# Руководство по использованию программы «Классификация примеров на плоскости»

## 1. Основные сведения

### 1.1 Системные требования

Matlab R2014a и выше

### 1.2 Запуск программы

Для запуска программы необходимо перейти в Matlab папку с файлами pattern_reco_lab1_part3_par_est_examples.m и SetDataVarians.m и запустить первый файл. Если добавить эту папку в список папок для поиска (File->Set Path), то можно осуществлять запуск через командную строку командой:

>> pattern_reco_lab1_part3_par_est_examples

### 1.3 Файлы программы

**pattern_reco_lab1_part3_par_est_examples.m** – основной файл, содержащий элементы интерфейса и реализацию алгоритмов классификации;

**SetDataVarians.m** – вспомогательный файл, в котором задаются распределения классов на плоскости, используемые программой;

**mlp_kernel_mine.m** – файл для использования в SVM ядер типа MLP.

### 1.4 Отображение данных

Данные классов отображаются различными цветами:

1 класс – красный цвет;
2 класс – зеленый цвет;
3 класс – синий цвет;
4 класс – желтый цвет;
5 класс – розовый цвет (magenta);
6 класс – цвет морской волны (cyan);
7 класс – оранжевый цвет;
8 класс – цвет занаду (зелено-коричневый);
9 класс – красновато-коричневый цвет.

# 2. Интерфейс программы

## 2.1 Основное меню программы

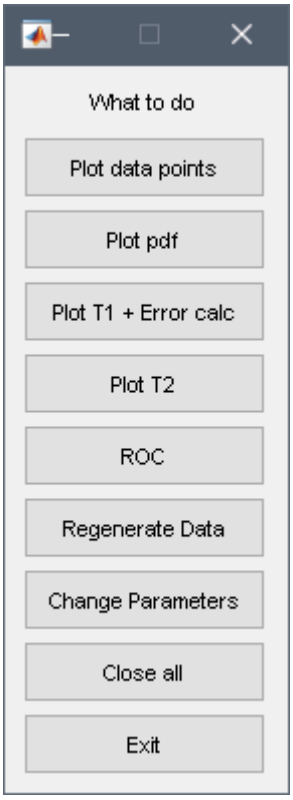



Рисунок 1 Основное меню выбора действий программы

Таблица 1

Описание пунктов основного меню программы

| **Пункт меню (название)** | **Описание** |
|---|---|
| Plot Data Points | Отображение исходных точек (тестовая или обучающая выборки) плюс граничные уровни истинной плотности и ее оценки |
| Plot pdf | Отображение плотности распределения вероятности – истинной, оценки (estimation) и ошибки оценивания |
| Plot T1 + Error calc | Построение графика типа T1 и расчет всех ошибок. На графике показаны результаты классификации примеров из тестовой выборки (каждый пример относится к одному из исходных классов) |
| Plot T2 | Построение графика типа T2 – диаграмма разбиения плоскости на классы классификатором (примеры взяты из прямоугольной области без привязки к классам) |
| ROC | Построение ROC-кривой для выбора порога у бинарного классификатора по одному из критериев (Байеса, Неймана-Пирсона, минимаксного) |
| Regenerate data | Повторная генерация исходных данных |
| Change parameters | Открытие диалогового окна с изменением параметров |
| Close all | Закрытие всех окон с графиками |
| Exit | Выход из программы |

При нажатии на крест в правом верхнем углу окна программа также закрывается.

## 2.2 Отображение исходных данных

При выборе пункта основного меню «Plot Data Points» появляется новое меню для уточнения параметров отображения исходных данных.

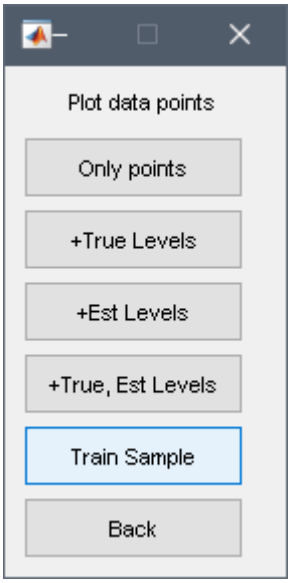



Рисунок 2 Меню выбора параметров отображения исходных данных

Таблица 2

Описание пунктов меню «Plot Data Points»

| **Пункт меню (название)** | **Описание** |
|---|---|
| Only points | Отображение только тестовой выборки (каждый пример каждого класса отображается символом, причем цвет и тип символа зависит от номера класса |
| +True Levels | В дополнении к тестовой выборке строятся уровни (границы) исходных (истинных) распределений классов – для нормального распределения и GMM – уровня 3σ, для равномерного распределения – границы классов |
| +Est Levels | В дополнении к тестовой выборке строятся уровни (границы) оцениваемых распределений классов |
| +True, Est Levels' | В дополнении к тестовой выборке строятся уровни (границы) исходных (истинных) и оцениваемых распределений классов |
| 'Train Sample | Отображение обучающей выборки |
| Back | Переход назад в основное меню |

При выборе любого пункта меню кроме «Back» создается новое окно и в нем отображается график выбранного типа.

## 2.3 Построение плотности распределения

При выборе пункта основного меню «Plot pdf» появляется новое меню для уточнения параметров построения плотности (рис. 3, слева). При выборе одного из пунктов нового меню появляется еще одно меню выбора класса, для которого строить плотность. Последнее меню может быть двух видов (рис. 3, в центре и справа).

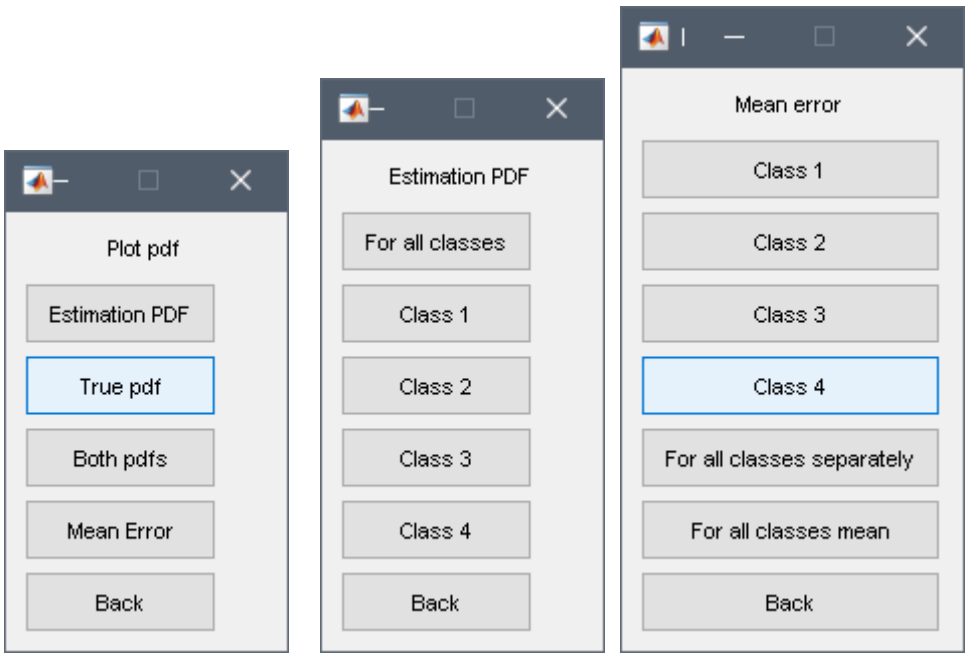



Рисунок 3 Меню выбора параметров построения плотности

Таблица 3

Описание пунктов меню «Plot pdf»

| **Пункт меню (название)** | **Описание** |
|---|---|
| Estimation PDF | Отображение оценки плотности распределения классов |
| True pdf | Отображение исходной (истинной) плотности распределения классов |
| Both pdfs | Отображение исходной (истинной) и оцениваемой плотности распределения классов |
| Mean Error | Отображение ошибки (средней разности между истинной и оцениваемой плотностями) |
| Back | Переход назад в основное меню |

На рисунках 4 и 5 показаны примеры графиков плотностей.

