

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**А. С. Потапов, О.В. Щербаков, И.Н. Жданов**  
**ТЕХНОЛОГИИ ИСКУССТВЕННОГО**  
**ИНТЕЛЛЕКТА:**  
**Учебно-методическое пособие по**  
**лабораторному практикуму**



**Санкт-Петербург**

**2013**

Потапов А.С., Щербаков О.В., Жданов И.Н. Технологии искусственного интеллекта: Учебно-методическое пособие по лабораторному практикуму. – СПб: НИУ ИТМО, 2013. – 35 с.

В учебно-методическом пособии предлагаются лабораторные работы, охватывающие основные понятия теории искусственного интеллекта. Особое внимание уделяется машинному обучению и распознаванию образов.

Предназначено для студентов, обучающихся по направлению подготовки 200700 – «Фотоника и оптоинформатика».

Рекомендовано советом факультета Фотоники и оптоинформатики НИУ ИТМО для использования в качестве учебно-методического пособия для студентов высших учебных заведений, обучающихся по направлению подготовки 200700 – «Фотоника и оптоинформатика».

В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена Программа развития государственного образовательного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» на 2009–2018 годы.



© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2013

© А. С. Потапов, 2013

© О.В. Щербаков 2013

© И.Н. Жданов 2013

# 1. Изучение классических методов поиска – градиентного спуска и моделирования отжига

Цель работы – ознакомиться с методами поиска в непрерывном пространстве состояний в случаях отсутствия и наличия вторичных минимумов. Данная работа имеет два варианта выполнения.

## Вариант 1

### Задание по работе:

1. Изучить теоретическую часть работы.
2. Реализовать методы градиентного спуска и моделирования отжига.
3. Для функций двух видов: вогнутой и с вторичными минимумами применить методы поиска, оценить скорость их сходимости и возможность нахождения глобального минимума.

### Теоретическая часть

#### Метод градиентного спуска

Метод градиентного спуска – это классический метод поиска минимума дифференцируемой функции с аргументами, принимающими вещественные значения. Данный метод, как правило, применяется для многомерных функций, поскольку в одномерном случае существуют более эффективные методы поиска.

Как известно, градиент некоторой функции  $f(\mathbf{x})$  в некоторой точке показывает направление локального наискорейшего увеличения функции. Этот факт используется в методах градиентного спуска (подъема).

Эти методы описываются следующей последовательностью действий:

1. Выбрать начальную точку  $\mathbf{x}_0 = (x_{1,0}, \dots, x_{n,0})$ . Установить номер итерации:  $i=0$ .
2. Для текущей точки определить значение градиента:

$$\nabla f(\mathbf{x}_i) = \left( \frac{\partial f}{\partial x_1}(\mathbf{x}_i), \dots, \frac{\partial f}{\partial x_n}(\mathbf{x}_i) \right). \quad (1)$$

В случае если градиент не может быть вычислен аналитически, его компоненты могут быть оценены:

$$\frac{\partial f}{\partial x_j}(\mathbf{x}_i) \approx \frac{f(x_{1,i}, \dots, x_{j-1,i}, x_{j,i} + \Delta x, x_{j+1,i}, \dots, x_{n,i}) - f(\mathbf{x}_i)}{\Delta x}. \quad (2)$$

3. Определить положение следующей точки:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - d \frac{\nabla f(\mathbf{x}_i)}{|\nabla f(\mathbf{x}_i)|}, \quad (3)$$

- где  $d$  – параметр, определяющий скорость спуска, и положить  $i=i+1$ .
4. Перейти к шагу 2, если не выполнен критерий останова.

Существует несколько способов ввода критерия останова. Самый простой – это наложить ограничение на количество итераций. Другие способы связаны с проверкой того, что текущая точка или значение функции  $f$  меняются мало. При фиксированном шаге  $d$  изменение положения текущей точки происходит всегда на одну и ту же величину. Однако в этом случае можно проверять изменение за несколько итераций и сравнивать с  $d$ :  $\frac{|\mathbf{x}_i - \mathbf{x}_{i-k}|}{kd}$ .

Существует также возможность адаптивного выбора шага  $d$ . Для этого на каждой итерации осуществляется выбор такого значения из  $d_0 = d, d_1 = d/w, d_2 = dw$  (где  $w$  – некоторый параметр, как правило  $w \in (1,2]$ ), что значение функции в точке  $\mathbf{x}_{i+1}^{(j)} = \mathbf{x}_i - d_j \frac{\nabla f(\mathbf{x}_i)}{|\nabla f(\mathbf{x}_i)|}$  минимально.

Таким образом, если при большом  $d$  метод градиентного спуска «проскакивает» минимум, то  $d$  будет уменьшаться. Уменьшение  $d$  ниже заданного порога также служит критерием останова. Напротив, на пологих участках значение  $d$  будет увеличиваться.

При условии существования глобального минимума функции  $f$  метод градиентного спуска обычно сходится (за исключением случаев, когда вдоль некоторого направления функция, монотонно убывая, стремится к некоторому конечному пределу при  $|\mathbf{x}| \rightarrow \infty$ ). Сходимость метода обеспечивается тем, что на каждой итерации выбирается такая точка  $\mathbf{x}_i$ , что  $f(\mathbf{x}_i) < f(\mathbf{x}_{i-1})$ . Метод, однако, не гарантирует нахождения глобального минимума, поскольку при достижении любого локального минимума метод не в состоянии определить направление на более глубокий минимум (и вообще обнаружить его существование) и останавливается в соответствии с выбранным критерием останова.

