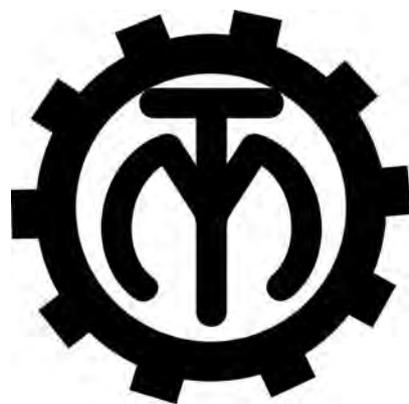


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В РОБОТОТЕХНИКЕ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
15.03.06 «Мехатроника и робототехника»
очной формы обучения*



Могилев 2019

УДК 004.8:621.01
ББК 32.813:43.4
М 54

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «30» октября 2018 г.,
протокол № 5

Составители: д-р техн. наук, проф. В. М. Пашкевич;
канд. техн. наук, доц. М. Н. Миронова

Рецензент канд. техн. наук, доц. А. П. Прудников

Методические рекомендации предназначены для выполнения лабораторных работ студентами направления подготовки 15.03.06 «Мехатроника и робототехника» очной формы обучения по дисциплине «Методы искусственного интеллекта в робототехнике».

Учебно-методическое издание

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В РОБОТОТЕХНИКЕ

Ответственный за выпуск	В. М. Шеменков
Технический редактор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 36 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий

№ 1/156 от 24.01.2014.

Пр. Мира, 43, 212000, Могилев.

© Белорусско-Российский
университет, 2019



Содержание

1 Лабораторная работа № 1. Проектирование базы знаний в форме семантической сети	4
2 Лабораторная работа № 2. Проектирование базы знаний в форме продукции.....	6
3 Лабораторная работа № 3. Проектирование базы знаний в форме фреймов.....	7
4 Лабораторная работа № 4. Процедуры поиска на графах.....	9
5 Лабораторная работа № 5. Проектирование экспертной системы.....	10
6 Лабораторная работа № 6. Моделирование нейронных сетей.....	12
7 Лабораторная работа № 7. Изучение алгоритмов поиска.....	17
8 Лабораторная работа № 8. Кластерный анализ.....	22
9 Лабораторная работа № 9. Сравнительный анализ методов нахождения решения на функциональной семантической сети.....	24
10 Лабораторная работа № 10. Управление процессом сборки с помощью конечного автомата.....	28
11 Лабораторная работа № 11. Метрическое распознавание объектов.....	33
12 Лабораторная работа № 12. Использование коэффициентов определенности.....	36
13 Лабораторная работа № 13. Обучение в интеллектуальных системах.....	40
14 Лабораторная работа № 14. Нахождение кратчайшего пути транспортным роботом.....	44
Список литературы.....	47



1 Лабораторная работа № 1. Проектирование базы знаний в форме семантической сети

Цель работы: изучение основных правил представления знаний в форме семантической сети.

Основные положения

База знаний – это компьютерная модель знаний экспертов в определенной предметной области.

Модели знаний: продукционная модель, семантическая сеть, фреймы, логическая модель.

Одним из способов представления знаний является семантическая сеть. В основе сетевых моделей представления знаний лежит идея о том, что любые знания можно представить в виде совокупности объектов (понятий) и связей (отношений) между ними.

Понятиями обычно выступают абстрактные или конкретные объекты, а отношения – это связи типа «это», «имеет частью», «принадлежит», «любит» и т. п.

Можно ввести несколько классификаций семантических сетей. Например, по количеству типов отношений:

- однородные (с единственным типом отношений);
- неоднородные (с различными типами отношений).

По типам отношений:

- бинарные (в которых отношения связывают два объекта);
- парные (в которых есть специальные отношения, связывающие более двух понятий).

В зависимости от типов связей, используемых в модели, различают классифицирующие, функциональные сети, сценарии.

В классифицирующих сетях используются отношения структуризации. Такие сети позволяют в базах знаний вводить разные иерархические отношения между информационными единицами.

Функциональные сети характеризуются наличием функциональных отношений. Их часто называют вычислительными моделями.

В сценариях используются каузальные отношения, а также отношения типов «средство – результат», «орудие – действие» и т. п.

Семантические сети делятся на интенциональные и экстенциональные.

Интенциональная сеть содержит интенциональные знания и описывает общую структуру модернизируемой предметной области на основе абстрактных объектов и отношений, т. е. обобщенных представителей некоторых классов объектов и отношений. Например, такие объекты как производственный участок, груз, деталь могут являться обобщенными понятиями множества значений, от которых образуется множество имен конкретных производственных участков (токарный, прессовый и т. п.), множество имен грузов (заготовка, кассета), множество классов деталей (болт, вал, гайка и т. п.).

Экстенсиональная семантическая сеть описывает экстенсиональные знания о модернизируемых объектах, являясь как бы «фотографией» его текущего состояния.

Интенсиональная сеть предложения «Робот сверлит отверстие в детали сверлом диаметром 10 мм» представлена на рисунке 1, экстенсиональная – на рисунке 2 [1].

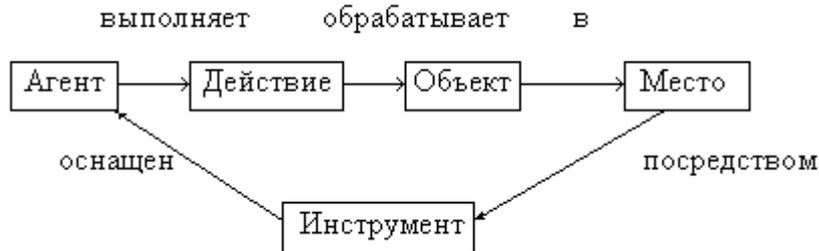


Рисунок 1 – Интенсиональная сеть

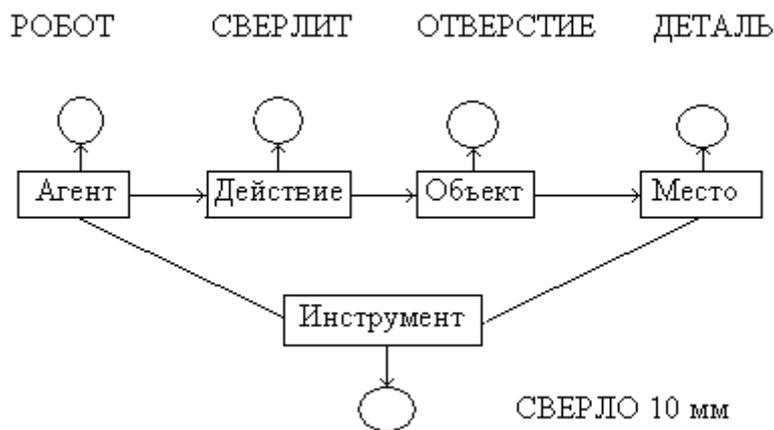


Рисунок 2 – Экстенсиональная сеть

Задание

Для предметной области сформировать базу знаний в форме семантической сети, содержащую не менее 12 отношений.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Понятия семантической сети.
- 4 Отношения семантической сети.
- 5 Семантическая сеть в виде графа.
- 6 Выводы.

Контрольные вопросы

- 1 Что такое экспертная система?
- 2 Что является информационной основой экспертной системы?
- 3 Какие модели знаний используются в базах знаний экспертных систем?
- 4 Что такое семантическая сеть?
- 5 Назовите виды семантических сетей.
- 6 Что такое функциональная семантическая сеть?

2 Лабораторная работа № 2. Проектирование базы знаний в форме продукций

Цель работы: изучение основных правил представления знаний в форме продукций.

Основные положения

Продукционная модель – модель, основанная на правилах, позволяющая представить знания в виде предложений типа «ЕСЛИ условие, ТО действие».

В качестве условия и действия в правилах может быть, например, предположение о наличии того или иного свойства, принимающее значение истина или ложь.

При использовании продукционной модели база знаний состоит из набора правил. Программа, управляющая перебором правил, называется машиной вывода. Чаще всего вывод бывает прямой (от данных к поиску цели) или обратный (от цели для ее подтверждения к данным). Прямые выводы реализуют стратегию «от фактов к заключениям». При обратных выводах выдвигаются гипотезы вероятностных заключений, которые могут быть подтверждены или опровергнуты на основании фактов, поступающих в рабочую память.

Примером продукции может служить следующее выражение:

ЕСЛИ клиент работает на одном месте более двух лет,
ТО клиент имеет постоянную работу.

Как условие, так и действие правила могут учитывать несколько выражений, объединенных логическими связками И, ИЛИ, НЕ:

ЕСЛИ клиент имеет постоянную работу,
И клиенту более 18 лет,
И клиент НЕ имеет финансовых обязательств,
ТО клиент может претендовать на получение кредита.

Помимо продукционных правил, база знаний должна включать и простые факты, поступающие в систему через интерфейс пользователя или выводимые в процессе поиска решения задачи. Факты являются простыми утверждениями

типа «клиент работает на одном месте более двух лет». И когда в процессе интерпретации правил машиной вывода какой-либо факт согласуется с частью правила ЕСЛИ, то выполняется действие, определяемое частью ТО этого правила. Новые факты, добавляемые в базу знаний в результате действий, описанных в правилах, также могут быть использованы для сопоставления с частями ЕСЛИ других правил. Последовательное сопоставление частей правил ЕСЛИ с фактами порождает *цепочку вывода*.

Задание

Для предметной области сформировать базу знаний, соответствующую следующим требованиям:

- включить не менее 12 правил, из которых не менее семи – сложные правила;
- для описания правил использовать не менее восьми переменных;
- число циклов просмотра правил для прямой цепочки рассуждений должно составлять не менее трех;
- для обратной цепочки рассуждений должны быть логически выведены не менее четырех переменных, прежде чем будет определена переменная вывода;
- пару последовательных правил.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Список переменных, их описание и принимаемые ими возможные значения.
- 4 Правила, составляющие базу знаний.
- 5 Выводы.

Контрольные вопросы

- 1 Что такое продукционная модель?
- 2 Как формируются продукционные правила?
- 3 Из каких правил может состоять база знаний?
- 4 Назовите типы продукционных систем.
- 5 Назовите достоинства и недостатки продукционных моделей.

3 Лабораторная работа № 3. Проектирование базы знаний в форме фреймов

Цель работы: изучение основных правил представления знаний в форме фреймов.

Основные положения

Фреймы – это структура данных, в которой в определенном порядке представлены сведения о свойствах объекта.



Основной структурной единицей фрейма является слот – вложенная во фрейм структура данных, который представляется в виде [4]

$$\langle \text{Имя слота} \rangle: \{(A_i, v_i)\}, \{r_i\},$$

где A_i – имя признака;
 v_i – его значение;
 r_i – связь с другими слотами.

Слоты – это некоторые незаполненные подструктуры фрейма, после заполнения которых конкретными данными, фрейм будет представлять ту или иную ситуацию, явление или объект предметной области.

При конкретизации фрейма ему и его слотам присваиваются конкретные имена и происходит заполнение слотов.

В качестве значений слотов могут выступать имена других фреймов, что обеспечивает построение сети фреймов.

В общем виде фрейм выглядит следующим образом:

⟨Имя фрейма⟩:
 [роль 1] (⟨имя слота 1⟩ : ⟨значение слота 1⟩);
 [роль 2] (⟨имя слота 2⟩ : ⟨значение слота 2⟩);

 [роль n] (⟨имя слота n⟩ : ⟨значение слота n⟩).

Имя фрейма служит для идентификации фрейма в системе и должно быть уникальным.

Имя слота должно быть уникальным в пределах фрейма.

Значение слота должно соответствовать указанному типу данных и условию наследования. Значением слота могут быть числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов.

Пример фрейма *РУКОВОДИТЕЛЬ*

<i>Имя слота</i>	<i>Значение слота</i>	<i>Тип значения слота</i>
Имя	Иванов И. И.	Строка символов
Рожден	01.01.1965	Дата
Возраст	age(data, рожден)	Процедура
Специальность	Юрист	Строка символов
Отдел	Отдел кадров	Строка символов
Зарплата	80000	Число
Адрес	ДОМ_АДРЕС	Фрейм

Основным преимуществом фреймов как модели представления знаний является способность отражать концептуальную основу организации памяти человека, а также гибкость и наглядность.



Задание

Для выбранной самостоятельно предметной области разработать фреймы в виде взаимосвязанных таблиц (таблица 1) и сложной иерархической структуры.

Таблица 1 – Описание фрейма

Имя фрейма			
Имя слота	Значение слота	Тип слота	Присоединяемая процедура

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Таблицы, содержащие описание фреймов.
- 4 Описание взаимосвязи фреймов.
- 5 Выводы.

Контрольные вопросы

- 1 Что такое фрейм?
- 2 Что является основной структурной единицей фрейма?
- 3 Назовите атрибуты фрейма.
- 4 Что может выступать в качестве значения слота?
- 5 Назовите типы фреймов.
- 6 В чем заключается преимущество использования фреймов?

4 Лабораторная работа № 4. Процедуры поиска на графах

Цель работы: изучение и исследование процедур поиска на графах.

Основные положения

При решении многих задач как для ориентированных, так и для неориентированных графов, для систематического обхода их вершин на практике применяется два принципиально различных порядка обхода, основанные на *поиске в глубину* и *поиске в ширину* соответственно.

Поиск в глубину начинается с произвольной вершины графа, например, с первой, обозначим ее v . При этом значение метки v меняется на противоположное (вершина уже посещалась). Затем для каждой вершины, смежной с v , которая ранее не посещалась, рекурсивно вновь применяется поиск в глубину. При этом все вершины, достижимые из начальной, будут пройдены. Если некоторые вершины оказались непройденными, то граф не связан. Для полного его обхода выбирается любая еще не посещенная вершина, и поиск продолжается.



