

4. Программный комплекс обучения распознаванию образов.

4.1. Общая характеристика программного комплекса

Предложенные алгоритмы обучению распознаванию образов были реализованы в виде программного комплекса под названием Space 5.3. Комплекс может работать как с тестовыми примерами для отладки предложенных алгоритмов, так и для обработки реальных данных из различных областей знаний, таких как, техника, медицина, социология и т.п. Также, описываемый программный комплекс может быть использован для проведения курса “Методы распознавания образов”, читаемого на кафедре автоматики и телемеханики ТулГУ.

Программный комплекс состоит из программной оболочки, позволяющей в диалоговом режиме ввести исходные данные, проверить их корректность, выполнить указанный алгоритм обработки и отобразить результат на экране дисплея, а так же подготовленные заготовки-функции, позволяющие на основе принципа наследования данных ООП легко встраивать новые задачи обработки

Программа-оболочка представляет собой 32-разрядное приложение Windows написанное на языке C++ (компилятор Visual C++ 5.0) в соответствии с принципами объектно-ориентированного подхода (ООП) и принципом свободной последовательности программ, управляемых событиями.

Space 5.3 позволяет:

- ввод и редактирование данных в наглядном текстовом виде
- просмотр многомерных данных в псевдо двумерном (2D) пространстве
- просмотр результатов работы в 2D-пространстве
- возможность работы с данными больших размеров (например, 1000 объектов в 200-мерном признаковом пространстве)
- отсутствие программных ограничений на размер данных
- возможность внедрения новых задач обработки на основе объектно-ориентированного подхода
- просмотр справки по системе в виде стандартного гипертекста Windows

Программный комплекс состоит из следующих модулей:

SPACE53.EXE
SPACE5.HLP
SPACE53.LGO

Исполнимый файл
Файл справки
Логотип программы

Входные данные

Стандартизованный файл данных представляет собой ASCII текстовый файл со строками различной длины и расширением .DAT. Этот файл содержит в себе информацию о размерности данных, информацию эксперта и непосредственно матрицу данных.

Технические требования к аппаратному обеспечению:

- процессор 80486 и выше
- ОЗУ – 8 МБ
- SVGA адаптер с разрешением 800x600 и 256 цветов
- устройство ввода типа «мышь»
- операционная система Windows'95 или Windows NT
- по крайней мере 2,5 МБ свободного дискового пространства

4.2. Структура файлов данных

Все алгоритмы, работающие в программно алгоритмическом комплексе, обращаются к данным, представленным в стандартизованном формате. Пользователю, при создании файла, содержащего новые данные, необходимо заботиться о соответствии этого файла существующему формату.

Стандартизованный файл данных представляет собой ASCII текстовый файл со строками различной длины и расширением .DAT. Этот файл содержит в себе информацию о размерности данных, информацию эксперта и непосредственно матрицу данных. Порядок следования данных изменять нельзя.

Входные последовательности разделены символом ":". В строке возможно наличие не более одного такого символа. Последовательность символов слева от ":" предназначена для комментариев, более наглядного представления, и может быть опущена. Сканируемое значение обычно представляет собой либо единственное число, либо последовательность чисел, разделенных пробелами. В качестве примера приводится часть файла A.DAT с данными, полученными при написании латинской буквы *a* различными людьми.

```
Number of objects: 6
Number of features: 6
Number of classes: 1
Cluster number: 6
Feature group number: 6
Actual features (>0 - used feature): 1 1 1 1 1 1
```

```

1 - a2-Seredin(#1): 0.000000 0.436457 0.579186 0.725564
1.049975 0.887381 1
2 - a3-Seredin(#1): 0.436457 0.000000 0.566639 0.461734
1.052930 0.975661 1
3 - a1-Dmitriev(#1): 0.579186 0.566639 0.000000 0.363853
1.156121 1.232744 1
4 - a2-Dmitriev(#1): 0.725564 0.461734 0.363853 0.000000
0.897810 1.135118 1
5 - a1-Dolgova(#1): 1.049975 1.052930 1.156121 0.897810
0.000000 0.542393 1
6 - a2-Dolgova(#1): 0.887381 0.975661 1.232744 1.135118 0.542393 0.000000 1

```

4.3. Загрузка и редактирование данных

Загрузка данных осуществляется в стандартном диалоге Windows посредством выбора соответствующего файла .DAT, для этого пользователю необходимо из главного окна программы (см. рис. 4.3.1) выбрать пункт меню “File|OpenTask”.

