

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

И.И. Кручинин
(к.т.н. доцент)

ЛАБОРАТОРНАЯ РАБОТА № 6
по курсу «Методы машинного обучения»
Методы классификации многомерных объектов пересекающихся классов
с использованием карт Кохонена

Приложение

Калуга
2018

Теоретические основы.

Самоорганизующиеся карты Кохонена

Самоорганизующиеся карты (SOM, Self Organizing Maps), разработанные Т. Кохоненом (Kohonen, 1982), представляют собой мощный инструмент, объединяющий две важные парадигмы анализа данных – кластеризацию и проецирование, т.е. визуализацию многомерных данных на плоскости. Процедура настройки SOM относится к алгоритмам обучения без учителя.

Сеть Кохонена имеет всего два слоя: входной и выходной, составленный из радиальных нейронов упорядоченной структуры (выходной слой называют также слоем топологической карты, или “экраном”). Нейроны выходного слоя располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. Количество нейронов в сетке p определяет степень детализации результата работы алгоритма, и, в конечном счете, от этого зависит точность обобщающей способности карты.

RBF – сети.

Также к алгоритмам классификации основанным на нейросетевой теории можно отнести сети с радиальными базисными функциями RBF. Обучение RBF-сети сводится к восстановлению плотностей классов $p_y(x)$ с помощью ЕМ-алгоритма. Результатом обучения являются центры μ_{yj} и дисперсии Σ_{yj} компонент $j = 1, \dots, k_y$. Оценивая дисперсии, мы фактически подбираем веса признаков в метриках $\rho_{yj}(x, \mu_{yj})$ для каждого центра μ_{yj} .

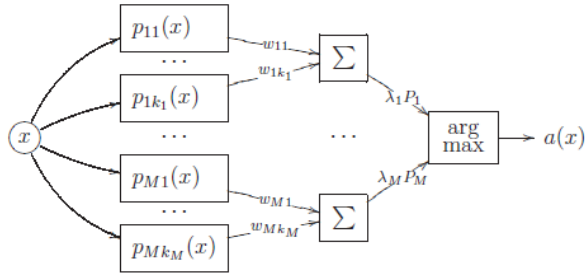


Рис. 5. Сеть радиальных базисных функций представляет собой трёхуровневую суперпозицию.

Пусть функции правдоподобия классов $p_y(x)$, $y \in Y$, представимы в виде смесей k_y компонент. Каждая компонента имеет n -мерную гауссовскую плотность с параметрами $\mu_{yj} = (\mu_{yj1}, \dots, \mu_{yjn})$, $\Sigma_{yj} = \text{diag}(\sigma_{yj1}^2, \dots, \sigma_{yjn}^2)$, $j = 1, \dots, k_y$:

$$p_y(x) = \sum_{j=1}^{k_y} w_{yj} p_{yj}(x), \quad p_{yj}(x) = \mathcal{N}(x; \mu_{yj}, \Sigma_{yj}), \quad \sum_{j=1}^{k_y} w_{yj} = 1, \quad w_{yj} \geq 0.$$

Примеры программного кода.

Карта Кохонена

```
#data(iris)
library(kohonen)
library(RSNNS)
```

```
ramFo = data.frame(read.table("cleverK.txt", header = TRUE, sep = ""))
#print("Исходные данные")
#print(ramFo)
```

```
alg <- matrix(0:0, nrow=150, ncol=4)
for (i in 1:50) {
  for (j in 1:4) {
    alg[i,j] = sample(5:50,1)
  }
}
```

```
for (i in 51:100) {
```

```

for (j in 1:4) {
  alg[i,j] = sample(51:95,1)
}
}

for (i in 101:150) {
  for (j in 1:4) {
    alg[i,j] = sample(1:4,1)
  }
}

#alg
write.table(alg, file="GMB1.txt")
ramFo2 = data.frame(read.table("GMB1.txt", header = TRUE, sep = ""))

#C1 <- c("Выгодная")
ramFoT = data.frame(read.table("org.txt", header = TRUE, sep = ""))
ramFo2 <- cbind(ramFo2, ramFoT$V1)
#ramFo2

inputs <- normalizeData(ramFo2[,1:4], "norm")
model <- som(inputs, mapX=26, mapY=26, maxit=400,
calculateActMaps=TRUE, targets=ramFo2[,5])
par(mfrow=c(3,3))
for(i in 1:ncol(inputs)) plotActMap(model$componentMaps[[i]],
col=rev(topo.colors(12)))
plotActMap(model$map, col=rev(heat.colors(12)))
plotActMap(log(model$map+1), col=rev(heat.colors(12)))
persp(1:model$archParams$mapX, 1:model$archParams$mapY,
log(model$map+1),
theta = 30, phi = 30, expand = 0.5, col = "lightblue")
plotActMap(model$labeledMap)
model$componentMaps
model$labeledUnits
model$map
names(model)