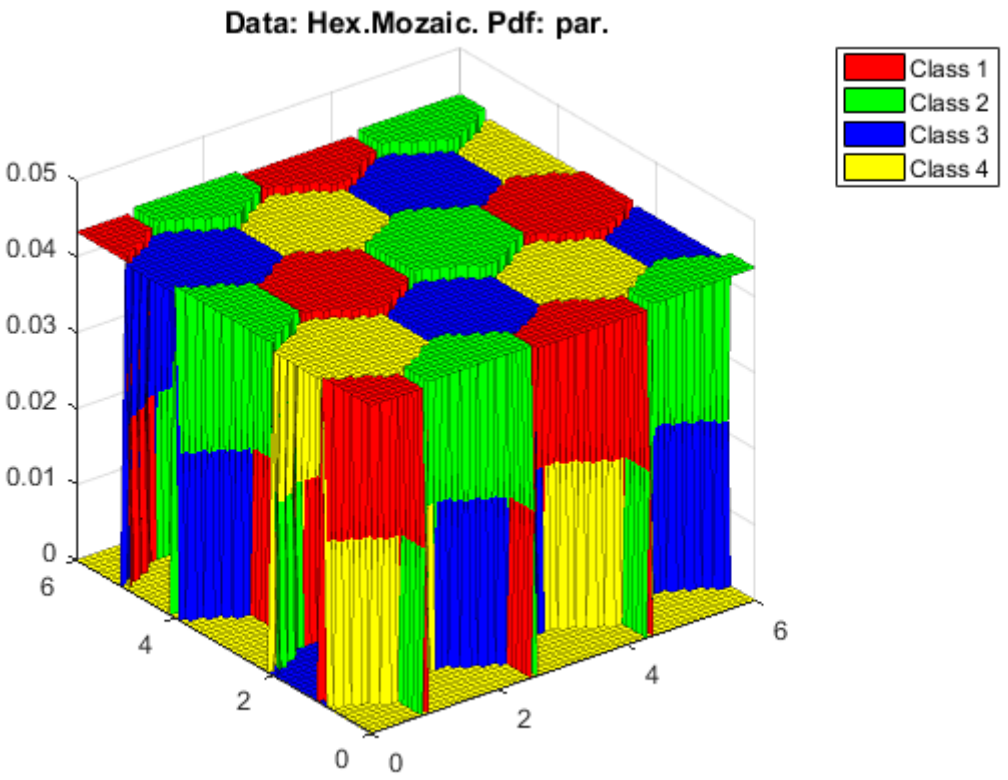



Рисунок 4 Пример отображения истинной плотности

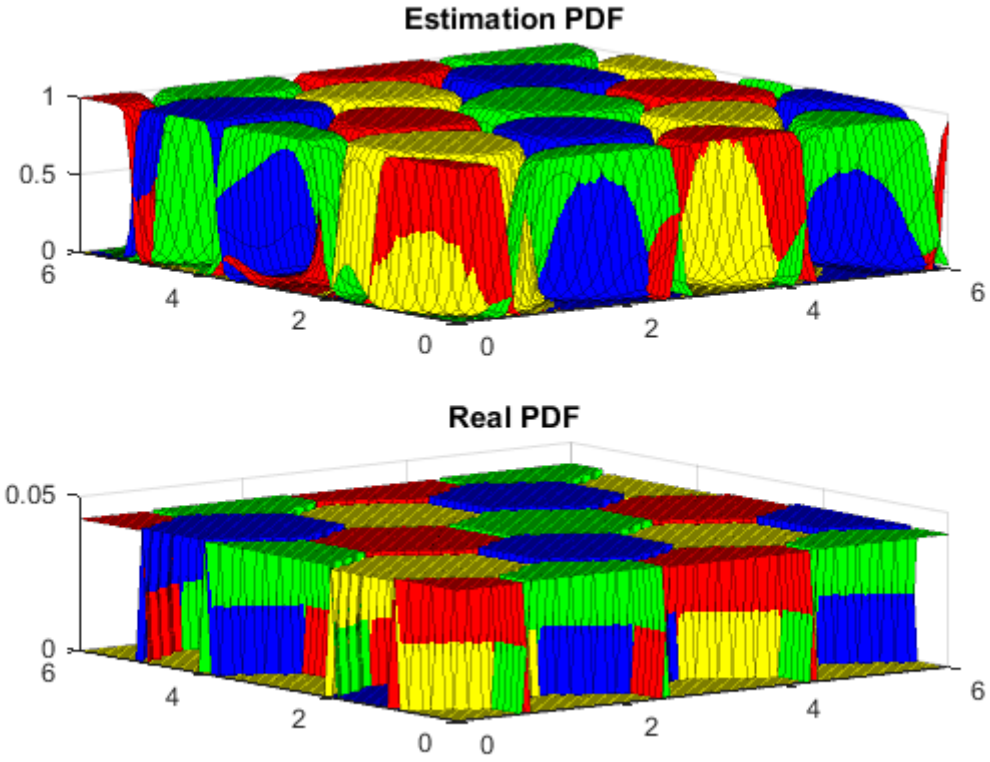



Рисунок 5 Пример отображения оцениваемой и истинной плотности на одном графике

После выбора пунктов Estimation PDF, True Pdf, Both Pdfs появляется еще одно меню (рис. 3, в центре), в котором можно выбрать, для какого класса выполнять построение.

Таблица 4

Описание пунктов меню выбора классов для построения ошибки плотности

| **Пункт меню (название)** | **Описание** |
|---|---|
| For All classes | Для всех классов |
| Class # | Только для класса с заданным номером |
| Back | Переход назад в основное меню |

После выбора пунктов Mean Error появляется еще одно меню (рис. 3, справа), в котором можно выбрать, для какого класса выполнять построение.

Таблица 5

Описание пунктов меню выбора классов для отображения ошибки плотности

| **Пункт меню (название)** | **Описание** |
|---|---|
| Class # | Построение ошибки только для класса с заданным номером |
| For All classes separately | Построение для каждого класса отдельно своим цветом ошибки |
| For All classes mean | Построение для всех классов средней ошибки |
| Back | Переход назад в основное меню |

Зам. 1 Плотность оценки действительно является плотностью только при использовании в качестве классификатора Байесовского классификатора с истинной плотностью, параметрического оценивания и ядерного оценивания плотностей. Во всех остальных случаях (kNN, нейронные сети, SVM, деревья решений) оцениваемая плотность на самом деле плотностью не является (интеграл от нее не равен 1). В этих случаях отображаются т.н. дискриминантные функции, поэтому сравнивать их с истинными плотностями следует лишь качественно, а не количественно.

Зам. 2 Графики плотностей содержат много точек и при их сохранении и копировании в векторном формате (emf) может потребоваться большой объем памяти. Поэтому следует:

- полезные и важные графики сохранять в векторном формате на жесткий диск в векторном формате (emf);
- сохранять все графики в растровом формате (png) для последующей вставки в отчет.

## 2.4 Построение графика типа T1

После выбора из основного меню пункта "Plot T1 + Error Calc" для тестовой выборки, составленной из примеров каждого из классов (поровну).

- решается задача классификации
- рассчитываются и выводятся в командное окно показатели качества распознавания:
    - средняя ошибка (mean_error)
    - ошибки 1 рода (p_e_1);
    - ошибки 2 рода (p_e_2);
    - матрица ошибок (confmatr);
- создается окно и в нем отображается график с результатами классификации (рис. 6).

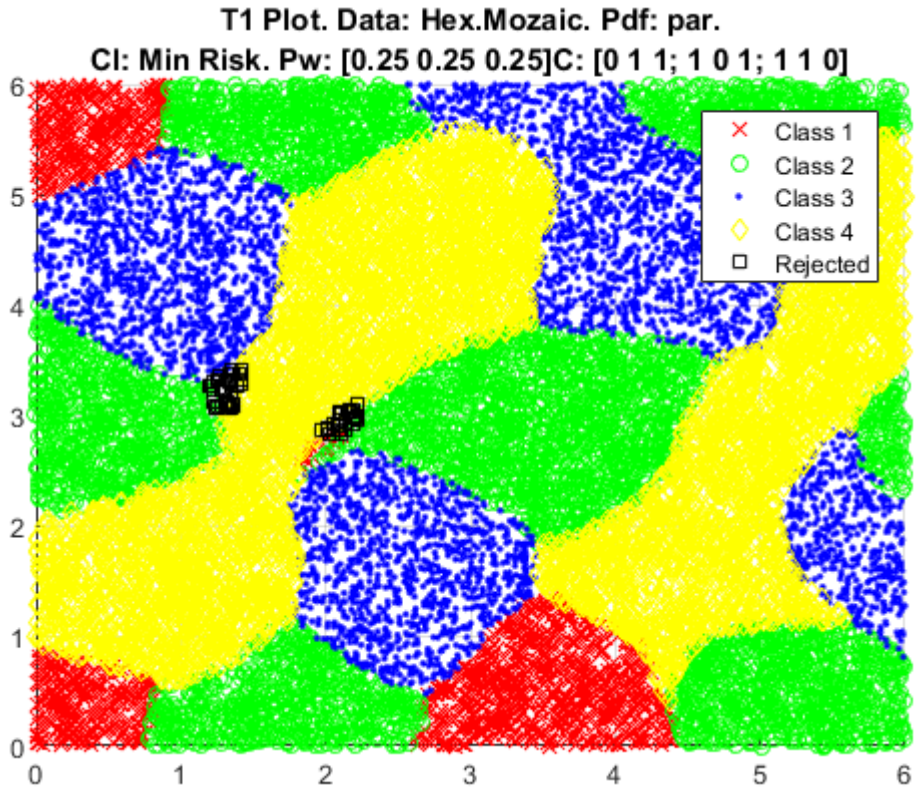



Рисунок 6 Пример графика типа T1

На графике символами различного цвета и типа отображаются метки классов, к которым классификатор присвоил соответствующие примеры. Примеры, не распознанные как один из заданных классов, помечаются черными прямоугольниками (в легенде помечаются как Rejected).

При использовании в качестве классификатора машин опорных векторов дополнительно отображаются опорные вектора с помощью специальных символов.

### 2.5 Построение графика типа T2

После выбора из основного меню пункта "Plot T2" открывается еще одно меню (рис. 7).

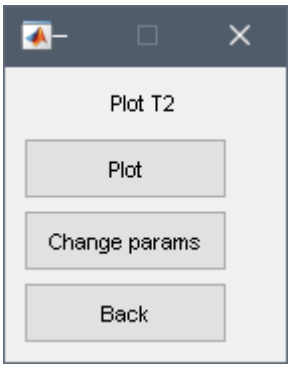



Рисунок 7 Меню при выборе пункта "Plot T2"

При нажатии кнопки Plot строится разбиение плоскости на классы (рис. 9). Диапазон значений задается в файле **SetDataVarians.m** параметром axis_data{i}, где i – номер выбранного распределения исходных данных. Шаг по умолчанию по каждой оси равен 0.05.

При нажатии кнопки Change Params появляется окно (рис. 8) с возможностью изменить шаг по умолчанию:

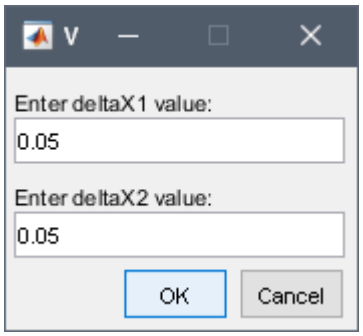



Рисунок 8 Окно для изменения шага построения графика T2

В поле "Enter deltaX1 Value" задается шаг по оси X, в поле "Enter deltaX2 Value" задается шаг по оси Y. При нажатии OK введенные данные сохраняются, при нажатии Cancel – нет. При введении в поля ввода некорректных данных они сбрасываются и устанавливаются последние корректно введенные значения.

На рис. 9 приведен пример графика типа T2. Плоскость раскрашивается в цвета класса, к которому классификатор относит соответствующие области значений. При построении выполняется интерполяция, т.е. если соседние точки области распознаются как один класс, то и промежуточные значения между этими точками относятся к этому же классу и раскрашиваются в тот же цвет. Области примеры в которых не распознаются как один из заданных классов, помечаются серым цветом (в легенде помечаются как Rejected).