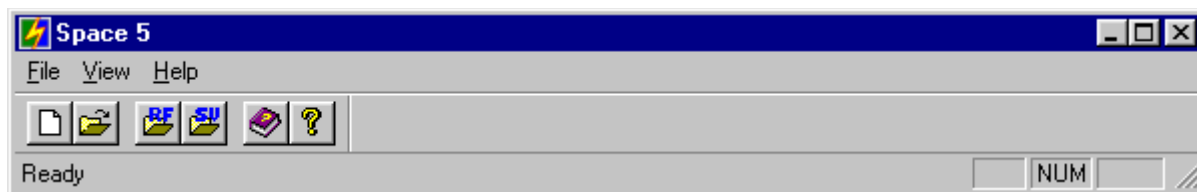


Рис. 4.3.1. Главное окно программного комплекса Space 5.3

В программном комплексе возможно редактирование данных с помощью встроенного многофункционального редактора. Для этого необходимо выбрать пункт меню “File|Open”. Необходимо помнить, что изменения, вносимые в файл данных .DAT, реально загружаются в динамические массивы памяти лишь после выбора задачи из меню и создания проблемного окна. Следовательно, после редактирования данных, они должны быть сохранены и проблемное окно открыто вновь. Алгоритмы обучения распознаванию образов имеют некоторые рабочие параметры. Начальные значения параметров хранятся в файле с тем же именем что и файл данных, но имеют расширение .DS5.

4.4. Отображение данных и решающего правила распознавания

4.4.1. Построение плоскости проецирования в многомерном пространстве

В процессе создания программного комплекса, реализующего алгоритм обучения распознаванию образов, реально встала проблема отображения результатов работы на дисплее ЭВМ. Если размерность признакового пространства

равна трем, уже приходится проводить визуализацию объектов распознавания с помощью громоздких изометрических построений. В случае же когда эта размерность больше трех, то вообще трудно себе представить геометрические аналогии. Реально же вектор состояния может включать в себя десятки, и даже сотни компонент. Вопрос: как отобразить многомерное признаковое пространство на плоскость экрана вычислительной машины? Разумеется, в прикладных задачах вряд ли возникнет такая необходимость, в них основой является непосредственно алгоритм распознавания, а способ интерпретации и показа результатов будет зависеть от конкретной решаемой задачи. В нашем случае программный комплекс выполняет демонстрационные функции, и было бы крайне полезно отобразить каким либо образом непосредственно объекты в многомерном пространстве.

Предлагается следующий способ визуализации. Пусть в признаковом пространстве существует разделяющая два класса гиперплоскость. Расположим плоскость экрана так, чтобы она была перпендикулярна гиперплоскости, и проекции объектов на нее располагались наименее плотно. В этом случае гиперплоскость будет отображаться на плоскость экрана в виде вертикальной линии, и основная задача будет заключаться в расчете проекций, которые будут отбрасывать объекты на плоскость экрана.

Пусть в пространстве \mathbf{R}^n найдена граничная гиперплоскость, например, как предложено в разделе 2, определяющая границу области ненулевого совместного распределения двух классов. Такая гиперплоскость задается ее направляющим вектором (вектор-нормаль к ней единичной длины) $\mathbf{c} = (c_1, \dots, c_n)^T$ и смещением c_0 , где $\|\mathbf{c}\| = 1$.

Будем полагать, что все векторы-строки \mathbf{x}^T матрицы данных $\mathbf{X}(N \times n)$, где N - число объектов, центрированы, то есть начало координат пространства \mathbf{R}^n перенесено в точку центра тяжести всей совокупности данных $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$. Тогда каждый объект будет представлен своим вектором $\mathbf{x} - \bar{\mathbf{x}}$. В дальнейшем будем полагать, что векторы $\mathbf{x} \in \mathbf{R}^n$ всегда центрированы, если это специально не оговорено.

