УДК 621.8

В.И.Литвиненко

ИММУННЫЙ КЛАССИФИКАТОР ДЛЯ РЕШЕНИЯ ЗАДАЧ БИНАРНОЙ КЛАССИФИКАЦИИ

(ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ)

Архитектура классифицирующей искусственной иммунной системы

Архитектурная схема иммунной системы, реализующей алгоритм отрицательного отбора состоит из семи блоков, каждый из которых представлен одним или несколькими классами. Блоки на схеме соответствуют основным функциональным узлам системы, а стрелки, соединяющие блоки – основным потокам передачи данных и управляющих параметров. Краткое описание элементов схемы представлено ниже.

- **интерфейс ввода/вывода данных.** Блок предназначен для загрузки обучающих и распознаваемых данных в систему, получения результата распознавания, а, также, сохранения текущего состояния множества детекторов для возможности быстрой перенастройки системы на другую задачу без дополнительного обучения;

- **база данных (БД).** Данные, предназначенные для обучения или распознавания хранятся в базе данных системы и выбираются оттуда по мере надобности. Здесь же хранятся результаты распознавания и текущее состояние множества детекторов;

- **генератор случайных чисел (ГСЧ).** Использует несколько видов распределений и может генерировать целые или вещественные числа в заданных диапазонах;

- **блок генерации кандидатов.** Используя последовательности случайных чисел, создаваемые ГСЧ, производит множество кандидатов детекторов для последующего отбора их в качестве детекторов;

- **блок проверки совпадения.** Во время обучения системы данный блок используется для создания множества детекторов. Проверяет два вектора на предмет совпадения их между собой. Для проверки использует заданное правило совпадения и порог совпадения, определяющий границы зоны совпадения. В режиме распознавания системы данный блок распознает поступающие на его вход тестируемые вектора;

- **интерфейс управления памятью.** Реализует набор процедур, необходимых для управления памятью системы и работы с пространственными формами – внутренним представлением данных;

- **интерфейс настройки алгоритма.** Блок предоставляет возможность настройки системы с использованием подгружаемого файла конфигурации или интерактивную настройку при помощи стандартных графических элементов управления операционной системы.