Поиск в ширину. Свое название он получил из-за того, что при достижении во время обхода любой его вершины v в очередь на рассмотрение попадают сразу все еще не просмотренные вершины, связанные с вершиной v . На каждом шаге из начала очереди извлекается один элемент, а в конец добавляются связанные с ним вершины, еще не находящиеся в очереди. Поэтому элементы метятся как обработанные в момент попадания в очередь, а не извлечения из нее. Так как максимальное количество элементов в очереди равно количеству вершин в графе, организовать ее можно и с помощью одномерного массива.

Порядок выполнения работы

- 1 Базу знаний, разработанную при выполнении лабораторной работы № 2, представить в виде графа.
- 2 Разработать схемы алгоритмов для исследуемых поисков.
- 3 Написать и отладить программы, реализующие разработанные алгоритмы.
- 4 Сравнить эффективность поисков. Сделать соответствующие выводы.
- 5 Оформить отчет.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Схема алгоритма поиска в глубину.
- 4 Схема алгоритма поиска в ширину.
- 5 Сравнительный анализ методов поиска.
- 6 Выводы.

Контрольные вопросы

- 1 Назовите методы эвристического поиска.
- 2 Назовите неинформированные процедуры поиска на графе.
- 3 Как осуществляется поиск в глубину?
- 4 Как осуществляется поиск в ширину?
- 5 Назовите достоинства и недостатки исследуемых поисков.

5 Лабораторная работа № 5. Проектирование экспертной системы

Цель работы: получение навыков проектирования и реализации экспертной системы на базе оболочки Inter.

Основные положения

Экспертная система – компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации. Современные



экспертные системы начали разрабатываться исследователями искусственного интеллекта в 1970-х гг., а в 1980-х получили коммерческое подкрепление.

Структура экспертных систем [5]:

- интерфейс пользователя;
- интеллектуальный редактор базы знаний;
- рабочая (оперативная) память;
- база знаний;
- механизм вывода;
- подсистема объяснений.

Экспертная система может функционировать в двух режимах.

В режиме ввода знаний эксперт с помощью инженера по знаниям посредством редактора базы знаний вводит известные ему сведения о предметной области в базу знаний экспертной системы.

В режиме консультации пользователь ведет диалог с системой, сообщая ей сведения о текущей задаче и получая рекомендации системы.

При разработке экспертных систем выделяют шесть этапов.

Этап идентификации проблем – определяются задачи, которые подлежат решению, выявляются цели разработки, определяются эксперты и типы пользователей.

Этап извлечения знаний – проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

Этап структурирования знаний – определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, моделируется работа системы, оценивается адекватность целям системы зафиксированных понятий, методов решений, средств представления и манипулирования знаниями.

Этап формализации – осуществляется наполнение экспертом базы знаний. Процесс приобретения знаний разделяют на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном экспертной системе. Процесс приобретения знаний осуществляется инженером по знаниям на основе анализа деятельности эксперта по решению реальных задач.

Реализация экспертной системы – создаётся один или несколько прототипов экспертной системы, решающих требуемые задачи.

Этап тестирования – производится оценка выбранного способа представления знаний в системе в целом.

Порядок выполнения работы

1 Наполнить базу знаний оболочки Inter, разработанной на кафедре «Технология машиностроения» Белорусско-Российского университета. Данные взять из лабораторной работы № 2.

2 Протестировать экспертную систему.

3 Дать оценку эффективности использования экспертной системы.

4 Оформить отчет.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 База знаний экспертной системы.
- 4 Результат работы экспертной системы.
- 5 Выводы.

Контрольные вопросы

- 1 Что такое экспертная система?
- 2 Что входит в состав экспертной системы?
- 3 Назовите режимы функционирования экспертной системы.
- 4 Назовите этапы разработки экспертной системы.
- 5 Что является информационной основой экспертной системы?
- 6 Какие модели знаний используются в базах знаний экспертных систем?

6 Лабораторная работа № 6. Моделирование нейронных сетей

Цель работы: ознакомление со средствами и методами MATLAB и пакета Simulink для моделирования и исследования нейронных сетей. Применение нейронных сетей для аппроксимации функций.

Основные положения

Нейронная сеть – это последовательность нейронов, соединенных между собой синапсами. Благодаря такой структуре машина обретает способность анализировать и даже запоминать различную информацию. Нейронные сети также способны не только анализировать входящую информацию, но и воспроизводить ее из своей памяти. Другими словами, нейросеть – это машинная интерпретация мозга человека, в котором находятся миллионы нейронов, передающих информацию в виде электрических импульсов.

Самыми распространенными применениями нейронных сетей являются:

- классификация (распределение данных по параметрам);
- предсказание (возможность предсказывать следующий шаг);
- распознавание.

Распознавание – это в настоящее время самое широкое применение нейронных сетей. Используется в Google, когда нужно найти фото, или в камерах телефонов, когда оно определяет положение лица и выделяет его и многое другое.

Нейрон – это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Они делятся на три основных типа: входной, скрытый и выходной.

В том случае, когда нейросеть состоит из большого количества нейронов,



вводят термин слоя. Соответственно, есть входной слой, который получает информацию, n скрытых слоев (обычно их не больше трех), которые ее обрабатывают, и выходной слой, который выводит результат. У каждого из нейронов есть два основных параметра: входные данные (input data) и выходные данные (output data). В случае входного нейрона $input = output$. В остальных случаях в поле input попадает суммарная информация всех нейронов с предыдущего слоя, после чего она нормализуется с помощью функции активации (пока что просто представим ее $f(x)$) и попадает в поле output.

Синапс – это связь между двумя нейронами. У синапсов есть один параметр – вес. Благодаря ему входная информация изменяется, когда передается от одного нейрона к другому. Допустим, есть три нейрона, которые передают информацию следующему. Тогда в наличии три веса, соответствующие каждому из этих нейронов. У того нейрона, у которого вес будет больше, та информация и будет доминирующей в следующем нейроне [6].

Порядок выполнения работы

1 Создание нейронной сети. Создать, используя графический интерфейс пользователя, нейронную сеть для выполнения операции $y = x^2$ при задании векторов входа $x = [-1; -0,8; -0,5; -0,2; 0; 0,1; 0,3; 0,6; 0,9; 1]$ и цели $y = [1; 0,64; 0,25; 0,04; 0; 0,01; 0,09; 0,36; 0,81; 1]$.

Для открытия основного окна интерфейса в командном окне MATLAB ввести команду `>> nntool`.

Выполнение команды приведет к открытию окна создания нейронной сети Network/Data Manager.

Сформировать последовательность входов и целей в рабочей области GUI-интерфейса, используя окно *Create New Data* (рисунок 3).

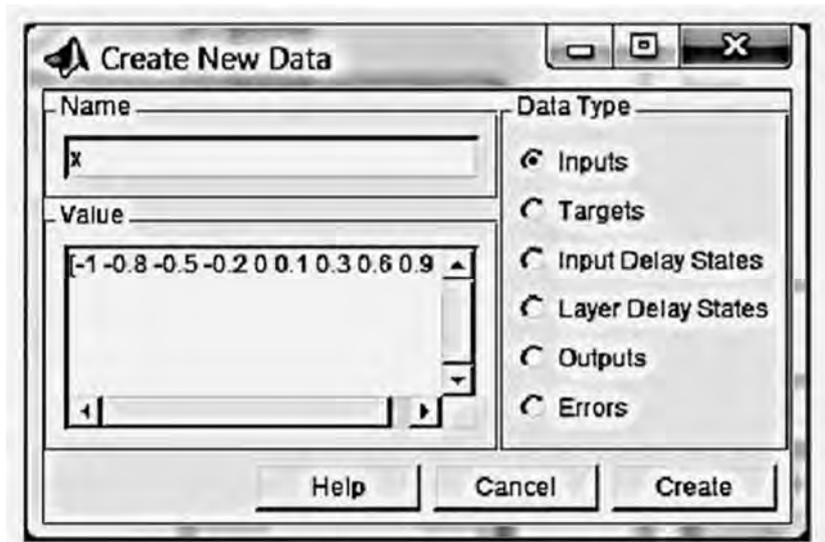


Рисунок 3 – Диалоговое окно задания векторов входа

С этой целью сначала нажать кнопку *New Data*, далее в поле *Name* окна *Create New Data* ввести имя переменной x , затем в области значений *Value*



вектор значений $[-1; -0,8; -0,5; -0,2; 0; 0,1; 0,3; 0,6; 0,9; 1]$ и, используя кнопку *Inputs* (в правой части окна), указать тип переменных (*Inputs* – Входы).

Ввод завершить нажатием кнопки *Create* (Создать).

Аналогичную операцию проделать для вектора y с указанием (с помощью кнопки *Targets*), что это вектор целевых данных.

Для создания новой нейронной сети в окне *Network/Data Manager* нажать кнопку *New Network*. В открывшемся окне *Create New Network* (рисунок 4) выбрать нейронную сеть типа *feed-forward backprop* с прямой передачей сигнала и с обратным распространением ошибки. При создании сети сохранить ей имя, даваемое по умолчанию (*network1*). Диапазон входов определить (в окне *Create New Network*) с помощью опции *Get from input*.

Количество нейронов (*Number of neurons*) первого слоя (*Layer 1*) установить равным двум. Остальные установки при создании сети оставить по умолчанию. Создание сети завершить нажатием кнопки *Create*.

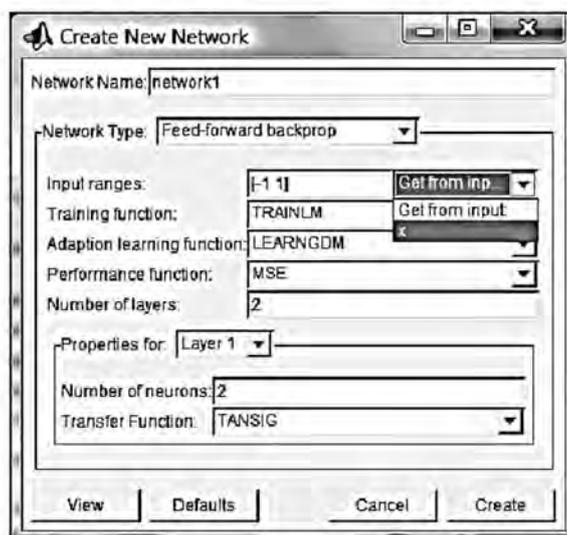


Рисунок 4 – Диалоговое окно создания нейронной сети

После этого в окне *Network/Data Manager*, в области *Networks* появится имя новой созданной сети – *network1*. Выбрать это имя с помощью мышки, что приведет к активизации всех кнопок указанного окна.

Выполнить инициализацию сети, для чего с помощью кнопки *Initialize...* выбрать закладку *Initialize*. Это приведет к открытию диалоговой панели *Network: network1*.

Для ввода в панели *Network: network1* установленных диапазонов и инициализации весов необходимо воспользоваться кнопками *Set Ranges* (Установить диапазоны) и *Initialize Weights* (Инициализировать веса). Если требуется вернуться к прежним диапазонам, то следует выбрать кнопки *Revert Ranges* (Вернуть диапазоны) и *Revert Weights* (Вернуть веса), но в условиях примера это не нужно.

2 Моделирование нейронных сетей при помощи Simulink. Пакет *Neural Network Toolbox* содержит ряд блоков, которые могут быть либо непосредственно использованы для построения нейронных сетей в среде Simulink, либо

применяться вместе с функцией *gensim*.

Для вызова этого набора блоков в командной строке необходимо набрать команду *neural*, после выполнения которой появляется окно, представленное на рисунке 5. Этому же результату можно добиться с помощью кнопки *Simulink* меню MATLAB и далее – кнопки *Neural Network Blockset*.

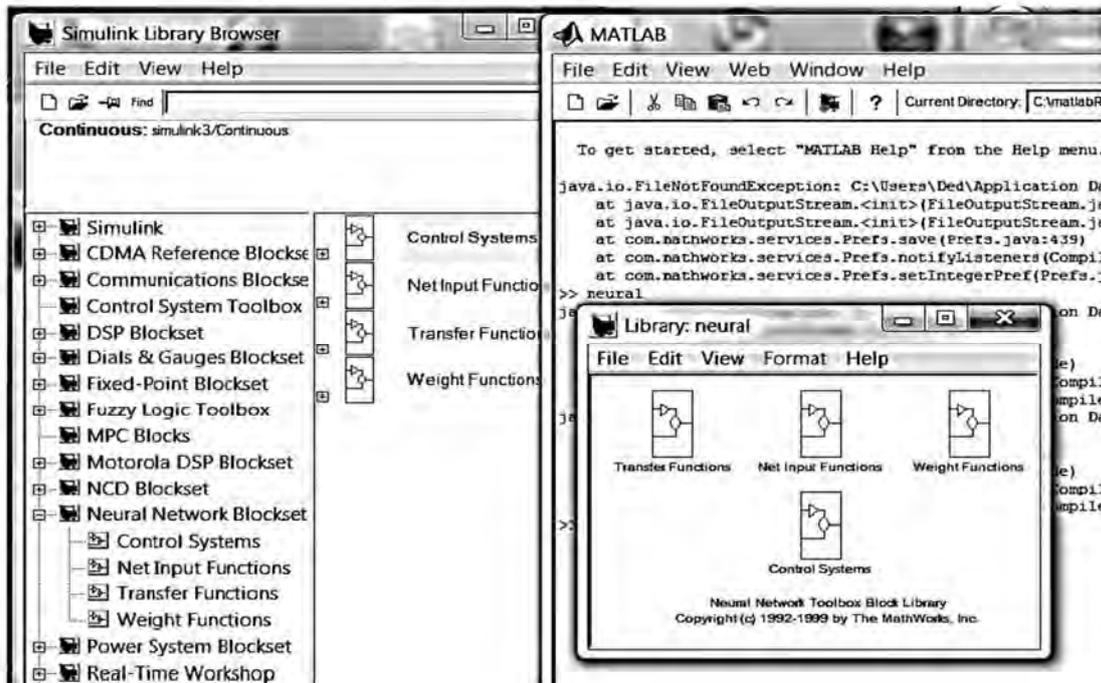


Рисунок 5 – Библиотека нейросетевых блоков

Каждый из нейросетевых блоков, в свою очередь, является набором (библиотекой) некоторых блоков. Рассмотрим их.