```

j=3

```
for (i in 1:20000000) {
  j=j+1
}
```

```
#data(iris)
ramFo2
ramFo2[sample(1:nrow(ramFo2),length(1:nrow(ramFo2))),1:ncol(ramFo2)] <-
ramFo2Values <- ramFo2[,1:4]
ramFo2Targets <- decodeClassLabels(ramFo2[,5])
ramFo2 <- splitForTrainingAndTest(ramFo2Values, ramFo2Targets,
ratio=0.15)
ramFo2 <- normTrainingAndTestSet(ramFo2)
model <- rbfDDA(ramFo2$inputsTrain, ramFo2$targetsTrain)
summary(model)
#plotIterativeError(model)
model
```

```
par(mfrow=c(2,2))
```

```
weightMatrix(model)
extractNetInfo(model)
par(mfrow=c(2,2))
plotIterativeError(model)
```

```
predictions <- predict(model,ramFo2$inputsTest)
plotRegressionError(predictions[,2], ramFo2$targetsTest[,2])
```

```
confusionMatrix(ramFo2$targetsTrain,fitted.values(model))
confusionMatrix(ramFo2$targetsTest,predictions)
plotROC(fitted.values(model)[,2], ramFo2$targetsTrain[,2])
```

```
plotROC(predictions[,2], ramFo2$targetsTest[,2])
```

```
#confusion matrix with 402040-method
confusionMatrix(ramFo2$targetsTrain,
encodeClassLabels(fitted.values(model),
```

```
method="402040", l=0.4, h=0.6))
```

Сеть РБФ

```
ramFo2 <- ramFo2[sample(1:nrow(ramFo2),length(1:nrow(ramFo2))),1:ncol(ramFo2)]
ramFo2Values <- ramFo2[,1:4]
ramFo2Targets <- decodeClassLabels(ramFo2[,5])
ramFo2 <- splitForTrainingAndTest(ramFo2Values, ramFo2Targets,
ratio=0.15)
ramFo2 <- normTrainingAndTestSet(ramFo2)
model <- rbf(ramFo2$inputsTrain, ramFo2$targetsTrain, size=40,
maxit=1000,
initFuncParams=c(0, 1, 0, 0.01, 0.01),
learnFuncParams=c(1e-8, 0, 1e-8, 0.1, 0.8), linOut=TRUE)
summary(model)
#plotIterativeError(model)
model
par(mfrow=c(2,2))

weightMatrix(model)
extractNetInfo(model)
par(mfrow=c(2,2))
plotIterativeError(model)

predictions <- predict(model,ramFo2$inputsTest)
plotRegressionError(predictions[,2], ramFo2$targetsTest[,2])

confusionMatrix(ramFo2$targetsTrain,fitted.values(model))
confusionMatrix(ramFo2$targetsTest,predictions)
plotROC(fitted.values(model)[,2], ramFo2$targetsTrain[,2])

plotROC(predictions[,2], ramFo2$targetsTest[,2])

#confusion matrix with 402040-method
confusionMatrix(ramFo2$targetsTrain,
encodeClassLabels(fitted.values(model),
method="402040", l=0.4, h=0.6))
```

Задания для учащихся

Наименование рыбы	Калории	Жиры	Холестерин	Натрий	Калий	Белок	Коэффициент скорости
Рыба мечь	172	8	78	97	499	23	69
Тунец	184	6	49	50	323	30	68
Акула	130	4.5	51	79	160	21	70
лосось	208	13	55	59	363	20	58
скупия	262	18	75	83	401	24	59
треска	69	6.1	40	55	340	16	29
кефаль	88	2.2	53	72	468	23	28
пикша	90	0.6	66	261	351	20	37
палтус	186	14	46	80	268	14	55
камбала	70	1.9	45	296	160	12	44