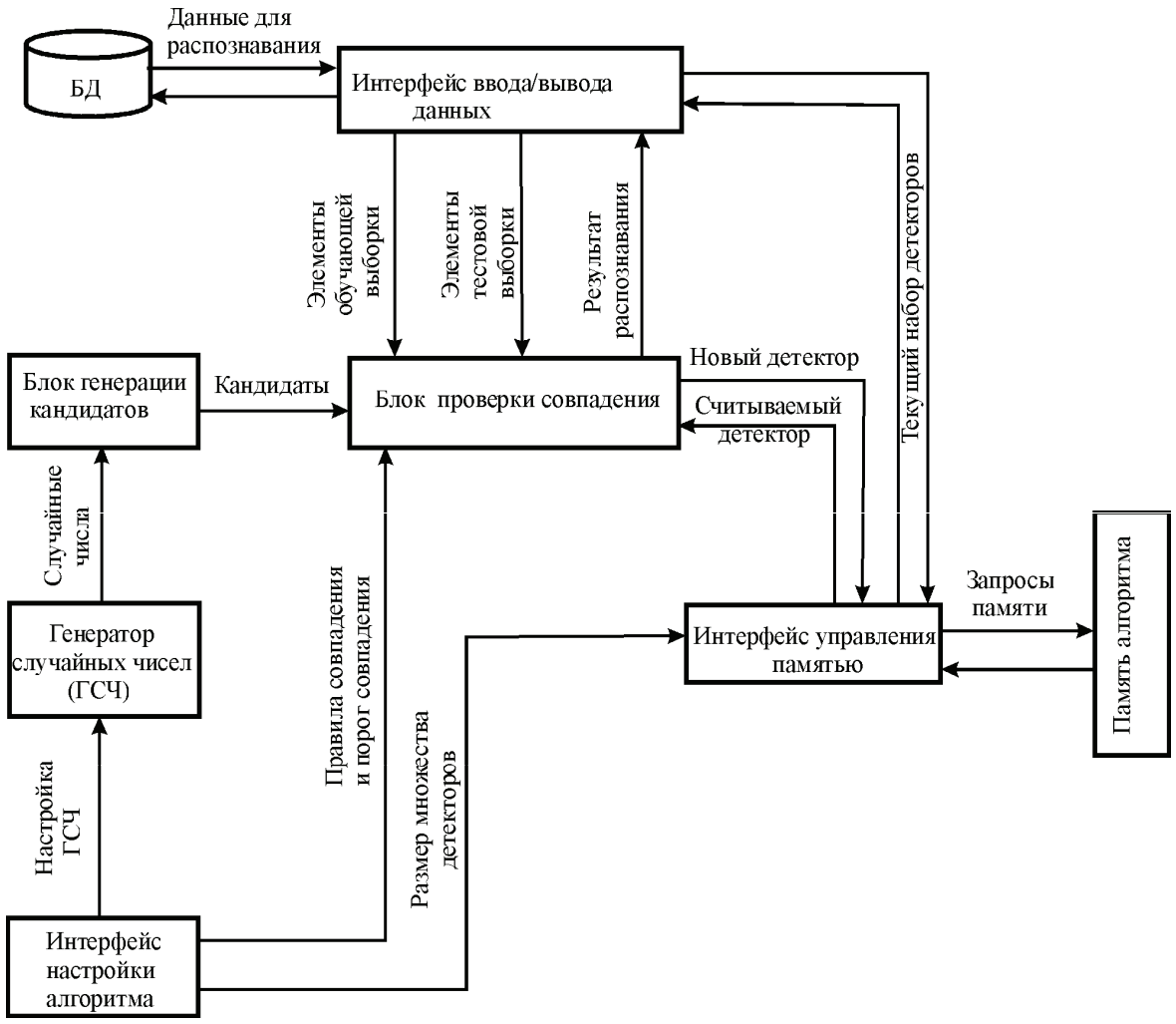


Рисунок 1 - Архитектура иммунной системы реализующей алгоритм отрицательного отбора

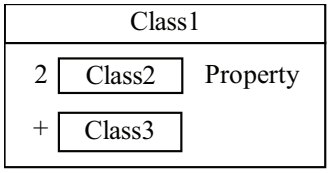
Объектно-ориентированное представление алгоритма отрицательного отбора

Прежде чем приступить к описанию структуры библиотеки, остановимся на условных обозначениях, используемых в схемах (табл.1), остальные обозначения представлены в соответствии со стандартами языка UML.

 ISSN 1562-9945

Таблица 1

Используемые условные обозначения

Обозначение	Описание
	Показывает, что Class1 является контейнером для объектов Class2 и Class3. Цифра в левой части прямоугольника показывает, какое количество экземпляров объектов соответствующего класса может содержать контейнер. Знак «+» говорит о том, что должен содержаться как минимум один экземпляр

Основываясь на данных модели, рассмотренной в предыдущем разделе можно выделить следующие объекты предметной области.

1. Генератор случайных чисел (Класс CRandomGenerator). Генерирует случайные числа в различных интервалах и с различными функциями распределения.
2. Абстрактный класс объектов алгоритма отрицательного отбора (CNSObject). Содержит базовые свойства для всех объектов системы.
3. Класс параметров алгоритма (CParametrs). Содержит все настройки алгоритма отрицательного отбора.
4. Класс атрибут формы (CAttribute). Представляет собой минимальный элемент пространственной формы, описывающей образ.
5. Класс пространственной формы. (CShape). Класс, описывающий образ объектов.
6. Класс множества объектов пространственной формы (CShapeSet).Описывает множество объектов пространственной формы (множество детекторов, множество кандидатов, обучающую выборку) и операции над ними.
7. Класс алгоритма отрицательного отбора (CNSAlgorithm). Класс включающий в себя основные и вспомогательные объекты модели и обеспечивающий функционирование отрицательного отбора.
8. Класс с правилами совпадений (CMatchingRule). Класс содержащий в себе набор различных правил для сравнения образов (пространственных форм).

9. Класс внутреннего преобразования данных (CDataDriver). Класс предназначен для ввода/вывода данных (класс-интерфейс).

Пример диаграммы взаимного включения основных объектов для построения алгоритма отрицательного отбора показан на рисунке 2.

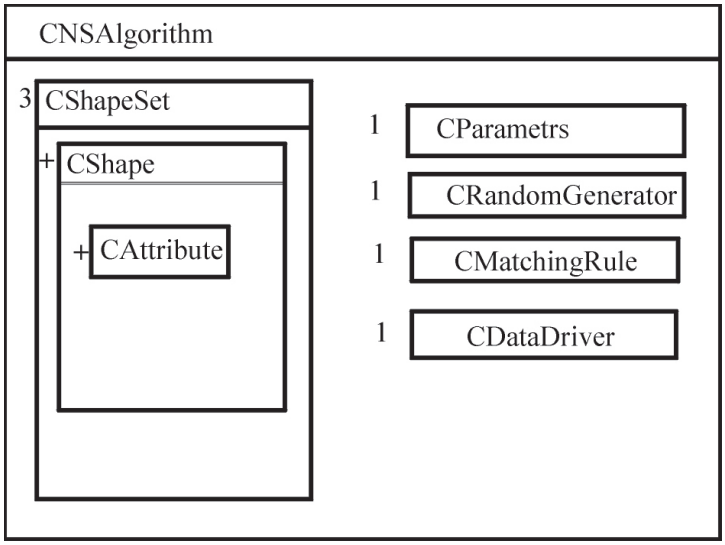


Рисунок 2 – Диаграмма взаимного включения объектов

Для описания логического аспекта алгоритма отрицательного отбора была разработана диаграмма классов (рисунок 2).