Двойной щелчок на блоке *Transfer Functions* приводит к появлению библиотеки блоков функций активации. Каждый из блоков данной библиотеки преобразует подаваемый на него вектор в соответствующий вектор той же размерности.

Блок *Net Input Functions* включает блоки, реализующие функции преобразования входов сети.

Блок *Weight Functions* содержит библиотеку блоков, реализующих некоторые функции весов и смещений.

В процессе работы со всеми приведенными блоками при задании конкретных числовых значений ввиду особенностей Simulink векторы необходимо представлять как столбцы, а не как строки.

Блоки, объединенные в библиотеку *Control Systems*, реализуют нейросетевые регуляторы трех различных структур – регулятор с предсказанием, регулятор, основанный на использовании модели нелинейной авторегрессии со скользящим средним (*Nonlinear Autoregressive-Moving Average* – NAR-MA-L2), и регулятор на основе эталонной модели, которые удобны при построении и исследовании моделей систем автоматического управления, а также блок просмотра результатов.

Основной функцией для формирования нейросетевых моделей в Simulink является функция *gensim*, записываемая в форме

$$gensim(net, st), \quad (1)$$

где *net* – имя созданной НС;

st – интервал дискретизации (если НС не имеет задержек, ассоциированных с ее входами или слоями, значение данного аргумента устанавливается равным -1).

В качестве примера использования средств Simulink рассмотрим следующую задачу.

Пусть входной и целевой векторы имеют следующий вид:

$$p = [1 \ 2 \ 3 \ 4 \ 5]; \quad t = [1 \ 3 \ 5 \ 7 \ 9].$$

Создать линейную нейронную сеть и протестировать ее по данным обучающей выборки:

```
>> p = [1 2 3 4 5];
>> t = [1 3 5 7 9];
>> net = newlind(p, t);
>> Y = sim(net, p);
Y = 1.0000 3.0000 5.0000 7.0000 9.0000.
```

Запустить Simulink командой

```
>> gensim(net, -1).
```

Это приведет к открытию окна с нейросетевой моделью.

Для проведения тестирования модели щелкнуть дважды на левом значке (с надписью *Input 1* – Вход 1) (рисунок 6), что приведет к открытию диалогового окна параметров блока. В данном случае блок *Input 1* является стандартным блоком задания константы (*Constant*). Изменить значение по умолчанию на «2» и нажать кнопку ОК.

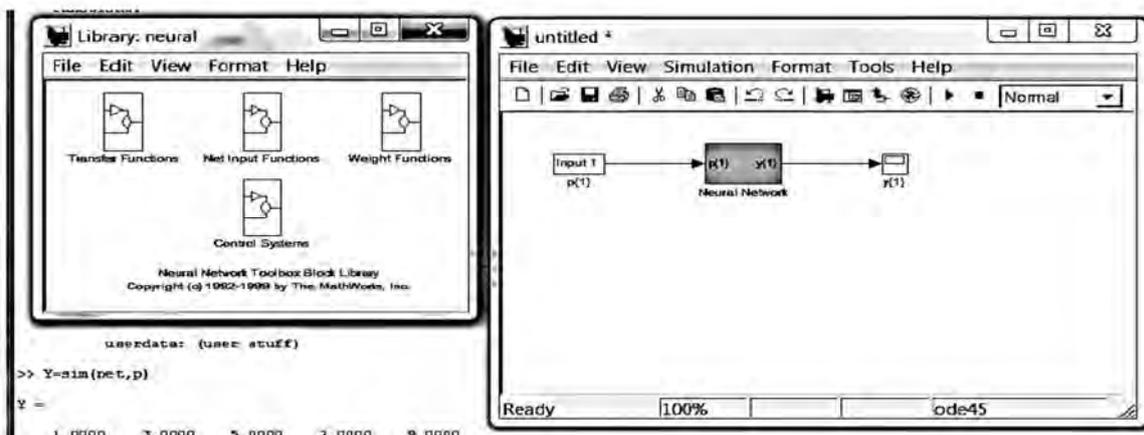


Рисунок 6 – Библиотека блока *neural*



Затем нажать кнопку *Start* в панели инструментов окна модели. Расчет нового значения сетью производится почти мгновенно.

Для его вывода необходимо дважды щелкнуть на правом значке (на блоке $y(1)$).

Результат вычислений равен 3 и выводится в виде графика. Для вывода результата в числовом виде к выходу модели следует подключить блок «дисплей».

Дважды щелкая на блоке *Neural Network*, а затем на блоке *Layer 1*, можно получить детальную графическую информацию о структуре сети.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Диалоговые окна с нейросетевыми моделями.
- 4 Результаты моделирования нейронных сетей.
- 5 Графическая информация о структуре сетей.
- 6 Выводы.

Контрольные вопросы

- 1 Что такое нейронная сеть?
- 2 Какие бывают нейронные сети?
- 3 Для чего нужны нейронные сети?
- 4 Что такое нейрон?
- 5 Что такое синапс?
- 6 Как работает нейронная сеть?

7 Лабораторная работа № 7. Изучение алгоритмов поиска

Цель работы: изучение алгоритмов поиска оптимальных решений.

Основные положения

Методы поиска оптимальных решений разделяются на два различных класса – детерминированные и случайные [8].

Основными детерминированными алгоритмами являются метод градиента и метод покоординатного спуска [9].

Метод градиента требует нахождения частных производных оптимизируемой функции по ее аргументам, определение которых аналитическим методом крайне затруднительно, к тому же нецелесообразно в силу чрезвычайной сложности функции $\Delta_{\Sigma}(\pi_1, \pi_2, \dots, \pi_n)$ и значительного количества ее аргументов.



Метод покоординатного спуска заключается в применении одномерной стратегии поиска по выделенной координате при фиксированных значениях остальных координат [9]. Геометрический смысл такого метода состоит в поочередном движении в направлениях, параллельных координатным осям (рисунок 7).

В данном алгоритме поиск параметров $\pi_1^*, \pi_2^*, \dots, \pi_n^*$, соответствующих экстремуму функции $\Delta_\Sigma(\pi_1, \pi_2, \dots, \pi_n)$ при условиях $\pi_{i\max} \leq \pi_i \leq \pi_{i\min}$, начинается из исходной точки $Z_{1\text{исх}}(\pi_{1\min}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}})$, в которой определяется значение целевой функции.

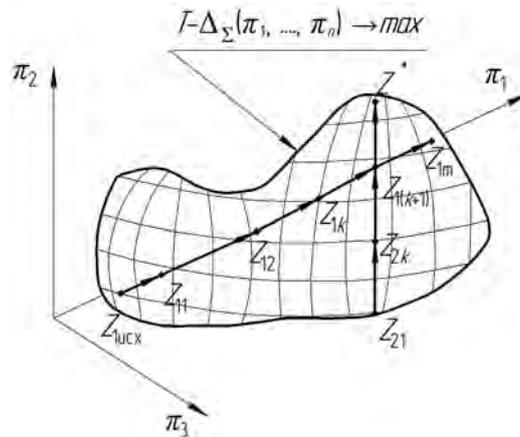


Рисунок 7 – Иллюстрация метода покоординатного спуска

Далее значения параметров фиксируются, кроме одного. При этом целевая функция превращается в функцию одной переменной. Изменяя, например, фактор π_1 , осуществляют переход от начальной $Z_{1k}(\pi_{1k}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}})$ к новой допустимой точке $Z_{1(k+1)}(\pi_{1(k+1)}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}})$, и для нее оценивается значение функции $\Delta_\Sigma(\pi_{1(k+1)}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}})$, которое сравнивается со значением $\Delta_\Sigma(\pi_{1k}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}})$, найденным предварительно в $Z_{1k}(\pi_{1k}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}})$.

Переход к новой точке факторного пространства осуществляется в соответствии с формулой

$$Z_{1(k+1)}(\pi_{1(k+1)}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}}) = Z_{1k}(\pi_{1k} + a, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}}), \quad (2)$$

где a – величина шага.

Далее нахождение целевой функции ведется поочередно по направлениям, параллельным осям координат.

В соответствии с описанным алгоритмом при последовательном изменении координат перемещаются к точкам факторного пространства Z_1, Z_2, \dots, Z_n до тех пор, пока не будет найдена точка глобального экстремума $Z^*(\pi_1^*, \pi_2^*, \dots, \pi_n^*)$.

Данный метод хорошо работает только в условиях, когда координаты мало влияют друг на друга, а также он требует большого объема вычислений, что

приводит к длительному процессу нахождения оптимальных параметров.

Для нахождения глобальных экстремумов сложных многоэкстремальных целевых функций могут использоваться методы случайного поиска, характеризующиеся небольшой продолжительностью нахождения оптимальных решений.

Случайный поиск с возвратом подразумевает соответствующее перемещение в факторном пространстве, выбираемое на каждом шаге поиска решения произвольным образом, и не требует процедуры нахождения упомянутых частных производных.

В качестве алгоритма поиска значений параметров на функциональной семантической сети можно использовать алгоритм случайного поиска с возвратом (рисунок 8) [10].

В данном алгоритме поиск значений факторов $\pi_1^*, \pi_2^*, \dots, \pi_n^*$, минимизирующих функцию $\Delta_{\Sigma}(\pi_1, \pi_2, \dots, \pi_n)$ при условиях $\pi_{i_{\max}} \leq \pi_i \leq \pi_{i_{\min}}$, начинается из исходной точки $Z_{\text{исх}}(\pi_{1\text{исх}}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}})$, выбранной случайным образом, в которой определяется значение функции $\Delta_{\Sigma_{\text{исх}}}(\pi_{1\text{исх}}, \pi_{2\text{исх}}, \dots, \pi_{n\text{исх}})$.

Далее выполняется переход от исходной к новой допустимой точке $X_{k+1}(\pi_{1(k+1)}, \pi_{2(k+1)}, \dots, \pi_{n(k+1)})$, и для нее оценивается значение функции $\Delta_{\Sigma(k+1)}(\pi_{1(k+1)}, \pi_{2(k+1)}, \dots, \pi_{n(k+1)})$, которое сравнивается со значением, найденным предварительно на предыдущем этапе.

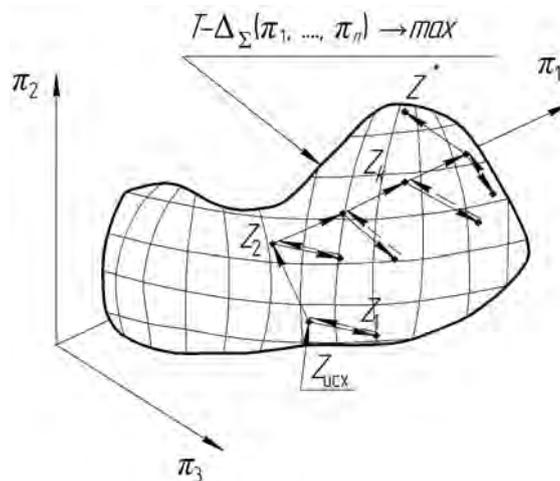


Рисунок 8 – Иллюстрация случайного поиска с возвратом

Переход к новой точке факторного пространства осуществляется в соответствии с формулой

$$Z_{k+1}(\pi_{1(k+1)}, \dots, \pi_{i(k+1)}, \dots, \pi_{n(k+1)}) = Z_k(\pi_{1k} \pm a_{1k} r_{1k}, \dots, \pi_{ik} \pm a_{ik} r_{ik}, \dots, \pi_{nk} \pm a_{nk} r_{nk}), \quad (3)$$

где a_{ik} – величина k -го шага для i -й переменной, определяемая случайным образом;

r_{ik} – единичный вектор, в направлении которого производится этот шаг.

Если оказывается $\Delta_{\Sigma(k+1)}(\pi_{1(k+1)}, \pi_{2(k+1)}, \dots, \pi_{n(k+1)}) < \Delta_{\Sigma k}(\pi_{1k}, \pi_{2k}, \dots, \pi_{nk})$, то совершается переход из $Z_k(\pi_{1k}, \pi_{2k}, \dots, \pi_{nk})$ в $Z_{k+1}(\pi_{1(k+1)}, \pi_{2(k+1)}, \dots, \pi_{n(k+1)})$, после чего $Z_{k+1}(\pi_{1(k+1)}, \pi_{2(k+1)}, \dots, \pi_{n(k+1)})$ становится новой исходной точкой для продолжения поиска глобального экстремума $Z^*(\pi_1^*, \pi_2^*, \dots, \pi_n^*)$.

Если же $\Delta_{\Sigma(k+1)}(\pi_{1(k+1)}, \pi_{2(k+1)}, \dots, \pi_{n(k+1)}) > \Delta_{\Sigma k}(\pi_{1k}, \pi_{2k}, \dots, \pi_{nk})$, то осуществляется возврат в исходную точку, т. к. полученное решение хуже исходного.

