

Министерство образования и науки Украины
Харьковский национальный университет радиоэлектроники

На правах рукописи

ХАУСТОВА ЯНА ВЛАДИМИРОВНА
(КУЦЕНКО ЯНА ВЛАДИМИРОВНА)

УДК 004.032.26

**МЕТОДЫ НЕЧЕТКОЙ КЛАСТЕРИЗАЦИИ НА ОСНОВЕ ЯДЕРНЫХ
ФУНКЦИЙ В ЗАДАЧАХ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА
ДАНЫХ**

05.13.23 – системы и средства искусственного интеллекта

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
Бодянский Евгений Владимирович,
доктор технических наук, профессор

Цей примірник дисертаційної роботи
ідентичний за змістом з іншими,
поданими до спеціалізованої вченої ради.

Учений секретар спецради Д 64.052.01

О.А. Винокурова

Харьков – 2016

СОДЕРЖАНИЕ

Введение	5
Раздел 1. Обзор состояния проблемы и постановка задачи исследования	12
1.1 Задача классификации и кластеризации	12
1.2 Алгоритмы кластерного анализа	13
1.2.1 Классификация алгоритмов кластеризации	14
1.2.2 Алгоритм k -средних	14
1.2.3 Применение нейронных сетей в задачах кластеризации	15
1.3 Нечеткая кластеризация	16
1.3.1 Пакетные методы нечеткой кластеризации	20
1.3.2 Адаптивные методы нечеткой кластеризации	26
1.4 Метрики кластерного анализа	28
1.5 Основные парадигмы и правила обучения искусственных нейронных сетей	29
1.5.1 Обучение нейронных сетей, основанное на оптимизации	32
1.5.2 Мгновенно обучаемые нейронные сети	33
1.5.4 Нейронные сети опорных векторов	35
1.5.5 Конкурентные нейронные сети	36
1.7 Постановка задачи исследования	40
Выводы по разделу 1	42
Раздел 2. Многослойные нечеткие кластеризующие системы для анализа потоков данных	43
2.1 Постановка задачи кластеризации	43
2.2 Мягкий вероятностный нечеткий EM-метод кластеризации многомерных данных	44

2.3 Нечеткая вероятностная кластеризация на основе WTA-правила самообучения	48
2.4 Многослойная ядерная кластеризующая нейро-фаззи система и алгоритмы ее самообучения	51
2.4.1 Архитектура ядерной нечеткой кластеризующей сети Т.Кохонена	52
2.5 Метод самообучения кластеризирующей нейро-фаззи системы	57
Выводы по разделу 2	63
Раздел 3. Последовательная кластеризация на основе ядерных нейронных сетей	64
3.1 Ядерная кластеризующая нейронная сеть на основе радиально-базисной нейронной сети	64
3.2 Архитектура ядерной кластеризующейся нейронной сети на основе радиально-базисной нейронной сети	66
3.3 Обучение ядерной самоорганизующейся карты и радиально-базисной нейронной сети	69
3.4 Ядерная кластеризация на основе обобщенной регрессионной нейронной сети и самоорганизующейся карты Кохонена	74
3.4.1 Архитектура ядерной самоорганизующейся карты на основе обобщенной регрессионной нейронной сети	74
3.4.2 Обучение ядерной самоорганизующейся карты на основе обобщенной регрессионной нейронной сети	77
Выводы по разделу 3	85
Раздел 4. Адаптивный метод комбинированного обучения-самообучения нейронных сетей и нейро-фаззи систем	87
4.1 Последовательная ядерная кластеризация на основе нейро-фаззи подхода	87
4.1.1 Обучения ядерной кластеризующей нейро-фаззи системы	94

4.2 Последовательная ядерная нечеткая кластеризация больших массивов данных на основе гибридной системы вычислительного интеллекта	101
4.2.1 Архитектура ядерной нечеткой кластеризующей системы	102
4.2.2 Самообучение ядерной нечеткой кластеризующей системы	104
Выводы по разделу 4	113
Раздел 5. Имитационное моделирование и решение практических задач	115
5.1 Имитационное моделирование ядерной кластеризующей нейронной сети	115
5.2 Имитационное моделирование многослойной гибридной нейро-фаззи системы вычислительного интеллекта	119
5.3 Имитационное моделирование модифицированного EM нечеткого вероятностного метода кластеризации	122
5.4 Имитационное моделирование усовершенствованной искусственной нейронной сети для анализа главных компонент	125
5.5 Решение практической задачи анализа политематических текстов с целью категоризации на кластеры для ускорения информационного поиска	127
Выводы по разделу 5	135
Выводы	137
Список использованных источников	138
Приложение А. Акты о внедрении	154

ВВЕДЕНИЕ

Актуальность темы. В настоящее время для решения широкого класса задач интеллектуального анализа данных, и прежде всего кластеризации, что является неотъемлемой частью общей проблемы Data Mining, известен ряд методов вычислительного интеллекта на основе нейро-фаззи подхода. Разработан мощный математический аппарат, позволяющий решать задачи кластеризации в различных областях: медицине, науке и технике, при управлении производством и др.

Однако, большинство из известных методов по своей сути являются четкими процедурами, где предполагается, что кластеры линейно делимы, данные поступают в пакетном режиме. Также разработан ряд методов нечеткой кластеризации, где предполагается, что каждое отдельное наблюдение может с разными уровнями принадлежности относиться к нескольким классам. Однако, при этом, формируемые кластеры, имеют выпуклую форму. Вместе с тем, сейчас на первый план выходят задачи, связанные с Dynamic Data Mining, Data Stream Mining и Big Data, когда данные поступают на обработку в виде потока информации. Более того, в ситуациях, когда кластеры могут пересекаться и иметь произвольную форму, возникает необходимость разработки методов нечеткой кластеризации в их рекуррентной форме. Эта проблема может быть решена на основе синтеза нечетких методов и ядерного подхода, связанного с гипотезой, сформулированной Кавером, которая утверждает, что если задача линейно неразделима в исходном пространстве, то она может быть решена в пространстве повышенной размерности.

Таким образом, на сегодняшний день является актуальной научная задача разработки новых методов нечеткой ядерной кластеризации предназначенных для обработки данных в on-line режиме, когда данные

поступают на обработку последовательно, одно за другим, а кластеры могут пересекаться и иметь произвольную форму.

Связь работы с научными программами, планами, темами. Диссертационная работа выполнена в рамках госбюджетных НИР: «Нейро-фаззи системы для текущей кластеризации и классификации последовательностей данных в условиях их искаженности отсутствующими и аномальными наблюдениями» (№ГР 0113U000361); «Динамический интеллектуальный анализ последовательностей нечеткой информации в условиях существенной неопределенности на основе гибридных систем вычислительного интеллекта» (№ГР 0116U002539). В рамках указанных НИР соискателем разработаны методы нечеткой ядерной кластеризации, которые предназначены для обработки данных в on-line режиме, когда данные поступают на обработку последовательно, одно за одним, а кластеры могут пересекаться.

Цели и задачи исследования. Целью настоящего исследования является разработка методов нечеткой кластеризации на основе ядерных нейронных сетей и нейро-фаззи систем, настраивающих свою архитектуру в процессе обучения-самообучения в условиях пересекающихся кластеров произвольной формы. Достижение поставленной цели осуществляется путем решения следующих основных задач:

- анализ существующих методов и подходов для кластеризации данных разной физической природы;
- разработка модификации метода EM (ожидание-максимизация) кластеризации данных, которые последовательно поступают на обработку одно за одним в on-line режиме;
- разработка искусственной нейронной сети, которая объединяет в себе идеи ядерных систем и самообучения, и построена на основе радиально-базисной нейронной сети и самоорганизующейся карты;

- разработка архитектуры гибридной нейронной сети на основе обобщенной регрессионной сети и самоорганизующейся карты Т. Кохонена;
- разработка многослойной нейро-фаззи системы, являющейся гибридом системы Ванга-Менделя и нечеткой кластеризующей самоорганизующейся сети;
- имитационное моделирование разработанных методов и моделей, решение с их помощью ряда практических задач нечеткой кластеризации данных.

Объект исследования – процесс обработки данных, поступающих одно за одним, в последовательном режиме с помощью нечетких методов кластеризации на основе ядерных функций.

Предмет исследования – методы нечеткой кластеризации на основе ядерных функций в задачах интеллектуального анализа данных.

Методы исследования базируются на теории вычислительного интеллекта, а именно на методах теории искусственных нейронных сетей и теории нечеткой логики для построения архитектур гибридных нейро-фаззи сетей, позволяющих проводить нечеткую кластеризацию; теории оптимизации и статистического анализа для синтеза эффективных методов нечеткой кластеризации и методов обучения и самоорганизации гибридных нейро-фаззи сетей. Имитационное моделирование применяется для проверки качества кластеризации с использованием синтезированных методов и архитектур.

Научная новизна полученных результатов:

1. Впервые предложены ядерные кластеризующие нейронные сети, которые основаны на радиально-базисной нейронной сети и обобщенной регрессионной сети, позволяющие обрабатывать потоки данных различной физической природы в последовательном режиме
2. Впервые предложено многослойную гибридную нейро-фаззи систему вычислительного интеллекта на основе системы Ванга-Менделя и

нечеткой кластеризующей самоорганизующейся сети, что позволяет в процессе самообучения настраивать не только параметры, но и архитектуру в on-line режиме и решать задачи кластеризации потока данных в условиях априорно неизвестной формы кластеров и уровней их пересечения.

3. Усовершенствован метод кластеризации EM за счет использования ядерных функций специального вида, что позволяет в отличие от стандартного подхода решать задачу кластеризации в условиях пересекающихся кластеров с расчетом оценки принадлежности каждого наблюдения к каждому кластеру.

4. Усовершенствована искусственная нейронная сеть для анализа главных компонент путем введения дополнительных слоев ядерных функций для повышения размерности входного пространства, что позволило обрабатывать информацию, которая содержится в классах произвольной формы.

Практическое значение полученных результатов. Предложенные в работе методы нечеткой кластеризации на основе ядерных функций и нейро-фаззи системы вычислительного интеллекта предназначены для on-line кластеризации потока информации в условиях неопределенности как о форме классов, так и об уровне их пересечения. Введенные нейро-фаззи системы достаточно просты в численной реализации и позволяют решать задачи динамического интеллектуального анализа данных (DDM) и интеллектуального анализа потоков данных (DSM). Использование разработанных методов кластеризации позволило повысить эффективность решения задач кластеризации и анализа документации программного обеспечения, уменьшить на 30% время на поиск того или иного типа документации, а также решить задачу автоматической каталогизации документации программного обеспечения в системах интеллектуального анализа данных в ООО «Академия СМАРТ».

Также основные результаты диссертационной работы используются в учебном процессе Харьковского национального университета радиоэлектроники на кафедре искусственного интеллекта в курсах «Искусственные нейронные сети: архитектуры, обучение и применение» и «Нейросетевые методы вычислительного интеллекта».

Личный вклад соискателя. Основные положения и результаты диссертационной работы получены соискателем лично. Вклад автора в публикациях, опубликованных в соавторстве такой: [1] - архитектура ядерной гибридной искусственной нейронной сети, на основе радиально-базисной нейронной сети и самоорганизующейся карты Кохонена; [2] - архитектура гибридной нейронной сети для ядерной кластеризации данных; [3] - архитектура для нечеткой ядерной кластеризации данных на основе EM-метода; [4] - архитектура гибридной нейро-фаззи системы вычислительного интеллекта для нечеткой on-line кластеризации потоков информации в условиях априорной неопределенности и произвольной формы кластеров; [5] - введение в ядерную эволюционирующую нейронную сеть слой радиально-базисных функций и слой восстановления входного пространства; [6] - гибридная архитектура на основе карты Кохонена и мягкого EM-метода; [7] - в качестве функции плотности предложены ядра Епанечникова; [8] - модель эволюционирующей нейро-фаззи системы; [9] - архитектура ядерной карты Кохонена; [10] - модификация EM-метода для вероятностной кластеризации; [11] - слой ядерных функций активации для повышения размерности исходного пространства; [12] - гибридная система для обработки больших массивов данных; [13] - архитектура ядерной гибридной нейронной сети.

Апробация результатов диссертации. Основные результаты диссертационной работы были представлены и обсуждены на: 20-м Международном молодежном форуме «Радиоэлектроника и молодежь в XXI веке» (Харьков, 2016), Международной научно-практической конференции «Интеллектуальные системы принятия решений и проблемы

вычислительного интеллекта» (Железный порт, 2015, 2016), 8-й Международной школе-семинаре «Теория принятия решений» (Ужгород, 2016), Xth International Scientific and Technical Conference «Computer Science and Information Technologies» (CSIT 2015) (Lviv, 2015), 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP 2016) (Lviv, 2016).

Публикации. Основные положения диссертационной работы опубликованы в 13 научных работах: в том числе 5 статьях, среди которых 4 опубликовано в периодических изданиях по техническим наукам, включенных в перечень МОН Украины и 1 за рубежом (3 статьи входят в международные наукометрические базы), 8 публикаций в материалах международных научных конференций и форумов (2 доклада входят в международные наукометрические базы Scopus, Web of Science).

РАЗДЕЛ 1

ОБЗОР СОСТОЯНИЯ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

1.1 Задача кластеризации

На сегодняшний день одной из основных задач в области интеллектуального анализа данных является задача кластеризации, то есть задача разбиения исходных данных на однородные в некотором смысле группы в режиме обучения без учителя (самообучения) [14].

Кластеризация – одна из наиболее важных проблем неконтролируемого обучения. Как и каждая проблема такого типа, она имеет дело с обнаружением скрытой структуры в совокупности данных. Объединение в кластеры – это процесс организации объектов в группы, члены которых подобны в некотором роде. Кластер – это множество объектов, близких между собой в смысле некоторой меры сходства. В пространстве переменных кластеры представляют собой скопления точек (объектов) различной формы (рис. 1.1).

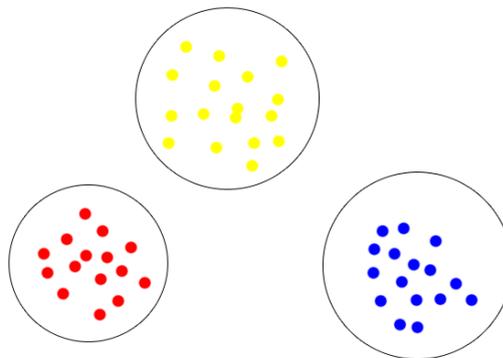


Рисунок 1.1 – Кластеризация

В случае, представленном на рисунке 1.1, мерой сходства является расстояние: два или более объектов принадлежат к одному кластеру, если они «близки» в соответствии с заданным расстоянием. Это кластеризация на основании расстояния.

Другой тип кластеризации – концептуальная кластеризация: два или более объектов принадлежат к одному кластеру, если они определяют понятие, общее для всех объектов. Другими словами, объекты группируются так, чтобы соответствовать определенному концепту.

Кластерный анализ занимает одно из центральных мест среди методов анализа данных и представляет собой совокупность подходов, методов и алгоритмов, предназначенных для нахождения некоторого разбиения исследуемой совокупности объектов на подмножества относительно сходных, похожих между собой объектов [15]. При этом исходным допущением для выделения таких подмножеств, получивших специальное название “кластер”, которые так же иногда называют “таксонами” или просто классами, служит лишь неформальное предположение о том, что объекты, относимые к одному кластеру, должны иметь большее сходство между собой, чем с объектами из других кластеров [16].

1.2 Алгоритмы кластерного анализа

1.2.1 Классификация алгоритмов кластеризации

Алгоритмы кластеризации могут быть классифицированы следующим образом:

- исключаяющая/нечёткая кластеризация;
- иерархическая/неиерархическая кластеризация;
- вероятностная кластеризация.

В случае исключаяющей кластеризации данные группируются особым образом так, что если некое наблюдение принадлежит к определенному

кластеру, то оно не может принадлежать к другому кластеру. Напротив, второй тип – нечёткая (фаззи, fuzzy) кластеризация – использует нечёткие наборы для кластеризации данных таким образом, что каждая точка может принадлежать к двум или более кластерам с разной степенью принадлежности. Результатом фаззи-кластеризации является построение матрицы нечёткого разбиения. Иерархические алгоритмы применяются, когда необходимо решить задачу таксономии. В ходе работы алгоритма крупные кластеры разбиваются на более мелкие (или мелкие кластеры объединяются в крупные). Результатом таксономии является древообразная иерархическая структура.

1.2.2 Алгоритм k -means

Одним из наиболее распространенных алгоритмов кластеризации является алгоритм k -средних (k -means), также называемый быстрым кластерным анализом. Полное описание алгоритма можно найти в работе Хартигана и Вонга [17]. В отличие от иерархических методов, которые не требуют предварительных предположений относительно числа кластеров, для возможности использования этого метода необходимо иметь информацию о наиболее вероятном количестве кластеров.

Алгоритм k -средних строит k кластеров, расположенных на возможно больших расстояниях друг от друга. Основной тип задач, которые решает алгоритм k -средних – наличие предположений (гипотез) относительно числа кластеров, при этом они должны быть различны настолько, насколько это возможно. Выбор числа k может базироваться на результатах предшествующих исследований, теоретических соображениях или интуиции.

Общая идея алгоритма: набор наблюдений сопоставляется k кластерам так, что среднее значение объектов в кластере (для всех наблюдений) максимально возможно отличаются друг от друга для каждого из кластеров.

Достоинствами алгоритма k -средних являются: простота использования; быстрота использования; понятность и прозрачность алгоритма.

Недостатки алгоритма k -средних:

- алгоритм слишком чувствителен к выбросам, которые могут исказить среднее. Возможным решением этой проблемы является использование модификации алгоритма – алгоритм k -медиан (k -medians);

- алгоритм может медленно работать на больших базах данных. Возможным решением данной проблемы является использование части выборки данных.

1.2.3 Применение нейронных сетей в задаче кластеризации

Методы кластеризации при помощи нейронных сетей являются развитием классических методов кластеризации.

Метод кластеризации с помощью сети Кохонена содержит в своей основе рассмотренный выше метод k -средних.

Нейронные сети Кохонена или самоорганизующиеся карты Кохонена (Kohonen's Self-Organizing Maps) предназначены для решения задач классификации без учителя [18]. Это двухслойная нейронная сеть, содержащая входной слой (слой входных нейронов) и слой Кохонена (слой активных нейронов). Слой Кохонена может быть одномерным, двумерным или трехмерным. В первом случае активные нейроны расположены в цепочку. Во втором случае они образуют двумерную сетку (обычно в форме квадрата или прямоугольника), а в третьем случае они образуют трехмерную конструкцию.

1.3 Нечёткая кластеризация

Теория нечётких множеств, основные идеи которой были предложены американским математиком Лотфи Заде в 1965 году, позволяет описывать качественные, неточные понятия и наши знания об окружающем мире, а также оперировать этими знаниями с целью получения новой информации [19, 20]. Основанные на этой теории методы построения информационных моделей существенно расширяют традиционные области применения компьютерных систем и образуют самостоятельное направление научно-прикладных исследований, которое получило специальное название – нечёткое моделирование. Оно оказывается особенно эффективным, когда в описании технических систем и процессов присутствует неопределенность, которая затрудняет или даже исключает применение точных количественных методов.

В области анализа данных нечёткое моделирование часто позволяет получать более адекватные результаты по сравнению с результатами, которые основываются на использовании традиционных аналитических моделей и алгоритмов.

Нечёткое множество (fuzzy set) представляет собой совокупность элементов произвольной природы, относительно которых нельзя с полной определенностью утверждать – принадлежит ли тот или иной элемент рассматриваемой совокупности данному множеству или нет [21].

Взаимосвязь между кластерным анализом и теорией нечётких множеств основана на том обстоятельстве, что при решении задач структуризации сложных систем большинство объектов оказываются размытыми по своей природе. Эта размытость состоит в том, что переход от принадлежности к непринадлежности элементов к данным классам скорее постепенен, чем скачкообразен.

Требование нахождения однозначной кластеризации элементов исследуемой проблемной области является достаточно грубым и жестким, особенно при решении плохо или слабо структурированных задач системного анализа. Методы нечёткой кластеризации ослабляют это требование. Ослабление требования осуществляется за счет введения в рассмотрение нечётких кластеров и соответствующих им функций принадлежности, принимающих значения из интервала $[0, 1]$. На рисунке 1.2 изображена нечёткая кластеризация с различными степенями принадлежности наблюдений [22].

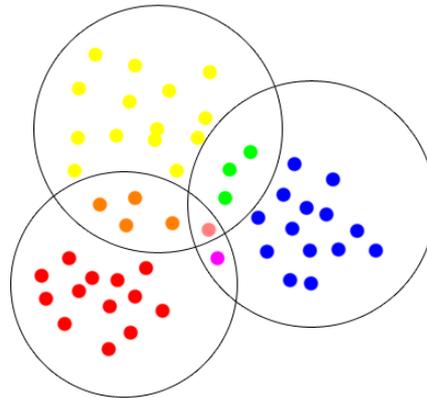


Рисунок 1.2 – Нечёткая кластеризация

Для любой меры сходства величина принадлежности наблюдения кластеру зависит от схожести объекта и прототипа этого кластера. В случае если мерой сходства является расстояние, то величина принадлежности объекта обратно пропорциональна его расстоянию до центроида кластера. Сумма принадлежностей наблюдения кластерам в любой момент времени должна быть равна 1.

Таким образом, в общем случае задачей нечёткой кластеризации является нахождение нечёткого разбиения или нечёткого покрытия множества элементов исследуемой совокупности, которые образуют

структуру нечётких кластеров, присутствующих в рассматриваемых данных [23]. Эта задача сводится к нахождению степеней принадлежности элементов искомым нечётким кластерам, которые в совокупности и определяют нечёткое разбиение или нечёткое покрытие исходного множества рассматриваемых элементов.

Основные идеи алгоритма для решения задачи нечёткой кластеризации были предложены Дж. К. Данном в 1973 году [24]. В дальнейшем алгоритм был развит Джеймсом Бездеком [25] и получил название нечётких *c*-средних (FCM).

Наряду с традиционным вероятностным подходом к нечеткой кластеризации, когда каждый объект с некоторой вероятностью принадлежит к каждому из кластеров, существует возможностный подход к кластерному анализу. Возможностная кластеризация также рассматривает нечёткие кластеры и соответствующие им функции принадлежности, принимающие значения из интервала $[0, 1]$. Разница состоит в том, что вероятностная кластеризация подразумевает наличие строгого ограничения, что сумма принадлежностей объекта ко всем кластерам равна 1, а возможностный кластерный анализ не подразумевает подобного ограничения.

Преимущество возможностного кластерного анализа над вероятностным состоит в том, что объекты, которые имеют низкий уровень сходства с любым из кластеров (outliers), будут иметь значение принадлежности близкое нулю для всех кластеров, в то время как вероятностный нечеткий кластерный анализ будет явно отдавать предпочтение одному или нескольким кластерам (хотя все они должны быть достаточно плохи).

Часто результаты возможностной кластеризации обращают в вероятностный вид при помощи нормализации, предварительно избавившись от outlier-объектов. Обычно, результаты полученные таким образом, лучше результатов, полученных при прямом использовании вероятностной

кластеризации (однако, это также зависит от исходных данных, от конкретных алгоритмов нечеткой кластеризации и инициализации их параметров) [26].

Алгоритм FCM имеет итеративный характер последовательного улучшения некоторого нечёткого разбиения, которое задается пользователем или формируется автоматически по некоторому эвристическому правилу [25]. На каждой из итераций рекуррентно пересчитываются значения функций принадлежности объектов нечётким кластерам и их типичные представители (центроиды).

Алгоритм закончит работу в случае, когда произойдет выполнение заданного априори некоторого конечного числа итераций, либо когда минимальная абсолютная разность между значениями функций принадлежности (или центроидами кластеров) на двух последовательных итерациях не станет меньше некоторого априори заданного значения.

Формально алгоритм FCM определяется в форме итеративного выполнения следующей последовательности шагов:

а) задаётся количество искомым нечётких кластеров m , максимальное количество итераций алгоритма n , параметр сходимости алгоритма ε . В качестве текущего нечёткого разбиения на первой итерации алгоритма для матрицы данных D задается некоторое исходное нечёткое разбиение на m непустых нечётких кластеров, которые описываются совокупностью функций принадлежности;

б) для исходного текущего нечёткого разбиения рассчитываются центры нечётких кластеров и значение целевой функции. Количество выполненных итераций устанавливается в 1;

в) формируется новое нечёткое разбиение исходного множества объектов кластеризации на m непустых нечётких кластеров, характеризующихся совокупностью функций принадлежности;

г) для нового нечёткого разбиения рассчитываются центры нечётких кластеров и значение целевой функции;

д) если количество выполненных итераций превышает заданное число n или же значение модуля разности между значениями функций принадлежности (или центроидами кластеров) на двух последних итерациях меньше значения параметра сходимости алгоритма ε , то в качестве искомого результата нечёткой кластеризации принимается последнее нечёткое разбиение и выполнение алгоритма прекращается.

Алгоритм FCM по своему характеру относится к приближенным алгоритмам поиска экстремума для целевой функции при наличии ограничений. В результате выполнения данного алгоритма определяется локально-оптимальное нечёткое разбиение, которое описывается совокупностью функций принадлежности.

1.3.1 Пакетные методы нечеткой кластеризации

Методы кластеризации, основанные на целевых функциях [25], предназначены для решения задачи кластеризации путем оптимизации некоторого наперед заданного критерия качества кластеризации и являются наиболее строгими с математической точки зрения.

Вероятностный подход. Целевая функция, подлежащая минимизации при вероятностном подходе, имеет вид:

$$E(w_j(k), c_j) = \sum_{k=1}^N \sum_{j=1}^m w_j^{\beta}(k) d^2(x(k), c_j) \quad (1.1)$$

при ограничениях

$$\sum_{j=1}^m w_j = 1, \quad k = 1, \dots, N, \quad (1.2)$$

$$0 < \sum_{j=1}^m w_j < N, \quad j = 1, \dots, m, \quad (1.3)$$

где $w_j(k) \in [0,1]$ – уровень принадлежности вектора $x(k)$ к j -му кластеру;

c_j – прототип (центр) j -го кластера;

β – неотрицательный параметр, называемый «фазификатором» («fuzzyfier»), обычно $\beta = 2$;

$d(x(k), c_j)$ – расстояние между $x(k)$ и c_j в принятой метрике.

Результатом кластеризации предполагается $N \times h$ матрица нечеткого разбиения (fuzzy partitioning matrix)

$$W = \{w_j(k)\}.$$

Необходимо заметить, что поскольку элементы матрицы W могут рассматриваться как вероятности гипотез принадлежности векторов данных определенным кластерам, процедуры, порождаемые (1.1) при ограничениях (1.1), (1.3), называются вероятностными методами кластеризации [28].

В качестве функции расстояния $d(x(k), c_j)$ обычно выбирается расстояние Минковского в L^p метрике [29]:

$$d^2(x(k), c_j) = \|x_i(k) - c_{ji}\|_{L^p}^p = \left(\sum_{i=1}^n |x_i(k) - c_{ji}|^p \right)^{\frac{1}{p}}, \quad p \geq 1,$$

где $x_i(k)$ – i -я компонента ($n \times 1$) вектора $x(k)$;

c_{ji} – i -я компонента ($n \times 1$) вектора c_j .

Рассмотрим функцию Лагранжа

$$\begin{aligned} L(w_j(k), c_j, \lambda(k)) &= \sum_{k=1}^N \sum_{j=1}^m w_j^\beta(k) d^2(x(k), c_j) + \sum_{k=1}^N \lambda(k) \left(\sum_{j=1}^m w_j(k) - 1 \right) = \\ &= \sum_{k=1}^N \left(\sum_{j=1}^m w_j^\beta(k) d^2(x(k), c_j) + \lambda(k) \left(\sum_{j=1}^m w_j(k) - 1 \right) \right), \end{aligned} \quad (1.4)$$

где $\lambda(k)$ – неопределенный множитель Лагранжа, обеспечивающий выполнение условий (1.2), (1.3).

Решая систему Куна-Таккера

$$\begin{cases} \frac{\partial L(w_j(k), c_j, \lambda(k))}{\partial w_j(k)} = 0, \\ \frac{\partial L(w_j(k), c_j, \lambda(k))}{\partial \lambda(k)} = 0, \\ \nabla_{c_j} L(w_j(k), c_j, \lambda(k)) = 0, \end{cases} \quad (1.5)$$

несложно получить искомое решение в виде [30]:

$$w_j^{pr}(k) = \frac{(d^2(x(k), c_j))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (d^2(x(k), c_l))^{\frac{1}{1-\beta}}}, \quad (1.6)$$

$$\lambda(k) = - \left(\sum_{l=1}^m (\beta d^2(x(k), c_l))^{\frac{1}{1-\beta}} \right)^{1-\beta}, \quad (1.7)$$

$$c_j^{pr} = \frac{\sum_{k=1}^N w_j^\beta(k) x(k)}{\sum_{k=1}^N w_j^\beta(k)}. \quad (1.8)$$

Уравнения (1.7) – (1.8) порождают широкий класс процедур кластеризации.

При $\beta = p = 2$, т.е. евклидовом расстоянии

$$d^E(x(k), c_j) = \|x(k) - c_j\| = \sqrt{(x(k) - c_j)^T (x(k) - c_j)}, \quad (1.9)$$

Получаем достаточно простую и эффективную процедуру кластеризации нечетких с-средних Бездека [25]:

$$w_j^{pr}(k) = \frac{\|x(k) - c_j\|^{-2}}{\sum_{l=1}^m \|x(k) - c_l\|^{-2}}, \quad (1.10)$$

$$c_j^{pr}(k) = \frac{\sum_{k=1}^N w_j^2(k) x(k)}{\sum_{k=1}^N w_j^2(k)}, \quad (1.11)$$

$$\lambda(k) = - \sum_{l=1}^m \left(\frac{\|x(k) - c_l\|^{-2}}{2} \right)^{-1}.$$

К вероятностным методам кластеризации относятся также процедуры Густафсона-Кесселя [30], Гата-Гевы [31], и ряд других. Несмотря на малую вычислительную сложность, процедура (1.10) - (1.11) обладает недостатком,

который заключается в необходимости выполнения общего для всех вероятностных методов нечеткой кластеризации условия (1.2).

В простейшем случае двух кластеров ($m=2$) несложно видеть, что наблюдение $x(k)$, равноправно принадлежащее обоим кластерам, и наблюдение $x(p)$, не принадлежащее ни одному из них, имеют одинаковые уровни принадлежности $w_1^{pr}(k) = w_2^{pr}(k) = w_1^{pr}(p) = w_2^{pr}(p) = 0.5$. Очевидно, что это обстоятельство может существенно снизить качество классификации в некоторых случаях. В то же время возможностный подход к нечеткой кластеризации [32-34] помогает избежать описанной выше ситуации и тем самым улучшить качество классификации.

Возможностный подход. При возможностном подходе к кластеризации минимизируемый критерий имеет вид

$$E(w_j(k), c_j) = \sum_{k=1}^N \sum_{j=1}^m w_j^\beta(k) d^2(x(k), c_j) + \sum_{j=1}^m \mu_j \sum_{k=1}^N (1 - w_j(k))^\beta, \quad (1.12)$$

где скалярный параметр $\mu_j > 0$ определяет расстояние, на котором уровень принадлежности принимает значение 0.5. т.е. если $d^2(x(k), c_j) = \mu_j$, то $w_j(k) = 0.5$.

Минимизация критерия (1.12) по параметрам $w_j(k)$, c_j , μ_j приводит к следующей системе уравнений:

$$\left\{ \begin{array}{l} \frac{\partial E(w_j(k), c_j)}{\partial w_j(k)} = 0, \\ \frac{\partial E(w_j(k), c_j)}{\partial \mu(k)} = 0, \\ \nabla_{c_j} E(w_j(k), c_j) = 0. \end{array} \right. \quad (1.13)$$

Решение первых двух уравнений приводит к известному результату:

$$w_j^{pos}(k) = \left(1 + \left(\frac{d^2(x(k), c_j)}{\mu_j} \right)^{\frac{1}{\beta-1}} \right)^{-1}, \quad (1.14)$$

$$\mu_j = \frac{\sum_{k=1}^N w_j^\beta(k) d^2(x(k), c_j)}{\sum_{k=1}^N w_j^N(k)}. \quad (1.15)$$

Решение третьего уравнения системы (1.13) для евклидовой нормы имеет вид:

$$c_j^{pos} = \frac{\sum_{k=1}^N w_j^\beta(k) x(k)}{\sum_{k=1}^N w_j^\beta(k)}. \quad (1.16)$$

Можно видеть, что возможностные и вероятностные методы достаточно похожи и переходят в друг друга заменой выражения (1.14) на формулу (1.6), и наоборот. Общим недостатком рассмотренных методов является их вычислительная сложность и невозможность работы в реальном времени.