Будем полагать, что плоскость проецирования (плоскость экрана) проходит через начало координат так, что вектор \mathbf{c} лежит в ней. Пусть вектор \mathbf{b} - некоторый вектор единичной длины $\|\mathbf{b}\| = 1$, проведенный из начала координат, лежащий в плоскости экрана и ортогональный вектору \mathbf{c} . Тогда векторы \mathbf{c} и \mathbf{b} образуют в плоскости экрана прямоугольную систему координат $y_1 O y_2$, где ось y_1 образована вектором \mathbf{c} , а ось y_2 образована вектором \mathbf{b} . Тогда в плоскости

экрана некоторый вектор \mathbf{X} будет представлен вектором $\mathbf{y} = (y_1, y_2)^T$, координаты которого определены как $y_1 = \mathbf{c}^T \mathbf{x} + a_0$, $y_2 = \mathbf{b}^T \mathbf{x}$.

Очевидно, что в плоскости экрана гиперплоскость вида $\mathbf{c}^T \mathbf{x} + a_0 = 0$ будет представлена вертикальной линией параллельной оси y_2 , пересекающей ось y_1 в точке $-c_0$.

Так как вектор \mathbf{c} уже найден, и представляет собой направляющий вектор разделяющей гиперплоскости, то найдем вектор \mathbf{b} . Еще раз заметим, что ориентация в пространстве вектора \mathbf{b} , который ортогонален вектору \mathbf{c} и лежит в плоскости экрана, определяет ориентацию плоскости экрана. Пусть вектор \mathbf{b} ориентирован так, что все проекции объектов на него располагаются наименее плотно. Другими словами, пусть координаты y_2 объектов в плоскости экрана занимают как можно больший диапазон, то есть различаются между собой как можно сильнее. Назовем такой вектор \mathbf{b} оптимальным в указанном здесь смысле.

Наше интуитивное представление об оптимальном векторе \mathbf{b} следует уточнить, то есть формализовать. Это можно сделать различными способами.

Пусть, например, среди всей совокупности объектов в исходном пространстве найдены два самых далеких объекта, то есть найден соединяющий их вектор \mathbf{z} , координатами которого являются разности координат данных объектов. Тогда оптимальный вектор \mathbf{b} ориентирован так, проекция вектора \mathbf{z} на него имеет наибольшую длину.

Следовательно, найдем вектор \mathbf{b} из условия $\mathbf{z}^T \mathbf{b} \rightarrow \max$ при ограничениях $\mathbf{c}^T \mathbf{b} = 0$, $\mathbf{b}^T \mathbf{b} = 1$, где первое из них означает ортогональность, а второе означает $\|\mathbf{b}\| = 1$.

Составим функцию Лагранжа $L(\mathbf{b}, \lambda_1, \lambda_2) = \mathbf{z}^T \mathbf{b} + \lambda_1 \mathbf{c}^T \mathbf{b} + \lambda_2 (\mathbf{b}^T \mathbf{b} - 1)$, и найдем ее минимум из условия равенства нулю ее производных по неизвестным координатам вектора \mathbf{b} :

$$\nabla_{\mathbf{b}} L(\mathbf{b}, \lambda_1, \lambda_2) = \mathbf{z} + \lambda_1 \mathbf{c} + 2\lambda_2 \mathbf{b} = 0.$$

Отсюда

$$\mathbf{b} = -\frac{\lambda_1}{2\lambda_2} \mathbf{c} - \frac{1}{2\lambda_2} \mathbf{z}.$$