Таким образом, сущность метода заключается в переходе из начальной точки в новую допустимую точку факторного пространства, в которой значение целевой функции улучшается. Этот процесс продолжается до тех пор, пока сохраняется возможность такого улучшения. Каждый шаг поиска базируется на использовании двух операций – выборе подходящего направления, двигаясь в котором можно достичь лучших значений $\Delta_{\Sigma}(\pi_1, \pi_2, \dots, \pi_n)$, и оценке случайной величины перемещения.

При большом количестве управляемых параметров, обеспечивающих наиболее точное решение, целесообразно использовать комбинированный алгоритм, сочетающий преимущества методов случайного поиска и покоординатного спуска.

В данном методе поиск значений n переменных $\pi_1, \pi_2, \dots, \pi_n$, доставляющих экстремум функции $\Delta_{\Sigma}(\pi_1, \pi_2, \dots, \pi_n)$ при условиях $\pi_{i\max} \leq \pi_i \leq \pi_{i\min}$, начинается из исходной точки $Z_{\text{исх}}(\pi_1, \pi_2, \dots, \pi_n)$, в которой определяется значение функции $\Delta_{\Sigma_{\text{исх}}}(\pi_1, \pi_2, \dots, \pi_n)$ (рисунок 9).

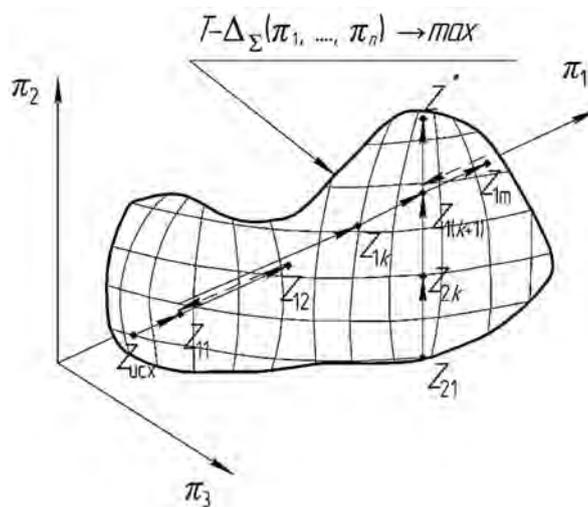


Рисунок 9 – Схема комбинированного метода

При этом из n переменных $\pi_1, \pi_2, \dots, \pi_n$ выбирается какая-то одна, например, π_1 , значения же остальных остаются фиксированными.

Далее осуществляется переход от начальной $Z_{1k}(\pi_{1k}, \pi_2, \dots, \pi_n)$ к новой допустимой точке $Z_{1(k+1)}(\pi_{1(k+1)}, \pi_2, \dots, \pi_n)$, и для нее оценивается значение

функции $\Delta_{\Sigma 1(k+1)}(\pi_{1(k+1)}, \pi_2, \dots, \pi_n)$, которое сравнивается со значением $\Delta_{\Sigma 1k}(\pi_{1k}, \pi_2, \dots, \pi_n)$, найденным предварительно в $Z_{1k}(\pi_{1k}, \pi_2, \dots, \pi_n)$.

Переход к новой точке осуществляется в соответствии с формулой

$$Z_{1(k+1)}(\pi_{1(k+1)}, \pi_2, \dots, \pi_n) = Z_{1k}(\pi_{1k} \pm a_k r_k, \pi_2, \dots, \pi_n), \quad (4)$$

где a_k – величина k -го шага, определяемая случайным образом;

r_k – единичный вектор, в направлении которого производится этот шаг.

Если оказывается $\Delta_{\Sigma 1(k+1)}(\pi_{1(k+1)}, \pi_2, \dots, \pi_n) \leq \Delta_{\Sigma 1k}(\pi_{1k}, \pi_2, \dots, \pi_n)$, то совершается переход из $Z_{1k}(\pi_{1k}, \pi_2, \dots, \pi_n)$ в $Z_{1(k+1)}(\pi_{1(k+1)}, \pi_2, \dots, \pi_n)$, после чего $Z_{1(k+1)}(\pi_{1(k+1)}, \pi_2, \dots, \pi_n)$ становится новой исходной точкой.

Если же $\Delta_{\Sigma 1(k+1)}(\pi_{1(k+1)}, \pi_2, \dots, \pi_n) > \Delta_{\Sigma 1k}(\pi_{1k}, \pi_2, \dots, \pi_n)$, то осуществляется возврат в исходную точку.

В дальнейшем в качестве исходных точек назначаются точки, в которых последовательно изменяются координаты относительно переменных π_2, \dots, π_n . Применительно к ним вся процедура поиска повторяется. Так продолжается до тех пор, пока не будет найдена точка экстремума $Z^*(\pi_1^*, \pi_2^*, \dots, \pi_n^*)$.

Порядок выполнения работы

- 1 Изучить метод покоординатного спуска.
- 2 Изучить случайный поиск с возвратом.
- 3 Изучить комбинированный метод поиска оптимальных решений.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Результаты сравнительного анализа методов поиска оптимальных решений.
- 4 Выводы.

Контрольные вопросы

- 1 В чем заключается проблема поиска локального минимума?
- 2 Дайте определение локального минимума, многоэкстремальной и одноэкстремальной задач.
- 3 Приведите описание общей структуры методов локальной оптимизации.
- 4 Приведите классификацию методов поиска.
- 5 В чем заключается метод покоординатного спуска? Сравните его свойства со свойствами случайных методов.
- 6 Какие методы относятся к методам случайного поиска?
- 7 В чем заключается комбинированный метод поиска решения?



8 Лабораторная работа № 8. Кластерный анализ

Цель работы: ознакомление с кластерным анализом и его методами, получение навыков применения методов ближайшего соседа и иерархической кластеризации.

Основные положения

Кластерный анализ – многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы [11]. Задача кластеризации относится к статистической обработке, а также к широкому классу задач обучения без учителя.

Цели кластеризации:

- понимание данных путём выявления кластерной структуры. Разбиение выборки на группы схожих объектов позволяет упростить дальнейшую обработку данных и принятия решений, применяя к каждому кластеру свой метод анализа (стратегия «разделяй и властвуй»);

- сжатие данных. Если исходная выборка избыточно большая, то можно сократить её, оставив по одному наиболее типичному представителю от каждого кластера;

- обнаружение новизны. Выделяются нетипичные объекты, которые не удаётся присоединить ни к одному из кластеров.

В первом случае число кластеров стараются сделать поменьше. Во втором случае важнее обеспечить высокую степень сходства объектов внутри каждого кластера, а кластеров может быть сколько угодно. В третьем случае наибольший интерес представляют отдельные объекты, не вписывающиеся ни в один из кластеров.

Во всех этих случаях может применяться иерархическая кластеризация, когда крупные кластеры дробятся на более мелкие, те, в свою очередь, дробятся ещё мельче и т. д. Такие задачи называются задачами таксономии. Результатом таксономии является древообразная иерархическая структура. При этом каждый объект характеризуется перечислением всех кластеров, которым он принадлежит, обычно от крупного к мелкому.

Задание

На основе наблюдений за неработоспособными объектами ($P_1, P_2, P_3, P_4, P_5, P_6, P_7$) установлено, что их характеристики описываются двухмерными признаками x_1 и x_2 .

Определить, имеются ли в этих наблюдениях закономерности, характеризующиеся сходными значениями признаков x_1 и x_2 .

Варианты заданий приведены в таблице 2 и выбираются по номеру студента в списке учебной группы.



Таблица 2 – Варианты заданий

Номер по списку	P_1		P_2		P_3		P_4		P_5		P_6		P_7	
	x_1	x_2												
1	3,5	6,8	2,9	2,9	1,5	7,7	14,4	22,8	16,3	21,3	19,3	23,0	34,5	47,4
2	1,4	1,6	0,8	6,5	2,1	4,1	14,3	9,8	12,7	6,2	12,2	11,4	3,2	27,5
3	0,8	7,8	6,8	7,5	6,0	8,3	0,4	6,3	0,9	3,2	6,6	3,8	4,5	32,2
4	3,8	7,0	2,5	1,6	1,1	3,3	15,7	42,7	10,4	42,4	14,4	49,8	38,9	48,3
5	25,5	9,8	24,3	2,4	22,4	6,3	3,7	23,0	1,1	17,8	4,4	17,5	24,9	38,9
6	1,3	2,8	3,2	2,8	2,9	9,9	18,2	23,5	15,4	23,8	16,6	16,0	4,3	60,0
7	2,6	9,6	2,0	4,4	4,7	3,6	10,5	31,6	7,9	32,0	9,4	37,7	24,9	38,9
8	6,7	2,1	2,9	2,9	0,5	24,4	5,7	23,8	5,9	5,9	4,1	26,7	39,2	24,1
9	1,4	1,6	0,8	6,5	2,1	4,1	14,3	9,8	12,7	6,2	12,3	10,0	3,2	27,5
10	13,4	25,2	31,7	3,8	34,2	9,2	12,6	18,8	35,3	2,9	11,2	20,8	36,5	50,1
11	0,3	3,9	1,8	4,9	0,8	2,3	5,1	18,9	21,2	20,7	18,8	19,6	20,2	22,7
12	3,3	5,4	2,1	0,8	1,0	6,8	5,2	42,1	3,0	42,1	18,6	15,9	3,6	45,5
13	1,4	6,7	1,3	0,9	0,2	3,2	15,8	8,9	16,9	14,4	17,2	10,2	1,8	28,9
14	1,0	8,9	5,0	7,5	4,6	3,3	10,9	28,6	12,7	32,5	15,6	28,0	36,9	37,2
15	1,7	1,5	3,6	4,2	5,0	1,3	7,1	48,5	11,0	47,2	10,8	43,5	33,9	49,6
16	3,4	7,2	5,0	3,4	2,5	4,1	13,9	13,9	13,3	16,3	16,3	16,2	17,1	30,5
17	2,5	6,2	17,8	6,8	4,4	3,7	16,2	8,8	17,9	14,0	4,5	6,6	14,5	52,6
18	2,4	26,0	3,4	7,5	3,1	4,3	3,2	24,2	4,1	28,7	1,3	3,2	31,7	22,8
19	2,3	1,2	0,6	1,7	1,6	7,1	13,4	16,8	12,8	21,7	14,9	22,6	16,0	54,2
20	3,7	3,7	3,2	7,1	0,1	4,3	30,0	13,4	12,0	24,1	33,5	14,9	29,4	16,6

Порядок выполнения работы

- 1 Построить матрицу евклидовых расстояний между точками P_i .
- 2 Используя метод ближайшего соседа, провести кластеризацию исходных данных и установить в них количество кластеров.
- 3 Провести кластеризацию методом иерархической кластеризации.
- 4 Построить графическое изображение исходных данных ($x_1; x_2$) и отразить результаты кластеризации (п. 2).
- 5 Сравнить результаты кластеризации (пп. 2 и 3).

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Матрица евклидовых расстояний между точками P_i .
- 4 Результаты кластеризации.
- 5 Сравнительный анализ результатов кластеризации.
- 6 Выводы.



Контрольные вопросы

- 1 В чем заключается кластерный анализ?
- 2 Назовите цели кластеризации.
- 3 Приведите описание метода иерархической кластеризации.
- 4 Приведите подходы кластерного анализа.
- 5 В чем заключается метод ближайшего соседа?
- 6 Сравните свойства метода ближайшего соседа со свойствами метода иерархической кластеризации.

9 Лабораторная работа № 9. Сравнительный анализ методов нахождения решения на функциональной семантической сети

Цель работы: исследование алгоритмов поиска решений при использовании функциональных семантических сетей и определение наиболее эффективного метода поиска оптимальных решений на сети.

Основные положения

Одним из методов представления знаний в интеллектуальных системах являются семантические сети, в которых знания отражаются в виде совокупности понятий и отношений между ними в некоторой предметной области [13].

В зависимости от типов отношений (связей), различают классифицирующие, функциональные семантические сети и сценарии.

Сети состоят из узлов и связывающих их дуг. Узлы в семантической сети соответствуют объектам, параметрам или событиям. Дуги описывают отношения между узлами.

Функциональную семантическую сеть изображают в виде графа, в котором вершины отображают понятия, а ребра или дуги – отношения между ними. Семантическую сеть можно представить тройкой объектов (V, E, θ) , где V – множество вершин графа, E – множество ребер, θ – функция инцидентности, которая каждому элементу множества E ставит в соответствие пару элементов из множества V .

У функциональной семантической сети множество вершин V является объединением непересекающихся подмножеств P и R , т. е.

$$V = P \cup R, \quad (5)$$

где P – множество параметров рассчитываемых задач, в том числе исходные данные;

R – множество отношений, определяющих расчетные зависимости решаемых с помощью семантической сети задач.



$$R_i = \{f(P_1, \dots, P_j, \dots, P_k) = 0\}, \quad (6)$$

где P_j – элемент множества параметров сети P ;

R_i – i -е отношение сети, определяющее функциональные зависимости между параметрами $P_1, \dots, P_j, \dots, P_k$ и имеющее вид

$$f(P_1, \dots, P_j, \dots, P_k) = 0. \quad (7)$$

На рисунке 10 приведена возможная структура функциональной семантической сети, представляющей собой в общем случае двудольный граф и состоящей из n отношений и k параметров. Здесь вершины-окружности являются параметрами проектируемой системы, а вершины-прямоугольники содержат расчетные зависимости (отношения).