Работа процедуры (1.6) - (1.7) начинается с задания начальной (обычно случайной) матрицы разбиения W^0 . На основе ее рассчитывается начальный набор прототипов c_j^0 , которые затем используются для вычисления новой матрицы W^1 . Затем в пакетном режиме пересчитываются c_j^1 ,

$W^2, \dots, W^t, c_j^t, W^{t+1}$ и так далее, пока разность $\|W^{t+1} - W^t\|$ не станет меньше некоторого наперед заданного порога ε . Таким образом, вся имеющаяся выборка данных обрабатывается многократно.

Решение, полученное с помощью вероятностного метода, рекомендуется использовать в качестве начальных условий для возможностного метода (1.14) - (1.15) [33,35].

1.3.2 Адаптивные методы нечеткой кластеризации

Вероятностный подход. Анализ уравнения (1.6) показывает, что для расчета уровней принадлежности $w_j(k)$ вместо функции Лагранжа (1.4) можно использовать его локальную модификацию:

$$L_k(w_j(k), c_j, \lambda(k)) = \sum_{j=1}^m w_j^\beta(k) d^2(x(k), c_j) + \lambda(k) \left(\sum_{j=1}^m w_j(k) - 1 \right). \quad (1.17)$$

Оптимизация выражения (1.17) с помощью Эрроу-Гурвица-Удзавы [34, 36] дает следующую процедуру:

$$w_j^{pr}(k) = \frac{(d^2(x(k), c_j(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (d^2(x(k), c_l(k)))^{\frac{1}{1-\beta}}}, \quad (1.18)$$

$$\begin{aligned} c_j^{pr}(k+1) &= c_j^{pr}(k) - \eta(k) \nabla_{c_j} L_k(w_j(k), c_j^{pr}(k), \lambda(k)) = \\ &= c_j^{pr}(k) - \eta(k) w_j^\beta(k) d(x(k+1), c_j^{pr}(k)) \nabla_{c_j} d(x(k+1), c_j^{pr}(k)), \end{aligned} \quad (1.19)$$

где $\eta(k)$ - параметр скорости обучения;

$c_j^{pr}(k)$ - прототипы j -го кластера, вычисляемые на выборке из k наблюдений.

Процедура (1.18)-(1.19) похожа на процедуру обучения Чанга-Ли [37] и для $\beta = p = 2$ совпадает с градиентной процедурой кластеризации Парка-Дэггера [38]:

$$w_j^{pr}(k) = \frac{\|x(k) - c_j(k)\|^{-2}}{\sum_{l=1}^m \|x(k) - c_l(k)\|^{-2}},$$

$$c_j^{pr}(k+1) = c_j^{pr}(k) - \eta(k)w_j^2(k)(x(k+1) - c_j^{pr}(k)). \quad (1.20)$$

Возможностный подход. В рамках возможностного подхода локальный критерий принимает форму

$$E_k(w_j(k), c_j) = \sum_{j=1}^m w_j^\beta(k) d^2(x(k), c_j) + \sum_{j=1}^m \mu_j (1 - w_j(k))^\beta,$$

а результат его оптимизации имеет вид

$$\mu_j(k+1) = \frac{\sum_{p=1}^k w_j^\beta(p) d^2(x(p), c_j(k+1))}{\sum_{p=1}^k w_j^\beta(p)}. \quad (1.21)$$

$$c_j^{pos}(k+1) = c_j^{pos}(k) - \eta(k)w_j^\beta(k) d(x(k+1), c_j^{pos}(k)) \nabla_{c_j} d(x(k+1), c_j^{pos}(k)),$$

$$w_j^{pos}(k) = \left(1 + \left(\frac{d^2(x(k), c_j(k))}{\mu_j(k)} \right)^{\frac{1}{\beta-1}} \right)^{-1}, \quad (1.22)$$

В квадратичном случае (при $\beta = 2$), процедура (1.21)-(1.22) преобразуется в достаточно простую конструкцию

$$w_j^{pos}(k) = \frac{\mu_j(k)}{\mu_j(k) + \|x(k) - c_j(k)\|^2}, \quad (1.23)$$

$$c_j^{pos}(k+1) = c_j^{pos}(k) - \eta(k)w_j^2(k)(x(k+1) - c_j^{pos}(k)),$$

$$\mu_j(k+1) = \frac{\sum_{p=1}^k w_j^2(p) \|x(p) - c_j(k+1)\|^2}{\sum_{p=1}^k w_j^2(p)}. \quad (1.24)$$

1.4 Метрики кластерного анализа

Сходство или различие между объектами кластеризации устанавливается в зависимости от выбранного метрического расстояния между ними. Если объект описывается n свойствами (признаками), то он может быть представлен как точка в n -мерном пространстве, и сходство с другими наблюдениями будет определяться как соответствующее расстояние. Мерой сходства (подобия) называется любая функция, которая может определить значение подобия или расстояние между двумя объектами в данном пространстве. Сходство между двумя объектами обратно пропорционально расстоянию между ними. Метрика – функция,

определяющая расстояние в метрическом пространстве. В данном пространстве определено расстояние между любой парой элементов.

В таблице 1.2 приведены основные метрики, используемые в задаче кластеризации.

Таблица 1.2 – Метрики, используемые в кластерном анализе

Наименование метрики	Тип признаков	Формула оценки меры близости
Евклидово расстояние	Количественные	$d_{ik} = \left(\sum_{j=1}^N (x_{ij} - x_{kj})^2 \right)^{\frac{1}{2}}$
Мера сходства Хэмминга	Качественные	$\mu_{ij}^H = \frac{n_{ik}}{N}$, где n_{ik} – число совпадающих признаков у образов X_i и X_k .
Манхэттенская метрика	Количественные	$d_{ik} = \sum_{j=1}^N x_{ij} - x_{kj} $
Расстояние Махаланобиса	Количественные	$d_{ik}^M = (x_{ij} - x_{kj})^T W^{-1} (x_{ij} - x_{kj})$, где W – ковариационная матрица выборки $X = \{X_1, X_2, \dots, X_n\}$

1.5 Основные принципы обучения искусственных нейронных сетей.

На сегодняшний день искусственные нейронные сети (ИНС) широко применяются в решении задач интеллектуального анализа данных таких, как кластеризация, распознавание образов, прогнозирование, интеллектуальное управление и т.д., в условиях неопределенности, нелинейности, стохастичности, хаотичности, разного рода возмущений и помех, благодаря своим универсальным аппроксимирующим способностям и возможности обучения по данным, характеризующим функционирование явления, которое исследуется [39].

Известно, что важное свойство, которым обладает биологический мозг, является способность к обучению, а так как искусственная нейронная сеть и есть модель мозга, понятие «обучение» является ключевым в теории ИНС. В общей теории искусственных нейронных сетей существует направление, которое занимается проблемами процессов обучения, получившее название «нейроматематика» [39-42].

Тип и характер обучения определяются прежде всего объемом априорной и текущей информации о среде, в которую «погружена» сеть, а также критерием обучения (целевой функцией), характеризующим степень соответствия нейросети решаемой ею задаче. Информация о внешней среде задана, как правило, в виде обучающей выборки образов или примеров, обрабатывая которую, сеть извлекает сведения, необходимые для получения искомого решения. Именно характер и объем этой информации определяют как тип обучения, так и конкретный алгоритм [43].

Наиболее популярной в настоящее время является парадигма обучения «с учителем».

Парадигму обучения «с учителем» схематически можно представить рис. 1.3

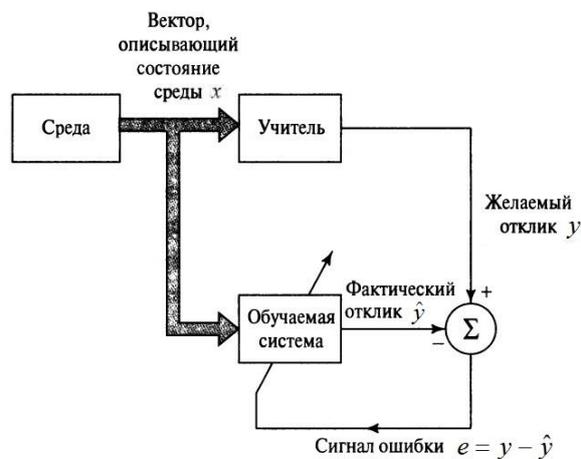


Рисунок 1.3 – Схема обучения с учителем

В данной схеме «учителю» известна информация о внешней среде, которая задана в виде последовательности входных векторов x , а также «правильная реакция» на эти сигналы, обозначенная желаемым сигналом y . Очевидно, что реакция необученной сети \hat{y} будет отличаться от «правильной» реакции учителя, в связи с чем возникает ошибка $e = y - \hat{y}$. Целью является настройка параметров ИНС так, чтобы некоторая скалярная функция от ошибки $E(e)$ (критерий обучения) достигала минимального значения. Так как данные о внешней среде имеют нестационарный характер, процесс обучения длится непрерывно, для чего и используются те или иные рекуррентные процедуры.

При обучении «без учителя» или самообучении нельзя прогнозировать выходные данные. Сети, которые реализуют парадигму самообучения, предназначены для анализа латентной структуры входных данных и решают задачу автоматической классификации (кластеризации), компрессии данных.

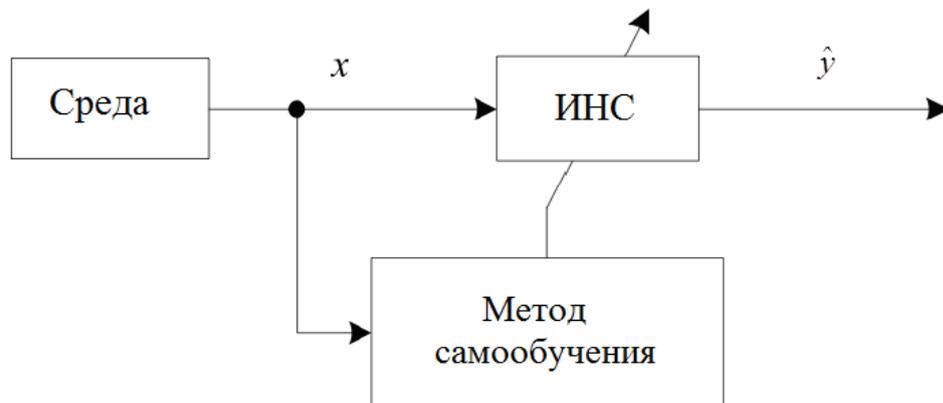


Рисунок 1.4 – Схема обучения без учителя

С вышеописанными парадигмами тесно связаны правила обучения, которые лежат в основе конкретных алгоритмов. С. Хайкин [44] определяет пять основных правил: обучение на основе коррекции по ошибке, обучение

по Больцману, обучение по Хэббу, обучение памяти и конкурентное обучение.

Своеобразным компромиссом между двумя парадигмами является обучение с подкреплением [45], при котором доступна лишь непрямая информация о правильной реакции на входной сигнал сети x . Достаточно известной также является парадигма смешанного обучения, когда часть параметров сети настраивается с помощью обучения с учителем, а вторая часть или архитектура в целом – с помощью самообучения. Этот подход получил широкое распространение в обучении радиально-базисных искусственных нейронных сетей.

Правило коррекции по ошибке – типичный случай обучения с учителем, при этом с помощью тех или иных процедур оптимизации и адаптивной идентификации минимизируется априори заданная скалярная целевая функция $E(e)$. Наибольшее количество известных алгоритмов основывается именно на этом правиле. В основе обучения по Больцману лежат принципы теоретической термодинамики, при этом настройка синаптических весов стохастической сети обеспечивает необходимое (желаемое) распределение вероятностей состояний отдельных нейронов. В некотором смысле обучение по Больцману может рассматриваться как расширение идей обучения с учителем на стохастический случай. С самообучением тесно связаны правило Хэбба и обучение, основанное на памяти, в основе которых лежит нейрофизиологический постулат, согласно которому, когда нейроны с обеих сторон синапса находятся в возбужденном состоянии, сила связи между ними возрастает (увеличивается синаптический вес) и, наоборот, когда соседние нейроны находятся в разных состояниях, связь между ними слабеет. В случае конкурентного обучения могут быть реализованы все вышеуказанные парадигмы, но отличием является процесс «соревнования» нейронов выходного слоя по принципу «победитель получает все», то есть возбуждается только один выходной нейрон – «победитель» [46].

1.5.1 Обучение нейронных сетей, основанное на оптимизации

Множество реальных задач характеризуется тем, что данные поступают в on-line режиме, их обработка должна проводиться в темпе функционирования объекта, а сам объект является нестационарным. Понятно, что классический многослойный персептрон, являющийся универсальным аппроксиматором, в данном случае не годится, а в качестве альтернативы ему в ряде случаев могут быть использованы радиально-базисные нейронные сети (РБНС) [47-50]. Эти сети также являются универсальными аппроксиматорами, а поскольку их выход линейно зависит от настраиваемых синаптических весов, для их обучения в реальном времени может быть использован рекуррентный метод наименьших квадратов или его модификации, являющиеся, по сути, алгоритмами оптимизации второго порядка, обеспечивающими квадратичную сходимость к оптимальному решению. Вместе с тем, практическое применение РБНС ограничивается, так называемым, проклятием размерности, а также возникновением «дыр» в пространстве радиально-базисных функций, приводящих к возникновению областей, где все нейроны сети не являются активными.

Избежать подобных «провалов» можно, воспользовавшись, так называемым, единичным разбиением пространства, реализуемым с помощью нормализованных радиально-базисных нейронных сетей (НРБНС) [51], в которых выходной сигнал сети нормируется на сумму выходов всех нейронов.

1.5.2 Мгновенно обучаемые нейронные сети

Эффективной альтернативой сетям, основанным на оптимизации, являются, так называемые, мгновенно обучаемые нейро-модели, в основе

которых лежит принцип «нейроны в точках данных» [48]. В рамках принципов обучения искусственных нейронных сетей подобные модели также называют «ленивыми» моделями обучения. Важной характерной чертой мгновенных нейросетей считается то, что этап формирования фактической модели длится до того момента, пока существует потребность в данной модели. Другими словами, это продолжается до тех пор, пока для данного входа необходим выходной сигнал, а до этого осуществляется только сбор и хранение данных.

Чаще всего, обучение классических моделей происходит в режиме off-line и для оценки применяется исключительно фиксированная модель. То есть, обработка всех данных, необходимых для обучения, изначально осуществляется в пакетном режиме. Подобная обучающая процедура отличается вычислительной сложностью и в отдельных ситуациях является фактически невозможной. В связи с этим используют методы сжатия данных. Более того, в качестве дополнительного варианта может применяться on-line адаптация, которая служит для настройки и адаптации модели посредством изменения ее характеристик (параметров) в течении фазы оценки. Необходимость в этом обусловлена важностью обеспечения процедуры следящего поведения.

Обобщенная регрессионная нейронная сеть (ОРНС), которую предложил Д. Шпехт [52], является самым ярким представителем моделей, обучаемых на основе этого принципа. Она базируется на принципах ядерных оценок Надарая-Ватсона [53-55], непараметрических моделей [56-59] и парзеновских окон [60], а процесс её обучения в конечном счете сводится к одноразовой установке многомерных радиально-базисных функций (РБФ) в точках единичного центрированного гиперкуба, которые задаются в однозначном порядке при помощи обучающей выборки. Это говорит о том, что подобные сети можно вполне отнести к тем самым мгновенно обучаемым

моделям [50, 52], настройка которых производится посредством единственного прохода обучающего алгоритма. За счет совпадения в плане архитектуры с нормализованными радиально-базисными нейронными сетями (НРБНС), обучение ОРНС происходит в более быстром темпе при одновременном установлении центров РБФ в точках, координаты которых определяются посредством входных сигналов объекта согласно принципу «нейроны в точках данных» [61]. При этом «высота» РБФ полностью совпадает с определенными параметрами выходного сигнала объекта. Благодаря высокой скорости обучения ОРНС, стало возможным их эффективное применение в реальных задачах [62, 63].

Основным «недостатком» ОРНС является необходимость больших объемов памяти и использование сложных вычислительных действий, которые требуются для оценки. Подобные сложности могут иметь место для on-line режима оценивания, к их числу относятся пропущенные данные, время необходимое для оценки возникающего отклика [64].

1.5.3 Нейронные сети опорных векторов

Нейронные сети опорных векторов, также именуемые машинами опорных векторов (Support Vector Machines – SVM), - это архитектуры с прямой передачей данных. Они представляют собой обобщенные варианты известных конструкций, к числу которых относятся: полиномиальные и радиально-базисные сети, а также многослойные персептроны.

Основным понятием при синтезе этих сетей считаются опорные векторы (крайние векторы согласно определению В. Н. Вапника [63]), которые представляют собой малое подмножество самых информативных сведений, включенных в обучающую выборку, и определяются во время обучающего процесса.

SVM имеет «слабое место», которое заключается в громоздком процессе обучения синаптических весов, который, в свою очередь, приводится к задачам нелинейного программирования.

Чтобы решить задачи в режиме онлайн, необходимо учитывать, что любое новое наблюдение в машине опорных векторов может привести к появлению нового обучающего критерия. Это считается наиболее важной проблемой, обусловленной операциями с SVM. Так как данная система имеет растущее количество ограничений, следовательно, на каждом новом этапе сеть будет переобучаться заново.

1.5.4 Конкурентные нейронные сети

Некоторые обучающие методы, используемые для традиционных архитектур искусственных нейронных сетей требуют значений оптимальных выходных сигналов сети в обучающей выборке информации, т.е. относятся к классу обучения с учителем.

Но, иногда имеют место реальные задачи, для целей которых в обучающей выборке нет необходимых желаемых значений, а присутствует исключительно набор наблюдений $x(k)$. Поэтому, для обучения сети, следует извлекать важные данные, основываясь исключительно на данном наборе $x(k)$. В подобных ситуациях речь идет о процессе самообучения или обучающей методах обучения без учителя. В большинстве случаев, обучающие процедуры без учителя для ИНС базируются на конкуренции между нейронами [65].

Охарактеризовать вышеупомянутые задачи лучшим образом помогут следующие примеры:

– кластеризация – в случае, когда входные данные группируются в кластеры, их необходимо выделить для дальнейшей обработки как четкой, так и нечеткой при существенном пересечении кластеров;

– понижение размерности – в ситуациях, когда обучающие данные находятся в меньшем по размерности подпространстве по сравнению с исходной выборкой. Для уменьшения размерности необходимо построить оптимальное отображение R^n в меньшее по размерности пространство с минимальными потерями данных (чаще всего информативными признаками считаются признаки с наибольшей дисперсией данных)[66];

Главное отличие методов кластеризации от квантования векторов заключается в том, что во время кластеризации определяют области входного пространства, которые включают «похожие» наблюдения [67], а во втором случае – основная задача состоит в разбиении всего входного пространства [40].

Конкурентное обучение. В данном случае изучаются процедуры самообучения, направленные на решение задач кластеризации информации в классическом понимании [67].

Процедура самоорганизации основана на методах конкурентного обучения, а ее работа начинается с инициализации синаптических весов сети, которые выбираются или в случайном порядке, или посредством какого-либо широко распространенного способа. Как правило, реализация процедуры самоорганизации включает три важных этапа [44]: конкуренция, кооперация и синаптическая адаптация.

Стандартная архитектура конкурентной сети (competitive neural network) представляет собой полносвязный слой из m нейронов, которые содержат по n рецепторов, характеризующихся n -мерными векторами синаптических весов

$$w_j(k) = (w_{j1}(k), w_{j2}(k), \dots, w_{jn}(k))^T, \quad j = 1, 2, \dots, m.$$

На вход сети подается входной вектор-наблюдение $x(k)$, $k = 1, 2, \dots$. На этапе конкуренции, во время подачи входного вектора, происходит активация лишь одного нейрона, именуемого как «нейрон-победитель». В случае с корректно обученной сетью для всех векторов $x(k)$, которые принадлежат единственному кластеру, активируется один и тот же нейрон-победитель $w^*(k)$.

В качестве функций активации нейронов конкурентной сети, в большинстве случаев, используются линейные функции. Также, нейроны включают веса w_j , которые задают топологию карты, и, за исключением непосредственного значения $y(k)$ выхода сети, в качестве результата классификации может выступать номер (координаты w^*) нейрона-победителя. Графически архитектура конкурентной сети изображена на рис. 1.6.

На сегодняшний день известно два основных метода нахождения нейрона-победителя как «ближайшего» к вектору текущего входного наблюдения $x(k)$ в соответствии с принятой метрикой. Рассмотрим каждый из них более подробно.

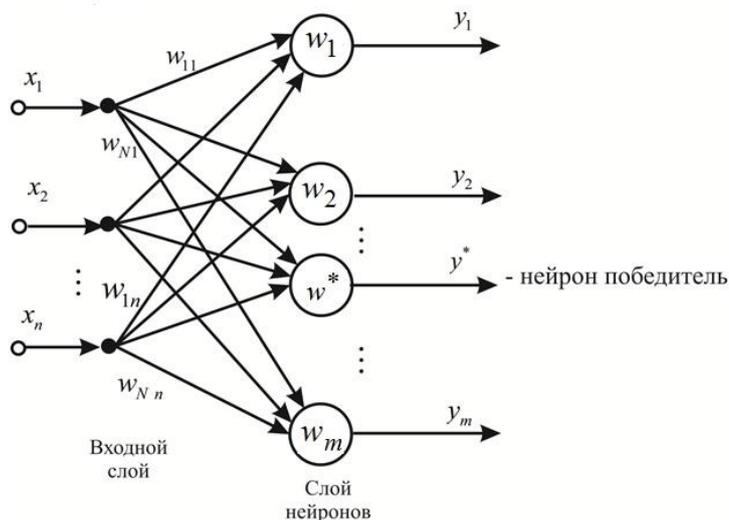


Рисунок 1.5 – Архитектура стандартной конкурентной нейронной сети

Осуществление выбора нейрона-победителя производится в соответствии с расстоянием в евклидовом пространстве. Скалярное произведение векторов не отображает степень их близости в случае с ненормированными данными и векторами синаптических весов. Таким образом, чтобы выбрать победителя, целесообразно применять метрику Евклида:

$$D_E(a,b) = \|a - b\| = \sqrt{(a - b)^T (a - b)},$$

в соответствии с которой

$$w^*(k) = w_j(k), \quad k = \arg \max_{j=1, \dots, m} \|w_j(k) - x(k)\|. \quad (1.25)$$

Настройка синаптических весов является следующим шагом. В общеизвестных случаях выбор «ближнего» для входного наблюдения нейрона-победителя согласно принятой метрике записывается так:

$$D(w^*(k), x(k)) = \min_j D(w_j(k), x(k)).$$

Следующим этапом после завершения фазы выбора победителя является настройка синаптических весов согласно последующим процедурам синаптической адаптации:

для ненормированного пространства весов:

$$w^*(k+1) = w^*(k) + \eta(k)(x(k) - w^*(k)). \quad (1.26)$$

для нормированного пространства весов:

$$w^*(k+1) = \frac{w^*(k) + \eta(k)(x(k) - w^*(k))}{\|w^*(k) + \eta(k)(x(k) - w^*(k))\|}. \quad (1.27)$$

В данном случае процедуры настройки весов (1.26), (1.27) «подтягивают» вектор синаптических весов нейрона-победителя ко входному вектору $x(k)$ на расстояние, определяемое параметром величины шага поиска $\eta(k)$. В случае нормированного пространства настройка веса выражается в его повороте в сторону $x(k)$.

1.6 Постановка задачи исследования

Как показал проведенный анализ, на сегодняшний день в рамках интеллектуального анализа данных существует большое количество методов кластеризации, отличающихся между собой как математическими постановками задач так и результатами. При этом большинство наиболее известных методов предполагают, что кластеры имеют выпуклую форму и линейно разделимы, но, к сожалению, в реальных задачах – это далеко не так, поскольку кластеры могут иметь произвольную форму. Для решения таких задач существует ряд методов, одним из наиболее популярных подходов при этом являются методы ядерной кластеризации [68, 69]. В рамках этого подхода предполагается, что кластеры могут иметь произвольную форму, но они не пересекаются и не перекрываются, а вся выборка, подлежащая кластеризации задана априори. На сегодняшний день существует множество методов кластеризации, но все они являются четкими, где допускается, что кластеры линейно разделимы, а данные поступают в пакетном режиме. В реальных задачах кластеры могут перекрываться, а каждое наблюдение может принадлежать нескольким кластерам с определенной принадлежностью. Для решения таких задач разработаны методы нечеткой кластеризации [25, 70], однако при этом формируемые кластеры имеют выпуклую форму. В ситуациях, когда кластеры могут перекрываться и иметь произвольную форму возникает необходимость разработки методов нечеткой кластеризации

На сегодняшний день, не существует систем, которые объединяли бы в себе on-line нечеткие методы кластеризации и ядерный подход, связанный с гипотезой, сформулированной Кавером [71], которая утверждает, что если задача линейно неразделима в исходном пространстве, то она может быть решена в пространстве повышенной размерности.

На сегодняшний день, не существует систем, которые объединяли бы в себе нечеткие методы кластеризации и ядерный подход, связанный с гипотезой, сформулированной Кавером [71], которая утверждает, что если задача линейно неразделима в исходном пространстве, то она может быть решена в пространстве повышенной размерности.

Таким образом, на данное время стоит актуальная задача по разработке нечетких on-line методов кластеризации на основе ядерных функций, когда кластеры могут перекрываться, а каждое наблюдение может принадлежать нескольким кластерам с определенной вероятностью. Использование специальных ядерных функций позволит повысить аппроксимирующие свойства предложенных гибридных нейро-фаззи сетей и методов кластеризации.

Для достижения указанной задачи необходимо:

- проанализировать существующие методы и подходы для кластеризации данных разной физической природы;
- разработать модификацию метода EM (ожидание-максимизация) кластеризации данных, которые последовательно поступают на обработку одно за одним в on-line режиме;
- разработать искусственную нейронную сеть, которая объединяет в себе идеи ядерных систем и самообучения, и построена на основе радиально-базисной нейронной сети и самоорганизованной карты;
- разработать архитектуры гибридной нейронной сети на основе обобщенной регрессионной сети и самоорганизованной карты Т. Кохонена;
- разработать многослойные нейро-фаззи системы, являющейся гибридом системы Ванга-Менделя и нечеткой кластеризующей самоорганизующейся сети;
- провести имитационное моделирование разработанных методов, моделей и решить с их помощью ряд практических задач нечеткой кластеризации данных.

Выводы по разделу 1

1. Проанализировано состояние проблемы кластеризации данных и рассмотрены существующие подходы к ее решению.
2. Проанализировано состояние современной теории ИНС и нейро-фаззи систем, предназначенных для решения задач обработки данных в том числе и задач интеллектуального анализа данных, а также рассмотрены основные подходы к её реализации.
3. Рассмотрены основные методы и процедуры обучения нечетких систем.
4. Сформулированы задачи исследования.

РАЗДЕЛ 2

МНОГОСЛОЙНЫЕ НЕЧЕТКИЕ КЛАСТЕРИЗИРУЮЩИЕ СИСТЕМЫ ДЛЯ АНАЛИЗА ПОТОКОВ ДАННЫХ

Задача кластеризации больших массивов многомерных наблюдений (векторов-образов) часто встречается во многих реальных практических задачах, а для ее решения разработано множество методов [68-69, 72], при этом в последние годы в рамках концепции Big Data особое внимание уделяется обработке информации, хранящейся либо в сверхбольших базах данных (VLDB), либо поступающей на обработку в on-line режиме в форме потока данных (Data Stream). Для решения этих задач с успехом может быть использован математический аппарат вычислительного интеллекта (Computational Intelligence) [73-76] и, прежде всего, искусственные нейронные сети и мягкие вычисления (Soft Computing), основанные на нечеткой логике.

2.1 Постановка задачи кластеризации

В наиболее общей постановке задачи кластеризации предполагается, что имеется массив из N многомерных наблюдений, описываемых n -мерными векторами признаков $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T \in R^n$, $k = 1, 2, \dots, N, \dots$, который необходимо разбить на m кластеров, при этом это число может быть заранее не известно, т.е. $1 < m < N$. Очевидно, что столь большое число известных методов решения задачи кластеризации связано с тем, что сегодня не существует универсального алгоритма пригодного для эффективного использования во всех возникающих реальных ситуациях. Одна из таких возможных и достаточно сложных ситуаций связана с предположением, что каждый вектор наблюдений может одновременно

относиться с различными уровнями возможностей, вероятностей или принадлежностей не к одному, а сразу к нескольким или ко всем формируемым кластерам. В этой ситуации целесообразно использование, так называемых, мягких алгоритмов (soft algorithms) [77], среди которых наибольшее внимание получили нечеткие методы кластеризации [25, 31, 78] и вероятностным алгоритмам (probabilistic model-based algorithms) [77], среди которых для обработки больших массивов широкое распространение получил, так называемый, EM-алгоритм (Expectation-Maximization algorithm) [79-84], в основе которого лежат сугубо вероятностные предпосылки. Каждый из отмеченных подходов имеет свои достоинства и недостатки, в связи с чем представляется целесообразной разработка численно простой мягкой процедуры кластеризации, объединяющей в себе достоинства вероятностного и фаззи-подходов и предназначенной для обработки потоков данных, поступающих в on-line режиме.

2.2 Мягкий вероятностный нечеткий EM-метод кластеризации многомерных данных

Задача вероятностной кластеризации в общей постановке сводится к проблеме самообучения при неизвестном числе областей [85], при этом предполагается, что плотность распределения данных в каждом кластере подчиняется многомерному нормальному (гауссовскому) закону

$$p_j(x) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{\det \Sigma_j}} \exp\left(-\frac{1}{2}(x - c_j)^T \Sigma_j^{-1}(x - c_j)\right), j = 1, 2, \dots, m, \quad (2.1)$$

а совместная плотность распределения всех данных описывается смесью

$$p(x) = \sum_{j=1}^m p_j p_j(x) = \sum_{j=1}^m \frac{p_j}{(2\pi)^{\frac{n}{2}} \sqrt{\det \Sigma_j}} \exp\left(-\frac{1}{2}(x - c_j)^T \Sigma_j^{-1} (x - c_j)\right), j = 1, 2, \dots, m, \quad (2.2)$$

где c_j – $(n \times 1)$ – мерный вектор-центроид j – го кластера, Σ_j – $(n \times n)$ – корреляционная матрица j – го кластера такая, что

$$\Sigma_j = \frac{1}{N} \sum_{k=1}^N (x(k) - c_j)(x(k) - c_j)^T, \quad (2.3)$$

p_j – априорные вероятности (веса), подчиняющиеся естественному условию

$$\sum_{j=1}^m p_j = 1, \quad (2.4)$$

при этом предполагается, что c_j, Σ_j и $p_j \forall j = 1, 2, \dots, m$ априори неизвестны и подлежат оцениванию в процессе кластеризации.

Здесь можно отметить, что показатель степени экспоненты в (2.1), (2.2) есть расстояние Махаланобиса между c_j и наблюдением $x(k)$, т.е.

$$d_M^2(c_j, x(k)) = (x(k) - c_j)^T \Sigma_j^{-1} (x(k) - c_j), \quad (2.5)$$

а условие (2.4) совпадает с условием, накладываемым на сумму принадлежностей в популярном алгоритме нечетких С-средних (FCM) Дж. Бездека [25]

$$\sum_{j=1}^m u_j(k) = 1, \quad (2.6)$$

где $u_j(k)$ – уровень нечеткой принадлежности наблюдения $x(k)$ j -му кластеру).