Пусть $\lambda'_1 = -\frac{\lambda_1}{2\lambda_2}$, $\lambda'_2 = -\frac{1}{2\lambda_2}$. Подставив эти значения и переобозначив $\lambda'_1 = \lambda_1$, $\lambda'_2 = \lambda_2$, окончательно получим

$$\mathbf{b} = \lambda_1 \mathbf{c} + \lambda_2 \mathbf{z}.$$

Используем первое ограничение и найдем λ_1 :

$$\mathbf{c}^T \mathbf{b} = \mathbf{c}^T (\lambda_1 \mathbf{c} + \lambda_2 \mathbf{z}) = \lambda_1 \mathbf{c}^T \mathbf{c} + \lambda_2 \mathbf{c}^T \mathbf{z} = \lambda_1 + \lambda_2 \mathbf{c}^T \mathbf{z} = 0, \quad \lambda_1 = -\lambda_2 \mathbf{c}^T \mathbf{z}.$$

Подставим найденное в выражение для \mathbf{b} . Заметим, что в соответствии с правилом согласования размерностей в матричных уравнениях, следует записать $\mathbf{b} = c\lambda_1 + z\lambda_2$, считая коэффициенты матрицами размера (1×1) . Тогда получим

$$\mathbf{b} = -\mathbf{c}\mathbf{c}^T z\lambda_2 + z\lambda_2 = (\mathbf{I} - \mathbf{c}\mathbf{c}^T)z\lambda_2,$$

где $\mathbf{I}(n \times n)$ - единичная матрица. Обозначим $\mathbf{C} = \mathbf{I} - \mathbf{c}\mathbf{c}^T$ и получим $\mathbf{b} = \mathbf{C}z\lambda_2$.

Найдем λ_2 из второго ограничения

$$(\mathbf{C}z\lambda_2)^T \mathbf{C}z\lambda_2 = \mathbf{z}^T \mathbf{C}^T \mathbf{C}z\lambda_2^2 = 1, \quad \lambda_2 = 1/\sqrt{\mathbf{z}^T \mathbf{C}^T \mathbf{C}z},$$

взяв положительное значение. Окончательно получим


$$\mathbf{b} = \mathbf{C}z/\sqrt{\mathbf{z}^T \mathbf{C}^T \mathbf{C}z}, \quad \text{где } \mathbf{C} = \mathbf{I} - \mathbf{c}\mathbf{c}^T.$$

4.4.2. Оптимальное проецирование данных на плоскость экрана

При проецировании возможны три случая. В первом случае исходное признаковое пространство является двухмерным. Тогда плоскость экрана $y_1 O y_2$ является просто исходным двухмерным пространством $x_1 O x_2$, в котором при пошаговом отображении работы алгоритма изменяется положение (наклон и смещения) трех параллельных границ (края и середина области совместного распределения данной пары классов), а объекты не изменяют своего положения. Положение границ определяется направляющим вектором $\tilde{\mathbf{a}}$ неединичной длины и соответствующими смещениями вдоль него от начала координат: $-a_0$, $0.5 - a_0$ и $1 - a_0$.

Во втором случае исходное признаковое пространство также является двухмерным, но плоскость экрана $y_1 O y_2$ образована вектором \mathbf{C} , где $\mathbf{c} = \tilde{\mathbf{a}}/\|\tilde{\mathbf{a}}\|$, $c_0 = a_0/\|\tilde{\mathbf{a}}\|$, $\|\mathbf{c}\| = 1$, и ортогональным вектором \mathbf{b} , где $\mathbf{b} = (-c_2, c_1)^T$, $\|\mathbf{b}\| = 1$. Тогда при пошаговом отображении работы алгоритма три границы изменяют только смещения, оставаясь взаимно параллельными и параллельными вертикальной оси $O y_2$. Объекты в плоскости экрана могут изменять свое положение, так как сам экран поворачивается в исходном пространстве соответственно поворотам вектора $\tilde{\mathbf{a}}$. Положение границ на оси $O y_1$ определяется смещениями $-a_0/\|\tilde{\mathbf{a}}\|$, $(0.5 - a_0)/\|\tilde{\mathbf{a}}\|$, $(1 - a_0)/\|\tilde{\mathbf{a}}\|$.

В третьем случае пространство является многомерным. Плоскость экрана находится как описано выше для многомерного пространства. Поэтому, все остается также как и во втором случае, за исключением того, что вектор \mathbf{b} находится специальным образом.