Классификация рыб по скорости движения: 1. Очень быстроплавающие – коэффициент скорости от 61 до 70; 2. быстроплавающие – коэффициент скорости от 31 до 60; 3. Умеренно быстрые – коэффициент скорости от 21 до 30

Классификация рыб по пищевой ценности: нежирные, умеренно жирные и жирные (характеристические показатели: калории, жиры, холестерин, натрий, калий, белок)

Вариант 1

Разработать классификатор для идентификации сортов рыбы с точки зрения пищевой ценности и скорости движения в воде, основанный на обучении без учителя. Визуализировать результаты с помощью карт Кохонена для оценки распределения сортов рыбы в рамках выбранного водного пространства. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fc="bubble"`, `topo="rectangular"`

Зададим число кластеров $k=5$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "mapping", "changes", "property", "quality".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Рыбы: тунец, лосось, пикша, палтус, треска, кефаль необходимо разделить по категориям: жирные и нежирные, очень быстроплавающие, быстроплавающие и умеренно быстрые.

Вариант 2

Разработать классификатор для идентификации сортов рыбы с точки зрения пищевой ценности и скорости движения в воде, основанный на обучении без учителя. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена для оценки распределения сортов рыбы в рамках выбранного водного пространства. Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcn="gaussian"`, `topo="hexagonal"`

Зададим число кластеров $k=6$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "mapping", "quality", "property", "count".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Рыбы: акула, рыба - мечь, кефаль, треска, пикша, палтус, скумбрия необходимо разделить по категориям: умеренно жирные и нежирные, очень быстроплавающие, быстроплавающие и умеренно быстрые.

Вариант 3

Разработать классификатор для идентификации сортов рыбы с точки зрения пищевой ценности и скорости движения в воде, основанный на обучении без учителя. Визуализировать результаты с помощью карт Кохонена для оценки распределения сортов рыбы в рамках выбранного водного пространства. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcn="gaussian"`, `topo="rectangular"`

Зададим число кластеров $k=7$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "changes", "count", "dist.neighbours", "codes".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Рыбы: акула, тунец, кефаль, треска, пикша, палтус, камбала, лосось необходимо разделить по категориям : умеренно жирные и жирные, очень быстроплавающие, быстроплавающие и умеренно быстрые.

Вариант 4

Разработать классификатор для идентификации сортов рыбы с точки зрения пищевой ценности и скорости движения в воде, основанный на обучении без учителя. Визуализировать результаты с помощью карт Кохонена для оценки распределения сортов рыбы в рамках выбранного водного пространства. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcn="bubble"`, `topo="hexagonal"`

Зададим число кластеров $k=9$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "changes", "count", "dist.neighbours", "property".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Рыбы: акула, рыба - мечь, кефаль, треска, пикша, палтус, камбала, скумбрия необходимо разделить по категориям : умеренно жирные и жирные, очень быстроплавающие, быстроплавающие и умеренно быстрые.

Вариант 5

Разработать классификатор для идентификации сортов рыбы с точки зрения пищевой ценности и скорости движения в воде, основанный на обучении без учителя. Визуализировать результаты с помощью карт Кохонена для оценки распределения сортов рыбы в рамках выбранного водного пространства. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcn="bubble"`, `topo="hexagonal"`

Зададим число кластеров $k=7$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "dist.neighbours", "changes", "codes", "quality".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Рыбы: тунец, лосось, пикша, палтус, треска, кефаль необходимо разделить по категориям : жирные и умеренно жирные, очень быстроплавающие, быстроплавающие и умеренно быстрые.

Вариант 6

Необходимо составить карты предпочтений при выборе услуг обслуживания автомобилей в трех сервисных автоцентрах «Дмитровка», «На Калужской», «Мичуринский».