В связи с этим методы кластеризации, связанные с ограничением (2.6), называются вероятностными алгоритмами нечеткой кластеризации.

Работа EM-алгоритма состоит из повторяющейся последовательности двух шагов, при этом на шаге E (expectation step) производится оценивание параметров совместного распределения (2.2), а на шаге M (maximization step) максимизируется критерий самообучения в виде логарифмической функции правдоподобия

$$E(c_j, \Sigma_j, p_j, x(k)) = \sum_{k=1}^N \log \left(\sum_{j=1}^m p_j p_j(x(k-1)) \right),$$

для чего могут быть использованы как традиционные градиентные, так и квазиньютоновские процедуры оптимизации [72].

И, наконец, для оценки вероятности принадлежности наблюдения $x(k)$ j -му кластеру используется выражение

$$p_j(x(k)) = \frac{p_j \exp \left(-\frac{1}{2} (x(k) - c_j)^T \Sigma_j^{-1} (x(k) - c_j) \right)}{\sum_{l=1}^m p_l \exp \left(-\frac{1}{2} (x(k) - c_l)^T \Sigma_l^{-1} (x(k) - c_l) \right)}, \quad (2.7)$$

удовлетворяющее условию (2.4).

Частным случаем EM-метода является известный метод кластеризации K-средних и совпадающий с ним при $p_j = m^{-1}$ и единичных корреляционных матрицах Σ_j . При этом метод K-средних является четкой процедурой, что означает, что каждое наблюдение может быть отнесено к единственному

кластеру. При этом метод К-средних существенно проще с вычислительной точки зрения чем EM-метод и связан с минимизацией критерия самообучения, основанного на евклидовых расстояниях

$$E(c_j, x(k)) = \sum_{k=1}^N \sum_{j=1}^m u_j(k) \|x(k) - c_j\|^2, \quad (2.8)$$

где

$$u_j(k) = \begin{cases} 1, & \text{если } x(k) \text{ принадлежит} \\ & j - \text{му кластеру,} \\ 0 & \text{в противном случае.} \end{cases} \quad (2.9)$$

EM-метод также относится к процедурам, основанным на расстояниях, и в этом смысле близок к, так называемому, методу К-средних Махаланобиса, являющемуся четкой процедурой, минимизирующей целевую функцию

$$E(c_j, \Sigma_j, x(k)) = \sum_{k=1}^N \sum_{j=1}^m u_j(k) (x(k) - c_j)^T \Sigma_j^{-1} (x(k) - c_j), \quad (2.10)$$

где Σ_j и $u_j(k)$ определяется выражениями (2.3), (2.9).

Принципиальная разница между EM-методом и методом К-средних Махаланобиса состоит в том, что последний дает однозначное решение о принадлежности каждого наблюдения единственному кластеру, в то время как вероятностная процедура учитывает возможность перекрытия классов с расчетом вероятностей согласно выражению (2.7).

Наряду с несомненными достоинствами EM-метод обладает и рядом существенных ограничений. Во-первых, исходные данные должны иметь

случайную природу и подчиняться нормальному закону распределения, во-вторых, на М-этапе возможно «застревание» процесса оптимизации в локальных экстремумах (эта проблема может быть преодолена с помощью использования процедур многоэкстремальной оптимизации), в-третьих, с вычислительной точки зрения это все-таки довольно громоздкая процедура [77] и, наконец, подразумевается, что весь массив данных, подлежащих кластеризации, задан заранее и не изменяется в процессе обработки.

В связи с этим представляется целесообразной разработка численно простого алгоритма кластеризации, основанного на метрике Махаланобиса, учитывающего возможность взаимного перекрытия формируемых кластеров и позволяющего анализировать поток данных, последовательно поступающих на обработку в on-line режиме.

2.3 Нечеткая вероятностная кластеризация на основе WTA-правила самообучения

Результат минимизации критерия самообучения (2.8) метода К-средних имеет вид среднего арифметического данных каждого кластера

$$c_j = \frac{\sum_{k=1}^N u_j(k) x(k)}{\sum_{k=1}^N u_j(k)} = \frac{1}{N_j} \sum_{x(k) \in Cl_j} x(k), \quad (2.11)$$

где N_j – количество наблюдений $x(k)$, отнесенных к кластеру Cl_j .

Если же данные поступают на обработку последовательно в on-line режиме, решение (2.11) может быть получено с помощью кластеризующей нейронной сети Т. Кохонена [18], настройка синаптических весов которой, являющихся по сути компонентами векторов-центроидов, производится с

помощью тех или иных алгоритмов конкурентного самообучения, наиболее популярным из которых является WTA-правило (“Winner Takes All”). При этом сама процедура настройки подобно EM-алгоритму состоит из последовательности двух этапов: конкуренции (соответствует E-шагу) и синаптической адаптации [44] (соответствует M-шагу).

Суть конкурентного обучения по Кохонену состоит в том, что если к моменту поступления наблюдения $x(k)$ рассчитаны координаты центроидов $c_1(k-1), \dots, c_j(k-1), \dots, c_m(k-1)$, среди них выбирается «победитель» ближайший в смысле евклидова расстояния

$$d_E^2(c_j(k-1), x(k)) = \|x(k) - c_j(k-1)\|^2 \quad (2.12)$$

к $x(k)$ (E-шаг), который и уточняется на M-шаге с помощью рекуррентной процедуры

$$c_j(k) = \begin{cases} c_j(k-1) + \eta(k)(x(k) - c_j(k-1)), \\ \text{если } c_j(k-1) - \text{"победитель"}, \\ c_j(k-1) \text{ в противоположном случае,} \end{cases} \quad (2.13)$$

где $\eta(k)$ параметр шага обучения, выбираемый обычно из эмпирических соображений, хотя несложно показать, что результат (2.11) может быть получен при $\eta(k) = k^{-1}$.

Несложно заметить, что первое соотношение (2.13) есть не что иное, как градиентная минимизация (2.8), т.е. может быть переписано в виде

$$c_j(k) = \begin{cases} c_j(k-1) - \eta(k) \nabla_{c_j} d_E^2(c_j(k-1), x(k)), \\ \text{если } c_j(k-1) - \text{"победитель"}, \\ c_j(k-1) \text{ в противоположном случае.} \end{cases} \quad (2.14)$$

Аналогично (2.14) может быть введена градиентная процедура минимизации критерия (2.10) на основе метрики Махаланобиса (2.5) в виде

$$c_j(k) = \begin{cases} c_j(k-1) - \eta(k) \nabla_{c_j} d_M^2(c_j(k-1), x(k)), \\ \text{если } c_j(k-1) - \text{"победитель"}, \\ c_j(k-1) \text{ в противоположном случае,} \end{cases}$$

или

$$c_j(k) = \begin{cases} c_j(k-1) + \eta(k) \times \\ \times \Sigma_j^{-1}(k-1) (x(k) - c_j(k-1)), \\ \text{если } c_j(k-1) - \text{"победитель"}, \\ \Sigma_j(k-1) = \frac{1}{k-1} \times \\ \times \sum_{l=1}^{k-1} (x(l) - c_j(k-1))(x(l) - c_j(k-1))^T, \\ c_j(k-1) \text{ в противоположном случае.} \end{cases} \quad (2.15)$$

Несложно заметить, что алгоритм самообучения (2.15) является по сути последовательной реализацией метода К-средних Махаланобиса, т.е. позволяет получить только четкое решение.

Для оценки уровня принадлежности отдельных наблюдений в случае перекрывающихся классов вместо громоздкого выражения (2.7) целесообразно воспользоваться оценкой принятой в стандартном FCM-алгоритме Дж. Бездека, используя вместо расстояния $d_E^2(c_j(k), x(k))$ метрику Махаланобиса $d_M^2(c_j(k), x(k))$ в виде

$$u_j(k) = \frac{d_M^{-2}(c_j(k), x(k))}{\sum_{l=1}^m (d_M^{-2}(c_l(k), x(k)))} = \frac{\left((x(k) - c_j(k))^T \Sigma_j^{-1} (x(k) - c_j(k)) \right)^{-1}}{\sum_{l=1}^m \left((x(k) - c_l(k))^T \Sigma_l^{-1} (x(k) - c_l(k)) \right)^{-1}}. \quad (2.16)$$

Таким образом, процедура нечеткой вероятностной кластеризации (2.15), (2.16), являясь своеобразным гибридом EM-алгоритма, метода К-средних Махаланобиса, алгоритмов нечеткой кластеризации Бездека [25] и Гата-Гевы [31] и нечеткой кластеризующей нейронной сети Кохонена в ее адаптивном варианте [86, 87], характеризуется вычислительной простотой и позволяет анализировать данные, последовательно поступающие на обработку в on-line режиме.

2.4 Многослойная ядерная кластеризующая нейро-фаззи система и алгоритмы ее самообучения

Как было выше описано, в традиционной постановке задачи кластеризации предполагается, что формируемые в процессе обработки данных кластеры линейно разделимы и имеют выпуклую форму. На сегодняшний день в реальных ситуациях кластеры, как правило, взаимно пересекаются, а для решения подобных задач был разработан нечеткий кластерный анализ [25, 32, 88], являющийся обобщением четких процедур кластеризации. Для преодоления затруднений, связанных с линейно неразделимыми кластерами произвольной формы, был разработан ядерный кластерный анализ [89-94] и, соответственно, ядерный нечеткий кластерный анализ [95-101]. В основе ядерных нечетких процедур кластеризации лежат идеи, связанные с радиально-базисными нейронными сетями (RBFN) и машинами опорных векторов (SVM) [44], а также традиционные процедуры нечеткой кластеризации и, прежде всего, метод нечетких с-средних (FCM) [25] и метод возможностных с-средних (PCM) [32, 88].

В связи с этим представляется целесообразной разработка on-line ядерной нейро-фаззи системы и адаптивных алгоритмов ее самообучения, позволяющих в последовательном режиме обрабатывать потоки данных и формировать пересекающиеся кластеры произвольной формы.

2.4.1 Архитектура ядерной нечеткой кластеризующей сети Т. Кохонена

Архитектура предлагаемой кластеризующей системы содержит четыре слоя обработки информации и приведена на рис. 2.1.

Входная информация в форме последовательности n -мерных векторов $x(k)$ поступает на первый скрытый слой PL (preprocessing layer), где исходные данные центрируются относительно текущего среднего и нормируются на гиперкуб с помощью элементарных соотношений (для on-line обработки)

$$\bar{x}(k) = \bar{x}(k-1) + \frac{1}{k}(x(k) - \bar{x}(k-1)), \quad (2.17)$$

или (в пакетном режиме)

$$\bar{x}(k) = \frac{1}{k} \sum_{p=1}^k x(p),$$

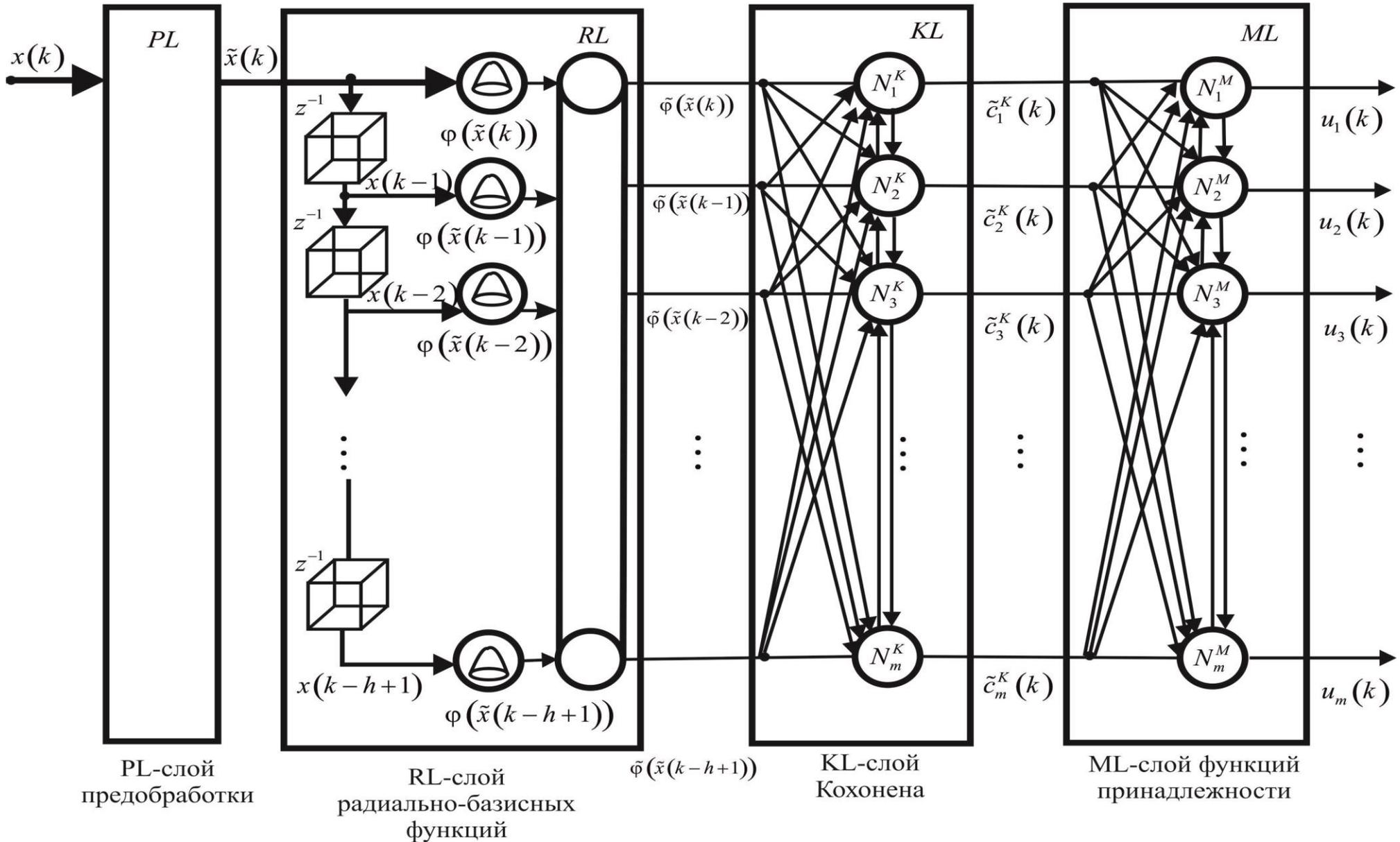


Рисунок 2.1 – Архитектура ядерной нечеткой кластеризующей сети Т.Кохонена

$$\tilde{x}(k) = x(k) - \bar{x}(k),$$

$$\tilde{x}_i(k) = 2 \frac{\tilde{x}_i(k) - \tilde{x}_{i\min}}{\tilde{x}_{i\max} - \tilde{x}_{i\min}} - 1, \quad \forall i = 1, 2, \dots, n, \quad (2.18)$$

в результате чего данные, поступающие на второй скрытый слой, удовлетворяют условию

$$N^{-1} \sum_{k=1}^N \tilde{x}(k) = 0, \quad -1 \leq \tilde{x}_i(k) \leq 1.$$

Последовательность преобразованных векторов $\tilde{x}(k)$ подается на второй скрытый слой радиально-базисных функций *RL* (radial-basis-function layer), где производится повышение размерности входного пространства данных R^n в пространство признаков [91] более высокой размерности R^h , $h > n$.

В общем случае для повышения размерности пространства могут быть использованы стандартные гауссианы

$$K(\tilde{x}(k), \tilde{x}(p)) = \exp\left(-\frac{\|\tilde{x}(k) - \tilde{x}(p)\|^2}{2\sigma^2}\right),$$

применяемые в RBFN и позволяющие обеспечить линейную разделимость формируемых кластеров согласно теореме Кавера [71]. При этом для оценки расстояния между двумя векторами $\tilde{x}(k)$ и $\tilde{x}(p)$ вместо традиционной евклидовой метрики

$$d_E^2(\tilde{x}(k), \tilde{x}(p)) = \|\tilde{x}(k) - \tilde{x}(p)\|^2, \quad (2.19)$$

обычно используемой в задачах кластеризации, с помощью «ядерного трюка» (distance kernel trick) может быть введена оценка несходства (dissimilarity measure) [100]

$$\begin{aligned} d_D^2(\tilde{x}(k), \tilde{x}(p)) &= K(\tilde{x}(k), \tilde{x}(k)) + K(\tilde{x}(p), \tilde{x}(p)) - \\ &- 2K(\tilde{x}(k), \tilde{x}(p)) = 2(1 - K(\tilde{x}(k), \tilde{x}(p))) \end{aligned} \quad (2.20)$$

или сходства (similarity measure)

$$d_S^2(\tilde{x}(k), \tilde{x}(p)) = K(\tilde{x}(k), \tilde{x}(p))$$

в пространстве повышенной размерности R^h .

Наиболее сложным моментом при синтезе второго слоя является выбор количества и координат центров радиально-базисных функций. Проще всего, используя идею «нейроны в точках данных» [48], задать число радиально-базисных функций равным числу наблюдений N с центрами, совпадающими с координатами наблюдений $\tilde{x}(k) \forall k = 1, 2, \dots, N$ так, как это принято в нейронных сетях опорных векторов [102, 103] или обобщенных регрессионных нейронных сетях [52], однако при обработке потоков данных, когда число наблюдений постоянно растет, этот подход представляется бесперспективным.

Более эффективным в данном случае представляется использовать идею «скользящего окна» [104-105], когда вместо всей выборки данных используется только ее часть, содержащая h последних наблюдений $\tilde{x}(k)$, $\tilde{x}(k-1), \dots, \tilde{x}(k-h+1)$, при этом значение h может изменяться в процессе обработки данных.

В результате повышения размерности входного пространства формируется $(h \times 1)$ вектор признаков на основе радиально-базисных функций

$$\varphi(k, h) = \left(\varphi(\tilde{x}(k)), \varphi(\tilde{x}(k-1)), \dots, \varphi(\tilde{x}(k-q)), \dots, \varphi(\tilde{x}(k-h+1)) \right)^T,$$

где $\varphi(\tilde{x}(k-q)) = \exp\left(-\frac{\|\tilde{x} - \tilde{x}(k-q)\|^2}{2\sigma^2}\right)$.

Кроме радиально-базисных функций слой RL содержит $(h-1)n$ элементов чистого запаздывания z^{-1} , формирующих предысторию $\tilde{x}(k-1)$, $\tilde{x}(k-2), \dots, \tilde{x}(k-h+1)$, и блок нормирования N , преобразующий вектор признаков так, чтобы

$$\|\tilde{\varphi}(k, h)\|_2 = 1, \quad \tilde{\varphi}(k, h) = \varphi(k, h) \|\varphi(k, h)\|^{-1}$$

или

$$\|\tilde{\varphi}(k, h)\|_\infty = 1$$

с помощью выражения (2.17).

Вектор признаков $\tilde{\varphi}(k, h)$ далее подается на третий скрытый слой (KL), представляющий собой по сути слой Кохонена самоорганизующейся карты [18], где в on-line режиме конкуренции и синаптической адаптации производится настройка центроидов-прототипов $\tilde{c}_j^k(k)$, $j = 1, 2, \dots, m$ формируемых кластеров.

И, наконец, четвертый выходной слой ML (membership layer) оценивает уровни принадлежности каждого вектора признаков $\tilde{\varphi}(k, h)$ к каждому из m формируемых кластеров.

Заметим также, что третий и четвертый слои являются по сути адаптивной нейро-фаззи сетью Т. Кохонена [87], предназначенной для решения задач нечеткой кластеризации данных, последовательно поступающих на обработку.

2.5. Метод самообучения кластеризирующей нейро-фаззи системы

Процесс самообучения-настройки синаптических весов рассматриваемой системы реализуется в третьем скрытом слое (KL) на основе принципов конкурентного обучения [18] и нечеткой кластеризации с использованием вероятностного подхода Дж. Бездека [25].

Согласно этому подходу процесс кластеризации реализуется в процессе минимизации целевой функции

$$E(u_j(k), \tilde{c}_j^k) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|\tilde{\varphi}(k, h) - \tilde{c}_j^k\|^2 \quad (2.21)$$

при ограничениях

$$\sum_{j=1}^m u_j(k) = 1, \quad (2.22)$$

$$0 \leq \sum_{k=1}^N u_j(k) \leq N, \quad (2.23)$$

где $u_j(k) \in [0, 1]$, β – неотрицательный параметр фаззификации (фаззификатор), определяющий «размытость» границ между кластерами.

Используя далее стандартную технику нелинейного программирования, основанную на неопределенных множителях Лагранжа и решении системы уравнений Каруша-Куна-Таккера, несложно получить известный результат:

$$\left\{ \begin{array}{l} \tilde{c}_j^k = \frac{\sum_{k=1}^N u_j^\beta(k) \tilde{\varphi}(k, h)}{\sum_{k=1}^N u_j^\beta(k)}, \\ u_j(k) = \frac{\left(\|\tilde{\varphi}(k, h) - \tilde{c}_j^k\|^2 \right)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m \left(\|\tilde{\varphi}(k, h) - \tilde{c}_l^k\|^2 \right)^{\frac{1}{1-\beta}}}, \end{array} \right. \quad (2.24)$$

при этом во втором соотношении может быть использована как стандартная метрика (2.19), так и мера несходства (2.20).

Полагая в (2.24) значение фаззификатора $\beta = 2$, приходим к популярному FCM-алгоритму

$$\left\{ \begin{array}{l} \tilde{c}_j^k = \frac{\sum_{k=1}^N u_j^2(k) \tilde{\varphi}(k, h)}{\sum_{k=1}^N u_j^2(k)}, \\ u_j(k) = \frac{\|\tilde{\varphi}(k, h) - \tilde{c}_j^k\|^{-2}}{\sum_{l=1}^m \|\tilde{\varphi}(k, h) - \tilde{c}_l^k\|^{-2}}. \end{array} \right. \quad (2.25)$$

Как можно видеть, процедуры (2.24), (2.25) реализуют пакетный режим обработки информации. Для обработки данных в on-line режиме могут быть использованы рекуррентные алгоритмы нечеткой кластеризации [34, 86, 106], в основе которых лежит процедура нелинейного программирования Эрроу-Гурвица-Удзавы:

$$\begin{cases} u_j(k+1) = \frac{\left(\|\tilde{\varphi}(k+1, h) - \tilde{c}_j^k(k)\|\right)^{\frac{2}{1-\beta}}}{\sum_{l=1}^m \left(\|\tilde{\varphi}(k+1, h) - \tilde{c}_l^k(k)\|\right)^{\frac{2}{1-\beta}}}, \\ \tilde{c}_j^k(k+1) = \tilde{c}_j^k(k) + \eta(k) u_j^\beta(k+1) (\tilde{\varphi}(k+1, h) - \tilde{c}_j^k(k)), \end{cases} \quad (2.26)$$

где $\eta(k) > 0$ – параметр шага, определяющий скорость сходимости алгоритма (2.26).

Несложно заметить, что сомножитель $u_j^\beta(k+1)$ соответствует функции соседства в WTM-правиле самообучения [18]. При $\beta \rightarrow 1$ процедура (2.26) совпадает с рекуррентной формой алгоритма k -средних четкой кластеризации, а при $\beta = 0$ получаем стандартное WTA-правило самообучения

$$\tilde{c}_j^k(k+1) = \tilde{c}_j^k(k) + \eta(k) (\tilde{\varphi}(k+1, h) - \tilde{c}_j^k(k)), \quad (2.27)$$

совпадающее по сути с рекуррентной процедурой оценки среднего (2.17). Можно видеть, что (2.27) минимизирует целевую функцию

$$E(\tilde{c}_j^k) = \sum_k \|\tilde{\varphi}(k, h) - \tilde{c}_j^k\|^2$$

с параметром шага

$$\eta(k) = \frac{1}{k+1},$$

т.е. (2.27) в оптимальном варианте приобретает вид

$$\tilde{c}_j^k(k+1) = \tilde{c}_j^k(k) + \frac{1}{k+1}(\tilde{\varphi}(k+1, h) - \tilde{c}_j^k(k)). \quad (2.28)$$

Принимая во внимание (2.28), процедура кластеризации (2.26) окончательно принимает вид

$$\begin{cases} u_j(k+1) = \frac{\|\tilde{\varphi}(k+1, h) - \tilde{c}_j^k(k)\|^{\frac{2}{1-\beta}}}{\sum_{l=1}^m \|\tilde{\varphi}(k+1, h) - \tilde{c}_l^k(k)\|^{\frac{2}{1-\beta}}}, \\ \tilde{c}_j^k(k+1) = \tilde{c}_j^k(k) + \frac{u_j^\beta(k+1)}{k+1}(\tilde{\varphi}(k+1, h) - \tilde{c}_j^k(k)) \end{cases} \quad (2.29)$$

и структурно близка к алгоритмам нечеткого конкурентного самообучения, введенным в [38, 107].

И хотя алгоритмы вероятностной нечеткой кластеризации (2.24), (2.25), (2.26), (2.29) достаточно эффективны при решении многих практических задач, они не лишены некоторых существенных недостатков. Во-первых, при высоких размерностях h обрабатываемых сигналов они страдают от эффекта «концентрации норм», а, во-вторых, эти алгоритмы нельзя в полном смысле называть нечеткими, поскольку в них входит четкое значение фаззификатора $1 < \beta < \infty$, выбираемое обычно из эмпирических соображений и существенным образом влияющее на конечные результаты.

В связи с этим Ф. Клавонном и Ф. Хёппнером [108] было предложено вместо целевой функции (2.21) с четким фаззификатором β использовать критерий вида

$$E(u_j(k), \tilde{c}_j^k) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1-\alpha)u_j(k)) \|\tilde{\varphi}(k, h) - \tilde{c}_j^k\|^2 \quad (2.30)$$

с ограничениями (2.22), (2.23), где $0 < \alpha \leq 1$ – настраиваемый параметр, определяющий характер получаемого решения.

Вводя функцию Лагранжа

$$L(u_j(k), \tilde{c}_j^k, \lambda(k)) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1-\alpha)u_j(k)) \|\tilde{\varphi}(k, h) - \tilde{c}_j^k\|^2 + \sum_{k=1}^N \lambda(k) \left(\sum_{j=1}^m u_j(k) - 1 \right) \quad (2.31)$$

(здесь $\lambda(k)$ – неопределенные множители Лагранжа) и решая систему уравнений Каруша-Куна-Таккера

$$\begin{cases} \frac{\partial L(u_j(k), \tilde{c}_j^k, \lambda(k))}{\partial u_j(k)} = (2\alpha u_j(k) + 1 - \alpha) \|\tilde{\varphi}(k, h) - \tilde{c}_j^k\|^2 + \lambda(k) = 0, \\ \nabla_{\tilde{c}_j^k} L(u_j(k), \tilde{c}_j^k, \lambda(k)) = -2 \sum_{k=1}^N (\alpha u_j^2(k) + (1-\alpha)u_j(k)) (\tilde{\varphi}(k, h) - \tilde{c}_j^k) = \vec{0}, \\ \frac{\partial L(u_j(k), \tilde{c}_j^k, \lambda(k))}{\partial \lambda(k)} = \sum_{j=1}^m u_j(k) - 1 = 0, \end{cases}$$

приходим к решению

$$\begin{cases} u_j(k) = \frac{\alpha - 1}{2\alpha} + \left(1 + m \frac{1-\alpha}{2\alpha} \right) \frac{\|\tilde{\varphi}(k, h) - \tilde{c}_j^k\|^{-2}}{\sum_{l=1}^m \|\tilde{\varphi}(k, h) - \tilde{c}_l^k\|^{-2}}, \\ \tilde{c}_j^k = \frac{\sum_{k=1}^N (\alpha u_j(k) + (1-\alpha)u_j(k)) \tilde{\varphi}(k, h)}{\sum_{k=1}^N (\alpha u_j(k) + (1-\alpha)u_j(k))}, \\ \lambda(k) = \frac{1 + m \frac{1-\alpha}{2\alpha}}{\sum_{l=1}^m \left(2\alpha \|\tilde{\varphi}(k, h) - \tilde{c}_l^k\|^2 \right)^{-1}}. \end{cases} \quad (2.32)$$

Можно видеть, что при $\alpha = 1$ приходим к FCM-алгоритму (2.25), а уменьшение α придаёт полученному решению более четкий характер.

Применяя к (2.31) процедуру нелинейного программирования Эрроу-Гурвица-Удзавы, приходим к рекуррентному алгоритму нечеткой кластеризации по критерию (2.30) [72]:

$$\left\{ \begin{array}{l} u_j(k+1) = \frac{\alpha - 1}{2\alpha} + \left(1 + m \frac{1 - \alpha}{2\alpha} \right) \frac{\|\tilde{\varphi}(k+1, h) - \tilde{c}_i^k(k)\|^{-2}}{\sum_{l=1}^m \|\tilde{\varphi}(k+1, h) - \tilde{c}_l^k(k)\|^{-2}}, \\ \tilde{c}_j^k(k+1) = \tilde{c}_j^k(k) + \eta(k) \left(\alpha u_j^2(k+1) + (1 - \alpha) u_j(k+1) (\tilde{\varphi}(k+1, h) - \tilde{c}_j^k(k)) \right), \end{array} \right. \quad (2.33)$$

при этом при $\alpha = 1$ алгоритм (2.33) совпадает с (10) при $\beta = 2$. Несложно также заметить, что второе соотношение (2.33) также является WTM-правилом самообучения.

Выводы по разделу 2

1. Предложен мягкий вероятностный нечеткий метод кластеризации многомерных данных, последовательно поступающих на обработку в режиме реального времени. Развиваемый подход предназначен для решения задач Data Stream Mining в условиях перекрывающихся классов по сравнению со своими прототипами значительно проще в вычислительной реализации, не использует никаких вероятностных предположений о природе обрабатываемых данных.

2. Разработан численно простой алгоритм кластеризации, основанный на метрике Махаланобиса и учитывающий возможность взаимного перекрытия формируемых кластеров, а также позволяющий анализировать поток данных, последовательно поступающих на обработку в on-line режиме.

3. Предложена архитектура многослойной ядерной кластеризующей нейро-фаззи системы и процедура ее самообучения, предназначенные для решения задач восстановления пересекающихся кластеров произвольной формы в условиях, когда данные поступают на обработку последовательно в on-line режиме. Предложенная система предназначена для решения задач Data Stream Mining и достаточно проста в программной реализации.

РАЗДЕЛ 3

ПОСЛЕДОВАТЕЛЬНАЯ КЛАСТЕРИЗАЦИЯ НА ОСНОВЕ ЯДЕРНЫХ НЕЙРОННЫХ СЕТЕЙ

В настоящее время искусственные нейронные сети [73-76] получили широкое распространение для решения различного круга задач, возникающих в рамках интеллектуального анализа данных (Data Mining) таких, как прогнозирование, классификация, кластеризация и т.п. При этом кластеризация в этом ряду занимает особое место [68, 69], поскольку решение этой задачи основывается на парадигме самообучения [44], что существенно усложняет процесс поиска решения. Здесь наиболее популярными являются BSB - и ART-искусственные нейронные сети, предназначенные для обработки информации в пакетном режиме, и самоорганизующиеся карты Т. Кохонена (SOM) [18], предназначенные для решения задач кластеризации больших массивов информации, благодаря простоте вычислительной реализации и возможности последовательной on-line обработки данных, что позволяет их использовать в задачах Dynamic Data Mining и Data Stream Mining [105].

3.1 Ядерная кластеризирующая нейронная сеть на основе радиально-базисной нейронной сети

Среди самообучающихся систем вычислительного интеллекта особо следует отметить нейронные сети Т. Кохонена (самоорганизующиеся карты, SOM) [18], предназначенные для решения задач кластеризации больших массивов информации, благодаря своей вычислительной простоте,

эффективности и возможности работы в on-line режиме путем последовательной обработки данных по мере их поступления.

Процесс настройки этих сетей реализуется в режиме самообучения на основе принципов «Победитель получает все» (WTA) или «Победитель получает больше» (WTM), при этом априорно предполагается, что исходная структура данных такова, что формируемые кластеры взаимно не пересекаются и имеют выпуклую форму, т.е. в процессе обучения нейросети могут быть построены разделяющие гиперплоскости, четко разграничивающие разные классы.