Для дальнейшего анализа используемого алгоритма, а также для выражения поведения отдельных классов и их возможного взаимодействия были разработаны диаграммы состояний в режиме обучения (рисунок 4) и в режиме распознавания (рисунок 5).

В соответствии с представленной диаграммой состояния системы в режиме обучения, предполагается, что в исходном состоянии предполагается, что обучающая выборка уже загружена и множество кандидатов-детекторов создано в соответствии с параметрами алгоритма. Вводится первый кандидат. Осуществляется его проверка в соответствии с правилом совпадения с обучающей выборкой. В случае хотя бы одного совпадения система переходит в состояние удаления кандидата и возвращается к состоянию оценки нового кандидата. В случае несовпадения кандидата с обучающей выборкой система переходит в состояние записи кандидата в множество детекторов и также возвращается к проверке следующего кандидата. Процесс продолжается до тех пор, пока не будет исчерпано все множество кандидатов.

 ISSN 1562-9945

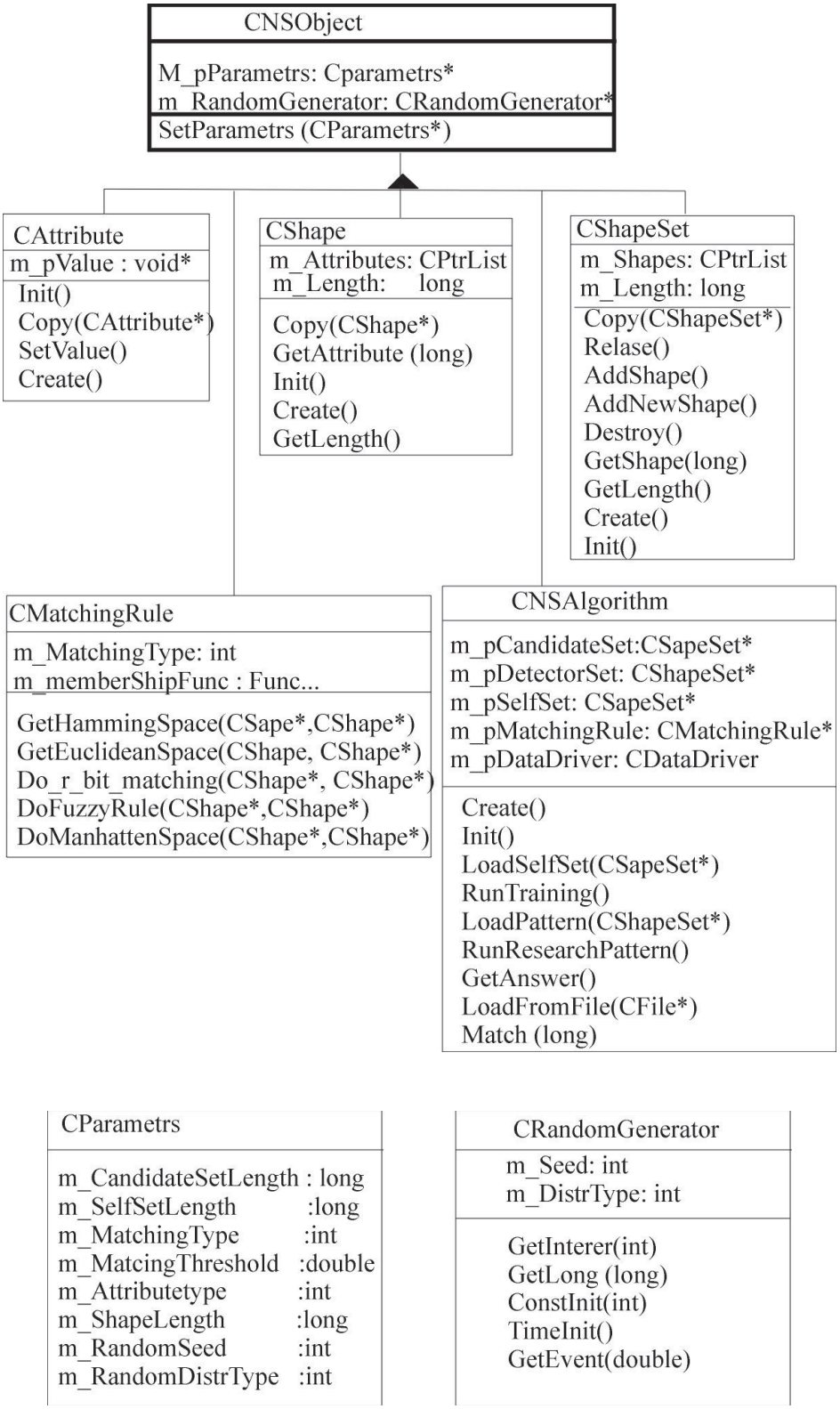


Рисунок 3 - Диаграмма классов

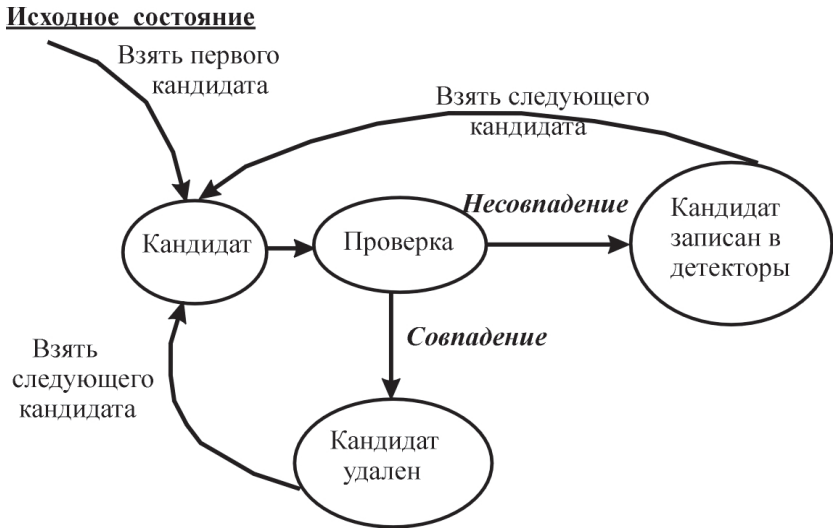


Рисунок 4 - Диаграмма состояний в режиме обучения

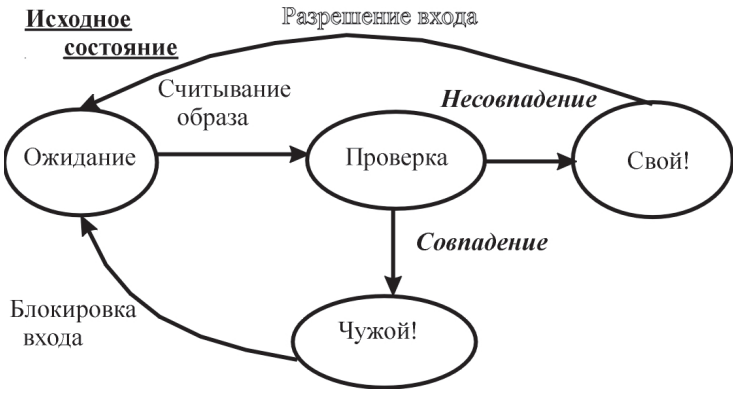


Рисунок 5 - Диаграмма состояний системы в режиме распознавания

В исходном состоянии предполагается, что множество детекторов уже сформировано. Находясь в состоянии ожидания система готова к считыванию входного образа. Если данное событие произошло, осуществляется проверка входного образа с множеством детекторов сформированным на этапе обучения. Если обнаружено совпадение входного образа хотя бы с одним элементом множества детекторов. Формируется состояние “чужой” и система блокирует вход. Иначе вход разрешается, и система переходит в состояние ожидания.