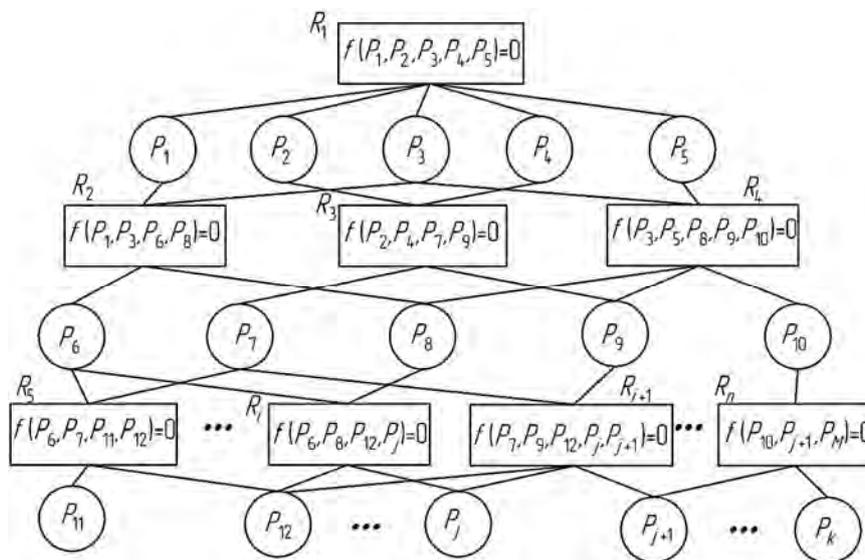


Рисунок 10 – Пример структуры функциональной семантической сети

Задание

Для определения наиболее эффективного метода поиска оптимальных решений на функциональной семантической сети с помощью интеллектуальной системы SEMANTIC провести сравнительный анализ алгоритмов, рассмотренных в лабораторной работе № 7.

Методы сопоставить по производительности (продолжительности поиска решений) и точности нахождения решений.

Порядок выполнения работы

1 Запустить интеллектуальную систему SEMANTIC, выбрав на рабочем столе файл SEMANTIC.exe.

2 С помощью процедуры *Загрузить СС* (рисунок 11) загрузить базу знаний в форме функциональной семантической сети. База знаний хранится в файле FSS.txt.

Функциональная семантическая сеть используется для определения суммарной погрешности расположения оси отверстия при сверлении. Исходными данными задачи являются: допуск на расположение оси отверстия $T_{obr} = 0,3$ мм; допуск базовой поверхности заготовки $TD_Z = 0,043$ мм; число установов $N = 50$; твердость материала заготовки по Бринеллю 241 НВ; диаметр инструмента $D_{in} = 12,8$ мм; подача сверла $S = 0,28$ мм/об; допуск на величину обратной конусности $T_k = 0,06$ мм; допуск на размер отверстия сменной втулки $T_{SV} = 0,011$ мм; допуск на размер направляющей части сверла $T_{IN} = 0,027$ мм; диаметральный зазор между сменной втулкой и сверлом $S_1 = 0,012$ мм; высота кондукторной втулки $l_{gm} = 12$ мм; максимальный зазор между шпонкой и Т-образным пазом станка $S_{shp} = 0,1$ мм; допуск на отклонение от соосности осей быстросменной и постоянной втулок $TB_1 = 0,006$ мм; допуск на отклонение от соосности осей постоянной втулки и отверстия в плите $TB_2 = 0,02$ мм; допуск на межцентровое расстояние между осями отверстий в плите $TB_3 = 0,03$ мм; допуск на отклонение от соосности осей отверстия во втулке и отверстия в плите $TB_4 = 0,04$ мм; допуск на отклонение от соосности осей отверстия во втулке и пальца $TB_5 = 0,036$ мм; допуск на расположение оси обрабатываемого отверстия относительно оси пальца приспособления $TB_7 = 0,02$ мм; допуск на эксцентricность пальца $TB_6 = 0,003$ мм и др.

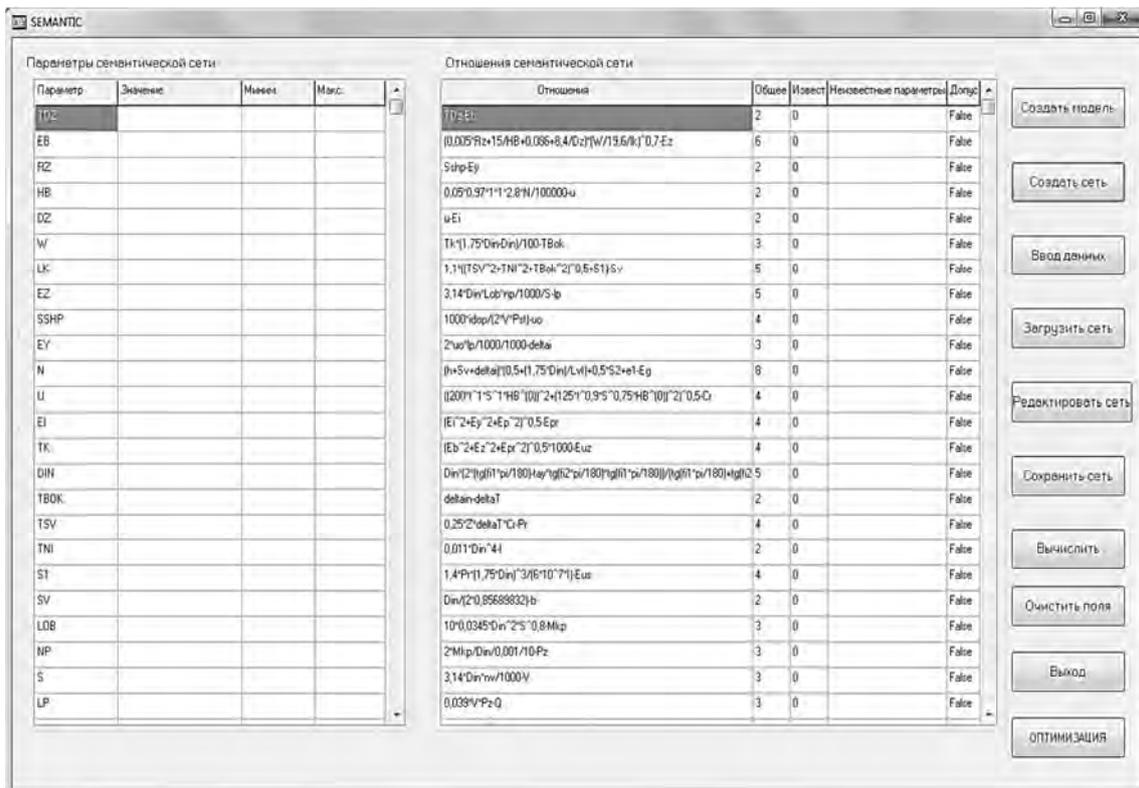


Рисунок 11 – Интерфейс программного обеспечения

3 С помощью процедуры *Оптимизация* (см. рисунок 11) выбрать оптимизируемый параметр сети, обеспечивающий наименьшую суммарную погрешность обработки отверстия.

4 С помощью процедуры *Метод оптимизации* выбрать метод по координатного спуска.

5 Определить продолжительность поиска решения. Результат записать в таблицу 3.

Таблица 3 – Продолжительность нахождения решений

Количество управляемых параметров	Время поиска, с						
	Метод по координатного спуска	Комбинированный метод	Случайный поиск с возвратом	Многолучевой поиск			
				5 лучей	10 лучей	20 лучей	100 лучей
1							
2							
5							
10							

6 Результат определения минимальной погрешности расположения оси отверстия (рисунок 12) записать в таблицу 4.

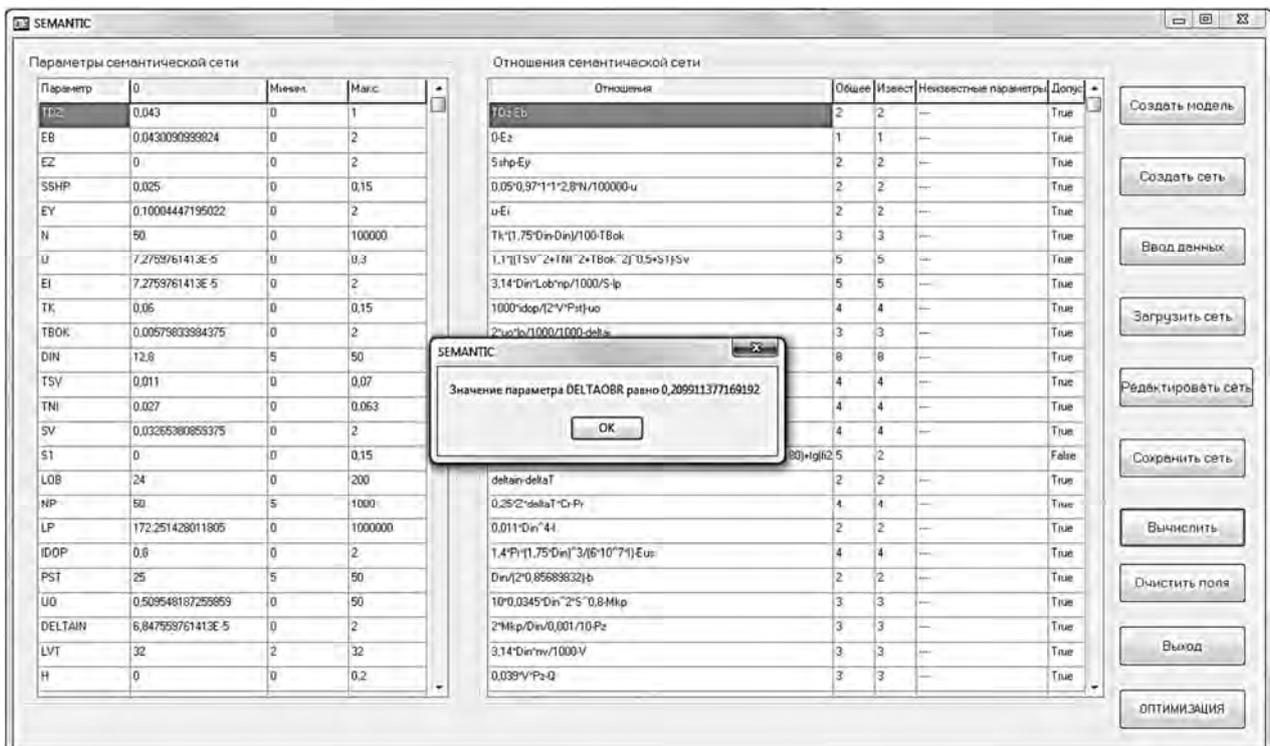


Таблица 4 – Значения целевой функции, соответствующие рациональным параметрам технологического оснащения

Количество управляемых параметров	Суммарная погрешность расположения оси отверстия, мм						
	Метод покоординатного спуска	Комбинированный метод	Случайный поиск с возвратом	Многолучевой поиск			
				5 лучей	10 лучей	20 лучей	100 лучей

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Результаты определения продолжительности поиска решения исследуемыми методами.
- 4 Результаты определения минимальной погрешности расположения оси отверстия.
- 5 Результаты сравнительного анализа алгоритмов.
- 6 Выводы.

Контрольные вопросы

- 1 Что такое функциональная семантическая сеть?
- 2 Приведите классификацию методов поиска на сети.
- 3 В чем заключается метод покоординатного спуска? Сравните его свойства со свойствами случайных методов.
- 4 Какие методы относятся к методам случайного поиска?
- 5 В чем заключается комбинированный метод поиска решения?
- 6 Дайте оценку исследуемых методов поиска.

10 Лабораторная работа № 10. Управление процессом сборки с помощью конечного автомата

Цель работы: ознакомление с конечным автоматом, управляющим процессом сборки узла или механизма.

Основные положения

Конечный автомат – это некоторая абстрактная модель, содержащая конечное число состояний чего-либо. Используется для представления и управления потоком выполнения каких-либо команд. Конечный автомат идеально подходит для реализации искусственного интеллекта, получая аккуратное решение без написания громоздкого и сложного кода.

Конечный автомат – это модель вычислений, основанная на гипотетической машине состояний. В один момент времени только одно состояние может быть активным. Следовательно, для выполнения каких-либо действий машина должна менять свое состояние.

Конечный автомат можно представить в виде графа, вершины которого являются состояниями, а ребра – переходами между ними. Каждое ребро имеет метку, информирующую о том, когда должен произойти переход.

Автомат – это математическая абстракция, которая предназначена для моделирования разнообразных устройств и процессов дискретной переработки информации. С помощью автомата можно описать поведение устройств дискретного действия или протекание дискретных процессов.

Универсальность теории автоматов позволяет рассматривать с единой точки зрения различные объекты, учитывать связи и аналогии между ними, переносить результаты из одной предметной области в другую.

Автомат описывается шестёркой элементов:

$$A = \{Q, X, Y, \delta, \lambda, q_1\}, \quad (8)$$

где Q – множество состояний, $Q = \{q_1, q_2, \dots, q_r\}$;

X – множество входных символов, $X = \{x_1, x_2, \dots, x_n\}$;

Y – множество выходных символов, $Y = \{y_1, y_2, \dots, y_m\}$;

δ – функция переходов, реализующая отображение множества $D_\delta \subseteq Q \times X$ в Q ($q_p = \delta(q_j, x_i), q_p \in Q$);

λ – функция выходов, реализующая отображение множества $D_\lambda \subseteq Q \times X$ на Y ($y_d = \lambda(q_j, x_i), y_d \in Y$);

q_1 – начальное состояние автомата.