В случае, когда число классов обучения больше двух пользователь может наблюдать проекцию данных лишь относительно одной из пар классов. Выбор соответствующей пары осуществляется посредством диалога "Mapping Style" из окна задачи, нажав кнопку .

4.5. Обучение и распознавание

Для запуска алгоритма обучения распознаванию образов пользователю необходимо из меню главного окна программы (рис. 4.3.1) выбрать пункт меню "File|OpenTask". После этого на экран будет выведено окно задачи (рис. 4.5.1) и произойдет загрузка динамических массивов памяти из файлов данных.

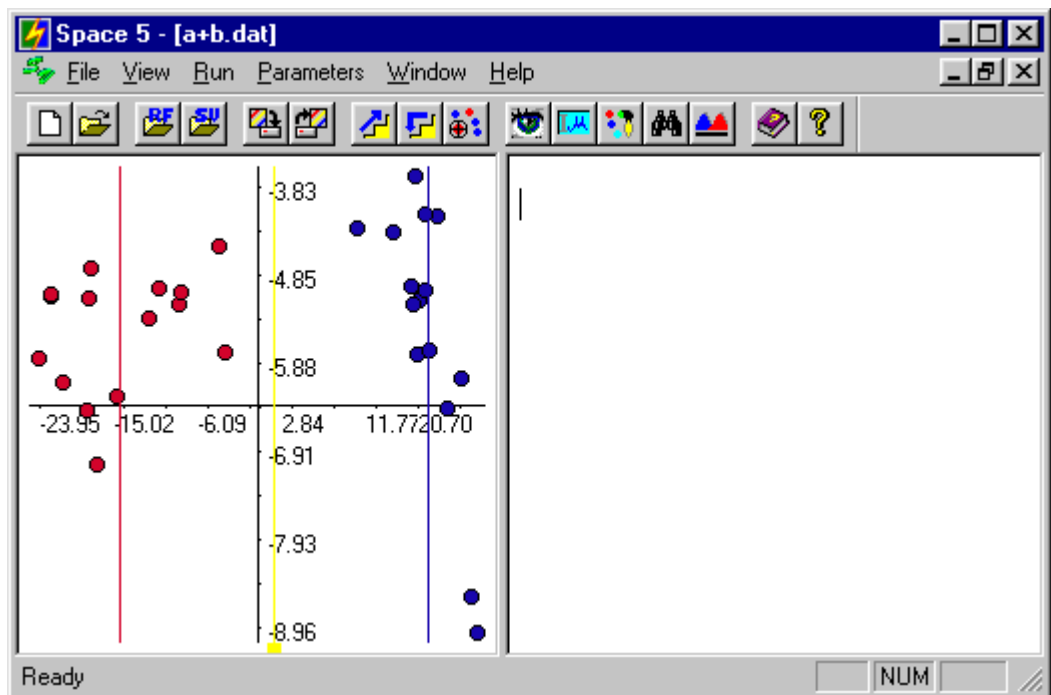


Рис. 4.5.1. Рабочее окно программы при выборе задачи обучения

Меню "Run" позволяет производить запуск итерационного алгоритма по шагам, до останова и возврат на начальное приближение. Кроме того, эти же функции можно выбрать посредством кнопок ускорителей:



- выполнение алгоритма до останова,



- возврат к начальному приближению.

Алгоритм, выполняемый в режиме "до останова", может быть остановлен нажатием кнопки "возврат".

Графическое соответствие реализуемого процесса выводится в виде двумерной проекции многомерного признакового пространства. Принцип

построения такой проекции описан в разделе 4.4. Направляющий вектор разделяющей гиперплоскости коллинеарен оси X экрана, и пользователь наблюдает след разделяющей гиперплоскости как вертикальную линию.

Одновременно с изменением графического представления решающих правил и данных на этапе обучения возможно отображение результатов в текстовом виде. Информация подобного рода содержится в текстовом буфере, который отображается в окне "Problem Status", и периодически обновляется (см. рис.4.5.1). Для просмотра более детальных сведений о решающем правиле пользователь может вывести подобную информацию в текстовое окно нажатием правой кнопки мыши на желтый квадрат внизу линии, отображающей разделяющую гиперплоскость (см. рис. 4.5.2).