 ISSN 1562-9945

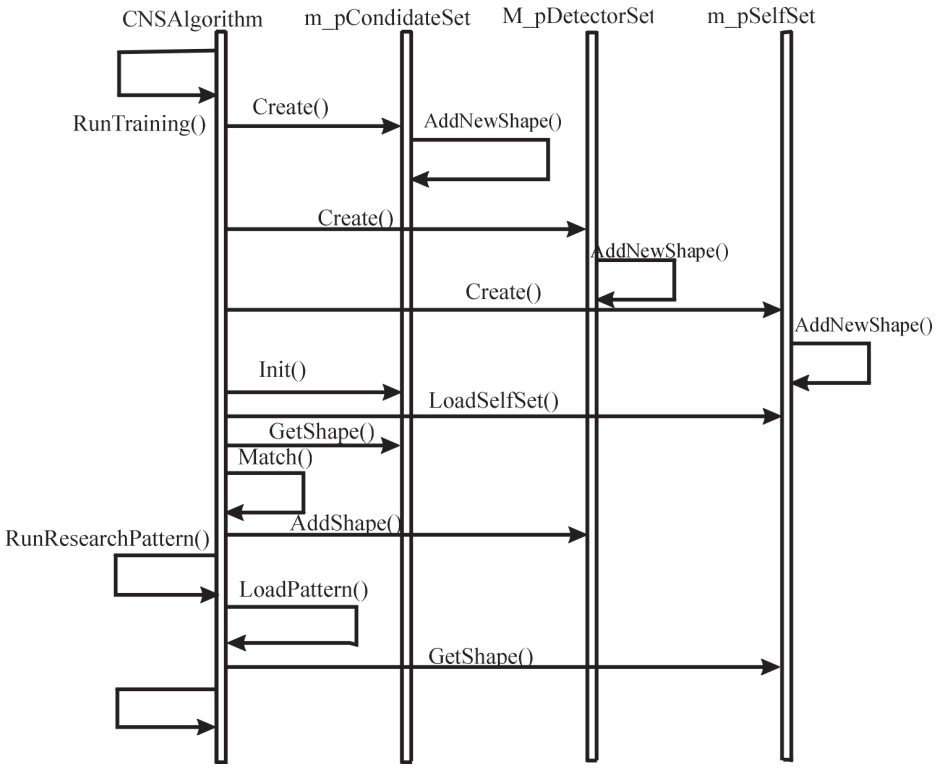


Рисунок 6 - Диаграмма взаимодействия основных объектов

Описание исследовательского прототипа программы реализующей алгоритм отрицательного отбора

Интерфейс:

Общий вид диалогового окна интерфейса программы представлен на рис. 7.

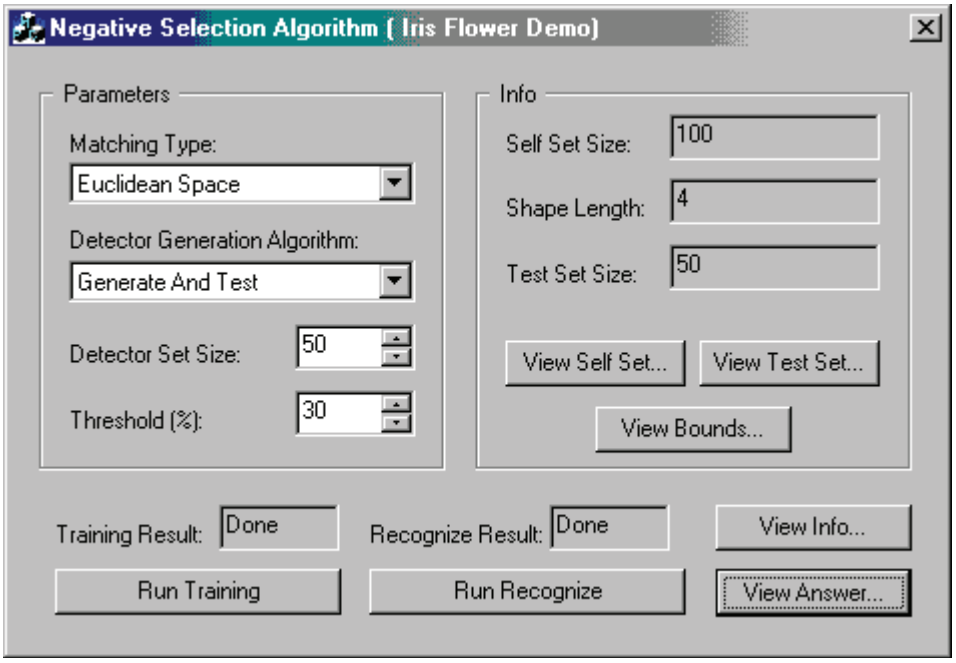


Рисунок 7 - Диалоговое окно интерфейса программы

Группа ***“Parametres”*** обеспечивает настройку 4-х основных параметров алгоритма:

Matching Type. Выбор правила сравнения векторов (рис. 8).

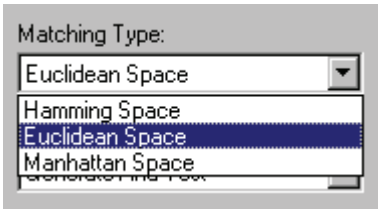


Рис. 8. Меню настройки правила сравнения векторов

Hamming Space – Хеммингово расстояние (используется только при бинарном кодировании векторов)

Euclidian Space – Евклидово расстояние.

Manhattan Space – Манхеттеновское расстояние.

Detector Generation Algorithm – алгоритм генерации множества детекторов

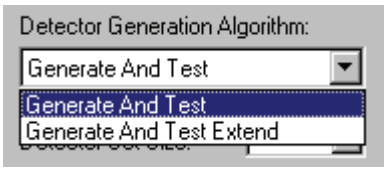


Рис. 9. Выбор алгоритма генерации детекторов

Generate and Test – генерируется один кандидат. Затем проверяется его “непересекаемость” с множеством ***S*** и в случае положительного ответа кандидат становится детектором, иначе он отбрасывается.

– ***Generate and Test Extended*** – аналогично предыдущему алгоритму с тем лишь дополнением, что проверяемый кандидат не должен попасть в зону действия других детекторов.

Detector Set Size – размер множества детекторов.

Threshold – параметр определяющий границы зон действия. Задается в процентах от максимально возможного расстояния между векторами в рамках множества **U**.