Автомат может иметь некоторое количество входов и выходов, и в этом случае каждому входу и каждому выходу может соответствовать свой алфавит.

Автомат называется конечным, если конечны множества Q , X и Y . Автомат называется полностью определённым, если $D_\delta = D_\lambda = Q \times X$. У частичного автомата функции δ и λ определены не для всех пар $(q_j, x_i) \in Q \times X$.

Символами обозначают *события* в процессе или *сигналы* в устройствах. Иногда используют вместо понятия «символ» понятие «буква», а последовательность букв называют *словом*.

В отличие от привычного рассмотрения времени (время непрерывно и задается на континууме), при изучении и проектировании автоматов удобно рассматривать воображаемое дискретное время (автоматное время, заданное на счётном множестве).

В реальных условиях символы (сигналы) представляются непрерывными функциями времени, поэтому для их надежного различения требуется, чтобы новые значения на входах появлялись после окончания переходных процессов, связанных с предыдущими значениями. При рассмотрении логической структуры автоматов обычно отвлекаются от существа этих процессов и считают, что переменные изменяются мгновенно в тактовые моменты.

Кроме входных и выходных переменных, можно выделить некоторую

совокупность промежуточных переменных, которые связаны с внутренней структурой автомата; именно они характеризуют состояние конечного автомата.

При рассмотрении конечного автомата значения символов состояний и входов существенны лишь в моменты тактов, и несущественны в промежутках между ними. Поэтому эту модель можно использовать и для описания непрерывных устройств (процессов), если фиксировать значения символов состояний и входов в моменты тактов. При этом важно, чтобы в рассматриваемые дискретные моменты времени множество возможных состояний было конечным и чтобы удовлетворялось требование однозначной связи между состояниями в соседних тактах.

Понятие «состояние автомата» определяет некоторую предысторию его поведения как реакции на символы, которые поступали ранее на его входы. Другими словами, состояние соответствует некоторой памяти о прошлом.

Строгое определение понятия состояния связывается с той ролью, которое оно играет при определении конечных автоматов.

Во-первых, значение выходной переменной на p -м такте (p – present – настоящее) $y(p)$ однозначно определяется значением входной переменной $x(p)$ и состоянием $q(p)$ на том же такте, т. е. $y(p) = \lambda(x(p), q(p))$.

Во-вторых, состояние $q(p + 1)$ в следующем, $(p + 1)$ -м такте однозначно определяется входной переменной $x(p)$ и состоянием $q(p)$ в рассматриваемом такте, т. е. $q(p + 1) = \delta(x(p), q(p))$.

Таким образом, состояние конечного автомата в любой тактовый момент характеризуется значением такой переменной, которая вместе с заданным значением входной переменной позволяет определить выходную переменную в данный тактовый момент и состояние в следующий тактовый момент.

Автоматы должны обладать способностью сохранять предыдущее состояние до следующего такта, в связи с чем их называют *автоматами с памятью*. В качестве памяти могут использоваться элементы задержки, на выходах которых повторяются входные воздействия со сдвигом во времени на интервал между тактами Δt .

Автомат без памяти называется *тривиальным автоматом* или *комбинационной схемой*. В таких автоматах значения выходных переменных определяются только комбинацией входных переменных в данный момент времени; для комбинационных схем функция переходов не имеет смысла, а функция выходов вырождается к виду $y(p) = \lambda(x(p))$.

Задание

Построить конечный автомат, управляющий процессом сборки узла механического редуктора. Сборочный робот, снабженный системой технического зрения, должен при этом обеспечить гибкий сценарий сборки (выполнять сборку с учетом произвольно появляющихся на конвейере деталей, а также с учетом деталей, уже установленных в сборочный узел).



Варианты заданий приведены в таблице 5 и выбираются по номеру студента в списке учебной группы.

Таблица 5 – Варианты заданий

Вторая цифра номера	Схема собираемого узла (рисунок 13)	Вторая цифра номера	Схема собираемого узла (рисунок 13)
0	а	5	е
1	б	6	ж
2	в	7	з
3	1	8	и
4	д	9	к

Порядок выполнения работы

1 Построить автоматную таблицу, содержащую набор команд для управления сборочным роботом, учитывающую возможные комбинации входных сигналов X (совокупность деталей на конвейере) и допустимых внутренних состояний автомата Y (совокупность деталей, уже установленных в узле). Базовой является деталь, обозначенная на схеме позицией 0; по конвейеру движутся детали, обозначенные позициями 1, 2, 3.

Пример входных сигналов:

«000» – на сборочном конвейере нет деталей;

«001» – по конвейеру движется деталь, обозначенная на схеме позицией 1;

«110» – по конвейеру движутся детали, обозначенные на схеме позициями 2 и 3.

Пример внутренних состояний:

«000» – в узле нет установленных деталей;

«001» – в узел установлена деталь, обозначенная на схеме позицией 1;

«111» – все детали установлены в узел.

Отобрать из общей совокупности внутренних состояний только те из них, которые допустимы исходя из возможных последовательностей сборки.

Пример команд робота:

« W » (wait) – не выполнять каких-либо действий, ожидать изменения входного сигнала;

« R » (release) – снять собранный узел и установить в сборочное приспособление базовую деталь, обозначенную на схеме позицией 0 (выйти в исходное положение для сборки);

«1» – манипулятором взять деталь, обозначенную на схеме позицией 1, и установить на базовую деталь;

«2» – манипулятором взять деталь, обозначенную на схеме позицией 2, и установить на базовую деталь;

«3» – манипулятором взять деталь, обозначенную на схеме позицией 3, и установить на базовую деталь.

2 Проанализировать автоматную таблицу и устранить возможные колли-

зии путем исключения из таблицы альтернативных команд.

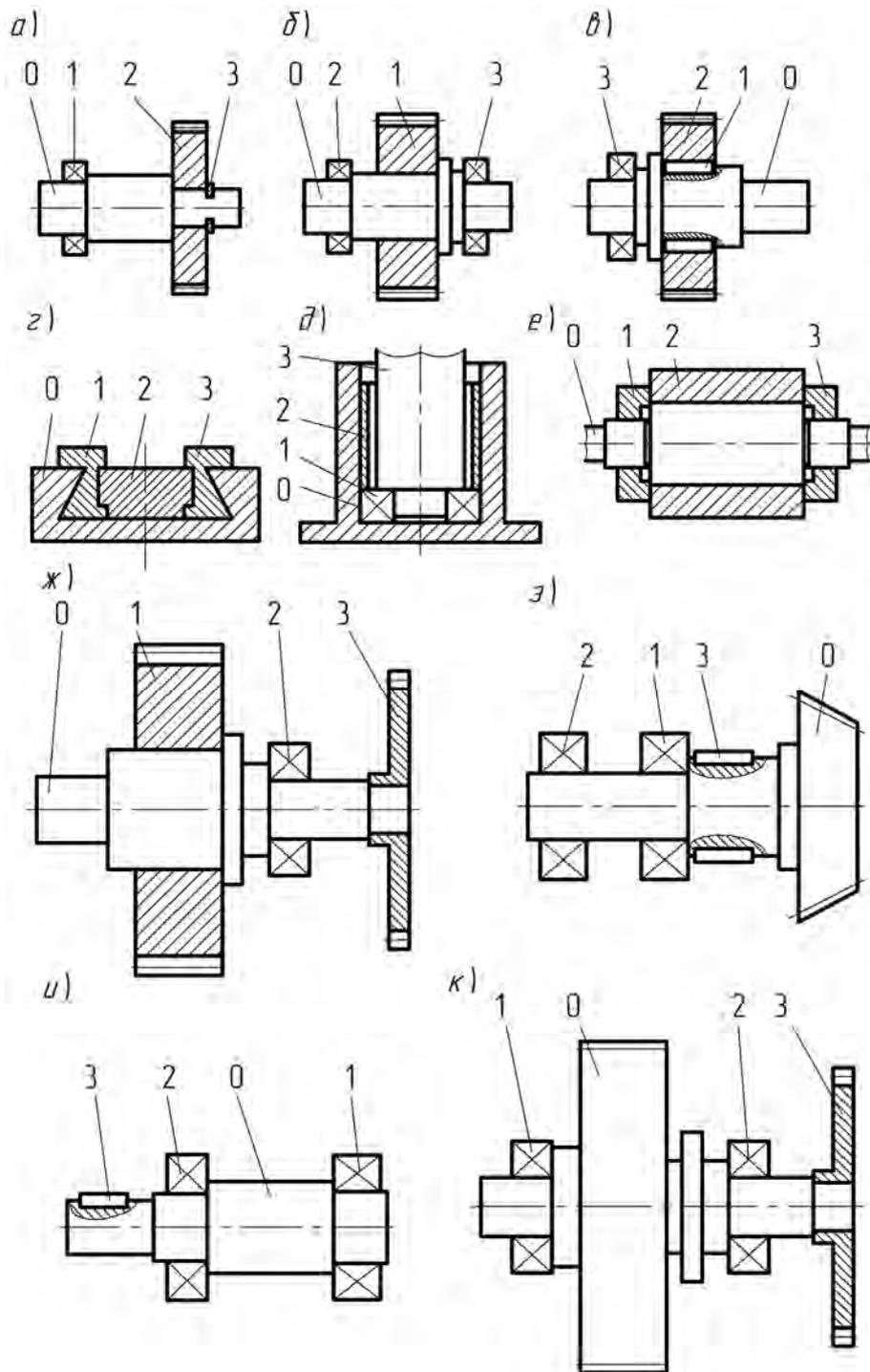


Рисунок 13 – Варианты схем собираемых узлов

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Автоматная таблица, содержащая набор команд для управления сборочным роботом.
- 4 Выводы.

Контрольные вопросы

- 1 Что такое автомат?
- 2 Дайте определение конечному автомату.
- 3 Как можно описать автомат?
- 4 Что характеризует состояние конечного автомата?
- 5 Что определяет понятие «состояние автомата»?
- 6 Какие задачи решаются в теории автоматов?
- 7 Охарактеризуйте автоматы с памятью и без нее.

11 Лабораторная работа № 11. Метрическое распознавание объектов

Цель работы: ознакомление с методикой метрического распознавания образов.

Основные положения

Метрические методы связаны с измерением расстояний в пространстве признаков. Будем характеризовать состояние системы (изделия) вектором параметров

$$x = (x_1, x_2, \dots, x_m). \quad (9)$$

Компоненты вектора x могут быть непрерывными или дискретными величинами. В последнем случае x_j представляет собой (многозарядный) диагностический признак.

Каждое состояние изделия в соответствии с данным равенством может быть представлено точкой в пространстве признаков, а вектор \vec{x} соединяет эту точку с началом координат. Предполагается, что точки с одним и тем же состоянием (диагнозом) группируются в компактной области пространства признаков («гипотеза компактности»).

Допустим, что имеется n_i образцов с диагнозом D_i (рисунок 14). Они образуют обучающую последовательность. Точки, входящие в области диагнозов, обычно располагаются более плотно в центральной части области.

Примем в качестве «типичного» изделия с данным диагнозом «среднюю точку», которую назовем эталоном.

Координаты эталона i -го диагноза

$$a_{ij} = \frac{1}{n_i} \sum_{s=1}^{n_i} a_{ij} S, \quad (10)$$



где $a_{ij}S$ – значение параметра x_j ($j = 1, 2, \dots, m$) для образца S , принадлежащего диагнозу D_i .

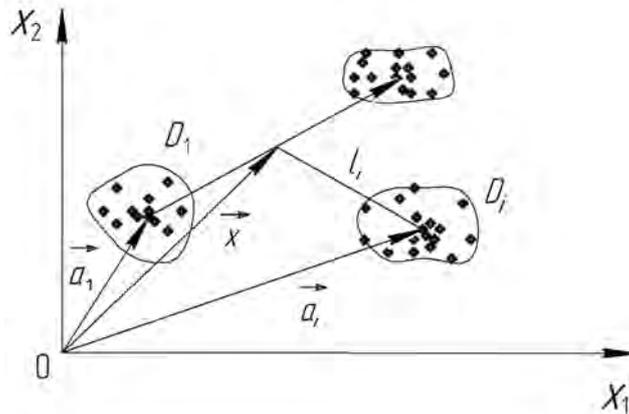


Рисунок 14 – Область диагнозов (состояний) в пространстве признаков

Пусть предъявлено для распознавания изделие, характеризующееся вектором \vec{x} в пространстве признаков. Решение вопроса об отнесении изделия к диагнозу D_i связано с измерением расстояния до эталонов.

Решающее правило принимается по минимальному расстоянию до эталона:

$$l_i = \min, x \in D_i, \quad (11)$$

Таким образом, если точка x ближе всего к эталону диагноза D_i , то вывод делается в пользу диагноза D_i .

Расстояния до i -го эталона

$$l_i = \left(\sum_{j=1}^m (x_j - a_{ij})^2 \right)^{\frac{1}{2}}. \quad (12)$$

Задание

Для оценки состояния подшипниковых опор редуктора в сборе предложено использовать сигналы двух преобразователей виброускорений x_1 и x_2 , смонтированных над передней и задней опорами ведомого вала соответственно. Установлено, что на основе анализа этих сигналов можно разделять редукторы на два следующих класса:

- класс 1: работоспособное состояние (замена подшипников не требуется);
- класс 2: критическое состояние (требуется замена подшипников, износ которых достиг предельного значения).

По результатам исследований партии редукторов определены эталоны (средние значения) E_1 и E_2 вышерассмотренных классов.

Определить состояние объектов O_1 и O_2 по сигналам преобразователей $x_1(O_1)$, $x_2(O_2)$ и $x_1(O_2)$, $x_2(O_1)$ на основе методики метрического распознавания образов.