Вид услуги	«Дмитровка», время выполнения	«На Калужской», время	«Мичуринский», время	«Дмитровка», цена	«На Калужской», цена	«Мичуринский», цена
Замена сальника коленчатого вала	12	11.8	12.4	19200	19400	21000
Замена клапанной	4.6	5.2	4.2	7360	6900	7700

крышки						
Замер компрессионного давления	2	2.1	1.89	3200	3400	3000
Диагностика двигателя	1.14	1.27	0.89	1600	1630	1737
Диагностика ходовой части	0.87	1.2	1.23	1630	1609	1670
Проверка угла установки колес	0.55	0.48	0.69	860	790	900
Замена амортизаторов	3	3.5	3.2	6200	6400	6100
Замена масла в резервуаре заднего моста	1.5	1.3	1.4	2400	2500	2600
Замена сальника хвостовика редуктора заднего моста	2.5	2.6	2.8	4000	4200	4150
Ремонт коробки передач	5	4.9	5.2	8000	7800	7940

Анализировать виды услуг: с 1 по 5 из таблицы. Выбрать лучший автосервис с точки зрения цены на выполненные работы. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.
Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcn="bubble"`, `topo="hexagonal"`
Проверить результаты классификации с помощью функций RBF и rbfDDA.

Зададим число кластеров $k=9$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "changes", "count", "dist.neighbours", "property".

Вариант 7

Необходимо составить карты предпочтений при выборе услуг обслуживания автомобилей в трех сервисных автоцентрах «Дмитровка», «На Ка-лужской», «Мичуринский».

Анализировать виды услуг: с 6 по 10 из таблицы. Выбрать лучший авто-сервис с точки зрения скорости выполнения работ. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.
Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcn="gaussian"`, `topo="hexagonal"`
Зададим число кластеров $k=6$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "mapping", "quality", "property", "count".
Проверить результаты классификации с помощью функций RBF и rbfDDA.

Вариант 8

Необходимо составить карты предпочтений при выборе услуг обслуживания автомобилей в трех сервисных автоцентрах «Дмитровка», «На Ка-лужской», «Мичуринский».

Анализировать виды услуг: с 3 по 7 из таблицы. Выбрать лучший авто-сервис с точки зрения ценовых затрат на выполнение работ. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.
Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcn="bubble"`, `topo="rectangular"`
Зададим число кластеров $k=5$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "mapping", "changes", "property", "quality".
Проверить результаты классификации с помощью функций RBF и rbfDDA.

\.

Вариант 9

Необходимо составить карты предпочтений при выборе услуг обслуживания автомобилей в трех сервисных автоцентрах «Дмитровка», «На Калужской», «Мичуринский».

Анализировать виды услуг: с 2, 4, 6, 8, 10 из таблицы. Выбрать лучший автосервис с точки зрения двух показателей скорость и цена на выполнение работ. Полный список из 35 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.
Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcn="bubble"`, `topo="hexagonal"`
Зададим число кластеров $k=10$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "changes", "count", "dist.neighbours", "property".
Проверить результаты классификации с помощью функций RBF и rbfDDA.

Вариант 10

Наименование неисправности	Компрессия Двигателя с открытой	Компрессия Двигателя с закрытой заслон-	Примечание
----------------------------	---------------------------------	---	------------

	заслонкой, мПА	кой, мПА	
Двигатель исправен	1 -1.2	0.6 -0.8	
Трещина в перемычке поршня	0.6 – 0.8	03. – 0.4	
Прогар поршня	0.5	0.1	
Залегание колец в канавках поршня	0.2 – 0.4	0.2	
Задир поршня	0.2 – 0.8	0.15 – 0.5	
Задир цилиндра	0.3 -0.8	0.4 -0.5	
Деформация клапана	0.3 – 0.7	0.2	
Прогар клапана	0.1 -0.4	0.01	
Дефект профиля кулачка распределвала	0.7 – 0.8	0.1 – 0.3	
Сильный нагар в камере сгорания	1.2 – 1.5	0.9 – 1.2	
Износ деталей поршневой группы	0.6 - 0.9	0.4 – 0.6	
Зависание клапана	0.4 – 0.8	0.2 - 0.4	

Необходимо идентифицировать неисправности двигателя автомобиля (задир поршня, Двигатель исправен, Зависание клапана, прогар поршня, деформация клапана, задир цилиндра) по двум показателям. Полный список из 40 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.