В случае пересекающихся классов могут быть использованы методы нечеткого (fuzzy) кластерного анализа [110, 11], в том числе нечеткие самоорганизующиеся карты [34, 112 -114], реализующие в той или иной форме метод нечетких С-средних (FCM) [25].

В случае невыпуклых классов ситуация представляется более сложной. Здесь следует отметить, что сети Кохонена фактически реализуют классический метод К-средних [69,115], а потому клетки Г. Вороного, формируемые в процессе самообучения, ограничены гиперплоскостями. Естественно, что существуют методы кластеризации для классов произвольной формы [72, 116-119], однако реализующие их алгоритмы достаточно сложны с вычислительной точки зрения и никоим образом не предназначены для решения задач Dynamic Data Mining и Data Stream Mining [105], где информация должна обрабатываться в реальном времени.

На сегодня известны, так называемые, ядерные самоорганизующиеся карты (Kernel SOM) [89, 91-92], построенные с использованием ядер Дж. Мерсера [120] (обычно типа потенциальных функций) [121]) и основанные на минимизации критерия эмпирического риска [122], лежащего в основе машин опорных векторов (SVM) [103, 123].

Безусловно, для решения различного круга задач Data Mining можно использовать нейронные сети-машины опорных векторов, однако такие сети

могут страдать от «проклятия размерности», так как количество нейронов данной сети растет с увеличением числа наблюдений в выборке данных. Именно поэтому следует использовать идеи, основанные на радиально-базисных нейронных сетях [49, 124-125].

3.2 Архитектура ядерной самоорганизующейся карты на основе радиально-базисной нейронной сети.

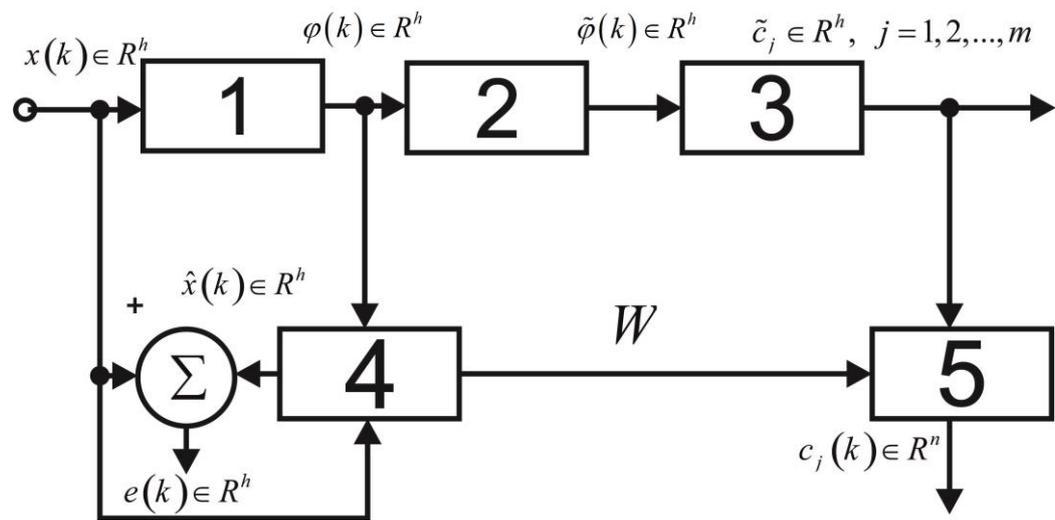


Рис. 3.1 – Ядерная кластеризующая нейронная сеть на основе радиально-базисной нейронной сети

На рис. 3.1 приведена архитектура рассматриваемой ядерной самоорганизующейся карты на основе радиально-базисной нейронной сети.

Исходной информацией в данном случае является центрированная выборка (возможно растущая) векторов наблюдений $x(1), x(2), \dots, x(k), \dots, x(N), \dots$; $x(k) = (x_1(k), \dots, x_i(k), \dots, x_n(k))^T \in R^n$ таких, что $-1 \leq x_i(k) \leq 1$,

$\frac{1}{N} \sum_{k=1}^N x_i(k) = 0$, которая должна быть разбита на m кластеров произвольной

формы, при этом k здесь может быть как номером наблюдений, так и моментом текущего времени.

Векторы наблюдений $x(k)$ последовательно поступают на слой радиально-базисных функций 1, полностью совпадающий по структуре с первым скрытым слоем стандартной радиально-базисной сети и сформированный ядерными (колоколообразными, потенциальными) функциями активации $\varphi_1, \varphi_2, \dots, \varphi_l, \dots, \varphi_h$ ($n+1 \leq h \leq N$), с помощью которых производится повышение размерности исходного пространства входов. В качестве таких функций могут быть использованы традиционные гауссианы:

$$\varphi_l = e^{-\frac{\|x-c_l\|^2}{2\sigma_l^2}}$$

хотя, конечно могут применяться и другие колоколообразные функции, а в качестве их центров c_l в простейшем случае могут быть взяты h произвольно выбранных векторов наблюдений $c_l = x(l)$ (концепция «нейроны в точках данных» [48]). Таким образом, при подаче на вход системы вектора наблюдений $x(k)$ на выходе первого слоя формируется векторный сигнал

$$\varphi(k) = (\varphi_1(k), \dots, \varphi_l(k), \dots, \varphi_h(k))^T \in R^h,$$

где

$$\varphi_l(k) = e^{-\frac{\|x(k)-c_l\|^2}{2\sigma_l^2}}.$$

Второй слой системы – слой нормализации 2 реализует элементарное преобразование

$$\tilde{\varphi}_l(k) = \frac{\varphi_l(k)}{\|\varphi_l(k)\|},$$

необходимое для эффективной работы третьего слоя – самоорганизующейся карты 3.

Именно в этом слое и решается задача кластеризации, т.е. разбиения последовательности образов $\tilde{\varphi}(1), \dots, \tilde{\varphi}(k), \dots$, на m кластеров с нахождением в процессе самообучения прототипов – центроидов классов $\tilde{c}_j^K \in R^h$, $j = 1, 2, \dots, m$.

Основная проблема состоит в том, чтобы эффективно сформировать базис, образованный радиально-базисными функциями, в котором можно было бы произвести кластеризацию с помощью карты Кохонена. Для этого, прежде всего, необходимо оценить насколько удачно было выбрано количество h и центры c_l радиально-базисных функций первого слоя.

Для этого предназначен слой восстановления входного пространства 4, представляющий собой по сути выходной слой радиально-базисной сети с h входами и n выходами и содержащий nh настраиваемых синаптических весов и n сумматоров.

Таким образом, первый и четвертый слои образуют многовыходовую радиально-базисную нейронную сеть, отличием которой от стандартной является то, что в качестве обучающего здесь используется входной сигнал, т.е. сеть работает в режиме автоассоциации [44]. На выходе четвертого слоя формируется сигнал $\hat{x}(k) \in R^n$, являющийся оценкой входного сигнала $x(k)$. Качество восстановления оценивается на основе векторной ошибки

$$e(k) = x(k) - \hat{x}(k)$$

с помощью скалярного критерия

$$\bar{e} = \frac{1}{N} \sum_{k=1}^N \frac{\|x(k) - \hat{x}(k)\|}{\|x(k)\|}. \quad (3.1)$$

Если окажется, что значение \bar{e} превышает некоторый заданный порог e_T , принимается решение о том, что количество нейронов в первом слое должно быть увеличено. Этот процесс продолжается до обеспечения требуемого качества восстановления входного пространства. Результатом обучения четвертого слоя является $(n \times h)$ - матрица синаптических весов $W(N)$, полученная на основании N наблюдений.

Эта матрица является исходной информацией для пятого слоя восстановления прототипов кластеров в исходном пространстве R^n . При этом прототипы, сформированные самоорганизующейся картой в h -мерном пространстве, проецируются в исходное n -мерное пространство с помощью элементарного преобразования

$$c_j^K(k) = W(N)\tilde{c}_j^K(k) \quad \forall j = 1, 2, \dots, m.$$

Таким образом, рассматриваемая здесь система является по сути объединением двух нейросетей: эволюционирующей радиально-базисной нейронной сети (ERBFN) и самоорганизующейся карты Кохонена (SOM), которые параллельно настраивают свои синаптические веса в режиме самообучения, одновременно с этим решая задачу кластеризации данных, образующих классы произвольной формы.

3.3 Обучение ядерной самоорганизующейся карты и радиально-базисной нейронной сети

Обучение введенной системы может рассматриваться как две относительно независимые задачи: самообучение радиально-базисной подсистемы и самообучение собственно самоорганизующейся карты.

Задача слоя восстановления входного пространства состоит в нахождении $(n \times h)$ -матрицы синаптических весов $W = \{w_{il}\}$ по выборке, содержащей N наблюдений путем минимизации критерия обучения

$$E^N = \sum_{k=1}^N E(k) = \frac{1}{2} \sum_{k=1}^N \|x(k) - W\varphi(k)\|^2 = \frac{1}{2} \sum_{k=1}^N \|e(k)\|^2. \quad (3.2)$$

Минимизация критерия (3.2) в пакетном варианте ведет к оценке наименьших квадратов вида

$$W(N) = \left(\sum_{k=1}^N x(k)\varphi^T(k) \right) \left(\sum_{k=1}^N \varphi(k)\varphi^T(k) \right)^{-1}$$

либо в рекуррентной форме [126] –

$$\begin{cases} W(k) = W(k-1) + \frac{(x(k) - W(k-1)\varphi(k))\varphi^T(k)P(k-1)}{1 + \varphi^T(k)P(k-1)\varphi(k)}, \\ P(k) = P(k-1) - \frac{P(k-1)\varphi(k)\varphi^T(k)P(k-1)}{1 + \varphi^T(k)P(k-1)\varphi(k)}. \end{cases} \quad (3.3)$$

Улучшить качество восстановления входного пространства можно, настраивая не только синаптические веса четвертого слоя, но и параметры центров c_l и ширины σ_l активационных функций первого слоя.

С учетом очевидных соотношений

$$\left\{ \begin{array}{l} \hat{x}(k) = \sum_{l=1}^h w_{il}(k-1) \varphi_l(k), \\ e_i^2(k) = (x_i(k) - \sum_{l=1}^h w_{il}(k-1) \varphi_l(k))^2, \\ E(k) = \frac{1}{2} \sum_{i=1}^n e_i^2(k), \\ \nabla_{c_i} E(k) = e_i(k) w_{il}(k-1) \varphi' \left(\frac{\|x(k) - c_i\|^2}{2\sigma_i^2} \right) \frac{x(k) - c_i}{\sigma_i^2}, \\ \frac{\partial E(k)}{\partial \sigma_i^{-2}} = -e_i(k) w_{il}(k-1) \varphi' \left(\frac{\|x(k) - c_i\|^2}{2\sigma_i^2} \right) \frac{\|x(k) - c_i\|^2}{2}, \end{array} \right.$$

можно ввести рекуррентные градиентные алгоритмы обучения вида:

$$\left\{ \begin{array}{l} c_l(k) = c_l(k-1) - \eta_c(k) e_i(k) w_{il}(k-1) \times e^{-\frac{\|x(k) - c_l(k-1)\|^2}{2\sigma_l^2(k-1)}} \frac{x(k) - c_l(k-1)}{\sigma_l^2(k-1)}, \\ \sigma_l^{-2}(k) = \sigma_l^{-2}(k-1) + \eta_\sigma(k) e_i(k) w_{il}(k-1) \times e^{-\frac{\|x(k) - c_l(k-1)\|^2}{2\sigma_l^2(k-1)}} \frac{\|x(k) - c_l(k-1)\|^2}{2}, \end{array} \right. \quad (3.4)$$

где $\eta_c(k), \eta_\sigma(k)$ – скалярные параметры шага обучения.

Таким образом, соотношения (3.3), (3.4) есть процедура обучения всех параметров радиально-базисной нейронной сети, при этом процесс обучения (включая изменение числа радиально-базисных функций) продолжается до достижения требуемого значения критерия (3.1).

Обучение слоя 3 – собственно карты Кохонена состоит в разбиении последовательности векторов - образов $\tilde{\varphi}(k)$ на m кластеров, каждый из которых характеризуется собственным прототипом-центроидом $\tilde{c}_j^K(k) \in R^h$,

$j = 1, 2, \dots, m$, непрерывно уточняемых при подаче очередного наблюдения $\tilde{\varphi}(k)$. Число же кластеров m полагается априорно заданным.

Как и любая другая процедура самообучения, процесс настройки начинается с инициализации синаптических весов сети, в качестве которых и выступают начальные значения прототипов $\tilde{c}_j^K(0)$. При этом эти значения в процессе обработки нормируются подобно входным образам, т.е.

$$\|\tilde{c}_j^K(k)\| = 1.$$

При подаче на вход третьего слоя сигнала $\tilde{\varphi}(k)$ вначале вычисляется m расстояний

$$D(\tilde{\varphi}(k), \tilde{c}_j^K(k-1)) = \|\tilde{\varphi}(k) - \tilde{c}_j^K(k-1)\| \quad \forall j = 1, 2, \dots, m, \quad (3.5)$$

при этом если в качестве расстояний используется евклидова метрика, то вместо (3.5) гораздо удобнее использовать меру подобия

$$SM(\tilde{\varphi}(k), \tilde{c}_j^K(k-1)) = \tilde{\varphi}^T(k) \tilde{c}_j^K(k-1) = \cos(\tilde{\varphi}(k), \tilde{c}_j^K(k-1)) = \cos \theta_j(k). \quad (3.6)$$

На основании (3.5) или (3.6) определяется нейрон-победитель, «ближайший» ко входному образу такой, что

$$D(\tilde{\varphi}(k), \tilde{c}_*^K(k-1)) = \min_j D(\tilde{\varphi}(k), \tilde{c}_j^K(k-1))$$

или

$$SM(\tilde{\varphi}(k), \tilde{c}_*^K(k-1)) = \max_j SM(\tilde{\varphi}(k), \tilde{c}_j^K(k-1)).$$

Далее настраиваются веса этого нейрона с помощью правила самообучения Кохонена [18] в форме

$$\tilde{c}_j^K(k) = \begin{cases} \frac{\tilde{c}_j^K(k-1) + \eta(k)(\tilde{\varphi}(k) - \tilde{c}_j^K(k-1))}{\|\tilde{c}_j^K(k-1) + \eta(k)(\tilde{\varphi}(k) - \tilde{c}_j^K(k-1))\|}, & \text{если } j\text{-й нейрон победил,} \\ \tilde{c}_j^K(k-1) & \text{в противном случае.} \end{cases} \quad (3.7)$$

Наличие знаменателя в первом соотношении (3.7) автоматически обеспечивает протекание процесса обучения на единичном гипершаре.

Процедура (3.7) реализует принцип «Победитель получает все» (WTA), при этом вектор синаптических весов нейрона-победителя $\tilde{c}_*^K(k-1)$ «подтягивается» ко входному образу $\tilde{\varphi}(k)$ на расстояние, определяемое шагом поиска $0 < \eta(k) < 1$. Регулирование шага поиска обычно производится, исходя из эмпирических соображений, а общая рекомендация сводится к тому, что шаг должен монотонно уменьшаться в процессе самообучения. Это условие может быть обеспечено при выборе шага согласно соотношению [127]

$$\eta(k) = r^{-1}(k), r(k) = \alpha r(k-1) + \|\tilde{\varphi}(k)\|^2 = \alpha r(k-1) + 1, 0 < \alpha \leq 1,$$

при этом при $\alpha = 1$ параметр шага $\eta(k) = \frac{1}{k}$ удовлетворяет условиям

А. Дворецкого, а получаемые результаты совпадают с оценками К-средних.

Варьируя фактором забывания α , можно обеспечить широкий интервал изменения шага

$$\frac{1}{k} \leq \eta(k) \leq 1.$$

В результате предъявления самоорганизующейся карте N образов $\tilde{\varphi}(k)$ будет получено m прототипов-центроидов $\tilde{c}_j^K(N)$, которые далее из пространства повышенной размерности R^h могут быть спроецированы в исходное пространство R^n с помощью простого соотношения

$$c_j^K(N) = W(N)\tilde{c}_j^K(N) \quad \forall j = 1, 2, \dots, m.$$

3.4 Ядерная кластеризация на основе обобщенной регрессионной нейронной сети и самоорганизующейся карты Кохонена.

Как уже отмечалось выше, для решения задач кластеризации в ситуациях, когда классы имеют произвольную форму, могут быть использованы, так называемые, ядерные самоорганизующиеся карты (KSOM) [89, 91-92], построенные с использованием ядер Дж. Мерсера [120] и основанные на минимизации критерия эмпирического риска [122], лежащего в основе, так называемых, машин опорных векторов (SVM) [103], введенных В.Н. Вапником. Недостатком нейронной сети опорных векторов, является то, что количество нейронов такой сети определяется объемом выборки данных, поэтому данная сеть не подходит для on-line обработки.

3.4.1 Архитектура ядерной самоорганизующейся карты на основе обобщенной регрессионной нейронной сети

Ниже на рис. 3.2 приведена архитектура рассматриваемой ядерной самоорганизующейся карты на основе обобщенной регрессионной нейронной сети.

Исходной информацией для данной сети является выборка (возможно растущая) векторов наблюдений

$$x(1), x(2), \dots, x(k), \dots, x(N), \dots; x(k) = (x_1(k), x_2(k), \dots, x_i(k), \dots, x_N(k))^T \in R^n,$$

которая должна быть разбита на m кластеров произвольной формы, при этом k здесь может быть как номером наблюдения, так и моментом текущего времени.

Векторы наблюдений $x(k)$ последовательно поступают на первый слой радиально-базисных функций (R -нейронов), полностью совпадающий по структуре с первым слоем (слоем образов) стандартной обобщенной регрессионной сети Д. Шпехта и сформированный ядерными колоколообразными функциями активации $\varphi_1, \dots, \varphi_k, \dots, \varphi_N$, с помощью которых производится повышение размерности входного пространства. В качестве таких функций обычно используются традиционные гауссианы, а настройка этого слоя обеспечивается с помощью «ленивого» обучения на основе концепции «нейроны в точках данных» [48]. При этом в качестве центров активационных функций принимаются сами обрабатываемые векторы-образы. Таким образом, при подаче на вход нейронной сети некоторого неклассифицированного образа x , на выходах R -нейронов первого слоя появляются значения

$$\varphi_k(x) = e^{-\frac{\|x-x(k)\|^2}{2\sigma^2}}, k = 1, 2, \dots, N$$

(здесь σ^2 - параметр рецепторного поля колоколообразной функции), а на выходе GRNN в целом – сигнал

$$\hat{y}(x) = \frac{\sum_{k=1}^N y(k)\varphi_k(x)}{\sum_{k=1}^N \varphi_k(x)}, \quad (3.8)$$

где $y(k)$ – внешний обучающий сигнал, соответствующий образу $x(k)$.
 Понятно, что в задаче кластеризации обучающий сигнал отсутствует как таковой, а сама GRNN в общем случае ориентирована на решение задач интерполяции, а не кластеризации.

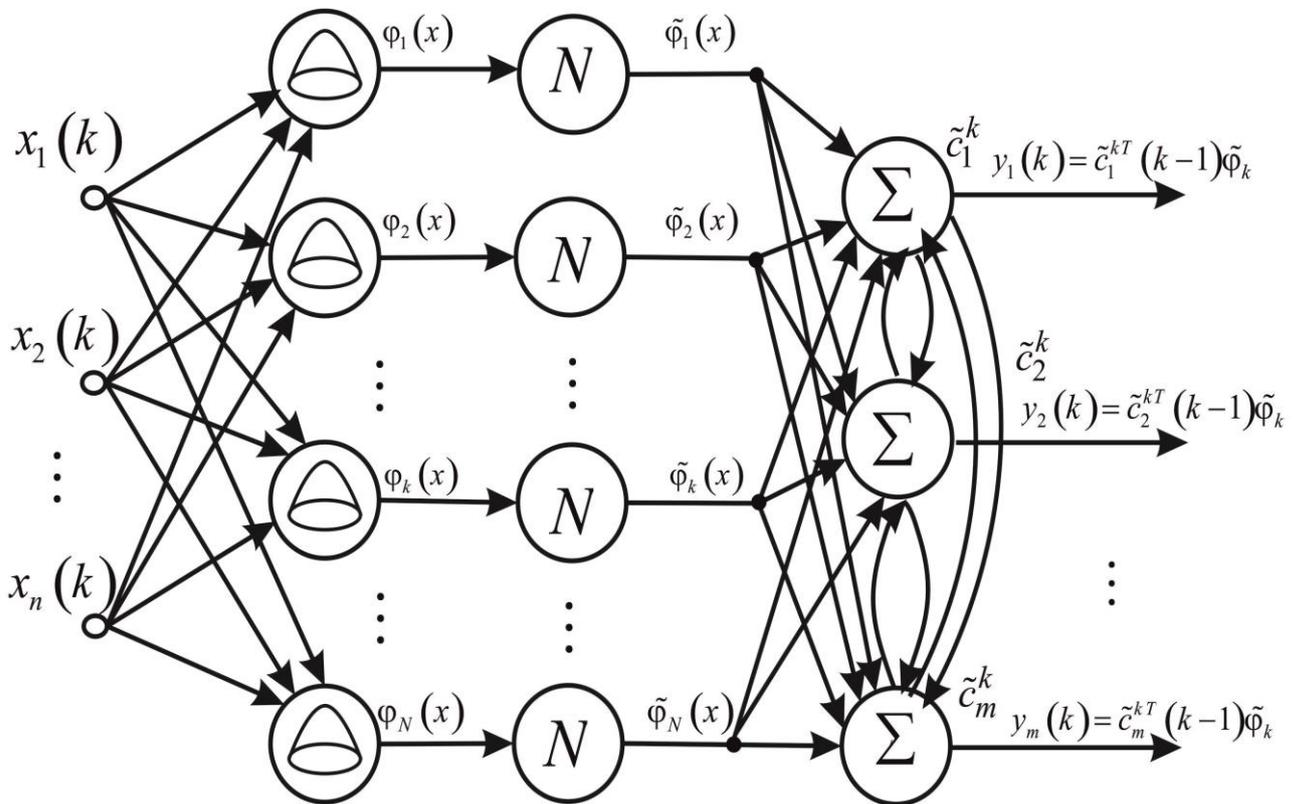


Рисунок 3.2 – Архитектура ядерной самоорганизующейся карты Т.Кохонена, основанная на обобщенной регрессионной нейронной сети

Второй скрытый слой рассматриваемой сети – слой нормализации реализует элементарное преобразование

$$\tilde{\varphi}(x) = \frac{\varphi(x)}{\|\varphi(x)\|}$$

(здесь $(\varphi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_N(x))^T$), необходимое для обработки информации выходным слоем, являющимся по сути кластеризующей нейронной сетью Т.Кохонена, настройка параметров которой производится на основе конкурентного самообучения. В этом выходном слое решается задача разбиения последовательности образов повышенной размерности $\tilde{\varphi}_1, \dots, \tilde{\varphi}_2, \dots, \tilde{\varphi}_k, \dots, \tilde{\varphi}_N$ на m кластеров с нахождением прототипов центроидов $\tilde{c}_1^K, \tilde{c}_2^K, \dots, \tilde{c}_m^K \in R^N$.

Несмотря на кажущуюся простоту, при реализации этого подхода могут возникать существенные вычислительные проблемы при большом объеме N обрабатываемой выборки, поскольку сеть, содержащая N нейронов, становится слишком громоздкой. В связи с этим представляется целесообразным вместо традиционной процедуры обучения GRNN ввести метод, позволяющий не только настроить параметры сети, но и существенно сократить число ее R -нейронов.

3.4.2 Обучение ядерной самоорганизующейся карты на основе обобщенной регрессионной нейронной сети

Для формирования первого слоя рассматриваемой гибридной нейронной сети можно воспользоваться идеями, лежащими в основе эволюционирующих систем вычислительного интеллекта [128-131], адаптированными к on-line обработке информации, последовательно поступающей в систему [132]. Реализация этого подхода производится в форме следующей последовательности шагов:

Шаг 0: задать порог неразличимости векторов центров активационных функций Δ , максимально допустимое количество нейронов в первом слое $H \leq N$ и параметр ширины рецепторного поля σ^2 .

Шаг 1: при поступлении наблюдения $x(1)$ формируется первый центр $c_1 = x(1)$ и сама активационная функция

$$\varphi_1(x) = e^{-\frac{\|x-c_1\|^2}{2\sigma^2}} = e^{-\frac{\|x-x(1)\|^2}{2\sigma^2}}.$$

Шаг 2: при поступлении наблюдения $x(2)$ проверяется условие

$$\|x(2) - c_1\|^2 \leq \Delta, \quad (3.9)$$

если оно выполняется, то наблюдение $x(2)$ не формирует новый центр и автоматически считается принадлежащим тому же кластеру, что и $x(1)$, если выполняется условие

$$\Delta < \|x(2) - c_1\| \leq 2\Delta,$$

то происходит коррекция c_1 согласно WTA-правилу самообучения Т.Кохонена [18] в форме

$$c_1(2) = c_1(1) + \eta(2)(x(2) - c_1(1))$$

(здесь $0 < \eta(2) < 1$ -параметр шага настройки), если же

$$2\Delta \leq \|x(2) - c_1\|,$$

то формируется вторая активационная функция

$$\varphi_2(x) = e^{-\frac{\|x-c_2\|^2}{2\sigma^2}} = e^{-\frac{\|x-x(2)\|^2}{2\sigma^2}}.$$

Шаг N: Если к k моменту поступления N-го вектора-образа $x(N)$ сформировано $h \leq H$ активационных функций и выполняется условие (2), процесс наращивания числа R-нейронов первого слоя заканчивается и в дальнейшем структура этого слоя остается неизменной.

Оценить качество функционирования первого слоя можно было бы, воспользовавшись выражением (3.8), которое для $h = H = N$ приобретает вид

$$\hat{x}(k) = \frac{\sum_{k=1}^N x(k)\varphi_k(x(k))}{\sum_{k=1}^N \varphi_k(x(k))}, \quad (3.10)$$

после чего оценить погрешность восстановления входных образов:

$$\varepsilon = \frac{1}{N} \sum_{k=1}^N \frac{\|x(k) - \hat{x}(k)\|}{\|x(k)\|}. \quad (3.11)$$

Однако, поскольку в процессе формирования первого слоя с помощью описанного выше подхода некоторые из центров активационных функций не совпадают с векторами наблюдений, использование выражения (3.10) корректного для «классической» GRNN, в нашем случае может оказаться неправомерным.

В этой ситуации можно воспользоваться модифицированной GRNN [45,133], чья архитектура приведена на рис. 3.3

Второй скрытый слой сети настраивается аналогично процессу обучения радиально-базисных нейронных сетей. При этом на выходах n адаптивных линейных ассоциаторов формируются сигналы

$$o_i(k) = \sum_{l=0}^h w_{il}(k) \varphi_l(x(k)), \quad i = 1, 2, \dots, n,$$

а на выходе сумматоров Σ^* появляется сумма $\sum_{l=0}^h \varphi_l(x(k))$.

Выходной слой производит нормирование выходного сигнала по типу нормированной радиально-базисной сети (NRBFN) так, что

$$\begin{aligned} \hat{x}_i(k) &= \frac{\sum_{l=0}^h w_{il}(k) \varphi_l(x(k))}{\sum_{l=0}^h \varphi_l(x(k))} = \sum_{l=0}^h w_{il}(k) \frac{\varphi_l(x(k))}{\sum_{l=0}^h \varphi_l(x(k))} = \\ &= \sum_{l=0}^h w_{il}(k) \varphi_l^*(x(k)), \end{aligned} \quad (3.12)$$

$$\varphi_0(x(k)) = 1, \quad \varphi_l^*(x(k)) = \varphi_l(x(k)) \left(\sum_{l=0}^h \varphi_l(x(k)) \right)^{-1},$$

или

$$\hat{x}(k) = W(k) \varphi^*(k), \quad (3.13)$$

где

$$\begin{aligned} \hat{x}_i(k) &= (\hat{x}_1(k), \dots, \hat{x}_i(k), \dots, \hat{x}_n(k))^T, \\ \varphi^*(k) &= (\varphi_0^*(x(k)), \varphi_1^*(x(k)), \dots, \varphi_h^*(x(k)))^T, \end{aligned}$$

$W(k) - (n \times (h + 1))$ - матрица синаптических весов, подлежащая определению.

Для настройки матрицы синаптических весов $W(k)$ может быть использован рекуррентный метод наименьших квадратов, являющийся по сути оптимальной по быстродействию гауссовско-ньютоновской процедурой оптимизации вида:

$$\begin{cases} W(k) = W(k-1) + \frac{(x(k) - W(k-1)\varphi^*(k))\varphi^{*T}(k)P(k-1)}{1 + \varphi^{*T}(k)P(k-1)\varphi^*(k)}, \\ P(k) = P(k-1) - \frac{P(k-1)\varphi^*(k)\varphi^{*T}(k)P(k-1)}{1 + \varphi^{*T}(k)P(k-1)\varphi^*(k)}. \end{cases}$$

Таким образом можно оценить качество работы первого слоя, используя в выражении (3.11) вместо стандартного соотношения (3.10) введенные выше оценки (3.12), (3.13).

Выходной сигнал синтезированного первого слоя в форме $(h \times 1)$ - вектора $\varphi(x) = (\varphi_1(x), \dots, \varphi_l(x), \dots, \varphi_h(x))^T$ во втором скрытом слое нормализуется к виду

$$\tilde{\varphi}(x) = \varphi(x) \|\varphi(x)\|^{-1},$$

т.е. проецируется на h - мерную гиперсферу единичного радиуса, после чего в виде последовательности $\tilde{\varphi}_1, \tilde{\varphi}_2, \dots, \tilde{\varphi}_k, \dots, \tilde{\varphi}_N$ поступает на вход самоорганизующейся карты Кохонена.

Настройка карты Кохонена, образованной m адаптивными линейными ассоциаторами, производится на основе WTM-правила самообучения («Победитель получает больше») и состоит в разбиении последовательности

нормированных векторов-образов $\tilde{\varphi}_1, \tilde{\varphi}_2, \dots, \tilde{\varphi}_N$ на m кластеров, каждый из которых характеризуется собственным прототипом-центроидом $\tilde{c}_j^K \in R^h$, $j = 1, 2, \dots, m$, непрерывно уточняемым при поступлении очередного образа повышенной размерности $\tilde{\varphi}_k$.

Процесс самообучения начинается с инициализации синаптических весов выходного слоя, в качестве которых выступают достаточно произвольно выбранные начальные значения прототипов $\tilde{c}_j^K(0)$ такие, что

$$\|\tilde{c}_j^K(0)\| = 1.$$

При подаче на вход третьего слоя сигнала $\tilde{\varphi}_1$ вычисляется m расстояний

$$D(\tilde{\varphi}_1, \tilde{c}_j^K(0)) = \|\tilde{\varphi}_1 - \tilde{c}_j^K(0)\| \quad \forall j = 1, 2, \dots, m,$$

на основании которых оценивается нейрон-победитель, для которого

$$D(\tilde{\varphi}_1, \tilde{c}_*^K(0)) = \min_j D(\tilde{\varphi}_1, \tilde{c}_j^K(0)).$$

После этого производится первый шаг настройки весов-центроидов

$$\tilde{c}_l^K(1) = \frac{\tilde{c}_l^K(0) + \eta(1)\psi(\tilde{c}_*^K(0), \tilde{c}_l^K(0))(\tilde{\varphi}_1 - \tilde{c}_l^K(0))}{\|\tilde{c}_l^K(0) + \eta(1)\psi(\tilde{c}_*^K(0), \tilde{c}_l^K(0))(\tilde{\varphi}_1 - \tilde{c}_l^K(0))\|}, \quad \forall l = 1, 2, \dots, m.$$

Аналогичным образом можно записать правило самообучения для k -го вектора – образа

$$\tilde{c}_l^K(k) = \frac{\tilde{c}_l^K(k-1) + \eta(k)\psi(\tilde{c}_*^K(k-1), \tilde{c}_l^K(k-1))(\tilde{\varphi}_k - \tilde{c}_l^K(k-1))}{\left\| \tilde{c}_l^K(k-1) + \eta(k)\psi(\tilde{c}_*^K(k-1), \tilde{c}_l^K(k-1))(\tilde{\varphi}_k - \tilde{c}_l^K(k-1)) \right\|}, \forall l = 1, 2, \dots, m, \quad (3.14)$$

где $\psi(\tilde{c}_*^K(k-1), \tilde{c}_l^K(k-1))$ - так называемая, функция соседства, определяющая локальную область топологического соседства, в которой настраиваются не только нейрон-победитель \tilde{c}_*^K , но и его ближайшее окружение, при этом более близкие к победителю нейроны подтягиваются по входному вектору $\tilde{\varphi}_k$ больше чем удаленные от \tilde{c}_*^K центроиды \tilde{c}_l^K .