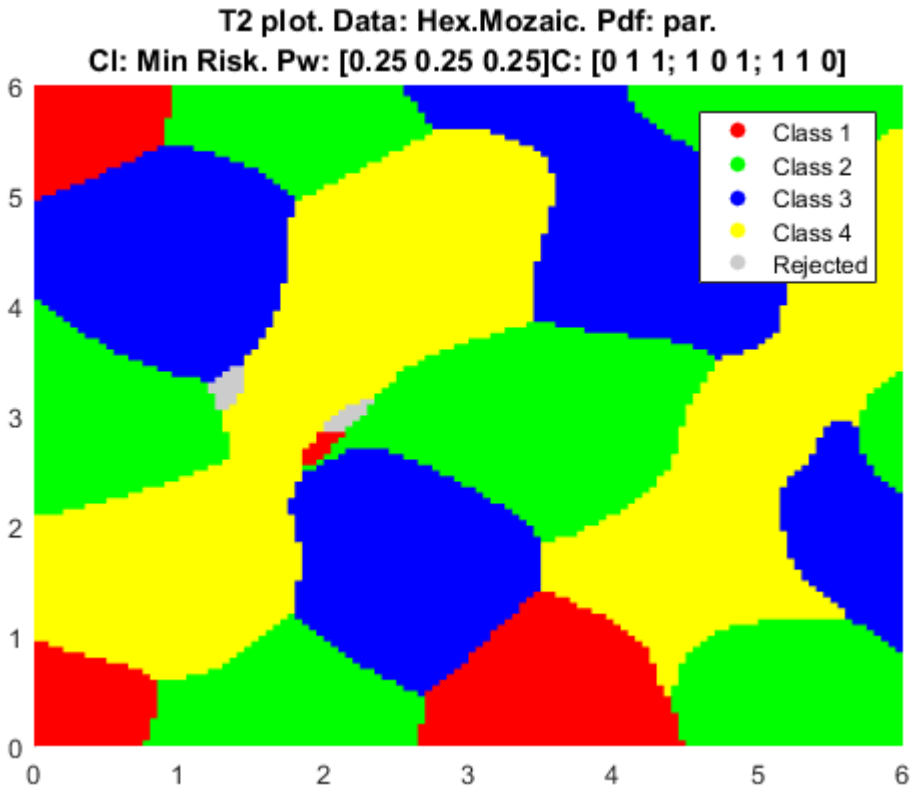


Рисунок 9 Пример графика типа T2

## 2.6 Построение ROC-кривой

После выбора из основного меню пункта "ROC" для тестовой выборки, составленной из примеров выбранных двух классов:

- решается задача классификации, причем классификатор по возможности дает т.н. мягкий (soft) ответ в форме вероятности (дискриминанта) принадлежности входного примера к каждому из классов

- создается окно и в верхней его части строится ROC-кривая, в которой по оси X откладывается ошибка первого рода (ложное срабатывание, false positive rate), а по оси Y вероятность правильного распознавания "положительного" класса (чувствительность, true positive rate);

- в нижней части окна строится еще один график – значения порога отношений правдоподобий (Threshold), обеспечивающие заданные значения ошибки 1 рода.

На рисунках 10 и 11 показаны примеры ROC-кривых.

Пересечение ROC-кривой и прямой красного цвета дает значение, при котором ошибки 1 и 2 рода равны (критерий минимакс). Соответствующая точка помечена красным кружком на обоих графиках.

Зеленым кружком отмечена точка для случая заданной вероятности ложного срабатывания (по умолчанию 5 %). Это соответствует критерию Пирсона.

Наконец, черным цветом отмечена точка, соответствующая критерию Байеса. Она имеет следующую особенность: если на ROC-кривой провести касательную, то коэффициент прямой совпадает с порогом отношения правдоподобий. Соответствующая прямая черным цветом отображается на графике ROC-кривой. Критерий Байеса обеспечивает минимальный риск. В случае ML-классификатора получается, что коэффициент наклона равен 1, что соответствует углу 45 градусов.

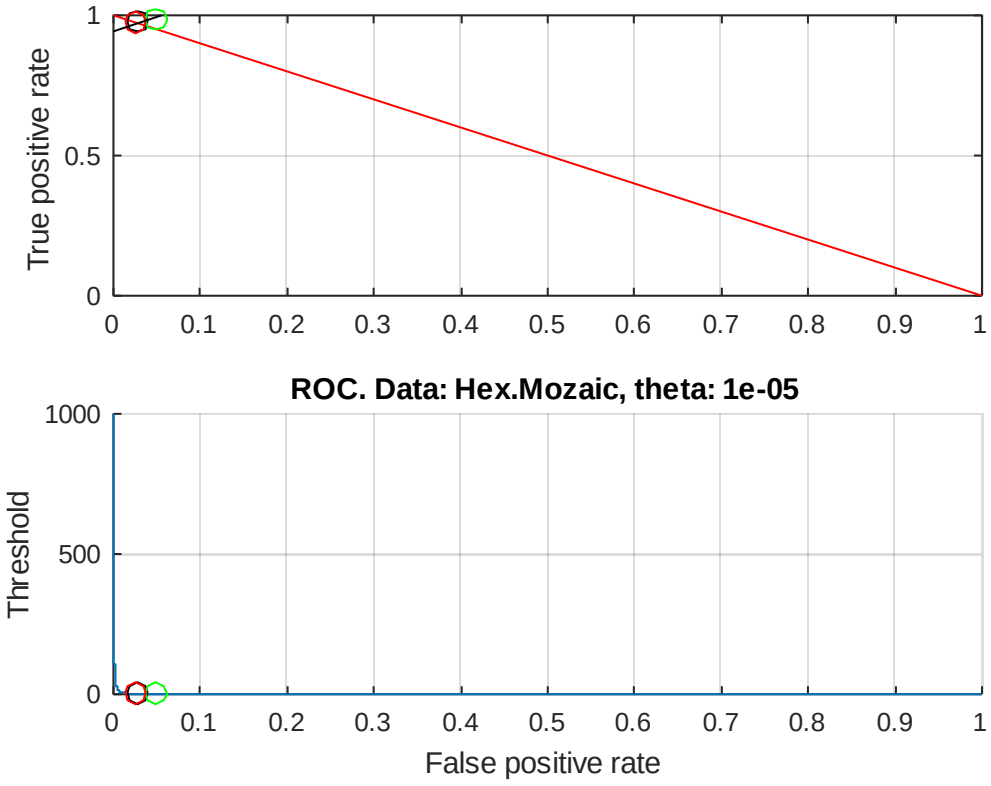



Рисунок 10 Пример графика с ROC-кривой

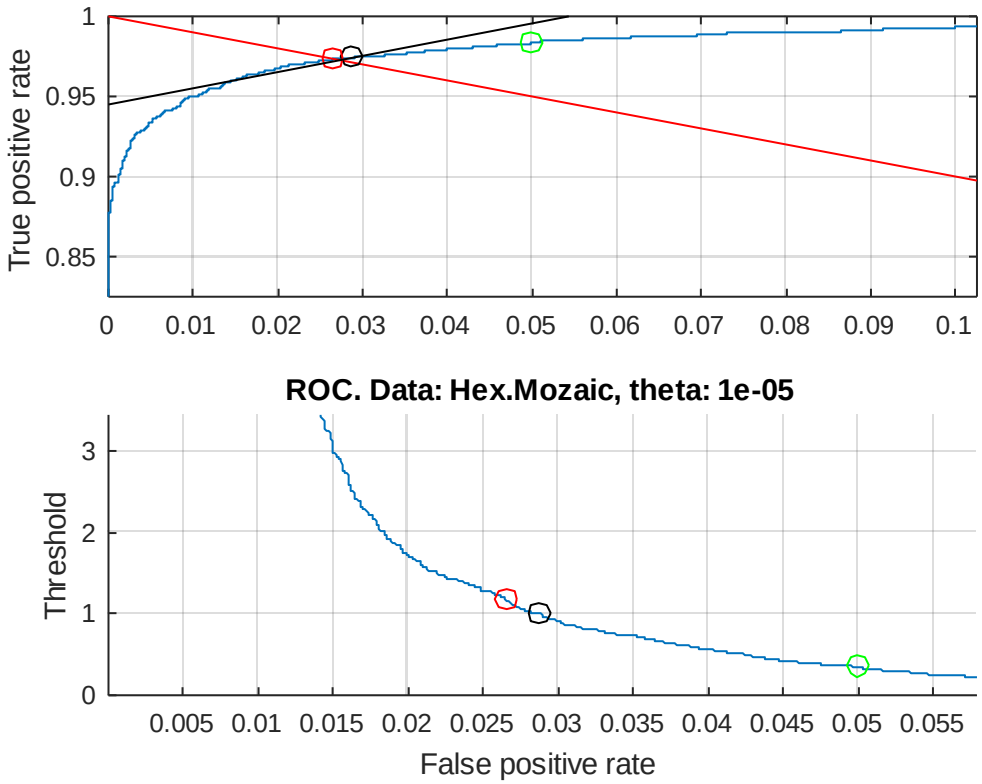



Рисунок 11 Пример графика с ROC-кривой (укрупненно)

## 2.7 Изменение основных параметров программы

После выбора из основного меню пункта "Change parameters" открывается диалоговое окно (рис. 12) для изменения основных параметров программы.