Варианты заданий приведены в таблицах 6 и 7 и выбираются по номеру студента в списке учебной группы.

Таблица 6 – Варианты заданий

Вторая цифра номера	$x_1(E_1)$	$x_2(E_1)$	$x_1(E_2)$	$x_2(E_2)$	$x_1(O_1)$	$x_2(O_1)$	$x_1(O_2)$	$x_2(O_2)$
0	3,6	3	12	12	14,3	9	10	4
1	4	4	13	15	5	12	5	5
2	4	2	18	14	15,3	5,3	5	6
3	2	6	14	12	6,5	10	4	5
4	5,6	4,1	18	13	9	12	4,3	5
5	6	2	18	10	5	10	17,3	7
6	1,5	1,2	10	5,5	9	4,5	4,5	4
7	1,5	1	10	12	1,5	4	10	5,5
8	1,7	6	14	14	5	4	5,7	12
9	1	7	18	10	4,3	5	11	10

Таблица 7 – Варианты заданий

Первая цифра номера	m_1	m_2	m_3
0	2	0,2	7
1	2	0,28	7
2	2	0,15	8
3	2	0,23	6

Порядок выполнения работы

1 Построить графическое изображение эталонов E_1 и E_2 , а также объектов O_1 и O_2 в пространстве признаков x_1 и x_2 .

2 Определить расстояния от исследуемых объектов до эталонов классов при показателях степени обобщенного расстояния m_1 , m_2 , m_3 , а также надежности принадлежности объектов классам 1 и 2.

В качестве меры расстояния между эталоном и объектом принять обобщенную метрику

$$L = \sum_{k=1}^n (E_k - O_k)^m, \quad (13)$$

где n – размерность пространства признаков;

m – показатель степени обобщенного расстояния.

Порог надежности для принятия гипотезы о принадлежности объекта выбранному классу состояний принять равным 0,7.

3 Оценить влияние показателя степени m (меры резкости пространства) на надежность распознавания.



Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Результаты построения графических изображений эталонов и объектов.
- 4 Расстояния от исследуемых объектов до эталонов классов.
- 5 Результаты оценки влияния меры резкости пространства на надежность распознавания.
- 6 Выводы.

Контрольные вопросы

- 1 Дать понятие о метрических методах распознавания.
- 2 Разъяснить суть и практическую направленность метрических методов распознавания.
- 3 На чем основаны методы разделения в пространстве признаков?
- 4 Поясните физический смысл метода эталонов.
- 5 Поясните физический смысл метода минимального расстояния до множества.

12 Лабораторная работа № 12. Использование коэффициентов определенности

Цель работы: построение системы для распознавания двух состояний на основе коэффициентов определенности и базы данных о состоянии механических передач.

Основные положения

Если должна быть численно выражена определенность, то используется интервал от -1 до $+1$. Границы интервала обозначают следующее: $+1$ – система в чем-то полностью определена, 0 – у системы нет знаний об обсуждаемой величине, -1 – высказанная гипотетическая посылка или заключение абсолютно неверно. Промежуточные величины отражают степень доверия или недоверия к указанным ситуациям.

Полная реализация идеи биполярных коэффициентов определенности требует сделать два обобщения.

- 1 Если в правиле есть отрицание, например,

если (e_1 и (не e_2)), то (c),

то нужно считать (не e_2) новым атомарным утверждением, например, e_3 . Для вычисления же коэффициента определенности (не e_2) достаточно просто поменять знак



$$c_1(\text{не } e) = -ct(e).$$

2 Правило получения коэффициентов определенностей в условиях поддержки двумя правилами одного и того же заключения:

– если оба коэффициента определенности положительны, то

$$ctotal = ct_1 + ct_2 - ct_1 \cdot ct_2;$$

– если оба коэффициента определенности отрицательны, то

$$ctotal = ct_1 + ct_2 + ct_1 \cdot ct_2.$$

Когда отрицателен один коэффициент, то

$$ctotal = (ct_1 + ct_2) / (1 - \min(\text{abs}(ct_1), \text{abs}(ct_2))).$$

В том случае, когда одна определенность равна +1, а другая –1, то $Ctotal = 0$.

Таблица 8 содержит все возможные способы комбинирования двух коэффициентов определенности в соответствии с указанными выше правилами.

Таблица 8 – Результат композиции коэффициентов определенности

Значение коэффициентов определенности для правила 2	Значение коэффициентов определенности для правила 1										
	-1	-0,8	-0,6	-0,4	-0,2	0	0,2	0,4	0,6	0,8	1
1,0	0	1	1	1	1	1	1	1	1	1	1
0,8	-1	0	0,5	0,67	0,75	0,8	0,84	0,88	0,92	0,96	1
0,6	-1	-0,5	0	0,33	0,5	0,6	0,68	0,76	0,84	0,92	1
0,4	-1	-0,67	0,33	0	0,25	0,4	0,52	0,64	0,76	0,88	1
0,2	-1	-0,75	-0,5	-0,25	0	0,2	0,36	0,52	0,68	0,84	1
0	-1	-0,8	-0,6	-0,4	-0,2	0	0,2	0,4	0,6	0,8	1
-0,2	-1	-0,84	0,68	-0,52	-0,36	-0,2	0	0,25	0,5	0,75	1
-0,4	-1	-0,88	-0,76	-0,64	-0,52	-0,4	-0,25	0	0,33	0,67	1
-0,6	-1	-0,92	-0,84	-0,76	-0,64	-0,6	-0,5	-0,33	0	0,5	1
-0,8	-1	-0,96	-0,92	-0,88	-0,84	-0,8	-0,75	-0,67	-0,5	0	1
-1,0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

Значения коэффициентов определенности для правила 1 отложены по горизонтальной оси, а для правила 2 – по вертикальной. Числа отражают результаты конкретных комбинаций. Здесь происходит следующее: если два правила с небольшими коэффициентами определенности поддерживают одно заключение, то коэффициент определенности заключения возрастает. Если же знаки не совпадают, то результат определяется сильнейшим, но влияние его несколько ослабляется. Применение биполярных коэффициентов опреде-

ленности может привести к нереальным результатам, если правила сформулированы неточно.

Задание

На основе коэффициентов определенности и базы данных о состоянии механических передач построить систему для распознавания двух состояний:

D_1 – нормальное состояние (работоспособное);

D_2 – критическое состояние (неработоспособное).

В качестве критериев распознавания были выбраны следующие признаки:

S_1 – шум передачи не превышает 80 дБ;

S_2 – уровень виброускорения на корпусе передачи свыше 3 м/с²;

S_3 – содержание железа в масле менее 150 мг/л;

S_4 – температура корпуса передачи не превышает 70 °С.

Варианты заданий приведены в таблицах 9 и 10, выбираются по номеру студента в списке учебной группы.

В таблицах 9 и 10 представлены типовые наблюдения за объектами семи типов, соответствующих состояниям D_1 и D_2 . При этом значение «1» соответствует наличию признака S_i , а значение «0» – отсутствию признака. В правом столбце указано количество объектов базы данных $n_{об}$, для которых установлены соответствующие сочетания признаков S_i .

Таблица 9 – Варианты заданий

Состояние		D_1	S_1	S_2	S_3	S_4	$n_{об}$
Первая цифра номера в списке	0	Объекты 1-го типа	1	0	1	0	10
		Объекты 2-го типа	0	0	1	1	25
		Объекты 3-го типа	1	0	0	1	5
	1	Объекты 1-го типа	1	0	0	1	10
		Объекты 2-го типа	1	0	1	0	20
		Объекты 3-го типа	1	0	1	1	10
	2	Объекты 1-го типа	0	1	0	1	15
		Объекты 2-го типа	1	1	0	1	25
		Объекты 3-го типа	1	1	0	0	10
	3	Объекты 1-го типа	1	0	1	0	10
		Объекты 2-го типа	1	0	0	1	25
		Объекты 3-го типа	1	0	1	1	5

Таблица 10 – Варианты заданий

Состояние		D_1	S_1	S_2	S_3	S_3	$n_{об}$
1	2	3	4	5	6	7	8
Вторая цифра номера в списке	1	Объекты 4-го типа	0	1	0	1	10
		Объекты 5-го типа	0	1	1	0	5
		Объекты 6-го типа	1	0	1	0	3
		Объекты 7-го типа	1	0	0	1	2



Окончание таблицы 10

1	2	3	4	5	6	7	8
Вторая цифра номера в списке	2	Объекты 4-го типа	1	0	0	1	10
		Объекты 5-го типа	1	1	0	1	5
		Объекты 6-го типа	1	0	0	0	3
		Объекты 7-го типа	0	1	1	0	7
	3	Объекты 4-го типа	1	0	1	1	10
		Объекты 5-го типа	0	1	0	0	5
		Объекты 6-го типа	1	0	1	0	5
		Объекты 7-го типа	1	1	0	0	5
	4	Объекты 4-го типа	0	1	0	1	8
		Объекты 5-го типа	0	1	1	1	5
		Объекты 6-го типа	1	0	0	0	7
		Объекты 7-го типа	1	0	0	1	5
	5	Объекты 4-го типа	0	1	1	1	10
		Объекты 5-го типа	0	1	0	0	5
		Объекты 6-го типа	1	0	0	0	3
		Объекты 7-го типа	0	1	0	1	2
	6	Объекты 4-го типа	0	1	1	1	8
		Объекты 5-го типа	0	1	0	0	5
		Объекты 6-го типа	1	0	0	0	5
		Объекты 7-го типа	0	1	0	1	2
	7	Объекты 4-го типа	1	0	1	1	7
		Объекты 5-го типа	0	0	0	0	5
		Объекты 6-го типа	1	0	1	0	5
		Объекты 7-го типа	0	0	1	0	8
	8	Объекты 4-го типа	1	0	1	1	10
		Объекты 5-го типа	1	1	0	0	5
		Объекты 6-го типа	1	0	0	0	3
		Объекты 7-го типа	0	1	1	0	2
	9	Объекты 4-го типа	0	1	1	1	10
		Объекты 5-го типа	0	1	0	0	5
		Объекты 6-го типа	1	0	0	0	3
		Объекты 7-го типа	0	1	0	1	2
0	Объекты 4-го типа	1	1	1	1	8	
	Объекты 5-го типа	0	0	1	0	7	
	Объекты 6-го типа	1	0	0	0	3	
	Объекты 7-го типа	0	0	1	1	2	



Порядок выполнения работы

- 1 Приняв в качестве базового состояния D_1 (нормальное состояние), рассчитать значения коэффициентов определенности для каждого из признаков S_i .
- 2 Проверить качество полученной системы коэффициентов определенности, оценив с их помощью принадлежность объектов типов 1...7 состояниям D_1 и D_2 . В качестве порога надежности для принятия гипотезы значение коэффициента определенности для комбинации признаков считать равным 0,7.
- 3 Определить величину ошибки системы распознавания.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Результаты определения коэффициентов определенности для каждого из признаков.
- 4 Результаты определения величины ошибки системы распознавания.
- 5 Выводы.

Контрольные вопросы

- 1 Назовите два обобщения при использовании коэффициентов определенности.
- 2 Какое правило называют обратимым, а какое – необратимым?
- 3 Каким образом на основе коэффициентов определенности и базы данных о состоянии механических передач строится система для распознавания двух состояний?

13 Лабораторная работа № 13. Обучение в интеллектуальных системах

Цель работы: изучение методов обучения в интеллектуальных системах.

Основные положения

Процесс приобретения знаний называется обучением. Смысл процесса машинного обучения можно пояснить следующим образом. В процессе повторения однотипных экспериментов происходит модификация программной системы, в результате которой система демонстрирует на следующем этапе экспериментов результаты лучшие, чем прежде.

Приобретение знаний реализуется с помощью двух функций: получения информации извне и ее систематизации. В зависимости от интеллектуальных способностей системы возможны разные формы приобретения знаний и представления исходной информации.



Классификация этапов обучения, соответствующих способностям компьютерных систем к формализации.

А. Получение информации без логических выводов.

- 1 Ввод программ.
- 2 Ввод фактических данных.

В. Получение знаний извне.

- 1 Получение готового набора знаний, представленных во внутреннем формате.
- 2 Получение знаний во внутреннем формате в режиме диалога.
- 3 Получение знаний во внешнем формате и их понимание.

С. Обучение по примерам.

- 1 Параметрическое обучение.
- 2 Обучение на основе выводов по аналогии.
- 3 Индуктивное обучение.
- 4 Обучение нейронных сетей.

Д. Приобретение знаний на метауровне.

Методы категории *А* можно назвать обучением без выводов или механическим запоминанием. Здесь полученная информация используется для решения задач в том же виде, в котором поступает в систему.

В методах категории *В* внешняя информация поступает в компьютерную систему в форме знаний, следовательно, ее можно использовать в процессе логического вывода. Примером такой информации являются правила, поступающие в экспертную систему продукционного типа. В интеллектуальной системе необходимо иметь функцию преобразования поступающих знаний в формат, удобный для дальнейшего использования. Эту функцию выполняют редакторы знаний, которые не только корректируют поступающую информацию, но и выполняют проверку знаний, содержащихся в базе знаний, на наличие конфликтов (противоречий) с новыми знаниями.

Одной из актуальных проблем искусственного интеллекта является *операционализация знаний*, связанная с пониманием, преобразованием и использованием советов и подсказок, которые человек сообщает интеллектуальной системе в ходе решения конкретных задач. Советы и подсказки, поступающие в систему на естественном языке в терминах предметной области, необходимо преобразовывать в процедуры, ориентированные на выполнение определенных действий.