В качестве функции соседства обычно используется все тот же гауссиан, приобретающий в данном случае вид

$$\psi(\tilde{c}_*^K(k), \tilde{c}_l^K(k)) = e^{-\frac{\|\tilde{c}_*^K(k) - \tilde{c}_l^K(k)\|^2}{2\sigma_c^2}},$$

где σ_c^2 определяет размеры области соседства, при этом в процессе обучения этот параметр должен монотонно уменьшаться [134].

В [18] для обучения самоорганизующейся карты предлагается вообще не определять победителя, а в качестве функции соседства использовать выходной сигнал каждого нейрона

$$y_l(k) = \tilde{\varphi}_k^T \tilde{c}_l^K, \quad (3.15)$$

при этом правило (3.14) может быть переписано в виде

$$\tilde{c}_l^K(k) = \frac{\tilde{c}_l^K(k-1) + \eta(k)y_l(k)(\tilde{\varphi}_k - \tilde{c}_l^K(k-1))}{\left\| \tilde{c}_l^K(k-1) + \eta(k)y_l(k)(\tilde{\varphi}_k - \tilde{c}_l^K(k-1)) \right\|}, \forall l = 1, 2, \dots, m. \quad (3.16)$$

Заметив, что

$$\|\tilde{\varphi}_k\| = \|\tilde{c}_l^K(k-1)\| = 1,$$

несложно установить, что выражение (3.15) есть не что иное, как косинус угла между входным образом $\tilde{\varphi}_k$ и вектором-центроидом $\tilde{c}_l^K(k-1)$, т.е. $\cos(\tilde{\varphi}_k, \tilde{c}_l^K(k-1))$. Тогда с учетом неотрицательности функции соседства можно окончательно переписать правило самообучения в виде

$$\tilde{c}_l^K(k) = \frac{\tilde{c}_l^K(k-1) + \eta(k)[\cos(\tilde{\varphi}_k, \tilde{c}_l^K(k-1))]_+ (\tilde{\varphi}_k - \tilde{c}_l^K(k-1))}{\|\tilde{c}_l^K(k-1) + \eta(k)[\cos(\tilde{\varphi}_k, \tilde{c}_l^K(k-1))]_+ (\tilde{\varphi}_k - \tilde{c}_l^K(k-1))\|}, \forall l=1, 2, \dots, m \quad (3.17)$$

где $[\cdot]_+$ проектор на положительный ортант.

Процесс кластеризации заканчивается либо по исчерпанию выборки, содержащей N наблюдений, либо происходит непрерывно, если данные поступают в форме потока в on-line режиме.

Выводы по разделу 3

1. Предложена гибридная искусственная нейронная сеть, объединяющая в себе идеи ядерных систем и самообучения и построенная на основе эволюционирующей радиально-базисной нейронной сети и самоорганизующейся карты

2. Предложена архитектура гибридной нейронной сети и метод ее самообучения, предназначенные для ядерной кластеризации потока наблюдений, последовательно в on-line режиме поступающих на обработку. Предлагаемая система построена на основе эволюционирующей обобщенной регрессионной нейронной сети и самоорганизующейся карты Т.Кохонена.

3. Предложенные системы позволяют решать задачу on-line кластеризации в условиях, когда образуемые исходными данными классы

имеют произвольную форму. Введенные нейронные сети просты в реализации и позволяют решать достаточно широкий класс задач динамического интеллектуального анализа данных (DDM) и интеллектуального анализа потоков данных (DSM).

РАЗДЕЛ 4

АДАПТИВНЫЙ МЕТОД КОМБИНИРОВАННОГО ОБУЧЕНИЯ- САМООБУЧЕНИЯ НЕЙРО-ФАЗЗИ СИСТЕМ

В данном разделе рассматриваются архитектуры и алгоритмы самообучения нейро-фаззи гибридных систем вычислительного интеллекта, которые предназначены для кластеризации данных в условиях, когда формируемые кластеры могут иметь произвольную форму и взаимно пересекаться. Введенные процедуры самообучения достаточно просты в численной реализации и предназначены для обработки данных, последовательно в on-line режиме поступающих в систему.

4.1 Последовательная ядерная кластеризация на основе нейро-фаззи подхода

На сегодняшний день разработано множество разнообразных методов кластеризации, при этом подавляющее их большинство обрабатывает информацию в предположении, что весь массив исходных данных сформирован заранее, его объем не меняется в процессе анализа, а каждое наблюдение-вектор признаков может обрабатываться многократно. Понятно, что именно это исходное предположение делает невозможным применение традиционного подхода к кластеризации в ситуациях, когда данные поступают на обработку последовательно в виде потока информации [105] либо их объем настолько велик (Big Data) [135,136], что многократный анализ отдельного наблюдения просто невозможен.

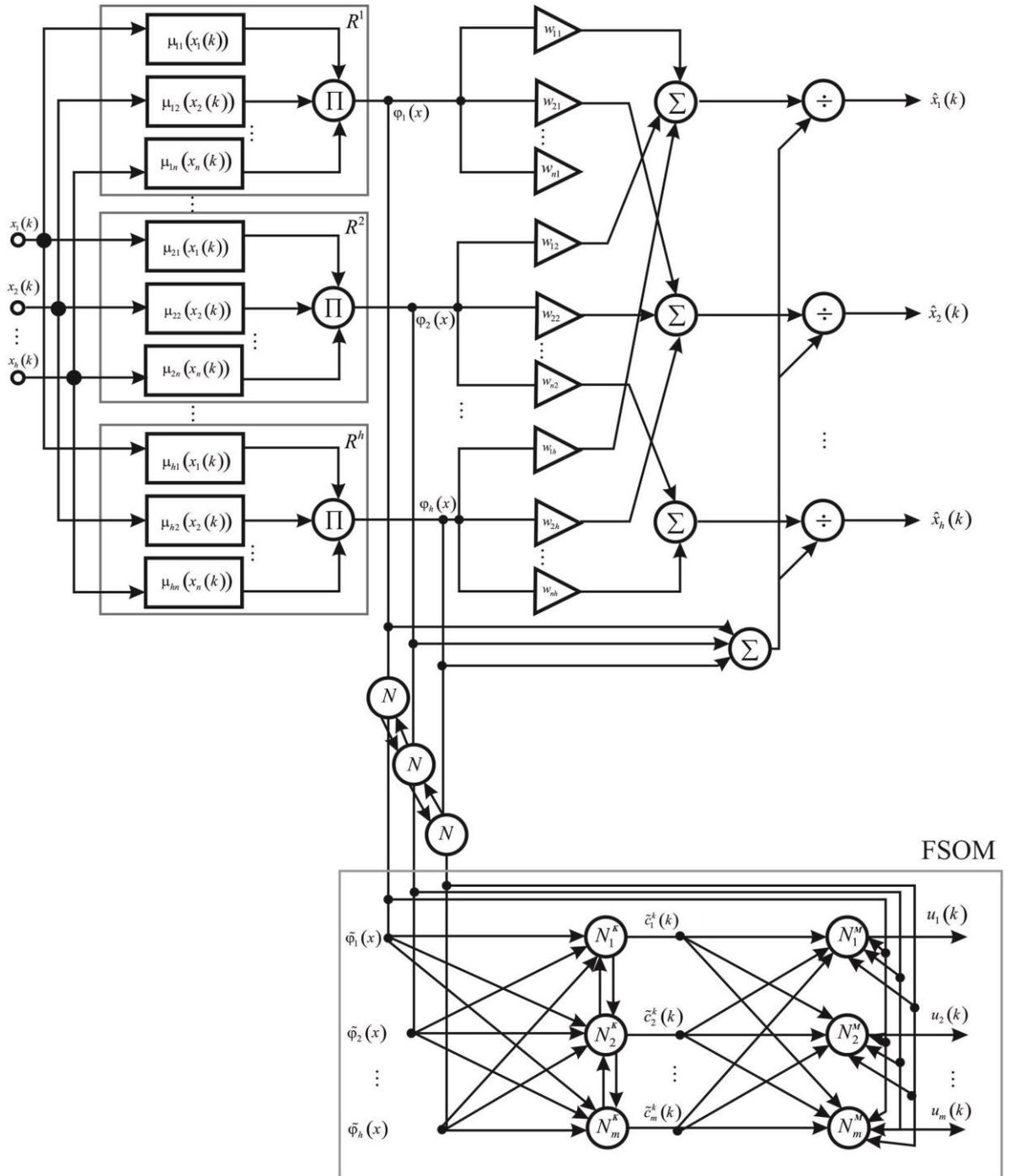
В связи с этим в задачах ядерной кластеризации более предпочтительным с позиций Big Data представляется использование подхода, основанного на оценках Э. Парзена [60], обобщенных

регрессионных нейронных сетях (GRNN) Д. Шпехта [41], основанных на «ленивом обучении» по принципу «нейроны в точках данных» [52], и теореме Т. Кавера [71] о возможности линейной разделимости классов в пространствах повышенной размерности. И хотя подобный подход также подвержен проклятию размерности и ориентирован на пакетный режим работы, он может быть адаптирован к on-line обработке больших массивов информации на основе гибридизации нейро-фаззи подхода и эволюционирующих систем вычислительного интеллекта [128-131], позволяющей в процессе самообучения «выращивать» архитектуру, наилучшим образом ориентированную на решение конкретной задачи нечеткой кластеризации данных, образующих классы произвольного вида.

В связи с этим представляется целесообразной разработка on-line нейро-фаззи систем для решения задачи последовательной нечеткой кластеризации данных, позволяющих обрабатывать векторы наблюдений любой размерности в условиях ограниченного числа наблюдений в обрабатываемой выборке.

На рис. 4.1 приведена архитектура предлагаемой ядерной кластеризующей on-line нейро-фаззи системы.

Исходной информацией для этой системы является центрированная относительно среднего выборка векторов наблюдений, возможно растущая $x(1), x(2), \dots, x(k), \dots, x(N), \dots$; $x(k) = (x_1(k), \dots, x_i(k), \dots, x_n(k))^T \in R^n$ таких, что $-1 \leq x_i(k) \leq 1$, $\frac{1}{N} \sum_{k=1}^N x_i(k) = 0$, которая должна быть разбита на m кластеров произвольной формы, при этом k здесь может быть как номером текущего наблюдения, так и моментом текущего времени.



4.1 – Архитектура ядерной кластеризующей нейро-фаззи системы.

Векторы наблюдений $x(k)$ последовательно поступают на нулевой (рецепторный) слой системы, откуда передаются на первый скрытый слой, образованный nh (по h на каждый вход) функциями

$\mu_{li}(x_i), l = 1, 2, \dots, h; i = 1, 2, \dots, n$ и выполняющий фаззификацию входного пространства R^n .

Второй скрытый слой обеспечивает агрегирование уровней принадлежности, рассчитанных в первом слое, и содержит h блоков умножения.

Таким образом, первые два слоя рассматриваемой системы полностью совпадают со слоями наиболее популярных нейро-фаззи систем ANFIS и TSK [137, 138], основным достоинством которых кроме универсальных аппроксимирующих свойств является то, что их выходной сигнал линейно зависит от настраиваемых параметров синаптических весов, что позволяет использовать для их настройки как множество известных линейных алгоритмов обучения [44, 55], так и известные методы адаптивной кластеризации [51, 139], оптимальные по быстродействию.

Итак, если на вход системы подан векторный сигнал $x(k)$, элементы первого скрытого слоя производят его фаззификацию, вычисляя уровни принадлежности $0 < \mu_{li}(x_i(k)) \leq 1$, при этом обычно в качестве функций принадлежности используются традиционные гауссианы

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma_i^2}\right),$$

где c_{li}, σ_i - параметры центров и ширины соответственно.

Заметим также, что предварительное кодирование данных на некоторый интервал, например, $-1 \leq x_i(k) \leq 1$ позволяет упростить расчеты, поскольку параметры ширины σ_i в этом случае могут быть приняты одинаковыми для всех входов, т.е. $\sigma_i = \sigma$.

Во втором скрытом слое вычисляются агрегированные значения

$$\varphi_l(k) = \prod_{i=1}^n \mu_{li}(x_i(k)),$$

при этом для гауссианов с одинаковыми параметрами ширины справедливо соотношение

$$\varphi_l(k) = \prod_{i=1}^n \mu_{li}(x_i(k)) = \prod_{i=1}^n \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = \exp\left(-\frac{\|x(k) - c_l\|^2}{2\sigma^2}\right)$$

(здесь $c_l = (c_{l1}, c_{l2}, \dots, c_{ln})^T$), т.е. сигналы на выходах блоков умножения второго скрытого слоя аналогичны сигналам на выходах нейронов R_l первого скрытого слоя обычных радиально-базисных нейронных сетей (RBFN).

Третий скрытый слой рассматриваемой системы – слой нормализации (N) реализует элементарное преобразование

$$\tilde{\varphi}_l(k) = \frac{\varphi_l(k)}{\|\varphi_l(k)\|}$$

(здесь $\varphi(x) = (\varphi_1(x), \dots, \varphi_l(x), \dots, \varphi_h(x))^T$), необходимое для эффективной работы четвертого выходного слоя, являющегося по сути кластеризующей нейро-фаззи сетью Кохонена (FSOM) [87], настройка параметров которой производится с помощью специализированной процедуры конкурентного самообучения. В этом слое решается задача разбиения последовательностей образов повышенной размерности $\tilde{\varphi}(x(1)), \dots, \tilde{\varphi}(x(k)), \dots, \tilde{\varphi}(x(N)), \dots$ на m кластеров с нахождением прототипов-центроидов $\tilde{c}_1^k, \dots, \tilde{c}_j^k, \dots, \tilde{c}_m^k$ (нейроны

N_j^k) и уровней принадлежности каждого $\tilde{\varphi}(k)$ к каждому j -му кластеру $u_j(k)$ (нейроны N_j^u), $\tilde{c}_1^k \in R^h$, $j=1,2,\dots,m$.

Важная проблема, возникающая в процессе синтеза системы, состоит в том, как эффективно организовать базис, образованный ядерными функциями, в котором можно было бы точно провести кластеризацию. Для этого необходимо организовать процесс адаптации количества $h > n$ и центров c_{ji} функций принадлежности первого слоя.

Для решения этой задачи предназначены пятый, шестой и седьмой дополнительные слои, образованные nh настраиваемыми синаптическими весами, $n+1$ сумматорами и n блоками деления, решающими задачи дефазификации.

Таким образом, первый, второй, пятый, шестой и седьмой слои рассматриваемой системы образуют по сути многовыходовую нейро-фаззи систему Ванга-Менделя [139] (TSK-система нулевого порядка), основным отличием которой является то, что в качестве обучающего здесь используется входной сигнал $x(k)$, т.е. система работает в режиме автоассоциации [140]. Выходом седьмого слоя является векторный сигнал $\hat{x}(k) \in R^n$, являющийся оценкой входного сигнала $x(k)$.

Качество восстановления оценивается на основе векторной ошибки

$$e(k) = x(k) - \hat{x}(k)$$

с помощью того или иного скалярного критерия, например,

$$\bar{e} = \frac{1}{N} \sum_{k=1}^N \frac{\|x(k) - \hat{x}(k)\|}{\|x\|^*}$$

(здесь $\|\bullet\|$ любая норма в метрике Минковского)

либо МАРЕ-оценки

$$\bar{e} = \frac{1}{N} \sum_{k=1}^N \frac{\sum_{i=1}^n |x_i(k) - \hat{x}_i(k)|}{\sum_{i=1}^n |x_i(k)|} \times 100\%.$$

Если окажется, что значение \bar{e} превышает некоторый априорно заданный порог e_T , принимается решение о том, что процесс настройки этих слоев должен продолжаться, т.е. h, c_{li}, w_{il} подлежат дальнейшему уточнению. Конечным результатом обучения этих слоев являются значения h, c_{li} и $(n \times h)$ -матрица синаптических весов $w = \{w_{il}\}$.

Необходимо также заметить, что на выходах пятого скрытого слоя формируется nh сигналов

$$w_{il} \prod_{i=1}^n \mu_{li}(x_i(k)) = w_{il} \varphi_l(k),$$

шестого - $n + 1$ сигналов

$$\sum_{l=1}^h w_{il} \prod_{i=1}^n \mu_{li}(x_i(k)) = \sum_{l=1}^h w_{il} \varphi_l(k),$$

$$\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k)) = \sum_{l=1}^h \varphi_l(k)$$

и, наконец, на выходах седьмого слоя n -сигналов

где

$$\hat{x}_i(k) = \frac{\sum_{l=1}^h w_{il} \prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \sum_{l=1}^h w_{il} \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} =$$

$$= \sum_{l=1}^h w_{il} \hat{\phi}_l(x(k)) = w_i^T \hat{\phi}^h(x(k)),$$

$$\hat{\phi}_l(x(k)) = \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))}, w_i = (w_{i1}, w_{i2}, \dots, w_{ih})^T,$$

$$\hat{\phi}^h(x(k)) = (\hat{\phi}_1(x(k)), \dots, \hat{\phi}_l(x(k)), \dots, \hat{\phi}_h(x(k)))^T.$$

Вводя далее в рассмотрение $(n \times 1)$ -вектор $\hat{x}(k) = (\hat{x}_1(k), \dots, \hat{x}_i(k), \dots, \hat{x}_n(k))^T$ и $(n \times h)$ -матрицу $W = (w_1, w_i, \dots, w_n)^T$, окончательно можно записать

$$\hat{x}(k) = W \hat{\phi}^h(x(k))$$

и

$$e(k) = x(k) - W \hat{\phi}^h(x(k)).$$

4.1.1 Обучение гибридной кластеризующей нейро-фаззи системы

Процесс обучения рассматриваемой нейро-фаззи системы сводится к самообучению – эволюции первого скрытого слоя, обучению с учителем матрицы синаптических весов W пятого скрытого слоя и конкурентному самообучению нейро-фаззи сети Кохонена четвертого выходного слоя.

В основу настройки первого скрытого слоя положены идеи эволюционирующих фаззи-систем и, прежде всего, адаптивный метод обучения эволюционирующей нейро-фаззи системы [131].

Пусть на вход системы, у которой в исходном состоянии в первом скрытом слое отсутствуют функции принадлежности, поступает первое наблюдение обучающей выборки $x(1) = (x_1(1), \dots, x_i(1), \dots, x_n(1))^T$. Это наблюдение формирует первый набор функций принадлежности $\mu_1 = (\mu_{11}, \dots, \mu_{1i}, \dots, \mu_{1n})^T$ так, что $c_{1i} = x_i(1)$.

Далее для векторной функции принадлежности μ_1 с центром $c_1(1)$ задается радиус соседства r , определяемый максимально возможным числом h функций принадлежности в системе. Так, если функции принадлежности по компонентам распределены равномерно, то

$$r = \frac{2}{h-1}.$$

Далее, при поступлении второго наблюдения $x(2)$ производится проверка условия

$$\max_i |c_{1i} - x_i(2)| \leq r. \quad (4.1)$$

Если это условие выполняется, производится коррекция центров функций принадлежности μ_1 согласно правилу

$$c_{1i}(2) = c_{1i}(1) + \eta(2)(x_i(2) - c_{1i}(1)),$$

где $\eta(2)$ -параметр шага обучения, например, при $\eta(2) = 0.5$

$$c_{1i}(2) = \frac{c_{1i}(1) + x_i(2)}{2}.$$

В том случае, если условие (4.1) не выполняется, формируется вторая векторная функция принадлежности с центрами

$$c_{2i}(2) = x_i(2).$$

Таким образом, формируется узел системы R_2 , образованный элементами первого и второго слоев.

Пусть к моменту поступления на вход системы наблюдения $x(k)$ сформировано p узлов R_l функций принадлежности $\mu_1, \mu_2, \dots, \mu_p, p < h$ с центрами $c_{li}(k-1), l=1, 2, \dots, p; i=1, 2, \dots, n$. С приходом $x(k)$ производится проверка условия

$$\max_i |c_{li} - x_i(k)| \leq r \quad \forall l=1, 2, \dots, p. \quad (4.2)$$

Если это условие выполняется, производится коррекция центров функций принадлежности, ближайших к соответствующим компонентам $x_i(k)$ согласно правилу

$$c_{li}(k) = c_{li}(k-1) + \eta(k)(x_i(k) - c_{li}(k-1)). \quad (4.3)$$

Несложно заметить, что (4.3) есть не что иное, как известное правило самообучения Т.Кохонена «Победитель получает все» [18] с той лишь разницей, что самообучение карты Кохонена реализуется на гиперсфере

$$\|x(k)\|_2 = 1,$$

а правило (4.3) – на гиперкубе

$$\|x(k)\|_\infty = 1.$$

В том случае, если условие (4.2) не выполняется, в системе формируется $(p+1)$ -й ($p+1 \leq h$) узел R_{p+1} с центрами функций принадлежности

$$c_{p+1,i}(k) = x_i(k).$$

Как видно, данная процедура есть гибрид эволюционирующего алгоритма Н.Касабова [129] и самоорганизующейся карты Т.Кохонена [18], при этом процесс эволюции архитектуры-самообучения функций принадлежности может протекать как непрерывно, так и до достижения числом функций принадлежности предельного значения nh .

Для настройки матрицы синаптических весов может быть использован либо рекуррентный метод наименьших квадратов, являющийся по сути оптимальной по быстродействию гауссовско-ньютоновской процедурой оптимизации вида

$$\begin{cases} W(k) = W(k-1) + \frac{(x(k) - W(k-1)\hat{\phi}^h(x(k)))\hat{\phi}^{hT}(x(k))P(k-1)}{1 + \hat{\phi}^{hT}(x(k))P(k-1)\hat{\phi}^h(x(k))}, \\ P(k) = P(k-1) - \frac{P(k-1)\hat{\phi}^h(x(k))\hat{\phi}^{hT}(x(k))P(k-1)}{1 + \hat{\phi}^{hT}(x(k))P(k-1)\hat{\phi}^h(x(k))}, \end{cases}$$

либо многомерная версия алгоритма Качмажа-Уидроу-Хоффа

$$\begin{aligned} W(k) &= W(k-1) + \frac{(x(k) - W(k-1)\hat{\phi}^h(x(k)))\hat{\phi}^{hT}(x(k))}{\|\hat{\phi}^h(x(k))\|^2} \hat{\phi}^{hT}(x(k)) = \\ &= W(k-1) + (x(k) - W(k-1)\hat{\phi}^h(x(k)))\hat{\phi}^{h+}(x(k)), \end{aligned}$$

где $(\square)^+$ -операция псевдообращения.

В основе конкурентного самообучения четвертого скрытого слоя лежит вероятностный алгоритм нечеткой кластеризации, основанный на оптимизации целевой функции вида [25]

$$E(u_i, \tilde{c}_j^K) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|\tilde{\varphi}(x(k)) - \tilde{c}_j^K\|^2$$

при ограничениях

$$\sum_{j=1}^m u_j(k) = 1, k = 1, 2, \dots, N,$$

$$0 \leq \sum_{k=1}^N u_j(k) \leq N, j = 1, 2, \dots, m,$$

где $u_j(k) \in [0, 1]$, β -неотрицательный параметр фаззификации, определяющий нечеткую границу между разными классами и влияющий на уровень нечеткости в окончательном разбиении данных по кластерам.

Применение стандартного аппарата нелинейного программирования, основанного на неопределенных множителях Лагранжа и решении системы уравнений Каруша-Куна-Таккера, ведет к результату

$$\left\{ \begin{array}{l} \tilde{c}_j^K = \frac{\sum_{k=1}^N u_j^\beta(k) \tilde{y}(x(k))}{\sum_{k=1}^N u_j^\beta(k)}, \\ u_j(k) = \frac{(\|\tilde{\varphi}(x(k)) - \tilde{c}_j^K\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (\|\tilde{\varphi}(x(k)) - \tilde{c}_l^K\|^2)^{\frac{1}{1-\beta}}}, \end{array} \right. \quad (4.4)$$

который при $\beta = 2$ совпадает с алгоритмом нечетких С-средних (FCM) Дж.Бездека:

$$\left\{ \begin{array}{l} \tilde{c}_j^K = \frac{\sum_{k=1}^N u_j^2(k) \tilde{\varphi}(x(k))}{\sum_{k=1}^N u_j^2(k)}, \\ u_j(k) = \frac{(\|\tilde{\varphi}(x(k)) - \tilde{c}_j^K\|^2)^{-2}}{\sum_{l=1}^m (\|\tilde{\varphi}(x(k)) - \tilde{c}_l^K\|^2)^{-2}}. \end{array} \right. \quad (4.5)$$

Алгоритмы (4.4), (4.5) получили широкое распространение при решении задач нечеткой кластеризации, однако их использование возможно только в пакетном режиме при фиксированном объеме N обрабатываемой выборки данных.

Если же данные поступают на обработку последовательно в on-line режиме, целесообразно использование рекуррентного варианта (4.4), основанного на процедуре нелинейного программирования Эрроу-Гурвица-Удзавы вида

$$\left\{ \begin{array}{l} \tilde{c}_j^K(k) = \tilde{c}_j^K(k-1) + \eta(k) u_j^\beta(k-1) (\tilde{\varphi}(x(k)) - \tilde{c}_j^K(k-1)), j = 1, 2, \dots, m, \\ u_j(k) = \frac{(\|\tilde{\varphi}(x(k)) - \tilde{c}_j^K\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (\|\tilde{\varphi}(x(k)) - \tilde{c}_l^K(k)\|^2)^{\frac{1}{1-\beta}}}. \end{array} \right. \quad (4.6)$$

Анализируя (4.6), можно заметить, что рассматривая сомножитель $u_j^\beta(k-1)$ в качестве функции соседства, приходим к правилу самообучения Кохонена на основе принципа «Победитель получает больше» (WTM), при $\beta \rightarrow 1$ получаем алгоритм типа K -средних, а $\beta \rightarrow 0$ соответствует стандартному правилу Кохонена типа «Победитель получает все» (WTA):

$$\tilde{c}_j^K(k) = \tilde{c}_j^K(k-1) + \eta(k) (\tilde{\varphi}(x(k)) - \tilde{c}_j^K(k-1)) \quad (4.7)$$

аналогичное (4.3). Несложно заметить также, что рекуррентная процедура (4.7) минимизирует целевую функцию вида

$$E(\tilde{c}_j^K) = \sum_k \|\tilde{\varphi}(x(k)) - \tilde{c}_j^K\|^2,$$

оптимизация которой ведет к обычной оценке среднего арифметического

$$\tilde{c}_j^K(k) = \frac{1}{k} \sum_{\varphi(\tilde{x}(k)) \in c_j} \tilde{\varphi}(x(k)),$$

или в рекуррентной форме:

$$\tilde{c}_j^K(k) = \tilde{c}_j^K(k-1) + \frac{1}{k} (\tilde{\varphi}(x(k)) - \tilde{c}_j^K(k-1)).$$

Такой выбор параметра шага $\eta(k)$ согласуется с правилами стохастической аппроксимации и придает результатам ясный физический смысл.

Таким образом, алгоритм самообучения (4.6) четвертого скрытого слоя окончательно может быть записан в виде

$$\begin{cases} \tilde{c}_j^K(k) = \tilde{c}_j^K(k-1) + \frac{u_j^\beta(k-1)}{k} (\tilde{\varphi}(x(k)) - \tilde{c}_j^K(k-1)), j = 1, 2, \dots, m, \\ u_j(k) = \frac{(\|\tilde{\varphi}(x(k)) - \tilde{c}_j^K(k)\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (\|\tilde{\varphi}(x(k)) - \tilde{c}_l^K(k)\|^2)^{\frac{1}{1-\beta}}}, \end{cases}$$

объединяющем в себе вычислительную простоту и последовательную обработку кохоненовского обучения с возможностями нечеткой кластеризации.

Векторы-центроиды кластеров $\tilde{c}_j^k(k)$, вычисляемые в пространстве повышенной размерности R^h , далее могут быть спроецированы в исходное пространство R^n с помощью матрицы синаптических весов на основе преобразования

$$c_j^k(k) = W(k)\tilde{c}_j^k(k) \quad \forall j = 1, 2, \dots, m,$$

$$c_j^k(k) \in R^n, \tilde{c}_j^k(k) \in R^h.$$

Таким образом, рассматриваемая здесь кластеризующая нейро-фаззи система является по сути объединением двух нейро-фаззи систем: эволюционирующей TSK-системы и нечеткой кластеризующей сети Кохонена, что позволяет в on-line режиме восстанавливать перекрывающиеся классы произвольной формы (линейно неразделимые), формируемые данными, поступающими на обработку в форме потока информации.

4.2 Последовательная ядерная нечеткая кластеризация больших массивов данных на основе гибридной системы вычислительного интеллекта

В задачах ядерной кластеризации более предпочтительным с позиций Big Data представляется использование подхода, основанного на оценках Э. Парзена [60], обобщенных регрессионных нейронных сетях (GRNN) Д. Шпехта [52], основанных на «ленивом обучении» по принципу «нейроны в точках данных» [48], и теореме Т. Кавера [71]. Подобный подход может быть адаптирован к on-line обработке больших массивов информации на основе гибридизации нейро-фаззи подхода и эволюционирующих систем вычислительного интеллекта [128], позволяющей в процессе самообучения

«выращивать» архитектуру, наилучшим образом ориентированную на решение конкретной задачи нечеткой кластеризации данных, образующих классы произвольного вида.

4.2.1 Архитектура ядерной нечеткой кластеризующей системы

Предлагаемая ядерная нечеткая кластеризующая система содержит четыре слоя обработки информации: первый скрытый слой фаззификации входного пространства, второй скрытый слой агрегирования, третий скрытый слой вычисления центроидов кластеров в пространстве повышенной размерности и четвертый выходной слой для расчета уровней принадлежности.

На входной нулевой слой системы подаются $(n \times 1)$ - мерные векторы входных сигналов-образов $x(k) = (x_1(k), \dots, x_i(k), \dots, x_n(k))^T$ (здесь $k = 1, 2, \dots, N, \dots$ - текущее дискретное время), предобработанные, так что

$-1 \leq x_i(k) \leq 1$, $\frac{1}{N} \sum_{k=1}^N x_i(k) = 0$, при этом поступающий поток данных должен быть разбит на m возможно пересекающихся кластеров произвольной формы. Первый скрытый слой содержит $n \cdot h$ (по h на каждый вход) функций принадлежности $\mu_{li}(x_i)$, $i = 1, 2, \dots, n$; $l = 1, 2, \dots, h$ и выполняет фаззификацию входного пространства, при этом число функций принадлежности на каждом входе $h > n$ задается либо априорно, либо определяется в процессе самообучения системы. Второй скрытый слой производит агрегирование уровней принадлежности, рассчитанных в первом слое, и состоит из h блоков умножения. Таким образом, первые два скрытых слоя рассматриваемой системы по архитектуре совпадают с соответствующими слоями известных нейро-фаззи систем Такаги-Сугено-Канга, Ванга-Менделя, ANFIS [73, 137, 139 - 141]. Основное отличие рассматриваемой системы от

указанных состоит в том, что обучение последних связано с настройкой синаптических весов с помощью тех или иных алгоритмов оптимизации, а настройка рассматриваемой системы основана на ленивом обучении и сводится к установке центров функций принадлежности. Можно заметить, что известные нейро-фаззи системы близки по своей природе к радиально-базисным нейронным сетям [69, 141], а предлагаемая система является нечетким аналогом GRNN [52], т.е. обобщенной регрессионной нейро-фаззи сетью [142-143].