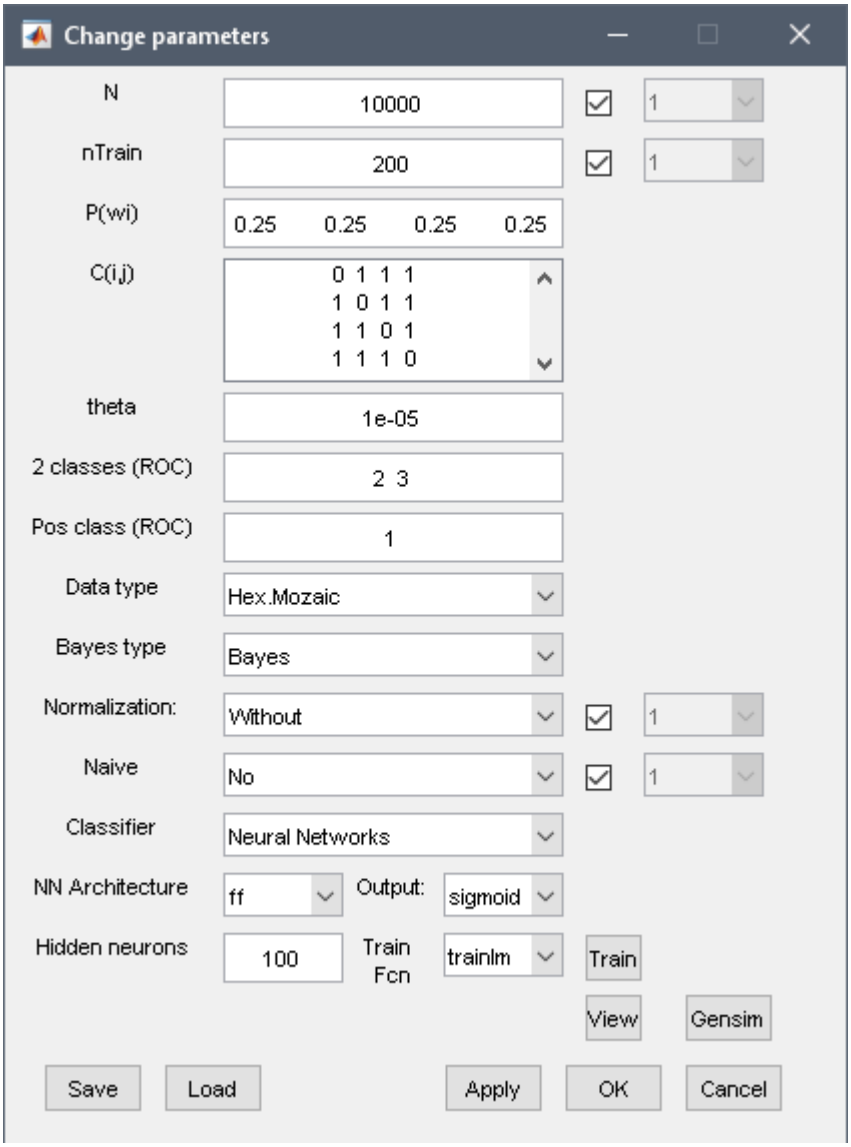



Рисунок 12 Диалоговое окно изменения параметров программы

Прежде чем переходить к описанию параметров, следует сказать, что для некоторых параметров в левой части окна используются флаг (checkbox, по умолчанию активен) и список выбора (по умолчанию не активен). Данная опция позволяет задать значение параметра либо одинаковым для всех классов, либо разным. Если флаг активен, то параметры для всех классов одинаковые и задаются соответствующим значением (par_all). Если флаг не активен, то активизируется список выбора, в котором можно выбрать класс и задать именно для этого класса то или иное значение параметра par_class(i), где i – номер класса. Общие (par_all) и частные (par_class) значения параметров не пересекаются и хранятся отдельно, что обеспечивает гибкость при выполнении различных экспериментов. В дальнейшем параметры, у которых есть такая возможность, в таблице будут отмечены символов + в графе "Задание для каждого класса".

Диалоговое окно сделано не модальным, поэтому имеется возможность параллельно изменять параметры и выполнять исследования. Следует только помнить, что все изменения параметров применяются только после нажатия кнопок OK или Apply. До нажатия этих кнопок при исследовании будут использоваться предыдущие параметры.

Таблица 6

Описание основных параметров диалогового окна Set Parameters

| **Пункт меню (название)** | **Описание** | Тип | Задание для каждого класса |
|---|---|---|---|
| N | Объем тестовой выборки для каждого класса | Целое число | + |
| nTrain | Объем обучающей выборки для каждого класса | Целое число | + |
| P(wi) | Априорные вероятности | Массив Cx1, $p_i \geq 0$ | - |
| C(i,j) | Матрица стоимости для обобщенного Байесового классификатора | Матрица CxC, $c_{ij} \geq 0$ | - |
| theta | Порог срабатывания классификатора – если для всех классов ответ классификатора меньше порога, то пример отклоняется | $\geq 0$ | - |
| 2 classes (ROC) | Номера классов, для которых строится ROC-кривая | 2 целых числа от 1 до C | - |
| Pos class (ROC) | Номер положительного класса для построения ROC-кривой среди классов, заданных в "2 classes (ROC)" | 1 или 2 | - |
| Data Type | Тип данных | Список | - |
| Bayes Type | Тип Байесовского классификатора – ML, MAP, Bayes | Список | - |
| Normalization | Есть ли предобработка входных данных – Without, Norm Variances, Whitening | Список | + |
| Naïve | Используется ли гипотеза о независимости признаков | Список | + |
| Classifier | Тип классификатора | Список | - |
| | | | |
| Save | Сохранение текущих настроек в файл | Кнопка | - |
| Load | Чтение настроек из файла | Кнопка | - |
| Apply | Применение текущих настроек | Кнопка | - |
| OK | Применение текущих настроек и закрытие окна | Кнопка | - |
| Cancel | Сброс несохраненных изменений и закрытие окна | Кнопка | - |

Далее немного рассмотрим пункты во всплывающих списках.

### Тип данных

**Тип данных (Data Type)** по умолчанию может быть одним из следующих:

- Normal – 3 класса с нормальным распределением;
- GMM – 3 класса с распределением типа гауссова смесь;
- Uniform – 3 класса с равномерным распределением;
- Norm+GMM+Uniform – 3 класса, 1 – с нормальным, 2 – с распределением типа гауссова смесь, 3 – с равномерным распределением;
- Conc Circles – 3 класса с равномерным распределением в форме концентрических окружностей;

- Hor. stripes – 3 класса с равномерным распределением в форме горизонтальных полос;
- Ver. stripes – 3 класса с равномерным распределением в форме вертикальных полос;
- Squares – 4 класса с равномерным распределением в форме прямоугольников;
- Circles+Lines – 9 классов с равномерным распределением в форме окружности в центре и 8 обрезанных этой окружностью треугольников внутри прямоугольной области;
- Spiral – 3 класса, имеющих случайное распределение вокруг спиралей;
- Hex Mozaic – 4 класса в форме гексагональной мозайки.

### Тип Байесовского классификатора

**Тип Байесовского классификатора (Bayes Type )** может быть одним из следующих:
- ML – максимального правдоподобия – max P(X/wi);
- MAP – максимальной апостериорной вероятности – max(P(wi/X)) или max(P(wi)*P(X/wi)) – т.е. с учетом априорных вероятностей P(wi);
- Bayes – обобщенный Байесовский классификатор минимального риска, где риск по принятию каждого решения задается в матрице C(i,j).

### Матрица стоимости

**Матрица стоимости C(i,j)** имеет размерность [число классов x число классов]. Элемент C(i,j) матрицы характеризует стоимость классификации класса i как класс j. По умолчанию матрица устанавливается в форме, когда диагональные элементы – нулевые, а внедиагональные – равны 1.

### Предобработка данных

Данные могут быть предобработаны (**Normalization**) следующим образом:
- Without – в исходной форме;
- Norm Variances – выравниваются дисперсии по каждому признаку;
- Whitening – т.н. отбеливание данных, когда в дополнение к выравниванию дисперсий осуществляется декорелляция, что в итоге приводит к единичной ковариационной матрице.

Данная настройка работает только для случая, когда в качестве классификатора выбран тип Parametric Estimation или Kernel Density Estimation.

### Гипотеза о независимости признаков

Гипотеза о независимости признаков (**Naïve**) дает две опции:
- No – признаки считаются зависимыми, оценка плотности осуществляется в 2-мерном пространстве признаков;
- Yes – признаки считаются независимыми, поэтому для каждого признака создается оценка плотности распределения отдельно, а затем они перемножаются.

Данная настройка работает только для случая, когда в качестве классификатора выбран тип Parametric Estimation или Kernel Density Estimation.

### Тип классификатора

В программе реализована возможность использования различных методов оценки плотности и последующей классификации. Для выбора того или иного алгоритма используется всплывающий список Classifier. Рассмотрим элементы этого списка:
- Bayessian Classifier – классификатор с использованием истинных (исходных) плотностей распределения;
- Parametric Estimation – классификатор с параметрической оценкой плотностей распределения (2 варианта – нормальное распределение и гауссовы смеси);
- Kernel Density Estimation – классификатор с оценкой плотностей при помощи ядер;
- k Nearest Neigbors – классификатор *k* ближайших соседей;
- Support Vector Machines – машины опорных векторов;
- Neural Networks – нейронные сети прямого распространения;
- Decision Tree – деревья принятия решений.

При выборе того или иного типа классификатора ниже появляются элементы управления, позволяющие изменить его параметры. Рассмотрим их более подробно ниже.

У Байесовского классификатора (**Bayessian Classifier**), использующего истинную плотность распределения, нет никаких дополнительных настроек (рис. 13). Данный классификатор в стандартном варианте (ML, MAP) по определению дает минимально возможную ошибку.


Рисунок 13 Настройки Байесовского классификатора

*Параметрическое оценивание*

Классификатор с параметрической оценкой (рис. 14) использует гипотезу о типе распределения, который можно выбрать во всплывающем списке (**Par distribution**). Возможны два варианта:

- Normal – нормальное распределение, параметры которого подбираются методом максимального правдоподобия;
- GMM – гауссова смесь, параметры которой подбираются с помощью алгоритма Expectation Maximization.

При использовании Гауссовых смесей в отдельном поле (**N components**) можно задать количество компонент в смеси (по умолчанию 4).

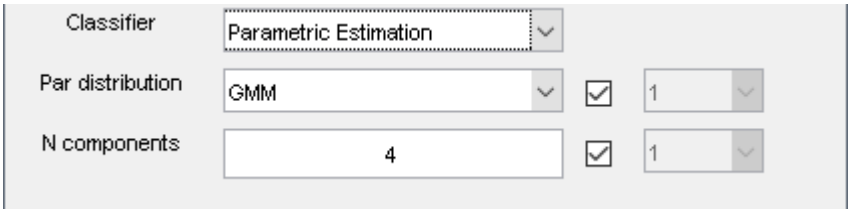



Рисунок 14 Настройки классификатора с параметрическим оцениванием

*Непараметрическое оценивание*

При выборе классификатора с непараметрической оценкой плотности с помощью ядер можно задать параметры ядер и подобрать оптимальные значения сглаживающих параметров (рис. 15).