Группа **“Info”** содержит информацию о входных данных алгоритма:

Self Set Size – размер обучающей выборки

Shape Length – размерность векторов

Test Set Size – размер распознаваемой выборки

View Self Set – показывает обучающую выборку

 ISSN 1562-9945

View Test Set – показывает распознаваемую выборку

View Bounds – показывает границы множества **U**, а также точность представления векторов

Результаты тестирования

Суть данной задачи заключается в следующем. Имеется три вида цветков ириса: Setosa, Versicolor и Virginica. По 50 экземпляров каждого вида. Из них измерены четыре величины: длина и ширина чашелистика, длина и ширина лепестка. В нашем случае цель состоит в том чтобы научить распознавать по данным этих четырех измерений два вида Setosa, Versicolor как "свои", а третий Virginica как "чужой". В нашем случае необходимо найти такие параметры алгоритма, позволяющий добиться 100 процентного распознавания как "своих" так и "чужих". Мы делали множество "своих" как обучающую выборку, а "чужие" как тестируемую выборку (для контроля в тестовую выборку добавляли несколько экземпляров “своих”.

С целью определения и выявления влияний параметров алгоритма на качество поставленной задачи нами было проведено несколько экспериментов. При этом мы пытались исследовать “экстремальные” ситуации, надеясь, что при них, алгоритмы покажут свои как положительные, так и отрицательные качества при решении поставленной задачи. Из всех групп проведенных экспериментов мы считаем, что наиболее яркими являются следующие две группы. В первой группе мы исследовали влияние параметра определяющего границы зон действия, заданного в процентах от максимально возможного расстояния между векторами ***Threshold*** (Порог %) на качество распознавания “чужих”. При этом было взято минимальное значение размера детектора (Detector Set Size) равное 5.

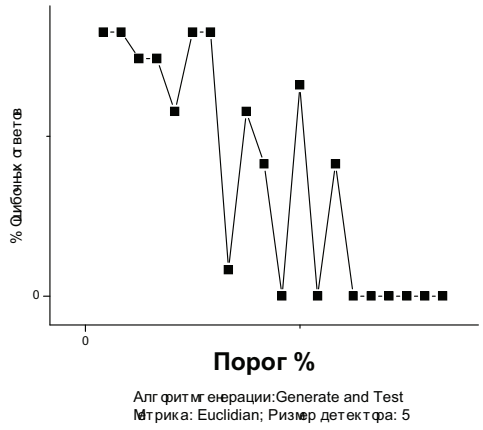


Рис. 10. Процент ошибочных ответов от параметра Threshold при режиме генерации Generate and Test, с использованием Евклидова расстояния

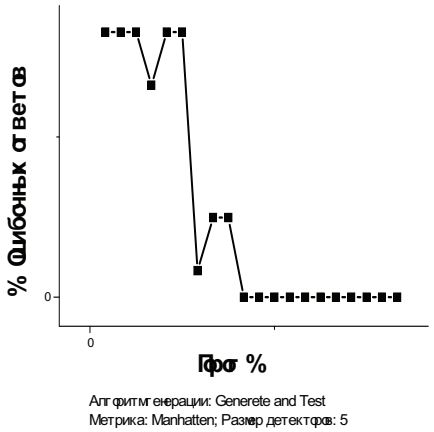


Рис. 11. Процент ошибочных ответов от параметра Threshold при режиме генерации Generate and Test, с использованием Манхетенского расстояния

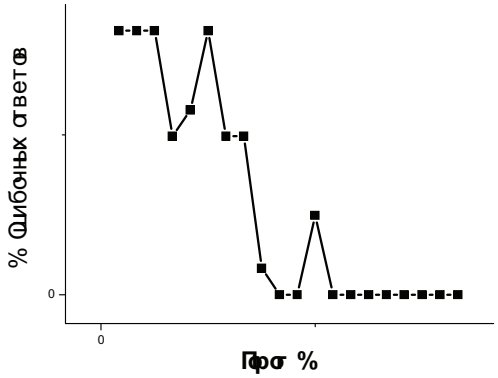


Рис. 12. Процент ошибочных ответов от параметра Threshold (Порог) при режиме генерации Generate and Test Extend, с использованием Евклидова расстояния

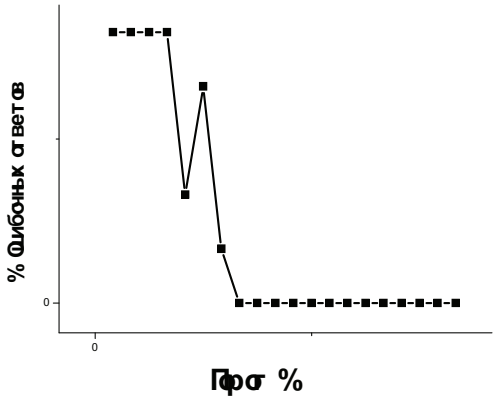


Рис. 13. Процент ошибочных ответов от параметра Threshold (Порог) при режиме генерации Generate and Test Extend, с использованием Манхетенского расстояния

Во второй группе экспериментов мы исследовали влияние множества максимальных размеров детекторов на качество распознавания “чужих” при постоянном минимально возможном значении параметра ***Threshold*** (Порог) и различных значениях метрики и различных видах алгоритма генерации детекторов.