Таким образом, если на вход нейро-фаззи системы подан векторный сигнал $x(k)$, элементы первого скрытого слоя вычисляют уровни принадлежности $0 \leq \mu_{li}(x_i(k)) \leq 1$, при этом в качестве функций принадлежности чаще всего используются одномерные гауссианы вида

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li}(k))^2}{2\sigma_i^2}\right),$$

где $c_{li}(k)$ -настраиваемый в процессе самообучения параметр центра l -той функции принадлежности на i -м входе, σ_i^2 - параметр ширины.

Во втором скрытом слое вычисляются агрегированные значения $\prod_{i=1}^n \mu_{li}(x_i(k)) = \varphi_l(k)$, при этом для гауссианов с одинаковыми параметрами ширины σ можно записать

$$\varphi_l(k) = \prod_{i=1}^n \mu_{li}(x_i(k)) = \prod_{i=1}^n \exp\left(-\frac{(x_i(k) - c_{li}(k))^2}{2\sigma_i^2}\right) = \exp\left(-\frac{\|(x(k) - c_l(k))\|^2}{2\sigma_i^2}\right),$$

где $c_l(k) = (c_{l1}(k), \dots, c_{li}(k), \dots, c_{ln}(k))^T$ - вектор центров многомерной активационной функции, сформированной на основе одномерных функций принадлежности на каждом входе.

Таким образом, если на вход системы подан $(n \times 1)$ - мерный вектор $x(k)$, на выходе второго скрытого слоя появится $(h \times 1)$ -мерный образ повышенной размерности $\varphi(k) = (\varphi_1(k), \dots, \varphi_l(k), \dots, \varphi_h(k))^T$. В дальнейшем нечеткой кластеризации подвергается последовательность $\varphi(1), \varphi(2), \dots, \varphi(N), \dots$.

Третий скрытый и четвертый выходной слои рассматриваемой системы образуют двухслойную нечеткую кластеризующую сеть Т. Кохонена [86, 87], в первом слое которой восстанавливаются центроиды кластеров $c_1^K, \dots, c_j^K, \dots, c_m^K \in R^h$, $j = 1, 2, \dots, m$, а в выходном - уровни принадлежности $u_j(k)$ каждого вектора $\varphi(k)$ к каждому кластеру с центроидом c_j^K .

4.2.2 Самообучение ядерной нечеткой кластеризующей системы

Процесс самообучения рассматриваемой системы состоит из настройки функций принадлежности первого скрытого слоя на основе ленивого обучения и методов эволюционирующих систем и настройки центроидов формируемых кластеров в третьем скрытом слое.

Процесс настройки функций принадлежности первого скрытого слоя для наглядности проиллюстрируем на примере двумерного вектора входов $x(k) = (x_1(k), x_2(k))^T$, $k = 1, 2, \dots$, при этом количество функций принадлежности на каждом входе $h = 5$. Таким образом, число настраиваемых центров $c_{li}(k)$ определяется значением $nh = 10$.

В простейшем случае ленивого обучения полагается $c_1(0) = x(1), c_2(0) = x(2), \dots, c_l(0) = x(l), \dots, c_5(0) = x(5)$, а дальнейшая коррекция

центров $c_l(k)$ по мере поступления данных в систему производится согласно технике, принятой в GRNN [51].

При последовательной обработке данных более удобным представляется подход, когда начальные положения центров функций принадлежности $c_{li}(0)$ равномерно распределены по координатным осям x_1, x_2 , при этом расстояние между ними определяется соотношением

$$\Delta(0) = \frac{x_{i\max} - x_{i\min}}{h-1} = \frac{2}{h-1} = 0.5$$

для $-1 \leq x_i(k) \leq 1$.

Данная ситуация иллюстрируется рис. 4.2

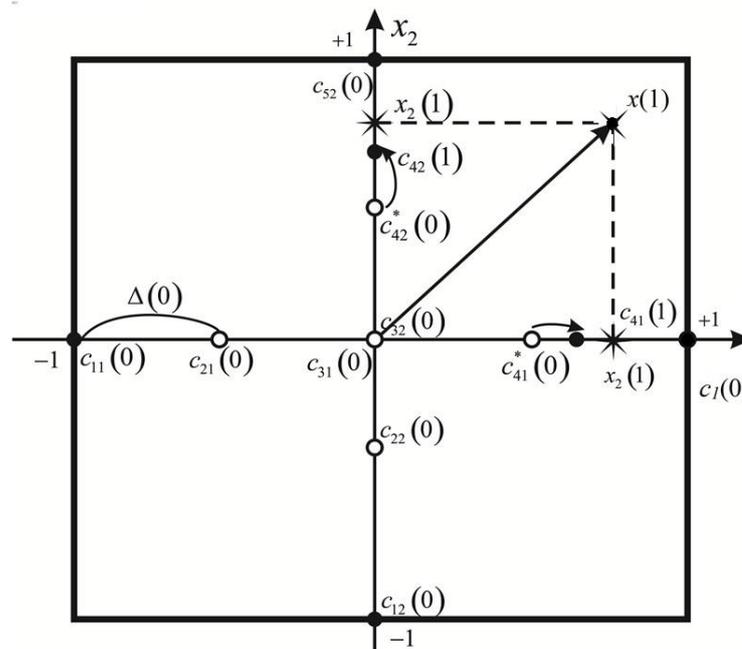


Рисунок 4.2 – Настройка функций принадлежности на первом шаге самообучения

Таким образом, в случае многомерного вектора входов $x(k) \in R^n$, центры $c_{li}(0)$ равномерно распределяются по осям гиперкуба $[-1, 1]^n$.

Пусть на вход системы подано первое наблюдение $x(1) = (x_1(1), x_2(1))^T$, обозначенное на рис. 1 символом *. Далее на каждой из осей находятся центры – «победители» (аналоги нейронов – «победителей» в SOM) $c_{li}^*(0)$ ближайšie к $x_i(1)$ в смысле расстояния

$$d_{li} = |x_i(1) - c_{li}^*(0)|,$$

т.е.

$$c_{li}^*(0) = \arg \min \{d_{1i}, d_{2i}, \dots, d_{hi}\}.$$

Поиск центра – «победителя» является по сути реализацией процесса конкуренции по Т. Кохонену [18] с той лишь разницей, что «победители» по различным осям могут принадлежать функциям принадлежности с разными индексами l . На рис.1 такими «победителями» являются центры $c_{41}^*(0)$ и $c_{42}^*(0)$.

Далее эти «победители» подтягиваются к компонентам входного сигнала $x_i(1)$ согласно WTA – правилу самообучения, которое для ситуации, приведенной на рис. 1, может быть записано в виде

$$c_{li}(1) = \begin{cases} c_{li}^* + \eta_{li}(k)(x_i(1) - c_{li}^*(0)) - \text{для победителя } (l = 4), \\ c_{li}(0) - \text{для всех остальных } l = 1, 2, 3, 5. \end{cases} \quad (4.8)$$

Для произвольных n, h и k WTA-правило самообучения первого скрытого слоя (4.8) имеет форму

$$c_{li}(k) = \begin{cases} c_{li}^*(k-1) + \eta_{li}(k)(x_i(k) - c_{li}^*(k-1)) - \text{для победителя при } l = 1, 2, \dots, h; i = 1, 2, \dots, n; \\ c_{li}(k-1) - \text{для всех остальных,} \end{cases} \quad (4.9)$$

где $\eta_{li}(k)$ -параметр шага обучения, монотонно уменьшающийся в процессе настройки.

В случае

$$\eta_{li}(k) = k_{li}^{-1}$$

(здесь k_{li} - число «побед» центра $c_{li}(k)$), правило (4.9) описывает последовательный on-line процесс кластеризации, соответствующий классическому методу K-средних.

Как уже отмечалось выше, последовательность $(h \times 1)$ - мерных векторов $\varphi(1), \varphi(2), \dots, \varphi(N), \dots$ подвергается нечеткой кластеризации в третьем и четвертом слоях рассматриваемой системы.

В классе процедур нечеткой кластеризации наиболее строгими с математической точки зрения являются алгоритмы, основанные на целевых функциях [25, 111] и решающие задачу их оптимизации при тех или иных априорных предположениях. Наиболее распространенным здесь является вероятностный подход, основанный на минимизации критерия (целевой функции)

$$E(u_j(k), c_j^K) = \sum_{k=1}^N \sum_{j=1}^m u_j^B(k) \left\| \varphi(k) - c_j^K \right\|^2 \quad (4.10)$$

при ограничениях

$$\sum_{j=1}^m u_j(k) = 1, \quad (4.11)$$

$$0 \leq \sum_{k=1}^N u_j(k) \leq N, \quad (4.12)$$

где β - неотрицательный параметр фаззификации (фаззификатор), определяющий «размытость» границ между кластерами.

Вводя функцию Лагранжа

$$L(u_j(k), c_j^K, \lambda(k)) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \left\| \varphi(k) - c_j^K \right\|^2 + \sum_{k=1}^N \lambda(k) \left(\sum_{j=1}^m u_j(k) - 1 \right)$$

(здесь $\lambda(k)$ - неопределенные множители Лагранжа) и решая систему уравнений Каруша-Куна-Таккера, несложно получить окончательный результат в виде

$$\left\{ \begin{array}{l} u_j(k) = \frac{\left(\left\| \varphi(k) - c_j^K \right\|^2 \right)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m \left(\left\| \varphi(k) - c_l^K \right\|^2 \right)^{\frac{1}{1-\beta}}}, \\ c_j^K = \frac{\sum_{k=1}^N u_j^\beta(k) \varphi(k)}{\sum_{k=1}^N u_j^\beta(k)}, \\ \lambda(k) = - \left(\sum_{l=1}^m \beta \left\| \varphi(k) - c_l^K \right\|^2 \right)^{\frac{1}{1-\beta} - 1}, \end{array} \right. \quad (4.13)$$

совпадающем при $\beta = 1$ с FCM Дж. Бездека [25], а при $\beta \rightarrow 1$ близкий к четкому методу К-средних.

В ситуациях, когда данные поступают на обработку последовательно в on-line режиме, может быть использована рекуррентная версия (4.17) вида [86, 106]

$$\begin{cases} u_j(k) = \frac{(\|\varphi(k) - c_j^K(k-1)\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (\|\varphi(k) - c_l^K\|^2)^{\frac{1}{1-\beta}}}, \\ c_j^K(k) = c_j^K(k-1) + \eta(k)u_j^\beta(k)(\varphi(k) - c_j^K(k-1)), \end{cases} \quad (4.14)$$

при этом несложно заметить, что второе соотношение (4.14) является по сути WTM-правилом самообучения Т. Кохонена, где множитель $u_j^\beta(k)$ играет роль функции соседства.

Основной недостаток вероятностного подхода связан с ограничением (4.12), требующим равенства единице суммы принадлежностей каждого наблюдения всем векторам. Этого недостатка лишен метод возможностной нечеткой кластеризации, введенный в [32].

В возможностном методе целевая функция имеет вид

$$E(u_j(k), c_j^K) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|\varphi(k) - c_j^K\|^2 + \sum_{j=1}^m \mu_j \sum_{k=1}^N (1 - u_j(k))^\beta, \quad (4.15)$$

где скалярный параметр $\mu_j > 0$ определяет расстояние, на котором уровень принадлежности принимает значение 0,5, т.е. если

$$\|\varphi(k) - c_j^K\|^2 = \mu_j,$$

то

$$u_j(k) = 0.5.$$

Прямая минимизация (4.15) по $u_j(k)$ и c_j^K ведет к известному решению

$$\left\{ \begin{array}{l} u_j(k) = \left(1 + \left(\frac{\|\varphi(k) - c_j^K\|^2}{\mu_j} \right)^{\frac{1}{1-\beta}} \right)^{-1}, \\ c_j^K = \frac{\sum_{k=1}^N u_j^\beta(k) \varphi(k)}{\sum_{k=1}^N u_j^\beta(k)}, \\ \mu_j = \frac{\sum_{k=1}^N u_j^\beta(k) \|\varphi(k) - c_j^K\|^2}{\sum_{k=1}^N u_j^\beta(k)}, \end{array} \right. \quad (4.16)$$

которое в рекуррентном варианте имеет вид:

$$\left\{ \begin{array}{l} u_j(k) = \frac{1}{1 + \left(\frac{\|\varphi(k) - c_j^K(k-1)\|^2}{\mu_j(k)} \right)^{\frac{1}{\beta-1}}}, \\ c_j^K(k) = c_j^K(k-1) + \eta(k) u_j^\beta(k) (\varphi(k) - c_j^K(k-1)), \\ \mu_j(k) = \left(\sum_{p=1}^k u_j^\beta(p) \right)^{-1} \left(\sum_{p=1}^k u_j^\beta(p) \|\varphi(p) - c_j^K(k)\|^2 \right), \end{array} \right. \quad (4.17)$$

где второе соотношение представляет собой WTM-правило самообучения с функцией соседства, определяемой первым соотношением процедуры.

Необходимо отметить, что использование процедур (4.13), (4.14), (4.16) и (4.17) в задачах кластеризации данных высокой размерности (а именно такие данные обрабатываются рассматриваемой системой) может не привести к желаемому результату из-за, так называемого, эффекта

«концентрации нормы» (CoN) [108, 144]. Для преодоления этого негативного эффекта Ф. Клавонном и Ф. Хёппнером [108] в задачах вероятностной нечеткой кластеризации было предложено вместо целевой функции (4.10) использовать выражение

$$E(u_j(k), c_j^K) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1-\alpha)u_j(k)) \|\varphi(k) - c_j^K\|^2 \quad (4.18)$$

с ограничениями (4.11), (4.12), где $0 < \alpha \leq 1$ - настраиваемый параметр, определяющий характер получаемого решения.

Вводя функцию Лагранжа

$$L(u_j(k), c_j^K, \lambda(k)) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1-\alpha)u_j(k)) \|\varphi(k) - c_j^K\|^2 + \sum_{k=1}^N \lambda(k) (\sum_{j=1}^m u_j(k) - 1) \quad (4.19)$$

и решая систему уравнений Каруша-Куна-Таккера, приходим к решению

$$\left\{ \begin{array}{l} u_j(k) = \frac{\alpha - 1}{2\alpha} + \frac{1 + m \frac{1-\alpha}{2\alpha}}{\sum_{l=1}^m \frac{\|\varphi(k) - c_j^K\|^2}{\|\varphi(k) - c_l^K\|^2}}, \\ c_j^K = \frac{\sum_{k=1}^N (\alpha u_j^2(k) + (1-\alpha)u_j(k)) \varphi(k)}{\sum_{k=1}^N (\alpha u_j^2(k) + (1-\alpha)u_j(k))}, \\ \lambda(k) = - \frac{1 + m \frac{1-\alpha}{2\alpha}}{\sum_{l=1}^m (2\alpha \|\varphi(k) - c_l^K\|)^{-1}}. \end{array} \right.$$

Несложно заметить, что при $\alpha = 1$ приходим к стандартной FCM-процедуре.

Применяя далее к лагранжиану (4.19) процедуру нелинейного программирования Эрроу-Гурвица-Удзавы, приходим к рекуррентному алгоритму нечеткой кластеризации по критерию (4.19):

$$\begin{cases} u_j(k) = \frac{\alpha - 1}{2\alpha} + \frac{1 + m \frac{1 - \alpha}{2\alpha}}{\sum_{l=1}^m \frac{\|\varphi(k) - c_l^K\|^2}{\|\varphi(k) - c_j^K\|^2}}, \\ c_j^K(k) = c_j^K(k-1) + \eta(k)(\alpha u_j^2(k) + (1 - \alpha)u_j(k))(\varphi(k) - c_j^K(k-1)), \end{cases} \quad (4.20)$$

при этом несложно заметить, что (4.14) и (4.20) совпадают при $\beta = 2$, $\alpha = 1$ и представляют собой рекуррентную версию FCM [71].

Подход к нечеткой кластеризации, основанный на критерии (4.19), может быть распространен на возмозможностные процедуры путем использования целевой функции вида

$$E(u_j(k), c_j^K) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1 - \alpha)u_j(k)) \|\varphi(k) - c_j^K\|^2 + \sum_{j=1}^m \mu_j \sum_{k=1}^N (\alpha(1 - u_j(k)))^2,$$

минимизация которой ведет к результату [145]:

$$\begin{cases} u_j(k) = \frac{\left(\alpha \mu_j + (1 - \alpha) \|\varphi(k) - c_j^K\|^2\right) + \mu_j}{2\alpha \left(\|\varphi(k) - c_j^K\|^2 + \mu_j\right)}, \\ c_j^K = \frac{\sum_{k=1}^N \left(\alpha u_j^2(k) + (1 - \alpha)u_j(k)\right) \varphi(k)}{\sum_{k=1}^N \left(\alpha u_j^2(k) + (1 - \alpha)u_j(k)\right)}, \\ \mu_j = \frac{\sum_{k=1}^N \left(\alpha u_j^2(k) + (1 - \alpha)u_j(k)\right) \|\varphi(k) - c_j^K\|^2}{\sum_{k=1}^N \left(\alpha u_j^2(k) + (1 - \alpha)u_j(k)\right)}, \end{cases}$$

а в рекуррентном варианте:

$$\left\{ \begin{array}{l} u_j(k) = \frac{\left(\alpha \mu_j(k) + (1 - \alpha) \|\varphi(k) - c_j^K(k-1)\|^2 \right) + \mu_j(k)}{2\alpha \left(\|\varphi(k) - c_j^K(k-1)\|^2 + \mu_j(k) \right)}, \\ c_j^K(k) = c_j^K(k-1) + \eta(k) \left(\alpha u_j^2(k) + (1 - \alpha) u_j(k) \right) \left(\varphi(k) - c_j^K(k-1) \right), \\ \mu_j = \frac{\sum_{p=1}^N \left(\alpha u_j^2(p) + (1 - \alpha) u_j(p) \right) \|\varphi(p) - c_j^K(k-1)\|^2}{\sum_{p=1}^k \left(\alpha u_j^2(p) + (1 - \alpha) u_j(p) \right)}. \end{array} \right. \quad (4.21)$$

Заметим, что в (4.19) и (4.20) алгоритмы настройки центроидов кластеров совпадают, являясь по сути WTM-правилом самообучения, а отличие состоит в определении уровней принадлежности к конкретным кластерам.

Таким образом, предлагаемая ядерная гибридная нейро-фаззи система реализует по сути двухэтапную процедуру нечеткой кластеризации, где в первых двух слоях производится грубая настройка центров функций принадлежности на основе обычной SOM, обучаемой с помощью WTA-правила, а в выходных слоях точная настройка на основе нечеткой on-line кластеризации данных повышенной размерности с использованием модифицированного WTM-правила.

Выводы по разделу 4

1. Впервые предложена многослойная гибридная нейро-фаззи система вычислительного интеллекта, на основе системы Ванга-Менделя и нечеткой кластеризующей самоорганизующей сети, которая позволяет в процессе самообучения настраивать не только свои параметры, но и архитектуру в он-

line режиме и решать задачи кластеризации потока данных при условии априорно неизвестной формы кластеров и уровней их перекрытия.

2. Предложена архитектура и алгоритмы самообучения нейро-фаззи гибридной системы вычислительного интеллекта для кластеризации данных в условиях, когда формируемые кластеры могут иметь произвольную форму и взаимно пересекаться. В основе предлагаемой системы лежат нечеткая обобщенная регрессионная нейронная сеть и нейро-фаззи кластеризующая сеть Т. Кохонена, настройка которых основана как на ленивом, так и на обучении, связанным с оптимизацией.

3. Введенные процедуры самообучения достаточно просты в численной реализации и предназначены для обработки данных, последовательно в on-line режиме поступающих в систему.

РАЗДЕЛ 5

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И РЕШЕНИЕ ПРАКТИЧЕСКИХ ЗАДАЧ

В данном разделе приведены результаты моделирования разработанных методов ядерной кластеризации. Имитационное моделирование выполнялось как на тестовых выборках, так и реальных данных. Результаты моделирования предложенных методов были сравнены со стандартными методами кластеризации с целью сравнительного оценки сравнительного качества решения рассматриваемых задач.

Моделирование систем кластеризации данных выполнялось на языке R в среде RStudio. Язык R является уникальным языком для научных вычислений, обладает богатейшим инструментарием для статистического моделирования, обработки данных, интеллектуального анализа данных, динамического программирования, визуализации и т.п., благодаря чему стремительно завоёвывает популярность в исследовательских кругах всего мира, постепенно вытесняя MATLAB в качестве универсальной платформы научных разработок.

5.1 Имитационное моделирование ядерной кластеризующей нейронной сети

Для подтверждения работоспособности разработанной ядерной кластеризующей нейронной сети, основанной на обобщенной регрессионной нейронной сети, была решена задача кластеризации на основе тестовых выборок из UCI-репозитория [146].

Были взяты наборы данных:

1. Тестовая выборка «Iris». Выборка состоит из данных о 150 экземплярах ириса, по 50 экземпляров из трёх видов — Ирис щетинистый

(*Iris setosa*), Ирис виргинский (*Iris virginica*) и Ирис разноцветный (*Iris versicolor*). Для каждого экземпляра измерялись четыре характеристики (в сантиметрах): длина чашелистика (англ. *sepal length*); ширина чашелистика (англ. *sepal width*); длина лепестка (англ. *petal length*); ширина лепестка (англ. *petal width*).

2. Тестовая выборка «Wine». Выборка включает 178 векторов наблюдений, разделённых на 3 класса, каждое наблюдение содержит 13 признаков.

С целью оценки эффективности ядерной кластеризующей нейронной сети (KSOM-GRNN) результаты кластеризации были сравнены со стандартным методом С-средних (FCM) с параметром фаззификации $\beta = 2$, стандартной картой Кохонена (SOM) и обобщенной регрессионной нейронной сетью (GRNN). Поскольку для каждой выборки существуют метки верной классификации, эффективность кластеризации измерялась в процентах точности относительно эталонного. В каждой ячейке таблицы 5.1 приведён средний, минимальный и максимальный результат для серии из 50 экспериментов.

Таблица 5.1 - Сравнение точности кластеризации на тестовых выборках

Исследуемые методы	Iris			Wine		
	avg	max	min	avg	max	min
Стандартная карта Т. Кохонена (SOM)	83%	92%	80%	85%	94%	77%
Метод нечётких С-средних (FCM with $\beta = 2$)	92%	95%	84%	93%	96%	85%
Ядерная кластеризующая нейронная сеть (KSOM-GRNN)	95%	97%	88%	95%	97%	90%

Результаты показывают, что эффективность ядерной кластеризующей нейронной сети (KSOM-GRNN) выше и стабильнее, чем у SOM, FCM и GRNN.

На рисунке 5.1 и рисунке 5.2 приведены результаты кластеризации с помощью ядерной кластеризующей нейронной сети (KSOM-GRNN) тестовых выборок «Iris» и «Wine».

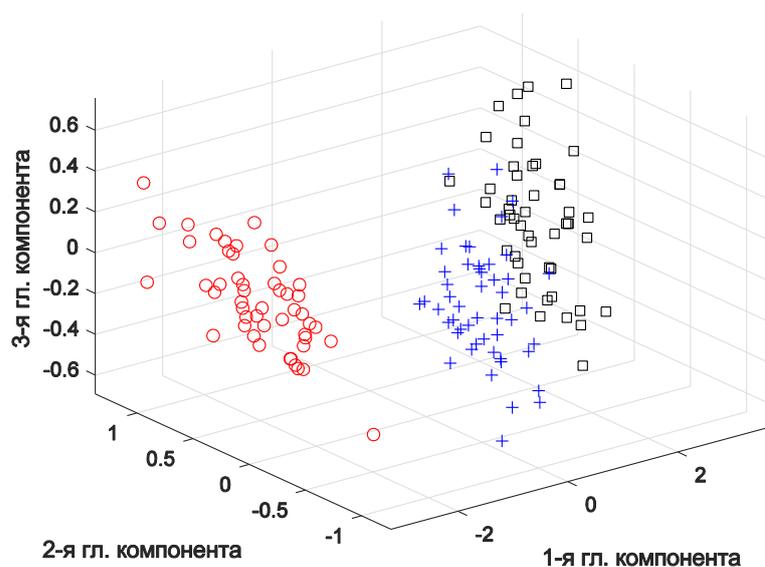


Рисунок 5.1 – Результаты кластеризации тестовой выборки «Iris» на основе ядерной кластеризующей нейронной сети (KSOM-GRNN) (визуализация первых трех главных компонентов)

В таблице 5.2 представлена эволюция центроидов разработанной нейронной сети.

Таблица 5.2 - Первые три этапа процедур обучения центроидов

Номер центроида	Координаты центроидов
C_{11}	-1.5886 0.4635 0.6624
C_{12}	1.3848 0.3887 1.8371
C_{13}	0.6149 0.5035 -1.9144

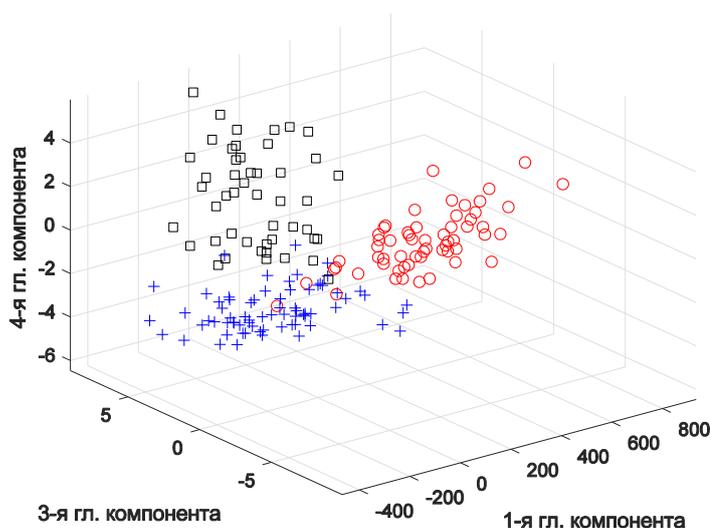


Рисунок 5.2 – Результаты кластеризации тестовой выборки «Wine» на основе ядерной кластеризующей нейронной сети (KSOM-GRNN)
(визуализация 1-й, 3-й и 4-й компонент)

Ввиду того, что точность качества кластеризации рассмотренных выше архитектур идентичны их пакетным аналогам, наиболее интересной для экспериментального исследования была признана скорость обучения системы. В качестве меры модельного времени, как минимальный общий квант пакетных и рекуррентных форм данных методов, принято количество проходов (эпох, итераций) по всей доступной выборке наблюдений. В проведённой серии экспериментов тестировалось время, за которое система достигает заданной точности кластеризации. Для тестирования использовалась традиционная выборка Wine (178 13-мерных наблюдений, разделённых на 3 класса). С каждым из рассмотренных методов и архитектур была проведена серия из 50 экспериментов. Каждый эксперимент включал 25 итераций обучения. Изначально система тестировалась на обучающем множестве, включающем 66% выборки, после чего измерялась точность кластеризации на всей выборке. На графиках на рис. 5.3 приведены средняя

точность кластеризации каждой из архитектур в зависимости от количества проходов по выборке. Заметим, что из-за большого количества вычисляемых параметров, ядерная кластеризирующая нейронная сети, требует для качественной настройки большего количества наблюдений, чем пакетные формы, что особенно выгодно выделяет синтезированную архитектуру на фоне других уже известных систем, традиционно отличающихся нестабильностью результатов.

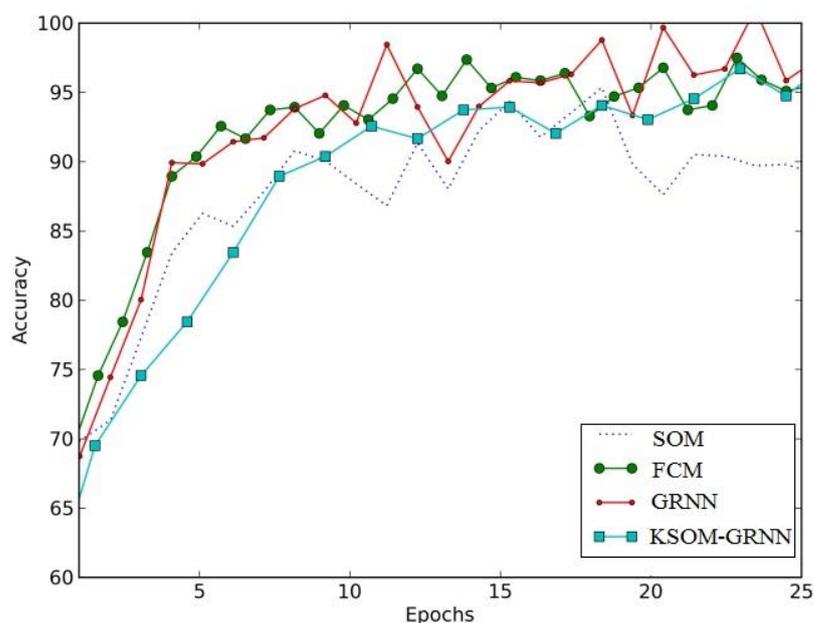


Рисунок 5.3 – Средняя точность кластеризации рассматриваемых методов

5.2 Имитационное моделирование многослойной гибридной нейро-фаззи системы вычислительного интеллекта

Эффективность предложенной многослойной гибридной нейро-фаззи системы и процедур ее самообучения продемонстрирована при решении задачи нечеткой кластеризации тестовых данных, на основе двух выборок:

1. выборка данных диагностирования рака молочной железы (Wisconsin Diagnostic Breast Cancer), которая включает 569

наблюдений, разделённых на 2 класса, каждое наблюдение содержит 30 признаков [146]

2. выборка данных диагностирования диабета (Pima Indians Diabetes), которая включает 768 наблюдений, разделённых на 2 класса, каждое наблюдение содержит 8 признаков [146].

Все данные были изначально пронормированы на гиперкуб в интервале $[-1,1]$ и отцентрированы относительно среднего значения.

Для сравнения эффективности кластеризации предложенной многослойной гибридной нейро-фаззи системы были выбраны самоорганизующаяся карта Т. Кохонена и метод нечетких С-средних (FCM) с параметром фаззификации $\beta = 2$.

Поскольку для каждой выборки существуют метки верной кластеризации, эффективность кластеризации измерялась в процентах точности кластеризации относительно эталонного. В каждой ячейке таблицы 5.3 приведён средний, минимальный и максимальный результат для серии из 50 экспериментов. Результаты сравнения с существующими методами представлены в табл. 5.3.

Таблица 5.3 - Сравнение точности кластеризации на тестовых выборках

Исследуемые методы	Wisconsin Diagnostic Breast Cancer			Pima Indians Diabetes		
	avg	max	min	avg	max	min
Стандартная карта Т. Кохонена (SOM)	80%	90%	71%	81%	90%	73%
Метод нечётких С-средних (FCM with $\beta = 2$)	89%	94%	81%	89%	96%	84%
Многослойная гибридная нейро-фаззи система	93%	96%	89%	92%	95%	91%

Как видно из табл. 5.3, предложенная гибридная нейро-фаззи система показывает преимущество по сравнению с исследуемыми методами. Стоит отметить, что с увеличением выборки ошибка кластеризации предложенной нейро-фаззи системы уменьшалась.

Результат кластеризации с помощью многослойной гибридной нейро-фаззи системы для тестовых выборок «Wisconsin Diagnostic Breast Cancer» и «Pima Indians Diabetes» приведены на рис. 5.4. и рис. 5.5.