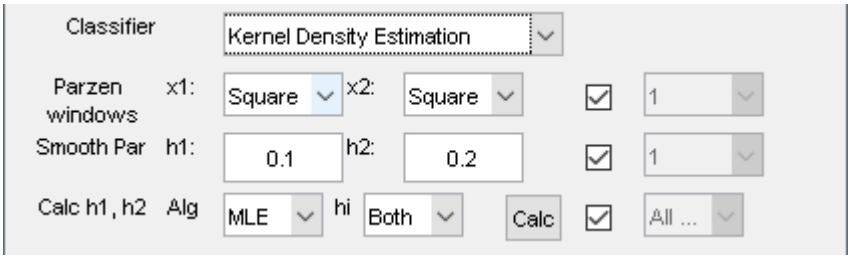



Рисунок 15 Настройки классификатора с оцениванием при помощи ядер

В таблице ниже приведено описание параметров данного классификатора.

Таблица 7

Описание основных параметров оценивания плотности с помощью ядер

| **Пункт меню (название)** | **Описание** | Тип | Задание для каждого класса |
|---|---|---|---|
| Parzen windows: | | | |
| x1, x2 | Тип окон Парзена для признаков x1, x2 | Список | + |
| Smooth Par | | | |
| h1, h2 | Значения сглаживающих параметров окон Парзена для признаков x1, x2 | ≥ 0 | + |
| Calc h1,h2 | | | |

| | | | |
|---|---|---|---|
| Alg | Алгоритм нахождения сглаживающих параметров | Список | + |
| hi | Какие параметры искать (по 1, 2 и или обоим признакам) | Список | + |
| Calc | Запуск процесса поиска сглаживающих параметров | Кнопка | + |

При выборе типа окон возможны следующие варианты:
- Square – квадратное (прямоугольное) окно;
- Gauss – гауссово окно (нормальная плотность N(0,1));
- Triangle – треугольное окно;
- Exp (Laplace) – экспоненциальное окно;
- Koshi – окно как плотность распределения Коши;
- Regen Filt – восстанавливающий фильтр.

Важно, что разные окно могут иметь различные оптимальные значения сглаживающих параметров. Поэтому значения этих параметров, подходящих для одних окон, могут давать неудовлетворительный результат для других окон.

Можно выбрать следующие алгоритмы поиска сглаживающих параметров:
- Par1 – параметрический алгоритм с гипотезой о нормальном распределении;
- Par2 – параметрический алгоритм с грубой оценкой разброса;
- MLE – правдоподобная кроссвалидация.

Поиск можно осуществлять по обоим (Both) параметров, либо только по одному из них (X1, X2) – это можно указать в списке hi.

При нажатии кнопки Calc запускается расчет оптимальных значений параметров. При использовании параметрических методов расчет осуществляется быстро по формуле.

При использовании метода правдоподобной кроссвалидации MLE поиск производится по сетке возможных значений параметров hi (сетка по умолчанию создается в логарифмическом масштабе), поэтому при нажатии Calc вначале открывается окно с вводом диапазонов значений параметров h1, h2 (рис. 16).

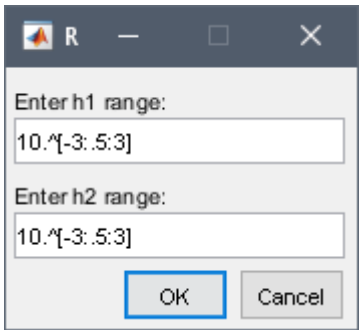



Рисунок 16 Задание диапазона поиска для метода MLE

При нажатии OK начинается поиск, при нажатии Cancel поиск не выполняется. При поиске оптимальных значений параметров целесообразно вначале найти грубую оценку на на небольшом диапазоне значений hi, а затем последовательно уточнять ее, суживая диапазон возможных значений hi.

При активном флаге слева от кнопки Calc поиск выполняется по всей выборке (Поиск типа 1) и для каждого класса ищутся одинаковые параметры. Если флаг неактивен, то поиск выполняется для каждого класса отдельно (Поиск типа 2), причем можно выбрать номер класса, для которого искать параметры или выбрать пункт All classes – тогда для каждого класса отдельно будет выполнен расчет сглаживающих параметров.

!!! При поиске параметров все три флага (checkbox) следует либо активировать, либо деактивировать. Поиск типа 1 в качестве типа окон использует глобальные (одинаковые)

настройки. Поиск типа 2 в качестве типа окон использует локальные (индивидуальные для каждого класса) настройки. Поэтому синхронизация флагов обеспечивает синхронизацию используемого типа окон и типа поиска.

*Метод k ближайших соседей*

При выборе классификатора k ближайших соседей можно задать различные параметры – число соседей, метрику (рис. 17).

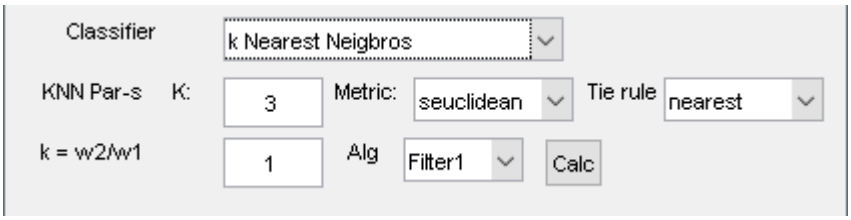



Рисунок 17 Настройки классификатора по методу *k* ближайших соседей

Число соседей **K** следует задавать нечетным.

При выборе метрики (**Metric)** возможны следующие варианты:

- euclidean – евклидово расстояние;
- seuclidean – взвешенное евклидово расстояние;
- cityblock – расстояние Манхэттена;
- chebychev – расстояние Чебышева;
- minkowski – расстояние Минковского;
- mahalanobis – расстояние Махаланобиса
- cosine – единица минус косинус угла между векторами;
- correlation – единица минус коэффициент корреляции между векторами;
- spearman – единица минус коэффициент ранговой корреляции Спирмена;
- hamming – расстояние Хэмминга (сколько координат отличается);
- jaccard – единица минус коэффициент Жаккарда (процент ненулевых отличающихся координат).

Возможно задание алгоритма разрешения споров, когда у нескольких классов находится одинаковое число ближайших соседей:

- smallest – выбирается класс с наименьшим индексом;
- nearest – выбирается класс с наименьшим расстоянием;
- random – выбирается случайный класс.

При выборе метрики Минковского активизируется поле (**p_exp**) для задания степени в этом расстоянии.

При выборе взвешенного расстояния активизируется несколько элементов управления:

- k=w2/w1 – поле для задания отношения весов по 2 и 1 признакам;
- Alg - выбор алгоритма подбора оптимального значения k;
- Calc – кнопка для запуска процесса поиска оптимального значения *k*.

В качестве алгоритмов предлагается на выбор три значения:

- Filter1 – фильтр-алгоритм первого типа;
- Filter2 – фильтр-алгоритм второго типа;
- Wrapper – враппер алгоритм (по минимуму ошибки классификации).

*Машины опорных векторов*

При выборе классификатора на основе машин опорных векторов можно задать ряд параметров (рис. 18, табл. 8).

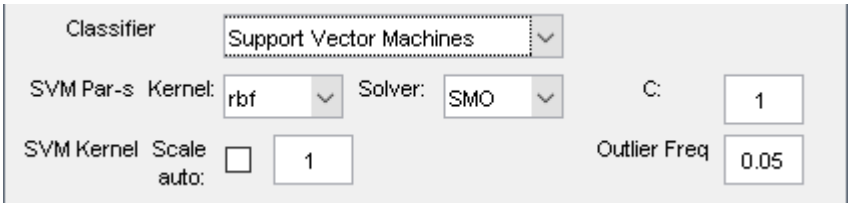



Рисунок 18 Настройки классификатора – машины опорных векторов

Таблица 8

Описание основных параметров машин опорных векторов

| **Пункт меню (название)** | **Описание** | Тип |
|---|---|---|
| Kernel | Тип ядра (линейное - linear, РБФ - rbf, полиномиальное - poly, нейронная сеть - mlp) | Список |
| Solver | Решатель (алгоритм оптимизации) | Список |
| Scale auto | Автоматическое масштабирование | Флаг |
| Scale | Коэффициент масштаба (особенно актуален для РБФ-ядер) | > 0 |
| C | Параметр алгоритма SVM – т.н. Box Constraint – его увеличение приводит к уменьшению числа опорных векторов, при этом увеличивается время обучения | > 0 |
| Outlier Freq | Процент примеров, которые можно считать выбросами | $0 \le x \le 1$ |
| Deg | Степень полинома при использовании полиномиальных ядер | > 0 |
| MLP | 2 параметра, характеризующие ядро – многослойный персептрон | $x1 \ge 0$, $x2 \le 0$ |

При выборе решателей возможны три варианта:

- ISDA (Iterative Single Data Algorithm);
- L1QP – L1-оптимизация с гладкими ограничениями с применением квадратичного программирования (требуется Optimization toolbox);
- SMO (Sequential Minimal Optimization).

Машины опорных векторов переобучаются при изменении каких-либо собственных параметров или при изменении исходных данных. Процесс обучения начинается при запуске решения задачи классификации. После обучения, если ничего не изменилось, SVM не будут еще раз обучаться.

### *Нейронная сеть прямого распространения*

При использовании в качестве классификатора НС прямого распространения можно задать ряд параметров (рис. 19, табл. 9).

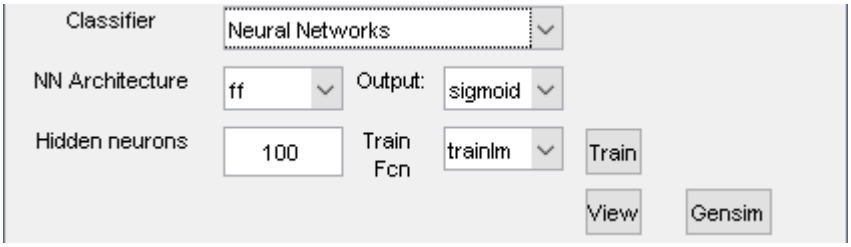



Рисунок 19 Настройки классификатора на основе НС прямого распространения

Таблица 9

Описание основных параметров НС прямого распространения

| **Пункт меню (название)** | **Описание** | Тип |
|---|---|---|
| NN Architecture | Архитектура НС ПР (обычная ff или каскадная cascade) | Список |
| Output | Тип нейронов в выходном слое – сигмоидальные sigmoid или с функцией softmax | Список |
| Hidden neurons | Количество скрытых нейронов в скрытых слоях | Массив, xi > 0 |
| Train Fcn | Функция обучения НС | Список |
| Train | Запуск обучения НС | Кнопка |
| View | Просмотр структуры НС | Кнопка |
| Gensim | Генерация Simulink-модели НС | Кнопка |

Каскадные НС имеют связи от входов ко всем скрытым и выходным слоям в отличие от обычных НС.