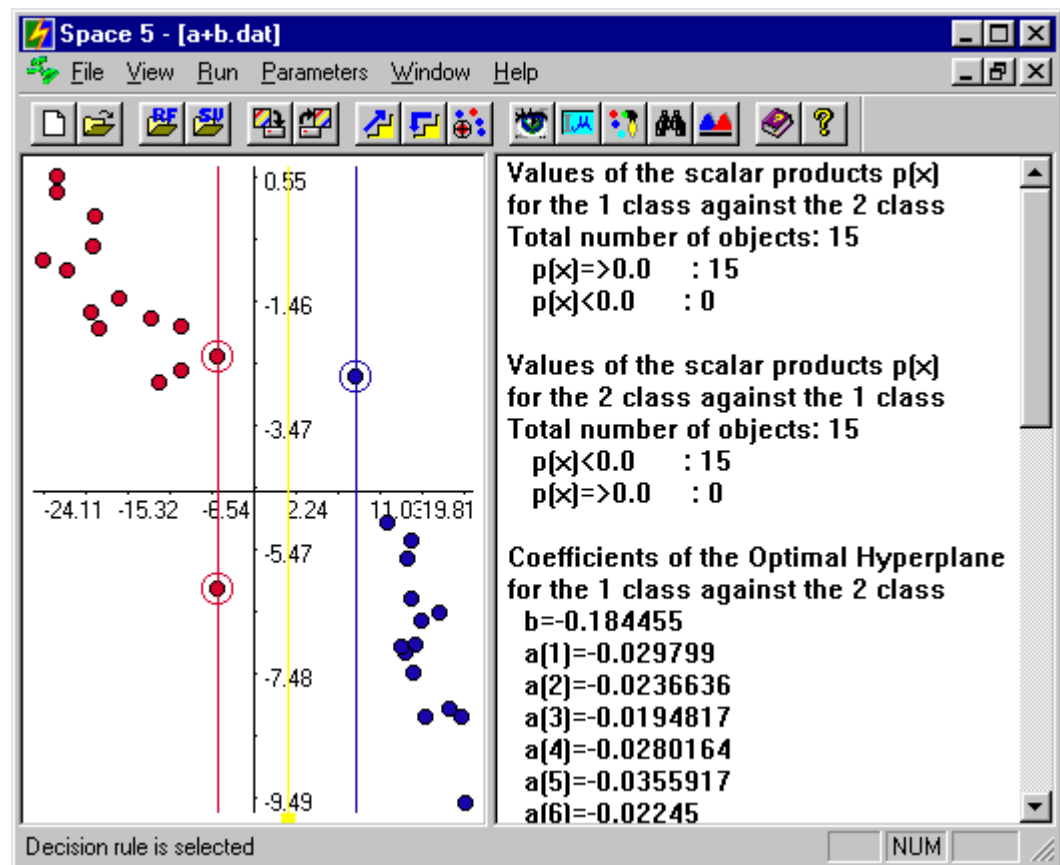




Рисунок 4.5.2 Информация о решающем правиле распознавания

Изменение параметров алгоритма осуществляется в диалоговом окне, выпадающем при нажатии кнопки . Начальные значения параметров загружаются из файла с тем же именем что и файл данных, и имеющего расширение .DS5. После изменения параметров в окне "Parameters of Algorithm" и завершения работы программы изменения не сохраняются. Для изменения начальных параметров в файле .DS4, пользователю необходимо воспользоваться каким либо внешним редактором, например NotePad.

Изменение вида отображения осуществляется нажатием кнопок  и .

После завершения процесса обучения пользователь может провести контроль качества решающего правила с помощью процедуры «скользящий контроль» (см. п.5.1.4., нажав кнопку ). Результаты процедуры скользящий контроль выводятся кратко в окне "Problem Status", и, полностью, в файле с расширением .loc. (см. рис. 4.5.3).