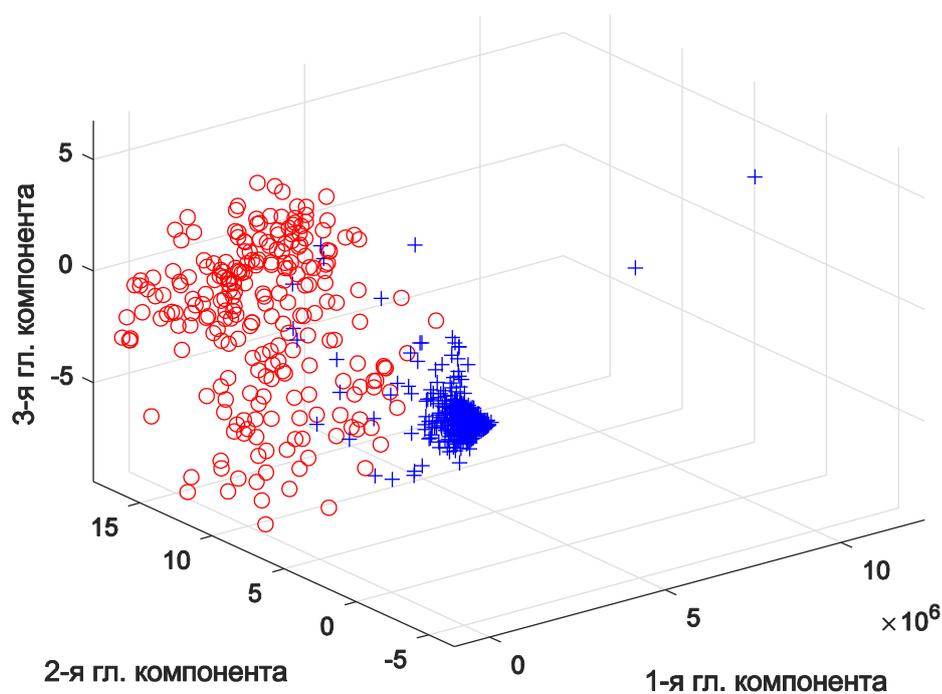


Рисунок 5.4 – Результаты кластеризации гибридной нейро-фаззи системой на основе выборки Wisconsin Diagnostic Breast Cancer (визуализация на первые три главные компоненты)

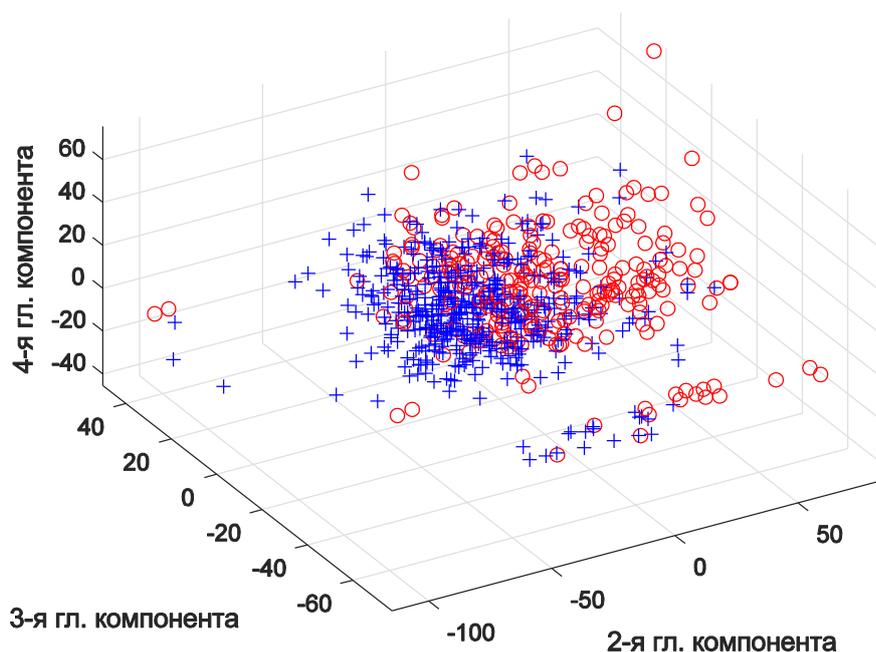


Рисунок 5.5 – Результаты кластеризации гибридной нейро-фаззи системой на основе выборки Pima Indians Diabetes (визуализация на первые три главные компоненты)

5.3 Имитационное моделирование модифицированного EM нечеткого вероятностного метода кластеризации

Эффективность предложенного модифицированного EM нечеткого вероятностного метода кластеризации, основанного на правиле самообучения «победитель получает все» (WTA - правило), была исследована на выборках из UCI репозитория [146]:

1. Выборка данных «Liver Disorders Data Set», которая содержала 345 наблюдений, разделённых на 2 класса, каждое наблюдение содержит 7 признаков;
2. Выборка данных «Glass Identification Data Set», которая содержала 214 наблюдений, разделённых на 7 классов, каждое наблюдение содержит 10 признаков.

Результаты кластеризации предложенного модифицированного EM нечеткого вероятностного метода кластеризации были сравнены со стандартной самоорганизующейся картой Т. Кохонена (SOM), нечетким алгоритмом кластеризации (FCM) с параметром фаззификации $\beta = 2$, EM вероятностным алгоритмом кластеризации.

Поскольку для каждой выборки существуют метки верной кластеризации, эффективность кластеризации измерялась в процентах точности кластеризации относительно эталонного. В каждой ячейке таблицы 5.4 приведён средний, минимальный и максимальный результат для серии из 50 экспериментов. Результаты сравнения с выше указанными методами представлены в табл. 5.4.

Таблица 5.4 - Результаты сравнения точности кластеризации на основе предложенного модифицированного EM нечеткого вероятностного метода кластеризации

Исследуемые методы	Liver Disorders Data Set			Glass Identification Data Set		
	avg	max	min	avg	max	min
Стандартная карта Т. Кохонена (SOM)	89%	91%	85%	90%	92%	88%
Метод нечётких С-средних (FCM $\beta = 2$)	94%	97%	91%	93%	96%	92%
EM вероятностный алгоритм кластеризации	94%	96%	90%	92%	96%	91%
Модифицированный EM нечеткий вероятностный метод кластеризации	96%	98%	94%	95%	98%	93%

Как видно из табл. 5.4, предложенный модифицированный EM нечеткий вероятностный метод кластеризации показывает преимущество по сравнению с исследуемыми методами.

На рисунке 5.6 и рисунке 5.7 представлены результаты визуализации работы модифицированного EM нечеткого вероятностного метода кластеризации на анализируемых тестовых выборках.

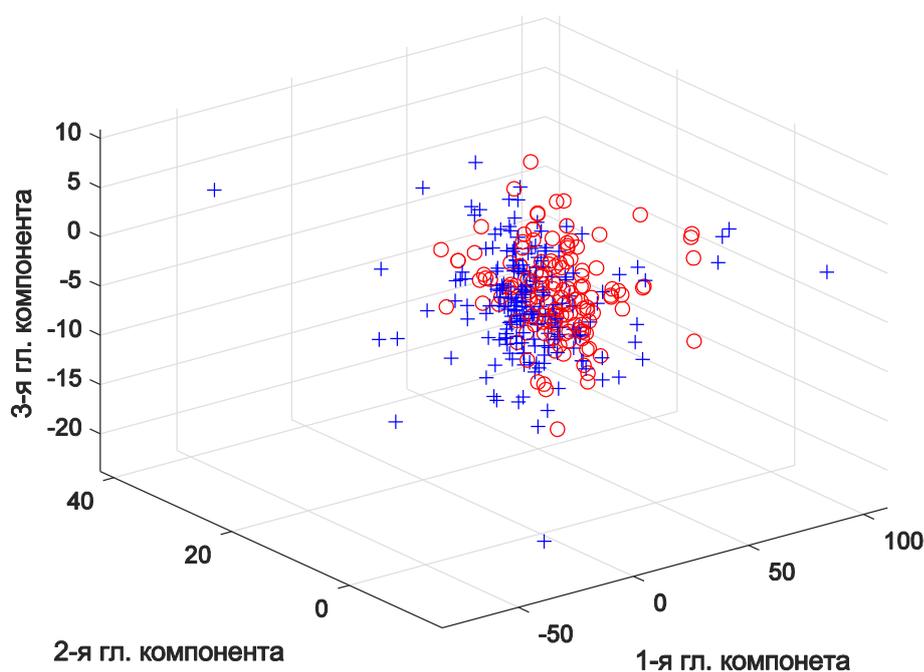


Рисунок 5.6 – Результаты кластеризации модифицированного EM нечеткого вероятностного метода кластеризации на основе выборки «Liver Disorders Data Set» (визуализация на первые три главные компоненты)

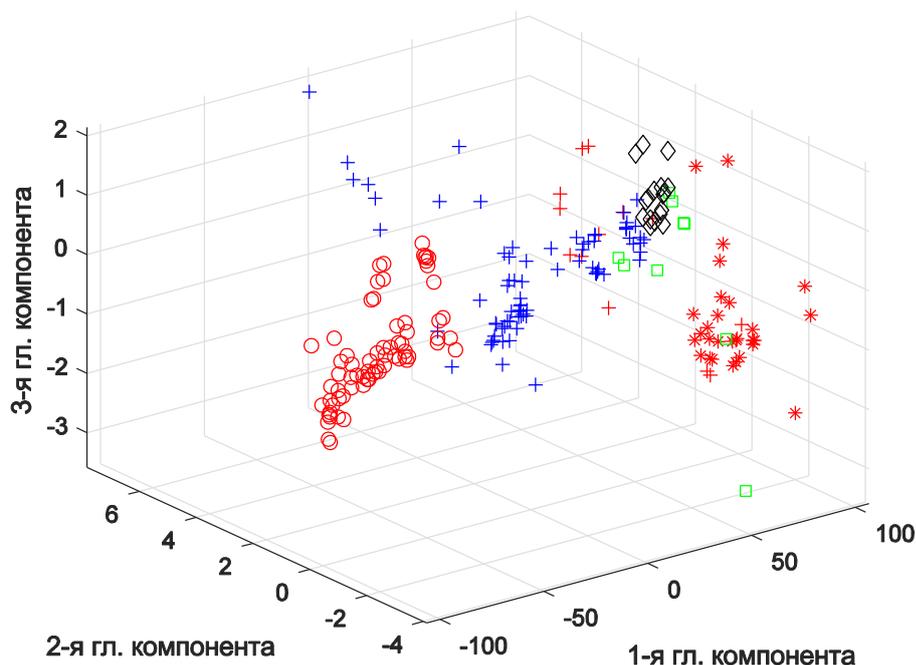


Рисунок 5.7 – Результаты кластеризации модифицированного EM нечеткого вероятностного метода кластеризации на основе выборки «Glass Identification Data Set» (визуализация на первые три главные компоненты)

5.4 Имитационное моделирование усовершенствованной искусственной нейронной сети для анализа главных компонент

Эффективность работы усовершенствованной искусственной нейронной сети для анализа главных компонент исследовалась в процессе решения задачи компрессии тестовых данных в виде таблицы «объект-свойство» с использованием выборки данных из репозитория UCI [146]: Nayes-roth. Данные содержат 132 наблюдения (по 5 признаков) в трех классах.

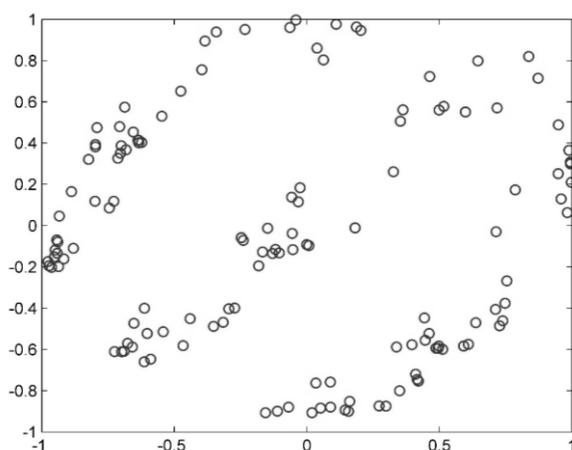
Для сравнения качества компрессии и скорости сходимости была исследована работа подобных систем компрессии данных. Обработка тестовых данных с помощью рассматриваемых методов проводилась до достижения наперед определенной (минимальной для каждого из методов) ошибки обучения. Для анализа были выбраны первые две главные компоненты.

Эксперимент проводился при разных начальных условиях 20 раз, результаты усреднены.

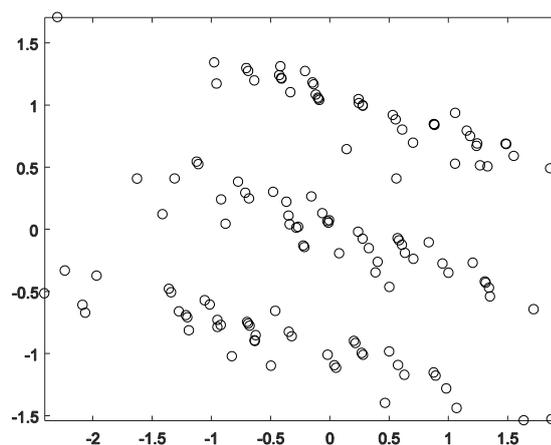
Таблица 5.5 – Результаты работы нейро-систем анализа главных компонент для тестовых данных «Hayes-roth»

Нейронная сеть	RMSE	Среднее время обучения, сек
Усовершенствованная искусственная нейронная сеть для анализа главных компонент	1,234	2,1
Нейро-фаззи-компрессор	0,7698	5,61
Нейронная сеть "Bottle neck"	0,6745	10,32

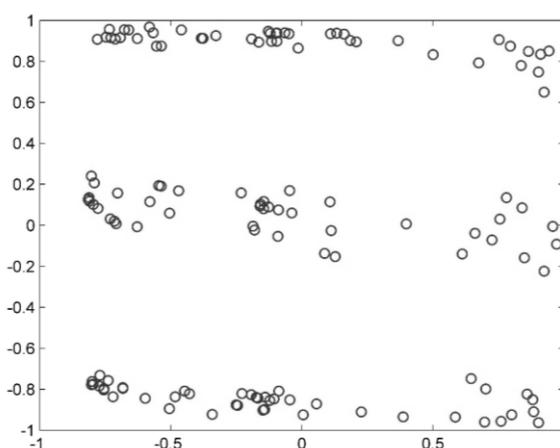
Как видно из рис.5.8, данные, которые сжаты с помощью усовершенствованной искусственной нейронной сети для анализа главных компонент, представляют более компактные кластеры, чем те что были сжаты нейро-фаззи компрессором и нейронной сетью "Bottle neck".



а)



б)



в)

Рисунок 5.8 – Результаты анализа главных компонент тестовых данных «Hayes-roth» после компрессии: а) на основе нейронной сети "Bottle neck", б) на основе нейро-фаззи-компрессора, в) на основе усовершенствованной искусственной нейронной сети для анализа главных компонент

Среднее время обучения усовершенствованной искусственной нейронной сети для анализа главных компонент на 6% меньше, чем время обучения нейро-фаззи-компрессора, и на 11% меньше, чем время обучения нейронной сети "Bottle neck".

5.5 Решение практической задачи анализа политематических текстов с целью категоризации на кластеры для ускорения информационного поиска

В условиях стремительного развития Web-пространства, увеличения хранимого в нем объема информации, появляются новые проблемы, связанные с обработкой столь большего количества данных. В частности, это проблема релевантного поиска информации в Web, категоризация и рубрикация документов (для каталоговых служб); подстройка персональных профилей интересов в тематическом пространстве; определение расстояния между персональными профилями, открывающие возможность плодотворного обмена информацией между близкими по интересам пользователями [147, 148].

Решение такого рода задач особо актуально для категоризации документации по программным продуктам (software), которые разрабатываются в IT-компаниях. Для их решения целесообразно использование методов кластеризации. Поскольку Web-пространство состоит из набора документов, содержащих некоторую текстовую информацию, либо текстовые документы различной направленности, то задачу кластеризации можно рассматривать как задачу кластеризации политематических текстовых документов [149, 150].

Под кластеризацией политематических текстовых документов понимают процесс классификации массива полнотекстовых документов, исходящий только из анализа и выявления внутренней тематической структуры этого массива без наличия априорной информации о нем, т.е. при отсутствии предопределенного рубрикатора и множества документов-образцов. Классификация документов с использованием методов кластеризации приводит к разбиению множества документов на однородные в соответствующем понимании группы, или кластеры, путем автоматического анализа тематической близости между ними.

Кластеризация текстов основывается на кластерной гипотезе, тесно связанные по смыслу документы стремятся быть релевантными одним и тем

же запросам, т.е. документы, релевантные запросу, отделимы от тех, которые нерелевантны этому запросу.

На рис. 5.9 представлена функциональная схема процесса кластеризации массивов текстовых документов в общем виде.

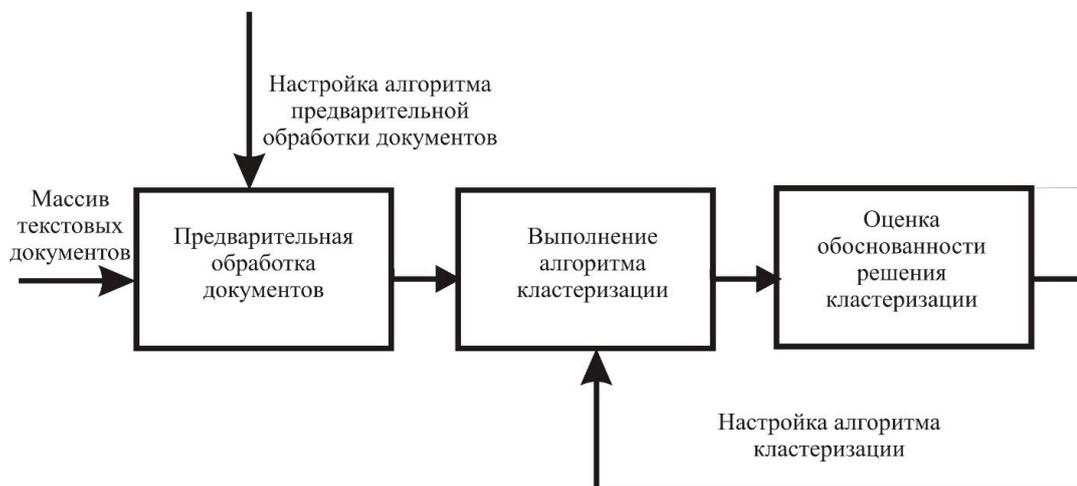


Рисунок 5.9 – Схема процесса кластеризации текстовых документов

Существующие алгоритмы автоматической классификации текстовых документов можно разделить на следующие две группы [151]:

1. Классификация полнотекстовых документов с обучением, или категоризация документов.
2. Классификация полнотекстовых документов без обучения, или кластеризация документов: документы классифицируются в условиях отсутствия predetermined классификационной схемы и множества документов-образцов.

Одним из способов решения такой задачи является определение тематической близости документов, которая определяется на основе нормированных последовательностей наиболее весомых ключевых слов, входящих в каждый документ.

Несмотря на большое число научных работ в области кластеризации, все еще существует проблема кластеризации политематических текстовых

документов, вызванная потребностью в алгоритмах кластеризации, работающих в условиях нечетких (существенно пересекающихся или перекрывающихся) кластеров, а также таких алгоритмах, которые могут работать в on-line режиме. [152] Соответственно, существует потребность в разработке таких алгоритмов и решении задач адаптивной нечеткой кластеризации.

Под каталогизацией документов принято понимать всю совокупность рабочих процессов, которые в конечном итоге приводят к созданию и полноценному функционированию более или менее сложной системы каталогов, удобных для хранения ваших документов. Чаще всего в средних и крупных компаниях создаётся система каталогов, при этом каждая фирма вправе создавать для собственного удобства дополнительные виды каталогов.

В данном случае решается задача кластеризации сопроводительной документации программного обеспечения, которое разрабатывается в IT-компаниях.

Существует четыре основных типа документации на программное обеспечение:

- архитектурная/проектная — обзор программного обеспечения, включающий описание рабочей среды и принципов, которые должны быть использованы при создании ПО;
- техническая — документация на код, алгоритмы, интерфейсы, API;
- пользовательская — руководства для конечных пользователей, администраторов системы и другого персонала;
- маркетинговая.

Проектная документация, как правило, программный продукт описывает в общих чертах. Например, программист в проекте может обосновать, почему структуры данных именно таким (а не иным) образом

организованы. Почему именно таким образом сконструирован тот или иной класс. В проекте выделяются паттерны. Часто даются указания, как выполнять модернизацию программы.

В технической документации не только указываются конкретные коды. Она, как правило, также описывает различные аспекты того, что этот код делает. Она имеет явно выраженный технический характер и используется в основном для описания и определения API, алгоритмов и структур данных. При её составлении возможно использование генераторов документации (Doxygen, NDoc, javadoc и др.), что даёт возможность постоянно поддерживать такую документацию в актуальном состоянии. В последнем случае техническая документация является частью исходного кода.

Пользовательская документация состоит из:

- вводного руководства, где рассматриваются общие вопросы выполнения типичных задач;
- тематического, где каждая глава посвящена разъяснению какого-либо раздела эксплуатации программы;
- алфавитного справочника для опытных пользователей, хорошо знающих, что нужно искать.

Маркетинговая документация используется для рекламы как самого программного продукта и его составляющих, так и других программных продуктов компании. Она часто информирует покупателя о свойствах продукта, объясняет его преимущества перед конкурирующими решениями. Часто бывает так, что именно коробка продукта и другая маркетинговая информация дают самое чёткое и простое представление о способах использования и возможностях программы.

Таким образом для решения задачи ускорения информационного поиска в базе данных с документацией будем использовать EM нечеткий вероятностный метод кластеризации с предварительной обработкой данных

на основе усовершенствованной искусственной нейронной сети для анализа главных компонент.

Каждому текстовому документу будет соответствовать вектор признаков, который будет получен на основе метода TMG (Text to Matrix Generator) [153].

Метод TMG предназначен для выполнения предварительной обработки и фильтрации, которые, как правило, выполняются в контексте применения информационно-поисковых систем [154].

В частности, TMG может использоваться для следующих типовых этапов информационно-поисковых систем:

- генерация tdm (term-document matrix) матрицы, которая соответствует входным политематическим текстам;
- создание векторов на основе пользовательских запросов для поиска;
- обновлять существующую tdm матрицу путем включения новых текстов;
- редуцировать существующую tdm матрицу путем удаления конкретных документов.

Обработка документов (политематических текстов) на основе TMG метода состоит из следующих шагов:

1. Лексический анализ;
2. Устранение стоп-слов, т.е. удаление очень частых слов, таких как артикли, предлоги, союзы и т.д., которые несут мало информации о содержании обрабатываемых документов;
3. Объединение и извлечение одних и тех же слов с разными окончаниями;
4. Индексирование терминов, т.е. выбор подмножества слов, встречающихся в документах, чтобы сформировать индекс документа;
5. Построение индекса.

После применения метода TMG мы получаем матрицу «объект-свойство», в которой содержится все текстовые документы и уникальный вектор признаков для этого документа.

Поскольку набор признаков может превышать количество документов, мы можем применить предложенную усовершенствованную искусственную нейронную сеть для анализа главных компонент с целью их сокращения. И в дальнейшем уже проводить анализ и кластеризацию документов на основе выделенных главных компонент.

Таким образом, выборка данных состоит из 1740 документов, которые относятся к 4 классам документации программного обеспечения. Анализ и кластеризации подвергались первые 10 и 25 главных компонент, которые составляли 89% и 98% информативности соответственно.

Результаты кластеризации данных и сравнение с известными подходами приведено в табл. 5.6, а на рис. 5.10 приведены результаты визуализации кластеризации текстовой документации к программным продуктам.

Таким образом, если провести анализ полученных кластеров, то можно увидеть, что техническая и проектная документация имеют наибольшее перекрытие среди рассматриваемых кластеров из-за тесной взаимосвязи самих документов.

Небольшое перекрытие имеет проектная и пользовательская документация, поскольку некоторые моменты из проектной документации могут освещаться в пользовательской документации, и маркетинговая документация, которая рекламирует производимое программное обеспечение близко к пользовательской, но перекрытия имеются только в тех документах, где маркетинговая документация для рекламы использовала описание некоторых пользовательских функций, как особенностей и преимуществ программного обеспечения.

Таблица 5.10 - Результаты сравнения точности кластеризации на основе предложенного модифицированного EM нечеткого вероятностного метода кластеризации

Исследуемые методы	10 главных компонент			25 главных компонент		
	avg	max	min	avg	max	min
Стандартная карта Т. Кохонена (SOM)	81%	89%	78%	87%	91%	81%
Метод нечётких С-средних (FCM $\beta = 2$)	89%	92%	82%	93%	94%	87%
EM вероятностный алгоритм кластеризации	89%	91%	80%	92%	95%	85%
Модифицированный EM нечеткий вероятностный метод кластеризации	94%	97%	90%	96%	98%	90%

Внедренный метод кластеризации и анализа документации программного обеспечения привел к уменьшению временных затрат на 30% на поиск того или иного типа документации, а также была решена задача автоматической каталогизации документации программного обеспечения.

Если провести более глубокий анализ полученных кластеров, то можно заметить, что внутри каждого кластера, есть небольшие подкластеры, которые характеризуют тот или иной проект по разработке программного обеспечения.

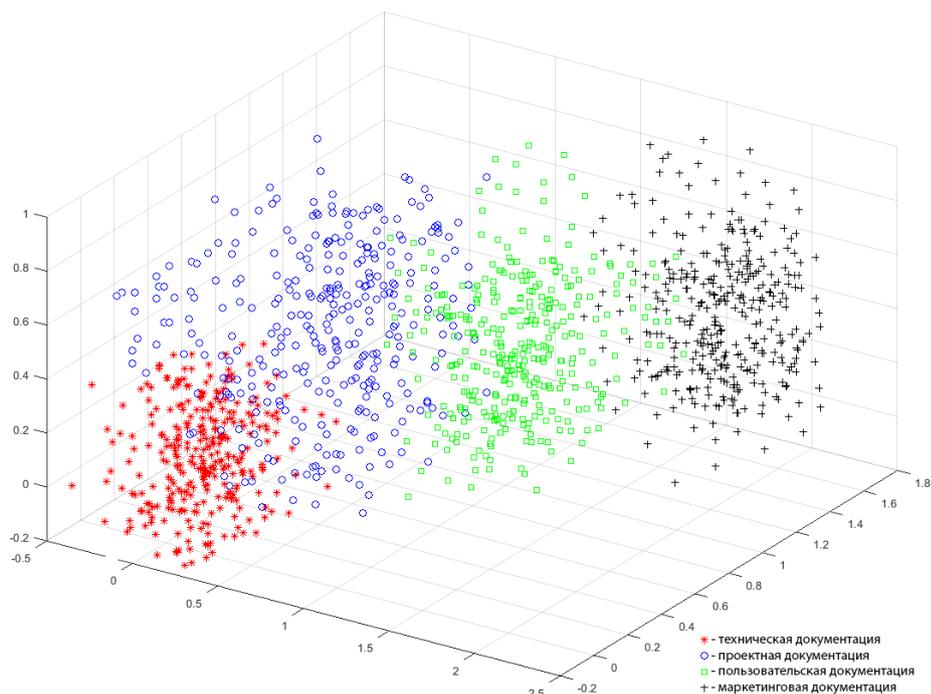


Рисунок 5.8 – Результаты кластеризации текстовых сопроводительных документов программного обеспечения на основе усовершенствованной искусственной нейронной сети для анализа главных компонент и EM нечеткого вероятностного метода кластеризации (по 25-и главным компонентам)

Выводы по разделу 5

1. Была решена задача кластеризации на основе тестовых выборок из UCI-репозитория с помощью разработанной ядерной кластеризующей нейронной сети, основанной на обобщенной регрессионной нейронной сети.
2. Проведено имитационное моделирование решения задачи нечеткой кластеризации тестовых данных, на основе двух выборок, с помощью предложенной многослойной гибридной нейро-фаззи системы и процедур ее самообучения.

3. Проведено имитационное моделирование кластеризации текстовых сопроводительных документов программного обеспечения на основе усовершенствованной искусственной нейронной сети для анализа главных компонент и EM нечеткого вероятностного метода кластеризации

4. Внедренный метод кластеризации позволил сократить на 30% время на поиск того или иного типа документации, а также была решена задача каталогизации документации программного обеспечения в ООО «Академия СМАРТ».

ВЫВОДЫ

В диссертационной работе представлены результаты, являющиеся в соответствии с поставленной целью решением актуальной задачи обработки многомерных массивов данных в условиях неопределенности с помощью нечетких методов ядерной кластеризации на основе ядерных функций. Проведенные исследования позволили сделать следующие выводы.

1. Разработано модификации метода EM (ожидание-максимизация) кластеризации данных, которые последовательно поступают на обработку одно за одним в on-line режиме.

2. Разработана искусственная нейронная сеть, которая объединяет в себе идеи ядерных систем и самообучения, и построена на основе радиально-базисной нейронной сети и самоорганизующейся карты.

3. Разработана архитектура гибридной нейронной сети на основе обобщенной регрессионной сети и самоорганизующейся карты Т. Кохонена.

4. Разработана многослойная нейро-фаззи система, являющаяся гибридом системы Ванга-Менделя и нечеткой кластеризующей самоорганизующейся сети;

5. Проведено имитационное моделирование разработанных методов и моделей, решен с их помощью ряд практических задач нечеткой кластеризации данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бодянский Е.В. Ядерная самоорганизующаяся карта на основе радиально-базисной нейронной сети / Е.В. Бодянский, А.А. Дейнеко, Я.В. Куценко (Я.В. Хаустова) // Электротехнические и компьютерные системы. – Одесса. – 2015. – 20 (96). – С. 97-105. (Входит до міжнародної наукометричної бази BASE, Index Copernicus).
2. Бодянский Е.В. Ядерная кластеризация на основе обобщенной регрессионной нейронной сети и самоорганизующейся карты Т.Кохонена / Е.В. Бодянский, А.А. Дейнеко, Я.В. Куценко (Я.В. Хаустова) // Інформаційно-керуючі системи на залізничному транспорті. – 2016. – 3 (118). – С. 15-22.
3. Бодянский Е.В. Нечеткая кластеризация потоков данных с помощью EM-алгоритма на основе самообучения по Т. Кохонену / Е.В. Бодянский, А.А. Дейнеко, А.А. Заика, Я.В. Куценко (Я.В. Хаустова) // Прикладная радиоэлектроника. – 2016. – Том 15. – № 1. – С. 80-83.
4. Бодянский Є.В. Послідовне нечітке кластерування на основі нейро-фаззі підходу / Є.В. Бодянский, А.О. Дейнеко, Я.В. Куценко (Я.В. Хаустова) // Радіоелектроніка, інформатика, управління. – Запоріжжя. -2016. – № 3(38) – С. 30-39. (Входит до міжнародних наукометричних баз Web of Science, Index Copernicus, BASE).
5. Deineko A. Neural Network for Kernel Principal Component Analysis / A. Deineko, I. Perova, O. Turuta, Y. Kutsenko (Y. Khaustova), M. Shalamov // International Journal of Computer Science and Mobile Computing. – 2015. – Vol.4 Issue 9. – P. 356-363. (Входит до міжнародних наукометричних баз Index Copernicus, BASE, INSPEC, World Cat, ESJT).
6. Bodyanskiy Ye. Data streams fast EM-fuzzy clustering based on Kohonen`s self-learning / Ye. Bodyanskiy, A. Deineko, Y. Kutsenko (Y. Khaustova), O. Zayika // The 1th IEEE International Conference on Data Stream

Mining & Processing (DSMP 2016): proc. of int. conf., Lviv, August 23-27, 2016. – Lviv. – 2016. – P. 309-313. (Входить до міжнародних наукометричних баз Scopus, Web of Science).

7. Дейнеко А.А. Гибридный EM-алгоритм вероятностной кластеризации потоков данных / А.А. Дейнеко, А.А. Заика, Я.В. Куценко (Я.В. Хаустова), И.П. Плисс // Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта: сб. трудов междунар. научн. конф., Железный порт, 24-28 мая, 2016 г. – Херсон: ХНТУ. – 2016. – С. 274-276.

8. Дейнеко А.О. Нечітке кластерування потоків даних на основі гібридних систем обчислювального інтелекту / А.О. Дейнеко, А.І. Долотов, Я.В. Куценко (Я.В. Хаустова), І.П. Плісс, Д.Р. Чигрин. // Теорія прийняття рішень: праці VIII міжнар. школи-семінару, Ужгород, 26 вересня - 1 жовтня, 2016 р. – Ужгород: УжНУ. – 2016. – С. 108-109.

9. Bodyanskiy Ye. Kernel clustering based on the hybrid neural network in Data Stream Mining tasks / Ye. Bodyanskiy, A. Deineko, Y. Kutsenko (Y. Khaustova) // Intellectual systems for decision making and problems of computational intelligence: proc. of int. conf. Zalizniy Port, May 24-28, 2016. – Kherson. – 2016. – P. 242-244.