Использование в выходных нейронах функции активации softmax и последующее округление – один из подходов к построению НС для классификации. Использование в выходных нейронах обычной сигмоидальной функции активации позволяет получить ответ в виде вероятности для каждого из классов.

При выборе функции обучения возможны несколько вариантов, которые показаны в табл. ниже.

Таблица 10

Алгоритмы обучения НС прямого распространения

| **Пункт меню (название)** | **Описание** |
|---|---|
| Градиентные алгоритмы | |
| traingd | Обычный градиентный спуск |
| traingdm | Градиентный спуск с моментом |
| traingda | Градиентный спуск с адаптивным шагом |
| traingdx | Градиентный спуск с моментом и адаптивным шагом |
| trainrp | Эластичное распространение |
| Алгоритмы сопряженных градиентов | |
| traincgf | Алгоритм Флетчера-Ривза |
| traincgb | Алгоритм Пауэлла-Билла |
| traincgp | Алгоритм Поляка-Рибери |
| trainscg | Алгоритм взвешенных сопряженных градиентов |
| Квазиньютоновские алгоритмы | |
| trainlm | Алгоритм Левенберга-Марквардта |
| trainbfg | Алгоритм BFGS Бройдена, Флетчера, Гольдфарба, Шенно |
| trainoss | Алгоритм одношаговой секущей |
| trainbr | Алгоритм Левенберга-Марквардта + Байесова регуляризация |

*Деревья принятия решений*

Классификатор на основе деревьев решений строит двоичное дерево принятия решений на основе простых операциях сравнения признаков. Данный классификатор в программе представлен набором параметров (рис. 20, табл. 11).

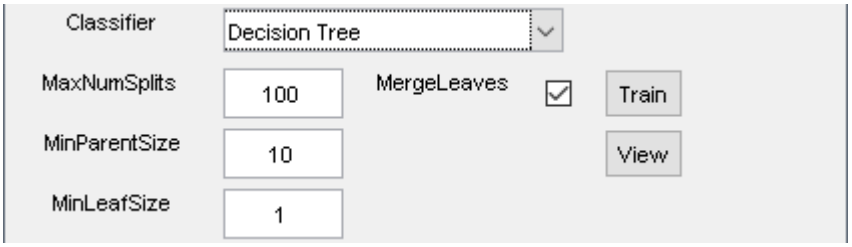


Рисунок 20 Настройки классификатора на основе деревьев принятия решений

Таблица 11

Описание основных параметров деревьев принятия решений

| **Пункт меню (название)** | **Описание** | Тип |
|---|---|---|
| MaxNumSplits | Максимальное количество разветвлений в дереве | > 0 |
| MinParentSize | Минимальное количество наблюдений на ветвь дерева | > 0 |
| MinLeafSize | Минимальное количество наблюдений за листом дерева | > 0 |
| MergeLeaves | Объединять листья дерева или нет | |
| Train | Запуск обучения НС | Кнопка |
| View | Просмотр структуры НС | Кнопка |
| Gensim | Генерация Simulink-модели НС | Кнопка |

Дерево состоит из узлов – ветвей и листьев (рис. 21). В ветвях осуществляется сравнение и ветвление, а каждый лист ассоциирован с тем или иным классом. С помощью MaxNumSplits можно ограничить максимальное число ветвлений. Увеличение MinParentSize ведет к тому, что каждая ветвь должна следовать к большему числу листьев, т.е. увеличивается густота листьев. Увеличение MinLeafSize ведет к тому, что каждый лист должен быть охвачен большим числом ветвей, т.е. чтобы было меньше отдельных листьев. Объединение листьев позволяет оптимизировать структуру дерева.

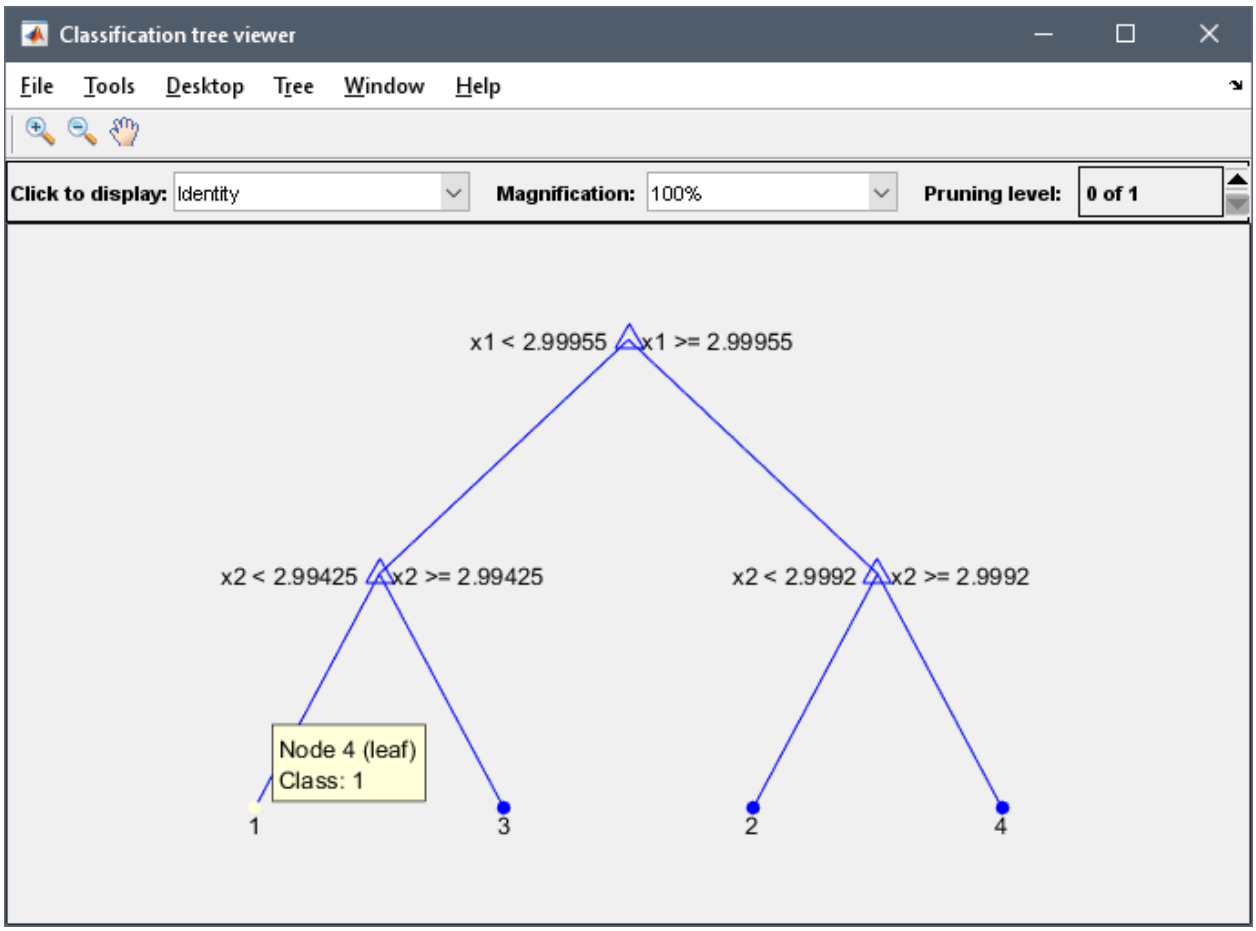



Рисунок 21 Пример структуры дерева принятия решений

# 3. Описание программной части

## 3.1 Задание исходных распределений классов

Задание вариантов исходных распределений классов выполняется в функции SetDataVariants файла SetDataVariants.m

Всего может быть сколько угодно вариантов распределений.

Основная информация записывается в переменной class_data. Дополнительная информация записывается в глобальные переменные: data_type_names, axis_rect.

Каждая из этих переменных представляет собой массив ячеек, количество элементов в котором совпадает с количеством вариантов исходных данных. Т.е. если всего *m* типов данных, то class_data, data_type_names, axis_rect являются массивами из *m* ячеек.

В class_data{i} содержится информация о распределении данных *i* варианта данных.

В axis_rect{i} содержится информация в формате [xmin xmax ymin ymax] о диапазоне значений по x и y для i варианта данных.

В data_type_names{i} содержится строка – название *i* варианта данных.

### Описание class_data{i}

class_data{i} является массивом ячеек, т.е. class_data{i} = {class1, class2, …, classC}, где каждая из ячеек classi описывает распределение *i* класса.

Каждый класс classi описывается структурой, которая может содержать три поля: type, par1, par2, par3. В следующей таблице приведены значения полей этих структур для различных распределений данных.

