$unit.classif]
print(table(obj_cluster, data_df$true_class))

# -----
# 6) RBF и rbfDDA
# -----

# 6.1 Нормализация признаков в [0,1]
normalize01 <- function(x) (x - min(x)) / (max(x) - min(x))

```

```
inputs <- as.matrix(apply(data_df[, c("speed", "flame_height")], 2, normalize01))
```

```
# 6.2 One-hot кодирование классов
```

```
targets <- decodeClassLabels(data_df$true_class)
```

```
colnames(targets) <- levels(data_df$true_class)
```

```
targets <- as.matrix(targets)
```

```
# 6.3 Разбиение на train/test (70%/30%)
```

```
set.seed(42)
```

```
train_idx <- sample(1:n, round(0.7 * n))
```

```
test_idx <- setdiff(1:n, train_idx)
```

```
inputsTrain <- inputs[train_idx, , drop = FALSE]
```

```
inputsTest <- inputs[test_idx, , drop = FALSE]
```

```
targetsTrain <- targets[train_idx, , drop = FALSE]
```

```
targetsTest <- targets[test_idx, , drop = FALSE]
```

```
# 6.4 Обучение RBF
```

```
rbf_model <- rbf(inputsTrain, targetsTrain, size = 10, maxit = 500, linOut = FALSE)
```

```
# 6.5 Предсказание на тестовой выборке
```

```
pred_test <- predict(rbf_model, inputsTest)
```

```
# 6.6 Перевод предсказаний в классы
```

```
if (is.null(dim(pred_test))) {
```

```
  # pred_test — вектор (1 объект)
```

```
  pred_test_class <- colnames(targets)[which.max(pred_test)]
```

```
} else {
```

```
  # pred_test — матрица (несколько объектов)
```

```
  pred_test_class <- apply(pred_test, 1, function(x) colnames(targets)[which.max(x)])
```

```
}
```

```
true_test_class <- apply(targetsTest, 1, function(x) colnames(targets)[which.max(x)])
```

6.7 Проверка длины и таблица соответствия

```
cat("\nRBF:\n")
```

```
cat("Длина предсказаний:", length(pred_test_class), "\n")
```

```
cat("Длина истинных классов:", length(true_test_class), "\n")
```

```
print(table(pred_test_class, true_test_class))
```

```
# -----
```

```
# 7) Результаты
```

```
# -----
```

```
result_table <- data.frame(obs = rownames(data_df),
```

```
    speed = data_df$speed,
```

```
    flame_height = data_df$flame_height,
```

```
    true_class = data_df$true_class,
```

```
    unit = som_model$unit.classif,
```

```
    som_node_cluster = som_cluster[som_model$unit.classif])
```

```
print(result_table)
```