Категория *С* принципиально отличается от *А* и *В* тем, что интеллектуальные системы приобретают знания самостоятельно, выполняя сбор отдельных фактов, их обобщение и систематизацию. В процессе решения задач определенного класса (примеров) компьютерная система выявляет понятия, выбирает формат их представления и проводит структуризацию. Подобные задачи вызывают сложности и у человека.

Проблемы понимания и преобразования знаний характерны для категорий *В* и *С*, причем последняя отличается наличием процедур обобщения, примерами которых являются замена постоянных атрибутов языка (констант) на переменные, исключение описаний с ограниченным применением, индук-



ция, абдукция и др. Способы обобщения тесно связаны с языком представления знаний в информационных системах.

Параметрическое обучение – самая простая форма в категории *C*. Оно заключается в определении общего вида правила, формирующего результат вывода, и в последующей корректировке входящих в него параметров, зависящих от конкретных данных. Пример обучающейся системы такого типа – известная экспертная система *Meta-Dendral*, предназначенная для построения структурных формул химических соединений на основе данных масс-спектрального анализа. В ней выводятся новые правила путем коррекции уже заложенных в базу знаний продукций.

Обучение по аналогии базируется на гипотезе о том, что «если две ситуации подобны по нескольким признакам, то они подобны еще по одному признаку». Подобие ситуаций распознается на основе обобщенной меры совпадения значений важнейших признаков, с помощью которых описаны ситуации.

Вопросы *приобретения знаний на метауровне* актуальны на современном этапе развития искусственного интеллекта, т. к. связаны с выработкой стратегий управления процессом решения задач в интеллектуальных информационных системах. Это направление активно развивается, но здесь пока не выработано устойчивых представлений и апробированных моделей. Вопросы приобретения метазнаний частично затрагиваются в новых направлениях *Data Mining* и *Knowledge Discovery*, которые связаны с извлечением знаний из данных.

Центроидный метод. Расстояние между двумя кластерами определяется как евклидово расстояние между центрами (средними) этих кластеров. Кластеризация осуществляется поэтапно: на каждом шаге объединяют два кластера, расстояние между которыми минимально.

Задание

Используя центроидный метод и метод, базирующийся на процедуре обучения, обучить систему распознаванию двух классов состояний механических передач:

- 1) работоспособное;
- 2) неработоспособное.

При этом в качестве представителей первого класса выбраны объекты P_1 и P_2 , второго – объекты P_3 и P_4 . Объекты характеризуются двухмерным набором признаков x_1 и x_2 .

Варианты заданий приведены в таблице 11 и выбираются по номеру студента в списке учебной группы.



Таблица 11 – Варианты заданий

Номер по списку	Работоспособное состояние				Неработоспособное состояние			
	P_1		P_2		P_3		P_4	
	x_1	x_2	x_1	x_2	x_1	x_2	x_1	x_2
1	0	1	2	1	6	7	8	9
2	2	3	1	2	8	6	7	9
3	1	1	2	2	6	7	8	5
4	1	2	1	0	9	8	6	7
5	2	1	1	3	7	6	9	8
6	1	2	3	1	6	6	7	8
7	0	1	1	3	5	6	7	7
8	1	1	2	3	6	8	8	9
9	2	0	0	1	7	6	9	7
10	1	1	3	1	6	8	7	5
11	0	1	2	1	5	7	6	8
12	3	2	1	1	8	9	9	10
13	2	3	4	0	7	8	10	9
14	2	4	3	2	7	9	8	6
15	1	1	3	2	8	7	8	10
16	2	3	3	4	8	8	9	9
17	2	2	4	3	8	7	9	8
18	1	0	2	2	6	8	8	9
19	1	0	2	1	7	6	8	7
20	0	1	2	2	7	6	8	8
21	2	3	4	2	8	9	9	10
22	1	3	2	1	8	7	9	9
23	1	0	2	1	8	9	9	10
24	1	1	0	3	7	9	8	10
25	4	3	3	2	9	9	10	11
26	2	1	3	3	7	8	9	8
27	2	0	1	3	8	7	9	11
28	2	1	3	1	8	7	8	10

Порядок выполнения работы

1 Рассчитать центроиды классов и найти выражение для дискриминантной функции. Определить принадлежность объектов $P_1 \dots P_4$ классу 1 или 2.

2 Используя алгоритм обучения, найти выражение для дискриминантной функции.

3 Сравнить результаты, полученные в пп. 1 и 2.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Выражения для дискриминантной функции.
- 4 Выводы.

Контрольные вопросы

- 1 В чем заключается процесс машинного обучения?
- 2 С помощью каких функций реализуется приобретение знаний в интеллектуальных системах?
- 3 Приведите этапов обучения, соответствующих способностям компьютерных систем к формализации.
- 4 Охарактеризуйте методы обучения категорий A, B, C .
- 5 В чем заключается параметрическое обучение?
- 6 На чем базируется обучение по аналогии?
- 7 С чем связано приобретения знаний на метауровне?
- 8 В чем заключается центроидный метод?

14 Лабораторная работа № 14. Нахождение кратчайшего пути транспортным роботом

Цель работы: изучение методов нахождения кратчайшего пути на графах; определение кратчайшего пути с помощью алгоритма Дейкстры.

Основные положения

Задача о кратчайшем пути – задача поиска самого короткого пути (цепи) между двумя точками (вершинами) на графе, в которой минимизируется сумма весов ребер, составляющих путь.

Задача поиска кратчайшего пути на графе может быть определена для неориентированного, ориентированного или смешанного графа. Далее будет рассмотрена постановка задачи в самом простом виде для неориентированного графа. Для смешанного и ориентированного графа дополнительно должны учитываться направления ребер.

Если все ребра в графе имеют единичный вес, то задача сводится к определению наименьшего количества обходимых ребер.

Алгоритмы для решения задачи поиска кратчайшего пути на графе [12].

Алгоритм Дейкстры находит кратчайший путь от одной из вершин графа до всех остальных. Алгоритм работает только для графов без ребер отрицательного веса.

Алгоритм Беллмана – Форда находит кратчайшие пути от одной вершины графа до всех остальных во взвешенном графе. Вес ребер может



быть отрицательным.

Алгоритм поиска A^* находит маршрут с наименьшей стоимостью от одной вершины (начальной) к другой (целевой, конечной), используя алгоритм поиска по первому наилучшему совпадению на графе.

Алгоритм Флойда – Уоршелла находит кратчайшие пути между всеми вершинами взвешенного ориентированного графа.

Алгоритм Джонсона находит кратчайшие пути между всеми парами вершин взвешенного ориентированного графа.

Алгоритм Ли (волновой алгоритм) основан на методе поиска в ширину. Находит путь между вершинами s и t графа (s не совпадает с t), содержащий минимальное количество промежуточных вершин (ребер).

Задание

Определить наиболее производительный технологический маршрут, учитывая время на обработку (матрица назначений) и время на транспортировку заготовки роботом от станка к станку (матрица транспортировки). Технологический процесс изготовления включает пять операций (O_1, \dots, O_5). ГПС состоит из пяти станков (C_1, \dots, C_5).

Варианты заданий приведены в таблицах 12 и 13, выбираются по номеру студента в списке учебной группы.

Матрицы транспортировки представлены на рисунке 15, а матрицы назначений – рисунке 16.

Таблица 12 – Варианты заданий

Первая цифра номера	Номер матрицы транспортировки
0	1
1	2
2	3

Таблица 13 – Варианты заданий

Вторая цифра номера	Номер матрицы назначений
0	1
1	2
2	3
3	4
4	5
5	1
6	2
7	3
8	4
9	5

а)

	C ₁	C ₂	C ₃	C ₄	C ₅
C ₁	0	6	7	3	5
C ₂	7	0	11	9	3
C ₃	1	5	0	4	2
C ₄	3	8	4	0	10
C ₅	12	5	2	6	0

б)

	C ₁	C ₂	C ₃	C ₄	C ₅
C ₁	0	4	5	3	5
C ₂	5	0	7	5	4
C ₃	1	6	0	10	8
C ₄	4	8	6	0	3
C ₅	2	3	9	6	0

в)

	C ₁	C ₂	C ₃	C ₄	C ₅
C ₁	0	10	12	7	6
C ₂	3	0	11	9	4
C ₃	6	5	0	7	8
C ₄	7	9	8	0	15
C ₅	10	5	7	6	0

а – № 1; б – № 2; в – № 3

Рисунок 15 – Матрицы транспортировки

а)

	C ₁	C ₂	C ₃	C ₄	C ₅
O ₁	5		8		10
O ₂		6	7	12	
O ₃	4	2		7	3
O ₄	8		4	9	
O ₅		7			5

б)

	C ₁	C ₂	C ₃	C ₄	C ₅
O ₁		4	5		2
O ₂	1	8		6	
O ₃	7		4		10
O ₄		9	7	3	
O ₅	8			9	7

в)

	C ₁	C ₂	C ₃	C ₄	C ₅
O ₁	3			3	
O ₂		9	12		4
O ₃	8	6		10	6
O ₄		5	6	8	
O ₅	10		5		7

г)

	C ₁	C ₂	C ₃	C ₄	C ₅
O ₁	3		10		5
O ₂		4	9	2	
O ₃	5	2		6	3
O ₄	2		12	9	
O ₅		1			7

д)

	C ₁	C ₂	C ₃	C ₄	C ₅
O ₁		4	9	2	
O ₂	7		5	3	6
O ₃	5	6	8		
O ₄		3		8	10
O ₅	9		4		9

а – № 1; б – № 2; в – № 3; г – № 4; д – № 5

Рисунок 16 – Матрицы назначений



Порядок выполнения работы

- 1 Построить модельный граф.
- 2 С помощью алгоритма Дейкстры определить кратчайший путь от источника до стока, совпадающий с наиболее производительным путем. Рассчитать метки для всех вершин модельного графа.
- 3 Выделить путь, соответствующий наиболее производительному технологическому процессу.
- 4 Определить общие затраты времени на обработку.
- 5 Определить узкие места технологического процесса. Предложить меры по повышению производительности технологического маршрута.
- 6 Предложить и обосновать вариант технологического маршрута в случае отказа станка, на котором выполняется третья операция.

Содержание отчета

- 1 Название работы.
- 2 Цель работы.
- 3 Модельный граф.
- 4 Путь, соответствующий наиболее производительному технологическому процессу, а также затраты времени на обработку.
- 5 Выводы.

Контрольные вопросы

- 1 В чем заключается задача о кратчайшем пути?
- 2 Назначение алгоритма Дейкстры?
- 3 Какие метки используются в алгоритме Дейкстры?

Список литературы

- 1 **Муромцев, Д. И.** Введение в технологию экспертных систем / Д. И. Муромцев. – Санкт-Петербург : С.-Петербур. гос. ун-т ИТМО, 2005. – 220 с.
- 2 **Минский, М.** Фреймы для представления знаний : пер. с англ. / М. Минский. – Москва : Энергия, 1979. – 151 с.
- 3 **Гаврилова, Т. А.** Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – Санкт-Петербург : Питер, 2000. – 384 с.
- 4 **Усков, А. А.** Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика / А. А. Усков, А. В. Кузьмин. – Москва : Горячая линия – Телеком, 2004. – 143 с.
- 5 **Васильев, Ф. П.** Численные методы решения экстремальных задач / Ф. П. Васильев. – 2-е изд., перераб. и доп. – Москва : Наука, 1988. – 552 с.
- 6 **Жиглявский, А. А.** Методы поиска глобального экстремума / А. А. Жиглявский, А. Г. Жилинскас. – Москва : Наука, 1991. – 248 с.



7 **Рассел, С.** Искусственный интеллект : современный подход : пер. с англ. / С. Рассел, П. Норвиг. – 2-е изд. – Москва : Вильямс, 2006. – 1408 с.

8 **Мандель, И. Д.** Кластерный анализ / И. Д. Мандель. – Москва : Финансы и статистика, 1988. – 176 с.

9 **Жамбю, М.** Иерархический кластер-анализ и соответствия / М. Жамбю. – Москва : Финансы и статистика, 1988. – 345 с.

10 **Сидоркина, И. Г.** Системы искусственного интеллекта : учебное пособие / И. Г. Сидоркина. – Москва : Кнорус, 2011. – 248 с.

11 **Малкин, В. С.** Техническая диагностика : учебное пособие / В. С. Малкин. – Краснодар : Лань, 2013. – 272 с.

12 **Рудинский, И. Д.** Методы и технологии искусственного интеллекта / И. Д. Рудинский. – Москва : Горячая линия – Телеком, 2010. – 520 с.

13 **Пашкевич, В. М.** Функциональные семантические сети для обеспечения точности механической обработки / В. М. Пашкевич, М. Н. Миронова. – Могилев : Белорус.-Рос. ун-т, 2015. – 200 с.

14 **Поспелов, Г. С.** Искусственный интеллект – основа новой информационной технологии / Г. С. Поспелов. – Москва : Наука, 1988. – 280 с.

15 **Галкина, В. А.** Дискретная математика. Комбинаторная оптимизация на графах. Гл. 4 : Построение кратчайших путей в ориентированном графе / В. А. Галкина. – Москва : Гелиос АРВ, 2003. – 232 с.

16 **Боровская, Е. В.** Основы искусственного интеллекта / Е. В. Боровская, Н. А. Давыдова. – 3-е изд. – Москва : Лаборатория знаний, 2016. – 130 с.

17 **Сергеев, Н. Е.** Системы искусственного интеллекта : учебное пособие / Н. Е. Сергеев. – Таганрог : Южный фед. ун-т, 2016. – Ч. 1. – 118 с.