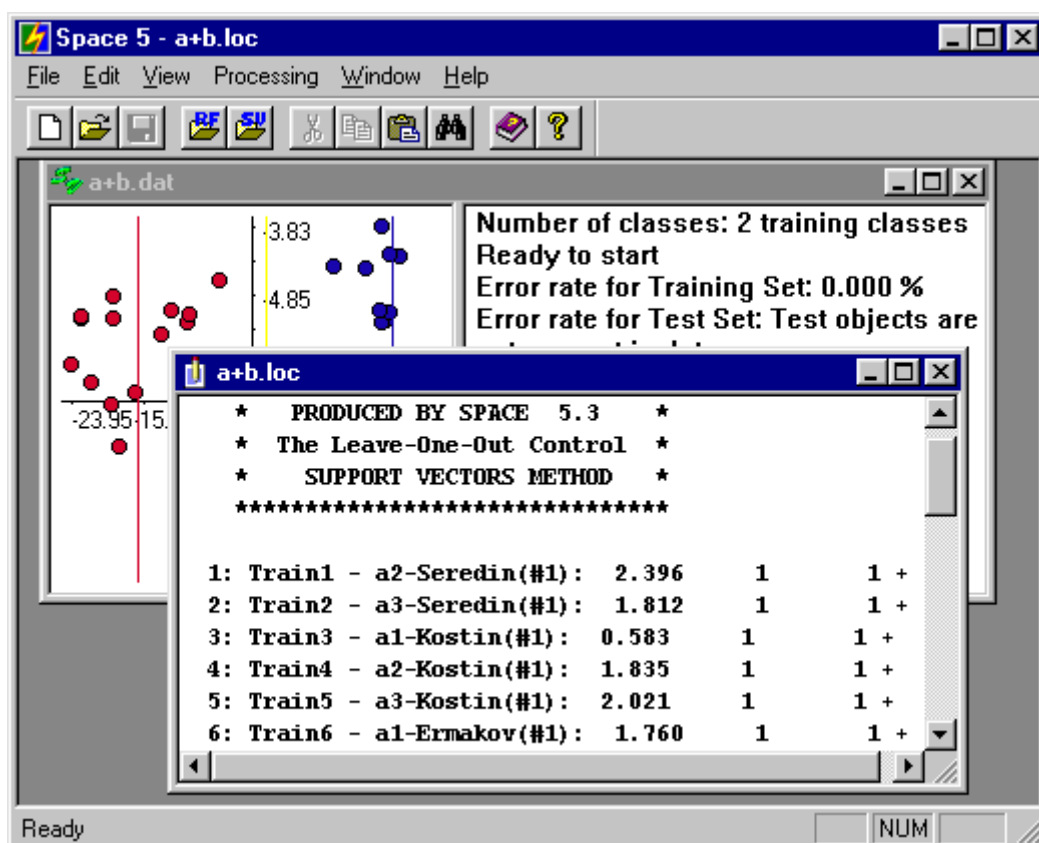



Рис. 4.5.3. Результат процедуры «скользящий контроль»


4.6. Сохранение и загрузка решающего правила

В программном комплексе Space 4.5 реализованы следующие функции:

- возможность сохранения текущего состояния задачи без выхода из программы,
- возможность загрузки предварительно сохраненных результатов и параметров,
- возможность применения полученного решающего правила к другим данным такой же природы.

Для обеспечения выполнения перечисленных функций в программный комплекс введено такое понятие как файл состояния задачи .STA. Пользователь

имеет возможность сохранить в этот файл полученное решающее правило, размерность данных, которым оно соответствует и параметры алгоритма. Для этого необходимо нажать кнопку  из окна задачи. Пользователю ни в коем случае не стоит редактировать этот файл, он является служебным. Кроме файла .STA на диске создается файл отчета с расширением .TXT. Именно этот файл, пользователь может использовать по своему усмотрению - редактировать, переименовывать, удалить.

Для вызова ранее сохраненного файла состояния для данных, по которым он был получен, либо для других данных такой же природы, пользователю необходимо выбрать кнопку . После этого будут загружены решающее правило и рабочие параметры алгоритма, и обновлено графическое отображение. В случае, если загружаемое решающее правило не соответствует размерности данных, будет выведено сообщение об ошибке.