 ISSN 1562-9945

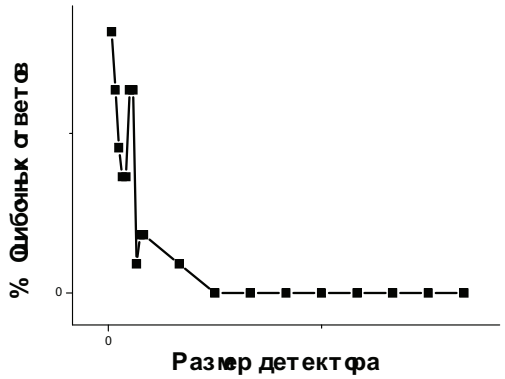


Рис. 14. Процент ошибочных ответов в зависимости от увеличения размера детекторов, при применении алгоритма генерации Generate and Test и Евклидовой метрики.

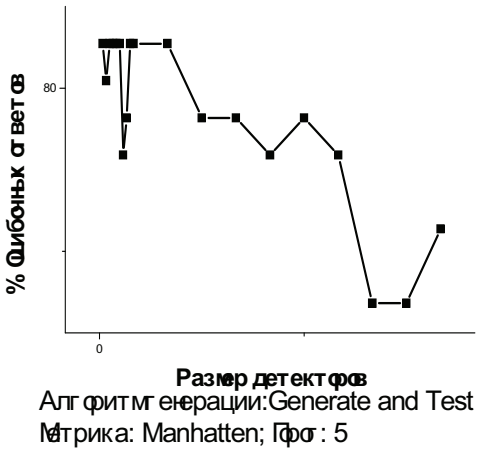


Рис. 16. Процент ошибочных ответов в зависимости от увеличения размера детекторов, при применении алгоритма генерации Generate and Test и Манхетенской метрики.

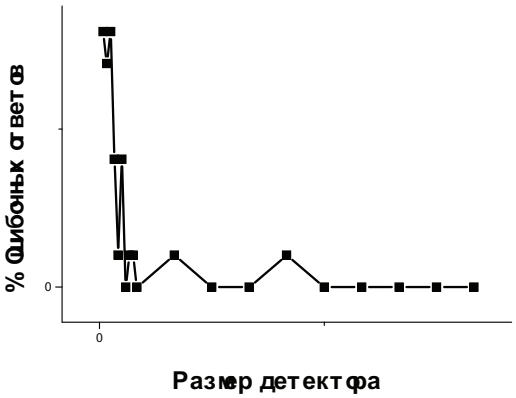


Рис. 17. Процент ошибочных ответов в зависимости от увеличения размера детекторов, при применении алгоритма генерации Generate and Tes Extend и Евклидовой метрики.

Рис. 18. Процент ошибочных ответов в зависимости от увеличения размера детекторов, при применении алгоритма генерации Generate and Test Extend и Манхетеновской метрики.

Для проведения третьей группы экспериментов мы помимо описанной ранее задачи IRIS использовали одну из задач из набора тестов PROBLEM1 задачу **CANCER и “задачу двух спиралей”. Первая представляет собой набор данных, связанных с диагностикой рака грудной железы. Задача заключается в классификации опухоли**

(доброкачественная или злокачественная) на основе полученном с помощью микроскопа наборе характеристик клетки. В качестве входных характеристик подается информация о диаметре группы клеток опухоли, единообразии размера и формы клетки, доли нуклеина, доле хроматина и др. Всего имеется 699 примеров выборки: 9 входов и 2 выхода. Все входы – вещественные значения, 65,5% всей выборки определяют доброкачественную опухоль.

Третья задача представляет собой весьма сложный для различных классификаторов синтетический тест. Обучающий набор данных состоит из 194 значений (x, y), половина из которых определяет точки одной спирали, а другая половина – точки второй спирали. Спирали плотно переплетаются друг с другом. Задача классификатора состоит в том, чтобы научиться различать принадлежность точек плоскости одной из спиралей. Сложность задачи двух спиралей определяется тем, что для подобных данных, граница решения, разделяющая представителей разных классов будет иметь сложный вид.

Таблица 6.1

Общие результаты работы на универсальных параметрах

ЗАДАЧА	ЧИСЛО ИТЕРАЦИЙ																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	min	max
	% ОШИБКИ КЛАСИФИКАЦИИ																
IRIS	18,46	8,46	6,15	5,38	9,23	9,23	6,15	6,92	7,69	8,46	7,69	7,69	7,69	5,38	6,92	5,38	9,23
CANCER	8,86	7,43	5,71	6,0	5,71	5,71	5,43	5,43	6,0	6,0	6,29	6,57	6,86	5,71	6,57	5,43	7,43
SPIRAL	44,24	43,46	50,79	50	49,74	45,29	50,52	48,17	45,29	47,38	50,52	47,91	45,03	47,12	49,21	43,46	50,79

6.2 Выводы

Проанализировав результаты работы программы, мы пришли к заключению, в соответствии с которым универсального набора регулирующих параметров для алгоритма не может быть выявлено, поскольку при решении каждой отдельной задачи необходимо учитывать специфику конкретного набора данных. В программе реализованная часть каблука границ мутации по максимальному и минимальному значению в каждом поле, однако это не может заменить полноценное исследование задачи для получения качественных результатов.