Таблица 12

Описание значений полей структуры classi для генерации исходных данных различного типа

| **type** | **par1, par2, par3** |
|---|---|
| 1 – нормальное распределение | **par1** - средние значения (строка [c1 c2 ... cn]<br>**par2** - ковариацоинная матрица |
| 2 – гауссова смесь | **par1** - матрица средних значений компонент (каждая строка - вектор средних [c1 c2 ... cn]<br>**par2** - 3-мерный массив с ковариацоинными матрицами компонент<br>**par3** - вектор с коэффициентами pi каждой компоненты |
| 3 – равномерное распределение | **par1** - массив ячеек, каждая из которых характеризует одну из областей, где расположено равномерное распределение {area1, area2, ..., areaN}<br>**par2** - то же, что par1, но описывает области, которные необходимо исключить<br>Возможны различные описания областей **areaj**:<br>• **{1, [x1min x1max x2min x2max], angle}** - прямоугольник с точками {x1min, x2min} и {x1max, x2max}, повернутый относительно центра на угол angle<br>• **{2, [c1 c2], r}** - окружность с центром в [c1 c2] радиуса r<br>• **{3, [c1 c2], [r1 r2], angle}** - эллипс с центром в [c1 c2], радиусами r1 и r2, повернутый относительно центра на угол angle<br>• **{4, [p11 p12], [p21 p22], [p31 p32], angle}** - треугольник с координатами вершин {p11, p12}, {p21, p22}, {p31, p32}, повернутый относительно центра на угол angle<br>• **{5, xv, yv, angle}** - внутренняя часть полигона, заданного точками {xv(i), yv(i)}, повернутого относительно центра на угол angle |
| 4 – распределение точек вокруг кривой на | **par1** - описание кривой в форме {curvePar1, curvePar2, ..., curveParN}<br>**curvePar** - описывает часть кривой - возможны следующие варианты:<br>• {1, tstart, tfinish, @fx(t), @fy(t)} - описание в параметрической форме x=fx(t), y=fy(t), tstart < t < tfinish |

| | |
|---|---|
| плоскости по заданному закону, положение на кривой (центр) имеет равномерное распределение | • {2, xv, yv} - описание в форме последовательности координат точек % {xv{i},yv{i}}<br>**par2** - описание распределения вокруг кривой в формате {distPar1, distPar2, ..., distparN}<br>**distPar** - описывает распределение точек вокруг соответствующей кривой, заданной curvePar<br>• {1, [s1 s2 r]} - нормальное распределение с СКО s1, s2 и к-том корреляции r<br>• {2, [r1 r2]} - равмномерное распределение в прямоугольнике, отстоящем на r1 вправо-влево, r2 - вверх-вниз<br>• {3, r} - равмномерное распределение в круге радиуса r<br>**par3** - коэффициенты (вероятности) каждых частей в формате:<br>• {1} - число точек пропорционально длине кривой<br>• {2, [p1 p2 ... pN]} - число точек для каждой кривой задается через pi |

### Примеры задания распределений исходных данных

#### *Нормальное распределение*

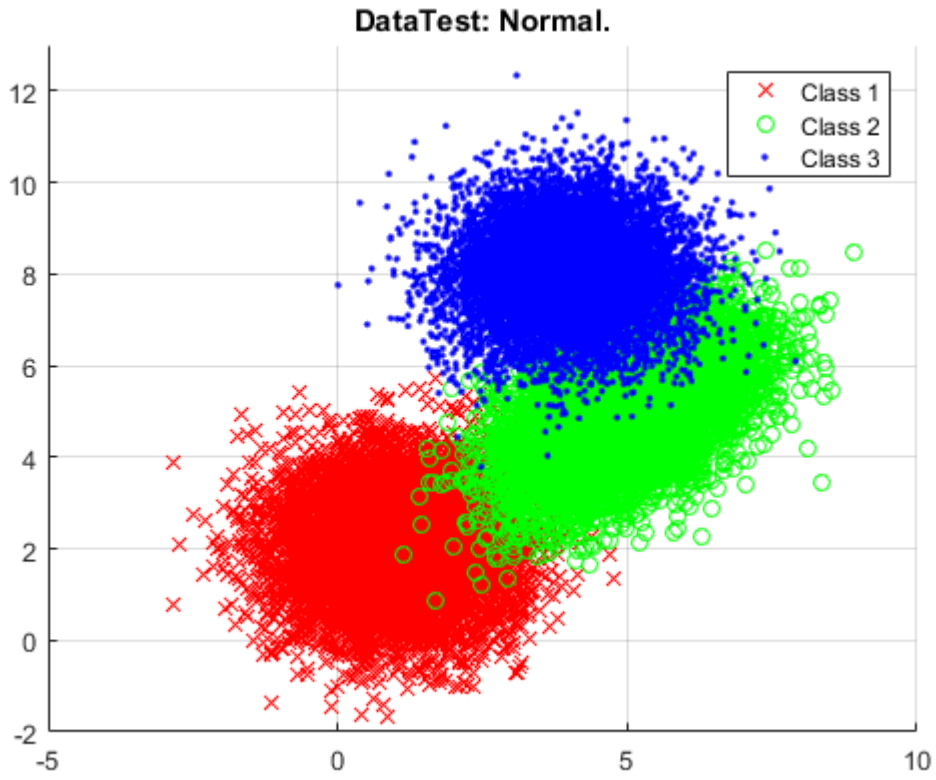
Задаются 3 класса, первый и третий классы имеют единичную ковариационную матрицу, у второго класса есть зависимость между признаками, поэтому эллипс рассеивания повернут.

```
class1.type = 1;
class1.par1 = [1 2];
class1.par2 = eye(2,2);

class2.type = 1;
class2.par1 = [5 5];
class2.par2 = [1 .5; .5 1];

class3.type = 1;
class3.par1 = [4 8];
class3.par2 = eye(2,2);

class_data{1} = { class1, class2, class3 };
axis_data{1} = [-5 10 -2 13];
data_type_names{1} = 'Normal';
```




*Гауссова смесь*

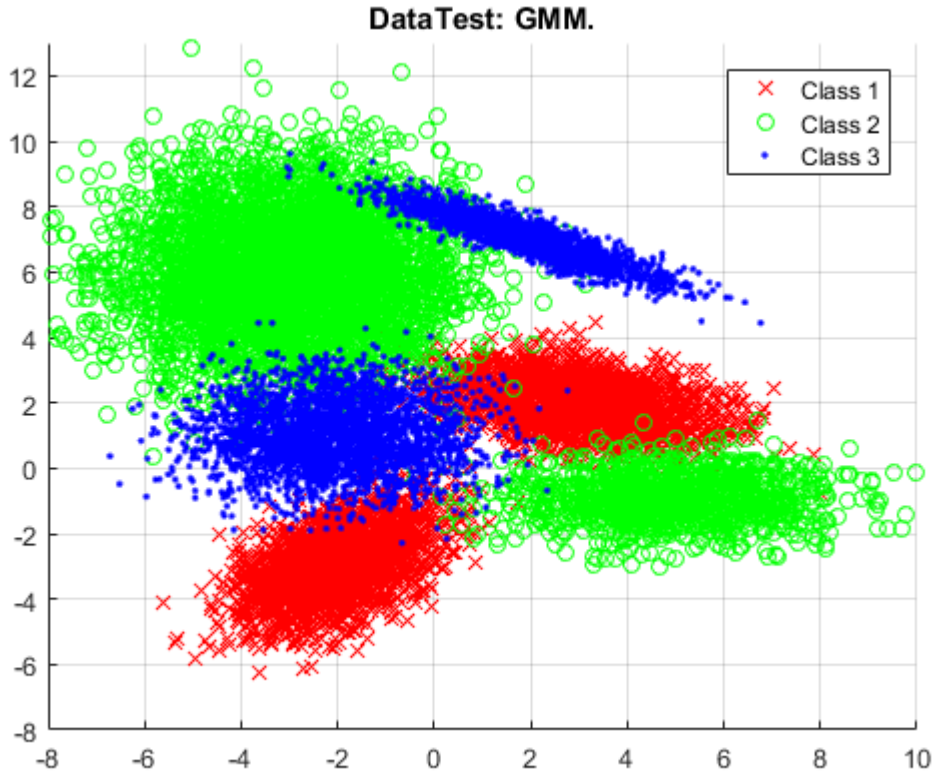
3 класса, каждый класс представлен 2 компонентами, причем компоненты удалены друг от друга так, что каждый класс разбивается на 2 части.

```
class1.type = 2;
class1.par1 = [3 2;-2 -3];
class1.par2 = cat(3,[2 -.4;-.4 .5],[1 .5;.5 1]);
class1.par3 = ones(1,2)/2;

class2.type = 2;
class2.par1 = [5 -1;-3 6];
class2.par2 = cat(3,[3 0;0 .5],[3 0;0 3]);
class2.par3 = [0.2 0.7];

class3.type = 2;
class3.par1 = [2 7;-2 1];
class3.par2 = cat(3,[2 -.9;-.9 .5],[2 0;0 1]);
class3.par3 = [0.4 0.6];

class_data{2} = { class1, class2, class3 };
axis_data{2} = [-8 10 -8 13];
data_type_names{2} = 'GMM';
```




*Равномерное распределение*

3 класса, один имеет распределение внутри круга, два других – внутри прямоугольника.

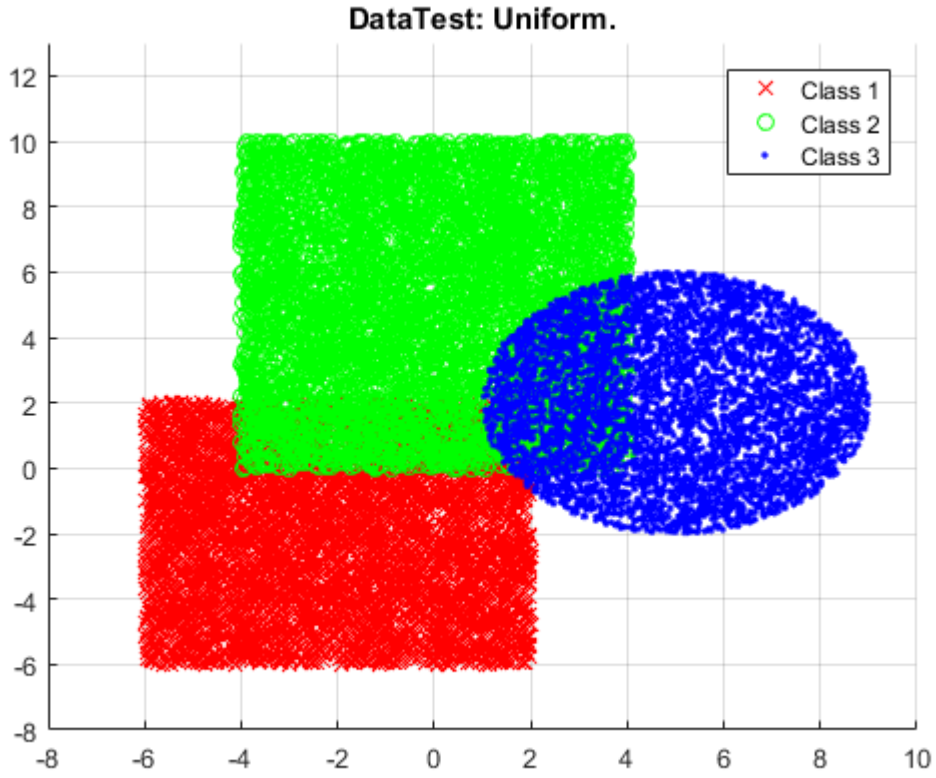
```
uni_c1_rect = {1, [-6 2 -6 2], 0};
uni_c2_rect = {1, [-4 4 0 10], 0};
uni_c3_circle = {2, [5 2], 4};

class1.type = 3;
class1.par1 = {uni_c1_rect};
class1.par2 = {};

class2.type = 3;
class2.par1 = {uni_c2_rect};
class2.par2 = {};

class3.type = 3;
class3.par1 = {uni_c3_circle};
class3.par2 = {};

class_data{3} = { class1, class2, class3 };
axis_data{3} = [-8 10 -8 13];
data_type_names{3} = 'Uniform';
```