10. Заика А.А. EM-алгоритм вероятностной кластеризации / А.А. Заика, Я.В. Куценко (Я.В. Хаустова) // 20-й Международный молодежный форум «Радиоэлектроника и молодежь в 21 веке»: матер. конф., Харьков, 22-24 апреля, 2016 г. – Харьков: ХНУРЭ. – 2016. – Том 6. – С. 19-20.

11. Deineko A. Kernel evolving neural networks for sequential principal component analysis and its adaptive learning algorithm / A. Deineko, Y. Kutsenko (Y. Khaustova), I. Pliss, M. Shalamov // Int. Scientific and Technical Conf. «Computer science and information technologies»: proc. of int. conf., Lviv, September 14-17, 2015. – Lviv: LPNU. – 2015. – P. 107-110. (Входить до міжнародної наукометричної бази Scopus).

12. Ганжа Д.Д. Ядерная самоорганизующаяся карта Т. Кохонена/ Д.Д. Ганжа, Я.В. Куценко (Я.В. Хаустова) // 20-й Международный молодежный форум «Радиоэлектроника и молодежь в 21 веке»: матер. конф., Харьков, 22-24 апреля, 2016 г. – Харьков: ХНУРЭ. – 2016. – Том 6. – С. 15-16.
13. Бодянский Є.В. Кластеризація даних на основі радіально-базисної самоорганізовної мапи / Є.В. Бодянский., А.О. Дейнеко, Я.В. Куценко (Я.В. Хаустова), М.О. Шаламов // Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту: матеріали міжнар. наук. конф., Залізний порт, 25-28 травня, 2015 г. – Херсон: ХНТУ. – 2015. – С. 257-259.
14. Borgelt, C. Prototype-based Classification and Clustering / C. Borgelt. –Magdeburg. – 2005. – 350 p.
15. Ball, G.H. A Clustering Technique for Summarizing Multivariate Data / G.H.Ball, D.J. Hall. // Behavioral Science. – 12(2). – 1967. – P.153–155.
16. Abonyi, J. Cluster Analysis for Data Mining and System Identification/ J.Abonyi, B. Feil. – Basel: Birkhauser, 2007. – p. 303.
17. Hartigan, J. A. A k-means clustering algorithm / J. A. Hartigan, M. A. Wong // Applied Statistics. – 1979. – 28. – P. 100–108.
18. Kohonen, T. Self-Organizing Maps / T. Kohonen – Berlin: Springer-Verlag. – 1995. – 362 p.
19. Zadeh, L. Fuzzy logic – a personal perspective / L. Zadeh // Fuzzy sets and Systems. – 2015. – 281. – P. 4-20.
20. Zadeh, L. Fuzzy sets / L. Zadeh // Information and Control. – 1965. – 8. – P. 338-353.
21. Tsoukalas, L. H. Fuzzy and Neural Approaches in Engineering / L.H.Tsoukalas, R.E. Uhrig. – N.Y.: John Wiley & Sons, Inc. – 1997. – 587 p.

22. Rhee, F. C.-H. Uncertain fuzzy clustering: insights and recommendations /F.C.-H. Rhee // IEEE. Comp. Intel. Mag. – 2007. – 2(1). – P.44-56.
23. Aliev, R.A. Type-2 fuzzy neural networks with fuzzy clustering and differentialevolution optimization / R. A. Aliev, W. Pedrycz, B. G. Guirimov, R. R. Aliev, U. Ilhan, M. Babagil, S. Mammadli // Information Sciences. – Elsevier. – 2011. – 181 – P.1591–1608.
24. Dunn, J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters [Text] / J. C. Dunn // Journal of Cybernetics. – 1973. – 3. – P. 32-57.
25. Bezdek, J.C. Pattern Recognition with Fuzzy Objective Function Algorithms / J.C. Bezdek. – N.Y.: Plenum Press. – 1981. – 272 p.
26. Mendel, J.M. Type-2 Fuzzy Sets and Systems: An Overview / J.M. Mendel// IEEE Comp. Intel. Mag. – 2007. – 2(1). – P.20-29.
27. Turksen, I.B. Type-II Fuzzy Possibilistic C-Mean Clustering / I.B. Turksen, M.H. Zarandi, M. Zarinbal // International Fuzzy Systems Association–Conference of the European Society for Fuzzy Logic and Technology (IFSLAEUSFLAT). – 2009. – P. 30-35.
28. Klawonn, F. Fuzzy shell cluster analysis / F. Klawonn, R. Kruse, H. Timm // In: Della Riccia G., Lenz H.J., Kruse R. (eds.), Learning, Networks and Statistics. – Wien: Springer-Verlag, 1997. – P. 105 – 120.
29. Pau, L. F. Failure diagnosis and performance monitoring / L.F. Pau // N.Y.: Marcel Dekker, 1981. – 448 p.
30. Gustafson, E.E. Fuzzy clustering with a fuzzy covariance matrix / E.E. Gustafson, W.C. Kessel // Proc. IEEE CDC. – San Diego, California, 1979. – P. 761 – 766.
31. Gath, I. Unsupervised optimal fuzzy clustering / I.Gath, A.V. Geva // IEEE Trans. on Pattern Analysis and Machine Intelligence, 1989. – Vol. 11. – P. 773 – 781.

32. Krishnapuram, R. A possibilistic approach to clustering / R. Krishnapuram, J. Keller // IEEE Trans. on Fuzzy System, 1993. – Vol. 1. – P. 98 – 110.
33. Krishnapuram, R., Fuzzy and possibilistic clustering methods for computer vision / R. Krishnapuram, J. Keller // Neural Fuzzy Systems, 1994. – Vol. 12. – P. 133 – 159.
34. Bodyanskiy, Ye. Recursive fuzzy clustering algorithms / Ye. Bodyanskiy, V. Kolodyazhniy, A. Stephan // Proc. East West Fuzzy Coll. 2002, 10-th East-West Fuzzy Coll. – Zittau/Goerlitz: HS, 2002. – P. 276 – 283.
35. Klawonn, F. Constructing a fuzzy controller from data / F.Klawonn, R. Kruse // Fuzzy Sets and Systems, 1997. – Vol. 85. – P. 117 – 193.
36. Arrow, K.J. Studies in Linear and Nonlinear Programming / K.J. Arrow, L. Hurwitz, H. Uzawa. – Stanford, CA: Stanford University Press, 1958. – 229p.
37. Chung, F.L. Fuzzy competitive learning / F.L. Chung, T. Lee // Neural Networks, 1994. – Vol. 7, No. 3. – P. 539 – 552.
38. Park, D.C. Gradient based fuzzy c-means (GBFCM) algorithm / D.C. Park, I. Dagher // Proc. IEEE Int. Conf. on Neural Networks. – Orlando, FL, USA: IEEE Press, 1994. – P. 1626 – 1631.
39. Golden, R. M. Mathematical Methods for Neural Network Analysis and Design. / R. M. Golden. – Cambridge, Massachusetts: The MIT Press, 1996. – 420 p.
40. Braun, H. Neuronale Netze. Optimierung durch Lernen und Evolution. / H. Braun – Berlin: Springer-Verlag, 1997. – 279 p.
41. Dracopoulos, D. C. Evolutionary Learning Algorithms for Neural Adaptive Control. / Evolutionary Learning Algorithms for Neural Adaptive Control. – Berlin: Springer-Verlag, 1997. – 211 p.
42. Shepherd, A. J. Second-Order Methods for Neural Networks. / A. J. Shepherd – London: Springer-Verlag, 1997. – 145 p.

43. Епанечников, В.А. Непараметрическая оценка многомерной плотности вероятности / Епанечников В.А. // Теория вероятностей и ее применение – 1968 – 14 – №1 – С. 156-161.
44. Haykin, S. Neural Networks. A Comprehensive Foundation. / S. Haykin // Upper Saddle River, N.J.: Prentice Hall, Inc. – 1999. – 842 p;
45. Бодянский, Е.В. Искусственные нейронные сети: архитектуры, обучение, применение / Е.В. Бодянский, О.Г. Руденко. – Харьков: ТЕЛТЕХ, 2004. – 372 с.
46. Goyal, S. A new scientific approach of intelligent artificial neural networkengineering for predicting shelf life of milky white dessert jeweled with pistachio / S. Goyal, G.K. Goyal // Scientific & Engineering Research, International Journal of. – 2(9). – 2011. – P.29-33.
47. Справочник по теории автоматического управления / Под. ред. А. А. Красовского. – М.: Наука, 1987. – 712 с.
48. Zahirniak, D. Pattern recognition using radial basis function network / D. Zahirniak, R. Chapman, S. Rogers, B. Suter, M. Kabritsky, V. Piaty // proc 6th Ann. Aerospace Application of Artificial Intelligence Conf. – Dayton, OH. – 1990. – P. 249-260.
49. Park, J. Universal approximation using radial-basis-function networks / J. Park, I.W. Sandberg // Neural Computation. – 1991. – 3. – P.246-257.
50. Schilling, R.J. Approximation of nonlinear systems with radial basis function neural networks / R.J. Schilling, J.J. Carrol, A.F. Al-Ajlouni // IEEE Trans. on Neural Networks. – 2001. – 12. – P.1-15.
51. Nelles, O. Nonlinear System Identification / O. Nelles. – Berlin: Springer. – 2001. – 785 p;
52. Specht, D.E. A general regression neural network / D.E. Specht // IEEE Trans. on Neural Networks. – 1991. – 2. – P.568-576.
53. Bishop, C.M. Neural Networks for Pattern Recognition / C.M. Bishop – Oxford: Clarendon Press. – 1995. – 482 p.

54. Nadaraya, E.A. About nonparametric probability density and regression estimates / E.A. Nadaraya // *Probability Theory and its Application*. – 1965. – 10. – №1. – P 199-203.

55. Friedman, J. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction* / J. Friedman, T. Hastie, R. Tibshirani – Berlin: Springer. – 2003. – 552 p.

56. Живоглядов, В.П. *Непараметрические алгоритмы адаптации* / В.П. Живоглядов, А.В. Медведев – Фрунзе: Илим. – 1974. – 214 с.

57. Медведев, А.В. *Адаптация в условиях непараметрической неопределенности* / А.В. Медведев // *Адаптивные системы и их приложения*. – Новосибирск: Наука. – 1978. – С. 4-34.

58. Раудис, Ш.Ю. *Оптимизация непараметрического алгоритма классификации* / Ш.Ю. Раудис // *Адаптивные системы и их приложения*. – Новосибирск: Наука. – 1978. – С. 57-61.

59. Медведев, А.В. *Непараметрические алгоритмы идентификации нелинейных динамических систем* / А.В. Медведев // *Стохастические системы управления*. – Новосибирск: Наука. – 1979. – С. 15-22.

60. Parzen, E. On the estimation of a probability density function and the mode / E. Parzen // *Ann. Math. Stat.* – 1962. – 38. – P.1065-1076.

61. Seng, T.L. Adaptive neuro-fuzzy control system by RBF and GRNN neural networks / T.L. Seng, M. Khalid, R. Yusof, S. Omatu // *J. of Intelligent and Robotic Systems*. – 1998. – 23. – P.267-289.

62. Guo, X.-P. A sub-stage moving window GRNN quality prediction method for injection molding process / X.-P. Guo, F.-L. Wang, M.-X. Jia // *“Lecture Notes in Computer Science”* – Berlin-Heidelberg: Springer-Verlag. – 2006. – P.1138- 1143.

63. Vapnik, V.N. *Statistical Learning Theory*. / V.N. Vapnik – Wiley Interscience. – 1998. – 736 p.

64. Розенблатт, Ф. Модель памяти на нейронных сетях / Ф. Розенблатт // Автоматика. - 1966. - 5(2). – С.2-4.
65. Kowaliw, T. Artificial Neurogenesis: An Introduction and Selective Review/ T. Kowaliw, N. Bredeche, S. Chevallier, R. Doursat // Studies in Computational Intelligence. – 2014. – 557. – P.1-60.
66. Dvoretzky, A. On stochastic approximation / A. Dvoretzky // Proc. 3-rd Berkley Symp. Math. Statistics and Probability. – 1956. – 1. – P. 39-55.
67. Bodyanskiy, Ye. Generalized regression neuro-fuzzy network / Ye. Bodyanskiy, N. Teslenko // Information Research and Applications i.TECH-2007: Proc. Fifth Int. Conf.-Varna, 2007. – V.1. – P.219-225.
68. Aggarwal, C.C. and Reddy, C.K., Data Clustering. Algorithms and Application, Boca Raton: CRC Press, 2014, 648 p.
69. Xu, R. Clustering / R. Xu, D.C. Wunsch // IEEE Press Series on Computational Intelligence. Hoboken, NJ: John Wiley & Sons, Inc., 2009, 370 p.
70. Vuorimaa, P. Fuzzy self-organizing maps / P. Vuorimaa // Fuzzy Sets and Systems. – 1994. – 66. – P.223–231.
71. Cover, T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition // IEEE Trans. on Electronic Computers. – 1965. – 14. – P. 326-334.
72. Gan, G. Data Clustering: Theory, Algorithms and Applications / G. Gan, Ch. Ma, J. Wu – Philadelphia, PA: SIAM, 2007, 466 p.
73. Rutkowski, L., Computational Intelligence. Methods and Techniques, Berlin-Heidelberg: Springer-Verlag, 2008, 514 p.
74. Mumford, C. Computational Intelligence. Collaboration, Fuzzy and Emergence / C.Mumford, L. Jain – Berlin: Springer-Verlag, 2009, 726 p.
75. Computational Intelligence. A Methodological Introduction / R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher, P. Held, P – Berlin: Springer, 2013, 488 p.

76. Du, K.-L. *Neural Networks and Statistical Learning* / K.-L. Du, M.N.S. Swamy – London: Springer-Verlag, 2014, 824 p.
77. Aggarwal, C.C., *Data Mining*, Cham: Springer, Int. Publ., Switzerland. – 2015, 734 p.
78. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing* / J.C. Bezdek, J. Keller, R. Krishnapuram, N. Paul // *The Handbooks of Fuzzy Sets*, Kluwer, Dordrecht, Netherlands: Springer. – 1999. – Vol. 4 – 776 p.
79. Dempster, A.P. Maximum likelihood from incomplete data via the EM algorithm / A.P. Dempster, N.M. Laird, D.B. Rubin // *J. of the Royal Statistical Society*. – 1977, Ser.B. – Vol. 39, no.1, P. 1-38.
80. Hathaway, R., Another interpretation of the EM algorithm for mixture distributions. *J. of Statistics & Probability Letters*. – 1986. – Vol. 4. – P. 53-56.
81. Meng, X.L. Maximum likelihood estimation via the ECM algorithm: a general framework / X.L. Meng, D.B. Rubin. *Biometrika*. – 1993. – Vol. 80. – P. 267-278.
82. Liu, C. The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence / C. Liu, D.B. Rubin. *Biometrika*. 1994. – Vol. 81. – P. 633-648.
83. Fessler, J.A. Space – alternating generalized EM algorithm / J.A. Fessler, A.O. Hero // *IEEE Trans. on Signal Processing*. – 1994. – Vol. 42. – P. 2664-2677.
84. Friedman, J. *The Elements of Statistical Learning. Data Mining, Inference and Prediction* / J. Friedman, T. Hastie, R. Tibshirani. Berlin: Springer. – 2003. – 552 p.
85. Tsytkin, Ya.Z., *Foundations of learning systems theory*, M.: Nauka, 1970.
86. Gorshkov, Ye. New recursive learning algorithms for fuzzy Kohonen clustering network / Ye. Gorshkov, V. Kolodyazhnyi, Ye. Bodyanskiy // *Proc. 17th*

Int. Workshop on Nonlinear Dynamics of Electronic Systems, Rapperswil, 21-24 June, - Switzerland. 2009, - P. 58-61.

87. Bodyanskiy, Ye. Adaptive neuro-fuzzy Kohonen network with variable fuzzifier / Ye. Bodyanskiy, B.V. Kolchygin, I.P. Pliss // International Journal «Information Theories and Applications», Sofia: ITHEA, 2011. vol.18, no3, P. 215-223.

88. Krishnapuram R. The possibilistic c-means algorithm: insights and recommendations / R. Krishnapuram, J.M. Keller // IEEE Trans. on Fuzzy Systems. – 1996. – 4(3). – P. 385–393.

89. Girolami, M. Mercer kernel based clustering in feature space // IEEE Trans. on Neural Networks. – 2002. – 13. – V. 3. – P. 789–784.

90. Kim D. W. Evaluation of the performance of clustering algorithms in kernel-based feature space / D. W. Kim, K. Lee, K.H. // Pattern Recognition. – 2002. – 35. – P. 2267–2278.

91. MacDonald, D., Clustering in data space and feature space / D. MacDonald, C. Fyfe C. // ESANN'2002 Proc. European Symp. on Artificial Neural Networks. Bruges (Belgium), 24-26 April. 2002. – P. 137-142.

92. Camastra F. A novel kernel method for clustering / F. Camastra, A. Verri // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2005. – 5. – P. 801-805.

93. Hastie, T. The Elements of Statistical Learning. Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman. – N.Y.: Springer Science & Business Media, LLC, 2009. – 758 p.

94. Kung, S.Y. Kernel Methods and Machine Learning. – Cambridge: University Press, 2014.– 603 p.

95. Zhang, D.-Q. Fuzzy clustering using kernel method / D.-Q. Zhang, S. - C. Chen // Proc. Int. Conf. on Control and Automation ICCA 2002. – P. 162 – 163.

96. Zhang D.-Q. Kernel based fuzzy and possibilistic c-means clustering / D.-Q. Zhang, S. - C. Chen // Proc. Int. Conf. Artificial Neural Networks ICANN 2003. – Turkey, 2003. – P. 122 – 125.
97. Miyamoto S. Fuzzy multiset space and c-means clustering using kernels with application to information retrieval / S. Miyamoto, K. Mizutani // Lecture Notes on Artificial Intelligence. – 2715. – Berlin Heidelberg: Springer-Verlag, 2003. – P.387 – 395.
98. Zhang, D.-Q. A novel kernelized fuzzy c-means algorithm with application in medical image segmentation / D.-Q. Zhang, S.-C. Chen // Artificial Intelligence in Medicine. 2004 – 32. – № 1. – P. 37 – 50.
99. Du, W. Robust kernel fuzzy clustering / W. Du, K. Inoue, K. Urahama // Lecture Notes on Artificial Intelligent. – 3613. – Berlin Heidelberg: Springer-Verlag, 2005. – P.454 – 461.
100. A survey of kernel and spectral methods for clustering / M. Filippone, F. Camastra, F. Masulli, S. Rovetta // Pattern Recognition. – 2008. – 41. – P. 176 – 190.
101. Havens T.S. Palaniswami M. Incremental kernel fuzzy c-means / T.S. Havens, Bezdek J.S. / Eds. K. Madani et. al. “Computational Intelligence”, SCI 399. – Berlin Heidelberg: Springer-Verlag, 2012. – P.3 – 18.
102. Vapnik, V. The Nature of Statistical Learning Theory. –N.Y.: Springer–Verlag, 1995 – 188 p.
103. Cortes, C. Support Vector Networks / C.Cortes, V. Vapnik // Machine Learning. – 1995. – 20. – P. 273 – 297.
104. Data Streams: Models and Algorithms / Ed. by C.C. Aggarwal. – N.Y.: Springer Science & Business Media, LLC, 2007 – 364 p.
105. Bifet, A. Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams. Amsterdam: IOS Press. – 2010. – 205 p.
106. Bodyanskiy Ye. Computational intelligence techniques for data analysis // Lecture Notes in Informatics. – P.-72 – Bonn: GI, 2005. – P. 15 – 36.

107. Chung F. - L. Unsupervised fuzzy competitive learning with monotonically decreasing fuzziness / F.-L. Chung, T. Lee // Proc. 1993 Int. Joint Conf. on Neural Networks, 1993. – P.2929 – 2932.

108. Klawonn, F. What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier./ Klawonn F., Höppner F. // Lecture Notes in Computer Science. – Berlin / Heidelberg: Springer, 2003. – 2811. – P. 254-264.

109. Mumford, C. Computational Intelligence. Collaboration, Fuzzy and Emergence. / C. Mumford, L. Jain // Berlin: Springer-Verlag. – 2009. – 726 p.

110. Höppner, F. Fuzzy-Clusteranalyse: Verfahren für die Bilderkennung, Klassifikation und Datenanalyse. / F. Höppner, F. Klawonn, R. Kruse // Braunschweig: Vieweg. – 1996. – 280 p.

111. Höppner, F. Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. / F. Höppner, F. Klawonn, R. Kruse, T. Runkler // Chichester: John Wiley & Sons. – 1999. – 289 p.

112. Tsao, E.C.-K. Fuzzy Kohonen clustering networks / E.C.-K. Tsao, J.C. Bezdek, N.R. Pal // Pattern Recognition. – 1994. – 27. – P.757–764.

113. Pascual – Marqui, R. D. Smoothly distributed fuzzy C-means: a new self-organizing map / R. D. Pascual – Marqui, A. D. Pascual - Montano, K. Kochi, J. M. Caroco // Pattern Recognition. – 2001. – 34. – P.2395–2402.

114. Deineko A.A. Adaptive method of combining learning-selflearning for neuro-fuzzy systems / A.A. Deineko, Sh.V. Deineko, A.P. Turuta, E.V. Bodyanskiy // System technologies. - Dnipropetrovsk, 2014. – № 2(91). – С. 145 – 153.

115. Kaufman, L. Finding Groups in Data. / L. Kaufman, P. Rousseeuw // N.Y.: John Wiley & Sons. – 1989. – 320 p.

116. Jain, A. K. Algorithms for Clustering Data. / A. K. Jain, R. C. Dubes // Englewood Cliffs, N.J.: Prentice-Hall. – 1988 – 318p.

117. Han, J. Data Mining: Concepts and Techniques. / J. Han, M. Kamber // Amsterdam: Morgan Kaufmann Publ. – 2006. – 754 p.

118. Abonyi, J. Cluster Analysis for Data Mining and System Identification. / J. Abonyi, B. Feil // Basel: Birkhauser. – 2007. – 303 p.
119. Olson, D. L. Advanced Data Mining Techniques. / D. L. Olson, D. Dursun // Berlin: Springer. – 2008. – 180 p.
120. Schölkopf, B. Learning with Kernels./ B. Schölkopf, A. Smola // Cambridge, M.A.: MIT Press. – 2002.
121. Aizerman, M. A. Theoretical foundation of potential functions method in pattern recognition learning / M. A. Aizerman, E. M. Braverman, L. I. Rozonoer // Automation and Remote Control. – 1964. – 25. – P. 821–837.
122. Вапник В.Н. Теория распознавания образов (Статистические проблемы обучения). / В.Н.Вапник, А.Я. Червоненкис // М.: Наука. – 1974. – 416 с.
123. Least Squares Support Vector Machines. / J.A.K. Suykens, T.V., Gestel, J.D. Brabanter, B.D. Moor, J. Vandewalle – Singapore: World Scientific. – 2002. – 294 p.
124. Moody, J. Fast learning in networks of locally-tuned processing units / J. Moody, C.J. Darken // Neural Computation. – 1989. – 1. – P.281-294.
125. Poggio, T. A Theory of Networks for Approximation and Learning / T. Poggio, F. Girosi // A. I. Memo № 1140, C.B.I.P. Paper № 31. – Massachusetts Institute of Technology. – 1994. – 63 p.
126. Ljung, L. System Identification: Theory for the Users. / L. Ljnnng // N.Y.: Prentice-Hall. – 1999. – 519 p.
127. Bodyanskiy, Ye. Adaptive generalized forecasting of multivariable random sequences / Ye.V. Bodyanskiy, I.P. Pliss, T.V. Solovyova // proc. AN USSR. – 1989. – A. – №9. – P. 73-75.
128. Angelov, P. Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems. / P. Angelov // Heidelberg-New York: Springer-Verlag. – 2002 – 227 p.

129. Kasabov, N. *Evolving Connectionist Systems*. / N. Kasabov // London: Springer-Verlag. – 2003 – 451 p.
130. Angelov, P. *Evolving computational intelligence systems* / P. Angelov, N. Kasabov // Proc. 1st Int. Workshop on Genetic Fuzzy Systems. – Granada, Spain. – 2005. – 76-82 p.
131. Lughofer, E. *Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications*. / E. Lughofer // Berlin-Heidelberg: Springer-Verlag. – 2011 – 456 p.
132. Bodyanskiy, Ye.V. *An Evolving Radial Basis Neural Network with Adaptive Learning of Its Parameters and Architecture* / Ye.V. Bodyanskiy, O.K. Tyshchenko, A.O. Deineko // *Automatic Control and Computer Sciences*. – 2015. – Vol. 49, No. 5. – 255–260 pp.
133. Bodyanskiy, Ye. *Evolving Neuro-Fuzzy Systems with Kernel Activation Functions. Their Adaptive Learning for Data Mining Tasks* / Ye. Bodyanskiy, O. Tyshchenko, A. Deineko // Saarbrucken: LAP LAMBERT Academic Publishing. – 2015 – 64 p.
134. Cottrel, M. *A stochastic model of retinotopy: a self-organizing process* / M.Cottrel, J. Fort // *Biological Cybernetics*. – 1986. – 53. – 405 – 411 pp.
135. Jin Y. *Computational Intelligence in Big Data* / Y. Jin, B. Hammer // *IEEE Computational Intelligence Magazine*. – 2014. – August. – P. 12-15.
136. *Information Granularity, Big Data, and Computational Intelligence* / Eds. W. Pedrycz, Shyi-Ming Chen. – Cham: Springer, 2015. – 444p.
137. Jang, J.-S. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence* / J.-S. Jang, C.-T. Sun, E. Mizutani. – Upper Saddle River: Prentice Hall. –1997. – 640 p.
138. Takagi, T. *Fuzzy identification of systems and its applications to modeling and control*. / T. Takagi, M. Sugeno // *IEEE Transactions on Systems, Man, and Cybernetics*. – 1985. – Vol.15, № 1. – P. 116-132.

139. Wang, L. X. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning / L. X. Wang, J. M. Mendel // IEEE Transactions on Neural Networks. – 1992. – Vol. 3, Issue 5. – P. 807-814.

140. Kacprzyk J. Springer Handbook of Computational Intelligence / J. Kacprzyk, W. Pedrycz. – Berlin Heidelberg: Springer – Verlag. – 2015. – 1634 p.

141. Cios K. J. Neuro-fuzzy algorithms. / Cios K. J., Pedrycz W. – Oxford: IOP Publishing Ltd and Oxford University Press. // Handbook of Neural Computation, – 1997. – D1. 3:1 – D1. 3:7.

142. Nonlinear process identification and modeling using generalized regression neuro-fuzzy network / Ye. Bodyanskiy, P. Otto, I. Pliss, N. Teslenko // Proc. 52nd Int. Sci. Coll. “Computer Science Meets Automation”. – TU Ilmenau (Thuer). – 2007. – P.27-33.

143. Bodyanskiy Ye. Generalized regression neuro-fuzzy network for identification of nonstationary plants / Ye. Bodyanskiy, N. Teslenko // Int. J. Information Technologies and Knowledge. – 2008. – 2. – №2. – P. 136-142.

144. Klawonn F. What are clusters in high dimensions and are they difficult to find?/ F. Klawonn, F. Höppner, B. Jayaram // Lecture Notes in Computer Science. – Berlin Heidelberg: Springer – Verlag, 2015. – Vol. 7627 – P. 14-33.

145. Kolchygin B. Adaptive fuzzy clustering with a variable fuzzifier / B. Kolchygin, Ye. Bodyanskiy // Cybernetics and System Analysis. – 2013. – 49. – № 3. – P. 176-181.

146. Murphy P. M. UCI Repository of machine learning databases. / P. M. Murphy, D. Aha. – URL: <http://www.ics.uci.edu/mllearn/MLRepository.html>. CA: University of California, Department of Information and Computer Science, 1994.

147. Bing Liu Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data/ Bing Liu. - Springer-Verlag Berlin Heidelberg. – 2011. – 624 p.

148. Aggarwal, C. C., Zhai, C.X. Mining Text Data / C.C. Aggarwal, C.X. Zhai. - Springer-Verlag New York. – 2012. – 524 p.
149. Sirmakessis, S. Text Mining and its Applications / S. Sirmakessis. - Springer-Verlag Berlin Heidelberg. – 2004. – 204 p.
150. Biemann, C. Text Mining / C.Biemann, A. Mehler . - Springer International Publishing. – 2014. – 238 p.
151. Cao, G. Suffix Tree Clustering on Post-retrieval Documents Information / G. Cao, D. Song, P. Bruza // The University of Queensland, 2003. – P. 234-240.
152. Zamir, O. Web Document Clustering: A Feasibility Demonstration / O. Zamir, O. Etzioni // In Proceedings of ACM/SIGIR'1998. – P. 30-40.
153. Baeza-Yates, M R. A. Modern Information Retrieval / M R. A. Baeza-Yates, B. Ribeiro-Neto // ACM Press: Addison-Wesley, 1999. – P. 367-395.
154. Sirmakessis, S. Text Mining and its Applications / S. Sirmakessis (Results of the NEMIS Launch Conference). Springer, Berlin, 2004. – P. 65-79.

ПРИЛОЖЕНИЕ А
АКТЫ О ВНЕДРЕНИИ

Затверджую
Перший проректор
Харківського національного
університету радіоелектроніки
_____ І. І. Ключник



_____ 2016 р.

А К Т

про впровадження в навчальний процес результатів дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук «Методи нечіткої кластеризації на основі ядерних функцій в задачах інтелектуального аналізу даних» аспірантки кафедри штучного інтелекту Харківського національного університету радіоелектроніки Куценко Яни Володимирівни

Комісія у складі декана факультету комп'ютерних наук, д. т. н., проф. Єрохіна А. Л., завідувача кафедри штучного інтелекту, д. т. н., проф. Філатова В. О., проф. каф. штучного інтелекту, д. т. н. Бодяньського Е. В. підтверджує, що результати дисертаційної роботи Куценко Я.В., що пов'язані із розробкою методів нечіткої кластеризації на основі ядерних функцій для інтелектуального аналізу даних, впроваджені в навчальний процес на кафедрі штучного інтелекту в курсах «Штучні нейронні мережі: архітектури, навчання та застосування» та «Нейромеревеві методи обчислювального інтелекту».

Декан факультету КН, д.т.н., проф.

_____ А. Л. Єрохін

Зав. каф. ШІ, д.т.н., проф.

_____ В. О. Філатов

Проф. каф. ШІ, д.т.н.

_____ Е. В. Бодяньський

ЗАТВЕРДЖУЮ
 Директор ТОВ «Академія СМАРТ»
 І.В. Пащенко
 «10» лютого 2016р

АКТ

про впровадження результатів дисертаційної роботи на здобуття
 наукового ступеня кандидата технічних наук
 КУЦЕНКО Яни Володимирівни

Комісія у складі:

Голова: Сокол В.Є. – директор ТОВ «Академія СМАРТ»,
 Члени комісії:
 Лисиченко В.С. – інженер-програміст,
 Пернальський А.М. – інженер-програміст.

склала даний акт про те, що у ТОВ «Академія СМАРТ» був застосований метод нечіткої кластеризації даних на основі ядерних функцій для використання в системах інтелектуального аналізу даних, що розробляються ТОВ «Академія СМАРТ». Розроблений програмний модуль, що заснований на результатах отриманих Я.В. Куценко, впроваджений на ТОВ «Академія СМАРТ» та дозволив покращити ефективність вирішення задач.

Результати впровадження довели, що розроблені Куценко Я.В. методи, які ґрунтуються на сучасних інтелектуальних технологіях, мають переваги над існуючими підходами до вирішення задач інтелектуального аналізу даних.

Комісія підтверджує доцільність розробленого програмного засобу для розв'язання задач, пов'язаних з аналізом даних.

Акт складений для пред'явлення до спеціалізованої вченої ради із захисту дисертацій і не є підставою для фінансових розрахунків.

Голова комісії:		Сокол В.Є.
Члени комісії:		Лисиченко В.С.
		Пернальський А.М.