В целому по задачам CANCER и IRIS полученные удовлетворительные результаты. В то же время в один момент на наборе данных к задаче IRIS было получено 100%-ное распознавание, которое говорит об эффективной работе алгоритма при

 ISSN 1562-9945

правильном настраивании параметров относительно каждой конкретной задачи.

Задача SPIRAL была приведена, так как она является классической сложной задачей классификации и распознавания, и показывает необходимость глубокого анализа исследуемых задач и функций.

Обсуждение

Как видно из рисунков 11–14 наиболее оправданным показало использование Манхетеновской метрики при применении обеих видов алгоритмов генерации детекторов (*Generate and Test Extend* и *Generate and Test*). Использование Евклидовой метрики характеризуется требованием увеличения сравнительно высоких значений параметров *Threshold* (т.е. параметра определяющего границы зон действия), при этом необходимо обратить внимание, что данный алгоритм неустойчиво работал при использовании Евклидовых расстояний на средних значениях параметрах *Threshold* ≈ 20 – 70% при обеих алгоритмах генерации детекторов.

На рисунках 15–18 показаны результаты численных экспериментов влияния “больших” размеров детекторов на процент ошибочных ответов при минимальном значении параметра *Threshold*. По данным рисункам видно, что в данном случае наблюдается обратная картина: при использовании Евклидовой метрики, при обеих видах генерации рецепторов наблюдались быстрые, устойчивые и положительные результаты при решении задачи определения “свой/чужой”. При этом необходимо отметить, при использовании Манхеттенской метрики особенно при режиме генерации детекторов *Generate and Test Extend* часто наблюдалось зависание компьютера. При этом время обучения при применении режима *Generate and Test Extend* при обеих видах метрики заметно увеличивалось особенно после увеличении размера детектора до 3000 и более, особенно ярко увеличение длительности режима обучения проявлялось при применении Манхетеновской метрики. Нами эмпирически установлено, что наиболее оптимальными параметрами для модифицированной нами задачи “Ирисы Фишера” являются:

Threshold ≈ 70–80 %

Detector Set Size ≈ 10 –20 %

Режима ***Generate and Test Extend***

Метрика ***Евклидовое расстояние***

ЛИТЕРАТУРА

1. D. Dasgupta (editor). Artificial Immune Systems and Their Applications. A book published by Springer Verlag Inc., January 1999.
2. Грицик В.В., Литвиненко В.І., Цмоць І.Г., Стех С.М. Теоретичні і прикладні проблеми застосування штучних імунних систем// Інформаційні технології і системи. –2003 –Т.6.-№1-2, с.7-45.
3. Литвиненко В.И. Бидюк П.И. Фефелов А.А., Баклан И.В.Программная реализация алгоритма отрицательного отбора для решения задач классификации //Тези доповідей Всеукраїнської конференції МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ, Дніпропетровськ,17-19 листопада 2003 р. с.10
4. Литвиненко В.И. Классифицирующая система на основе механизма отрицательного отбора // Матеріали ІІІ-ї міжнародної науково-практичної конференції “Динаміка наукових досліджень 2004 с.34-36”
5. Литвиненко В.И. Разработка и применение искусственных иммунных систем// Третя міжнародна науково-практична конференція МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ (MIZIS-2005), ТЕЗИ ДОПОВІДЕЙ, Дніпропетровськ, 16-18 листопада 2005 р. с.105.
6. Blake, C., and C.Merz. 1998. UCI reprository of machine learning databases.http://www.ics.uci.edu/~mlearn/MLRepository.html (Accessed 09 July 2001)
7. Баклан И.В., Бидюк П.И., Литвиненко В.И., Фефелов А.А. Архитектура классифицирующей иммунной системы на основе алгоритма клонального отбора// Тези доповідей участників VI Міжнародної науково-практичної конференції студентів, аспирантів та моложих вчених Системний аналіз та інформаційні технології /1-3 липня 2004 р., м.Київ, с.41-42.
8. Грицик В.В., Литвиненко В.І., Опотяк Ю.В., Фефелов А.О., Цмоць І.Г. Використання штучних імунних систем при розвязуванні задач мікроскопичної цитомітрії// Інформаційні технології і системи. –2005 –Т.8.-№1-2, с.37-51.

Получено __.__.2006 г.

 ISSN 1562-9945