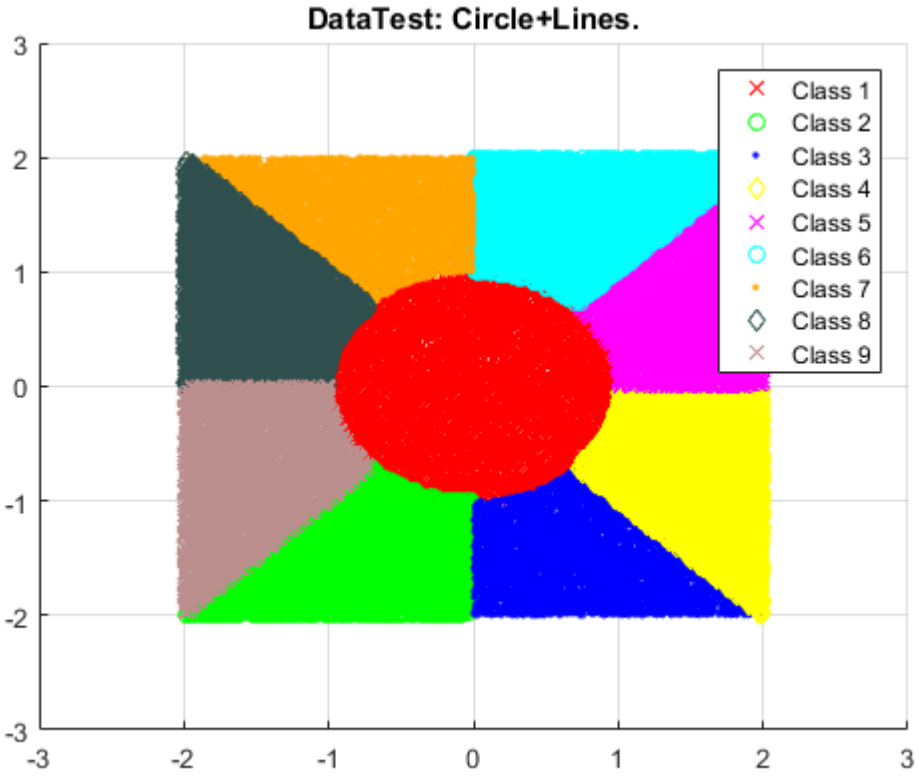



9 классов, 1 класс – окружность, 8 других – треугольники, из которых вычтена окружность первого класса.

```
circle_data = {{2, [0 0], 1}};
p = { [-2 -2],[0 -2],[2 -2], [2 0],[2 2],[0 2],[-2 2],[-2 0],[0 0] };

class.type = 3;
class.par1 = circle_data;
class.par2 = {};

class_data{9} = { class };
for i = 1:length(p) - 1
   class.type = 3;
   if i == length(p)-1
      class.par1 = {{4, p{i}, p{1}, p{length(p)},0}};
   else
      class.par1 = {{4, p{i}, p{i+1}, p{length(p)},0}};
   end;
   class.par2 = circle_data;
   class_data{9} = [class_data{9}, class ];
end;
axis_data{9} = [-3 3 -3 3];
data_type_names{9} = 'Circle+Lines';
```




*Распределение вокруг кривой*

Все 3 класса имеют распределение вокруг спирали. Причем сама спираль для первого класса задается набором значений

```
class1.par1 = {{2, fx(tmin:dt:tmax, k1, k2), fy(tmin:dt:tmax, k1, k2)}}
```

а для второго и третьего класса функтором

```
class3.par1 = {{1, tmin, tmax, @(x)fx(x,k1,k2), @(x)fy(x,k1,k2)}}; % {1, tstart, tfinish, @fx(t), @fy(t)}
```

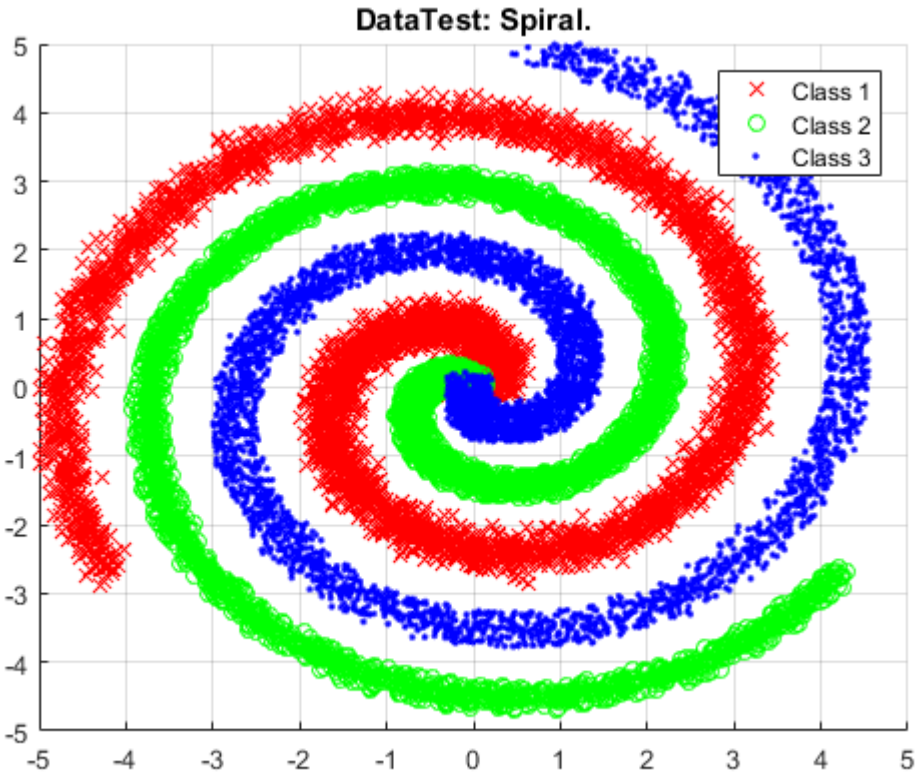
Также у всех 3 классов заданы различными способами распределения данных вокруг кривой. У 1 класса это нормальное распределение, у – 2 – равномерное внутри круга, у 3 – равномерное внутри прямоугольника.

```
class1.type = 4;
fx = @(t, k1, k2)t.*cos(k1*t+k2); fy = @(t, k1, k2)t.*sin(k1*t+k2);
tmin = 0; tmax = 5; dt = .2;
k1 = 2; k2 = 0;
s1 = .14; s2 = .14; r = 0;
class1.par1 = {{2, fx(tmin:dt:tmax, k1, k2), fy(tmin:dt:tmax, k1, k2)}};  % {2, xv, yv}
class1.par2 = {{1, [s1 s2 r]}};     % {1, [s1 s2 r]} - нормальное распределение с СКО s1, s2 и к-том корреляции r
class1.par3 = {1};

class2.type = 4;
k2 = 2;
r = 0.2;
class2.par1 = {{1, tmin, tmax, @(x)fx(x,k1,k2), @(x)fy(x,k1,k2)}}; % {1, tstart, tfinish, @fx(t), @fy(t)}
class2.par2 =  {{3, r}}; % равмномерное распределение в круге радиуса r
class2.par3 = {1};
```

```
class3.type = 4;
k2 = 4;
r1 = 0.25; r2 = 0.25;
class3.par1 = {{1, tmin, tmax, @(x)fx(x,k1,k2), @(x)fy(x,k1,k2)}}; % {1, tstart, tfinish, @fx(t), @fy(t)}
class3.par2 = {{2, [r1 r2]}};        % {2, [r1 r2]} - равмномерное распределение в прямоугольнике, отстоящем на
class3.par3 = {1};

class_data{10} = {class1, class2, class3};
axis_data{10} = [-5 5 -5 5];
data_type_names{10} = 'Spiral';
```




*Разные распределения*

1 класс имеет нормальное распределение, 2 класс – гауссову смесь, 3 и 4 класс имеют сложное по форме равномерное распределение. 3 класс состоит из 3 частей – треугольника и двух эллипсов, причем один эллипс и треугольник повернуты на некоторый угол. 4 класс состоит из эллипса, звезды и фигуры в форме окна. Для построения звезды использовалось представление данных в форме полигона class4.par1{3}={5, [6 7 8 6 8 6], [-4 -1 -4 -2 -2 -4], -10}. Для построения области типа окно из одной области вычиталась другая область class4.par1{2} = {1, [-7 0 -6 0], 20}; class4.par2{1} = {{1,[-5 -2 -5 -1], 20}}.

```
class1.type = 1;
class1.par1 = [1 2];
class1.par2 = eye(2,2);

class2.type = 2;
class2.par1 = [5 -1;-3 6];
class2.par2 = cat(3,[3 0;0 .5],[3 0;0 3]);
class2.par3 = [0.2 0.7];
```

```
class3.type = 3;
class3.par1 = {{4, [-7 0], [-6 4], [-4 2], 15},  {2, [5 5], 3}, {3, [5 -5], [4 1], -20} };
class3.par2 = {};

class4.type = 3;
class4.par1 = {{2, [4 10], 2}, {1, [-7 0 -6 0], 20}, {5, [6 7 8 6 8 6], [-4 -1 -4 -2 -2 -4], -10}};
class4.par2 = {{1,[-5 -2 -5 -1], 20}};

class_data{4} = { class1, class2, class3, class4 };
axis_data{4} = [-8 10 -8 13];
data_type_names{4} = 'Norm+GMM+Uniform';
```