Зададим число кластеров $k=12$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "dist.neighbours", "changes", "codes", "quality".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Вариант 11

Необходимо идентифицировать неисправности двигателя автомобиля (Трещина в перемычке поршня, Двигатель исправен, Зависание клапана, прогар поршня, износ деталей поршневой группы, сильный нагар в камере сгорания, дефект профиля кулачка распредвала)по двум показателям. Полный список из 37 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcf="bubble"`, `topo="rectangular"`

Зададим число кластеров $k=4$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "mapping", "changes", "property", "quality".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Вариант 12

Наименование показателя низового пожара	Слабый низовой	Средний низовой	Сильный низовой
Скорость распространения огня	До 1	1-3	Более 3
Высота пламени	До 0.5 м	0.5 – 1.5	Более 1.5

Наименование показателя верхового пожара	Слабый верховой	Средний верховой	Сильный верховой
Скорость распространения огня	До 3	3 - 100	Более 100

Высота пламени	До 2 м	2 – 4	Более 4 м
----------------	--------	-------	-----------

Наименование показателя подземного пожара	Слабый подземный	Средний подземный	Сильный подземный
Скорость распространения огня	До 0.65	0.65 - 2	Более 2
Высота пламени	До 0.55 м	0.55 – 1.2	Более 1.2 м

Необходимо идентифицировать вид низового пожара на выбранной лесистой местности (слабый, средний, сильный) по двум показателям: скорость огня и высота пламени. Полный список из 38 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fct="bubble"`, `topo="hexagonal"`

Зададим число кластеров $k=7$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "dist.neighbours", "changes", "codes", "quality".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Вариант 13

Необходимо идентифицировать вид верхового пожара на выбранной лесистой местности (слабый, средний, сильный) по двум показателям: скорость огня и высота пламени. Полный список из 42 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fct="gaussian"`, `topo="hexagonal"`

Зададим число кластеров $k=16$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "mapping", "quality", "property", "count".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Вариант 14

Необходимо идентифицировать вид подземного пожара на выбранной лесистой местности (слабый, средний, сильный) по двум показателям: скорость огня и высота пламени. Полный список из 33 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID fct="gaussian", topo="rectangular"

Зададим число кластеров k=10 и выполним иерархическую кластеризацию (по умолчанию используются method = "complete" и distance = "euclidean"). Построить карты "changes", "mapping", "dist.neighbours", "codes".

Проверить результаты классификации с помощью функций RBF и rbfDDA.

Вариант 15

Характеристика ветра	Значение показателя	Горизонтальный Барический градиент атмосферного давления	Вертикальный Барический градиент атмосферного давления
Слабый	0.5	1.1	1.24
Умеренный	6-14	1.65	1.54
Сильный	15-24	2.33	2.79
Очень сильный	25-32	2.64	2.89
Ураганный	Более 33	2.92	2.97

Показатель облачности	Значение
Малая	До 3 баллов

Переменная	4-7
С прояснениями	7-8
Облачно	8-10

Показатель насыщенности (плотности) дождя	Значение	Показатель продолжительности дождя	Значение
Отсутствует	Менее 0.2	Кратковременный	Менее 3 ч
Моросящий	0.3 – 10	Временный	От 3 до 6 ч
Сильный	11 – 49	Продолжительный	Более 6 ч

Необходимо разработать кратковременный прогноз погоды – наличие или отсутствие дождя на пять дней (дождь может быть временным и сильным, морозящим и продолжительным и т. Д.). Показатели прогноза погоды: облачность, сила ветра, барические градиенты горизонтальный и вертикальный. Полный список из 44 пунктов можно найти в приложении к лабораторной работе.

Визуализировать результаты с помощью карт Кохонена.

Использовать средства языка R – функции SOM и SOMGRID. Параметры для функции SOM GRID `fcf="bubble", topo="rectangular"`

Зададим число кластеров $k=9$ и выполним иерархическую кластеризацию (по умолчанию используются `method = "complete"` и `distance = "euclidean"`). Построить карты "quality", "count", "dist.neighbours", "property".

Проверить результаты классификации с помощью функций RBF и rbfDDA.