Тема: Парзеновское окно. Балицкая Анастасия 401-И

**1. алгоритм классификации «Парзеновское окно»**

Реализовали его:

a **=** **function(**u, xl, h, K, ro**)** **{**

l **=** dim**(**xl**)[**1**]**

n **=** dim**(**xl**)[**2**]-**1

facts **=** levels**(**xl**[**,n**+**1**])**

s **=** rep**(**0, times**=**length**(**facts**))**

**for** **(**i **in** 1**:**l**)** **{**

fact **=** xl**[**i,n**+**1**]**

score **=** K**(**ro**(**u, xl**[**i,1**:**n**])/**h**)**

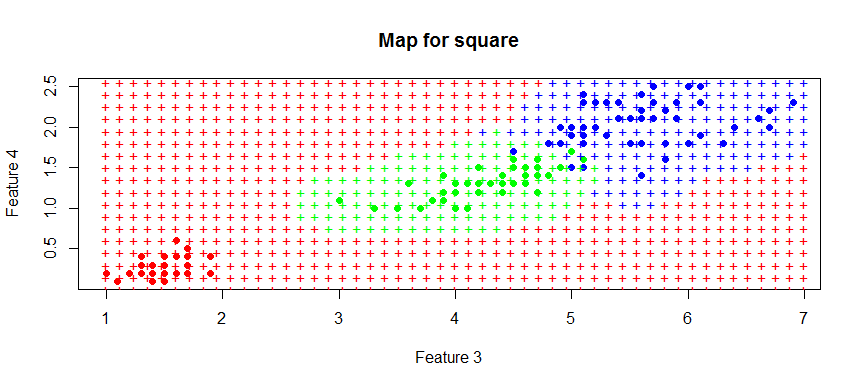
s**[**fact**]** **=** s**[**fact**]** **+** score

**}**

factor**(**facts**[**which.max**(**s**)]**, levels**=**facts**)**

**}**

но тут возникает такая проблема(долго с ней разбирались):



Она возникла потому, что примеру брали точку (1000,1000) каждый класс получал очки (0,0,0) и алгоритм просто брал первый 0, поэтому в таком случае не будем классифицировать:

**if** **(**sum**(**s**)** **==** 0**)** **{**

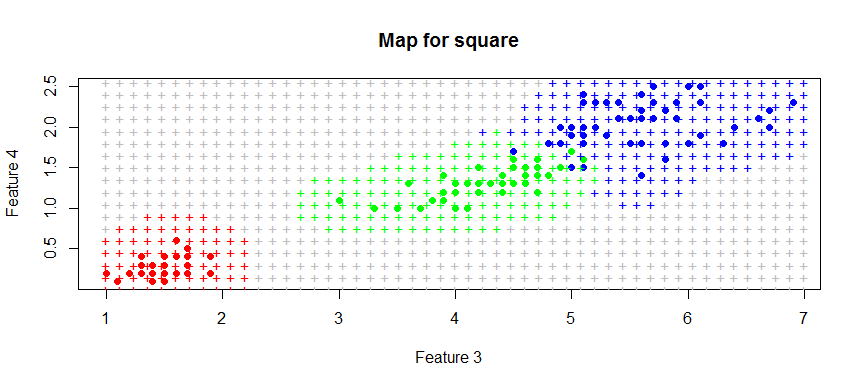
**-**1

**}** **else** **{**

factor**(**facts**[**which.max**(**s**)]**, levels**=**facts**)**

**}**

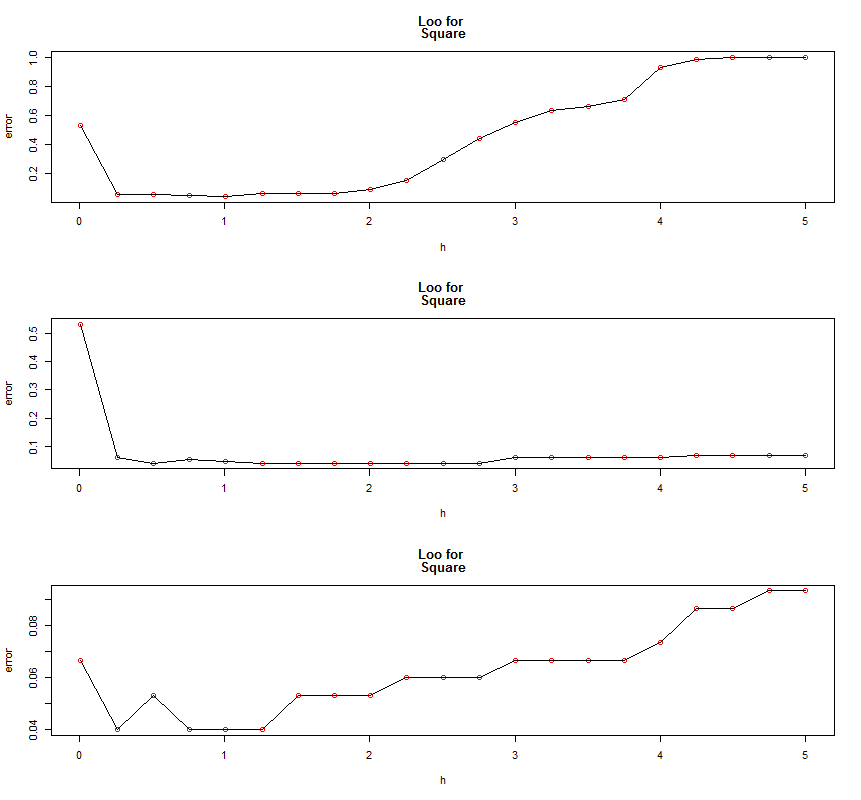
Тогда получим:



**2. подобираем параметр «ширина окна»**

Надо выбрать такое h, которое даст меньше всего ошибок. Для этого воспользуемся LOO:

Построим LOO(h) для 3х ядер: прямоугольное, треугольное и гауссовское:



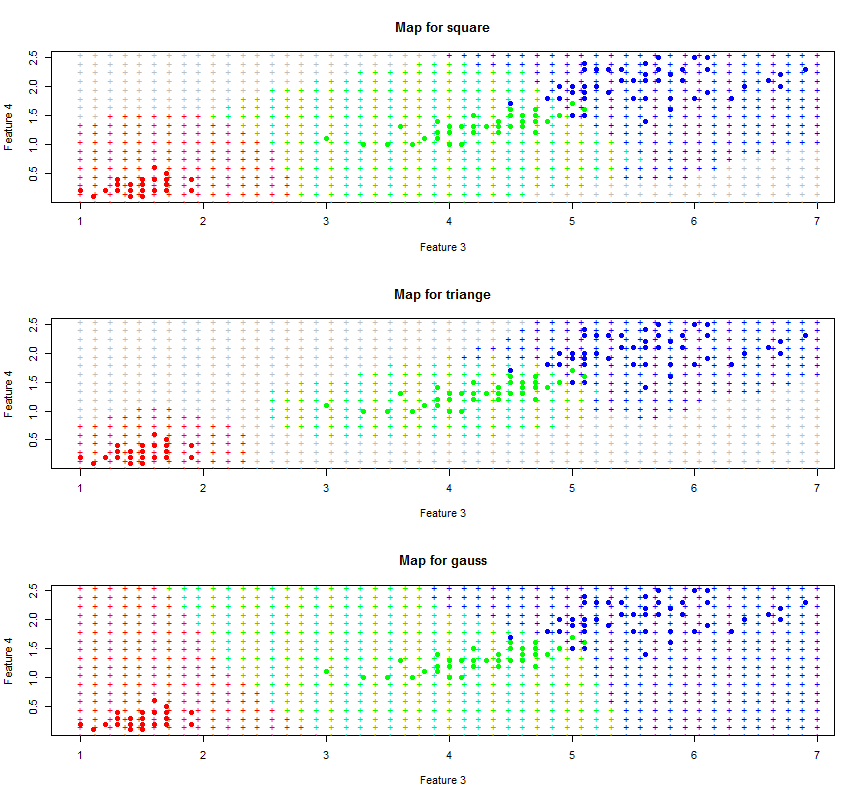
**3. попробуем алгоритм на разных ядрах, сравниваем их, строим карты классификации**

Результатами LOO оказались:

|  |  |  |
| --- | --- | --- |
| **Ядро** | **h** | **LOO(h)** |
| Прямоугольное | 1,008 | 0,04 |
| Треугольное | 0,509 | 0,04 |
| Гауссовское | 0,5333 | 0,04 |

Ошибка оказалась у всех ядер одинаковая, не знаю почему.

Посмотрим на карты классификации для разных ядер:



Вывод: Больше всего подходи гауссовское ядро. Оно однозначно разделило классы на всей плоскости. Поставленные задачи выполнены.