В связи с этим, выбор начальной точки может существенным образом сказываться на получаемом результате.

#### Метод моделирования отжига

Метод моделирования отжига предназначен для поиска глобального минимума некоторой функции  $f : S \rightarrow R$ , где  $S$  – некоторое пространство (необязательно непрерывное), элементы которого интерпретируются как состояния некоторой воображаемой физической системы, а значения самой функции – как энергия этой системы  $E=f(x)$  в состоянии  $x \in S$ .

В методе моделирования отжига система в каждый момент времени находится в некотором состоянии  $x_i$ , а также обладает некоторой температурой  $T$ , которая является управляемым параметром.

На каждой итерации система случайным образом переходит в новое состояние  $x_i \rightarrow x_{i+1}$ . Механизм выбора нового состояния состоит из двух частей:

1. Сначала выбирается  $x_{i+1}$  в соответствии с некоторой функцией распределения  $g(x_{i+1}, x_i, T)$ . Как правило, эта функция зависит только от расстояния  $|x_{i+1} - x_i|$ , причем с увеличением этого расстояния вероятность перехода понижается.
2. После случайного выбора  $x_{i+1}$  проверяется вероятность перехода в это новое состояние, исходя из разности энергий и текущей температуры:  $h(\Delta E, T)$ ,  $\Delta E = f(x_{i+1}) - f(x_i)$ . Здесь  $h(\Delta E, T)$  показывает вероятность перехода в состояние с другой энергией. Проверка производится следующим образом: выбрасывается случайное число из диапазона  $[0, 1]$ . Если это число оказывается меньше, чем значение вероятности  $h(\Delta E, T)$ , то новое состояние  $x_{i+1}$  принимается, в противном случае шаг 1 повторяется. Функция  $h(\Delta E, T)$ , как правило, стремится к 1 при  $\Delta E$ , стремящемся в минус бесконечность, и стремится к 0 при  $\Delta E$ , стремящемся в плюс бесконечность (то есть предпочтение в среднем отдается состояниям с меньшей энергией).

Поскольку метод моделирования отжига базируется на физических принципах, то и функции распределения вероятностей  $g(x_{i+1}, x_i, T)$  и  $h(\Delta E, T)$  также часто заимствуются из физики. В частности, достаточно популярен больцмановский отжиг, в котором:

$$g(x_{i+1}, x_i, T) = (2\pi T)^{-D/2} \exp(-|x_{i+1} - x_i|^2 / 2T), \quad (4)$$

где  $D$  – размерность пространства  $S$ ;

$$h(\Delta E, T) = \frac{1}{1 + \exp(\Delta E / T)} \approx -\Delta E / T. \quad (5)$$

Таким образом, температура  $T$  определяет, насколько в среднем может меняться текущее состояние  $x_i$ , а также то, насколько в среднем может меняться энергия системы при переходе в новое состояние.

Поскольку переход в состояния с меньшей энергией более вероятен, чем переход в состояния с более высокой энергией, то система будет больше времени проводить именно в низкоэнергетических состояниях.

Чтобы обеспечить сходимость системы к некоторому состоянию с наименьшей энергией, температуру системы понижают с переходом к следующей итерации. В больцмановском отжиге применяется следующий закон понижения температуры:

$$T_i = \frac{T_0}{\ln(1+i)}, \quad (6)$$

где номер итерации  $i > 0$ . Такой закон может, однако, потребовать большое число итераций, особенно при больших значениях начальной температуры  $T_0$ , в связи с чем используется более быстрое понижение температуры:

$$T_i = \frac{T_0}{1+i}. \quad (7)$$

Начальная температура неявно задает область, в которой будет осуществляться поиск глобального минимума, а также определяет необходимое для сходимости число итераций.

### Экспериментальная часть

В данной работе проводят сравнительный анализ методов градиентного спуска и моделирования отжига. Суть сравнительного анализа в данном случае заключается в построении графиков зависимости точности находимого решения от числа итераций для обоих методов и сопоставлении этих графиков для функций различных типов. Для этого необходимо последовательно выполнить следующие действия.

1. Реализовать методы градиентного спуска и моделирования отжига и описать детали выбранной реализации.

2. Для нескольких вогнутых (обладающих единственным минимумом) функций построить графики зависимости  $|\mathbf{x}_i - \mathbf{x}^*|$  (где  $\mathbf{x}^*$  - истинное положение глобального минимума) от номера итерации  $i$  для обоих методов. Выбранные для исследования функции должны различаться средним значением модуля градиента (крутизной). Следует обратить внимание на выбор начальной точки  $\mathbf{x}_0$ , которая должна отличаться от искомого минимума  $\mathbf{x}^*$ . В методе моделирования отжига следует также обратить внимание на задание начальной температуры  $T_0$  и ее влиянии на скорость сходимости.

3. Для нескольких функций с многими локальными минимумами определить условия сходимости обоих методов к глобальному минимуму. Для градиентного спуска определить, с какой вероятностью метод сходится к глобальному минимуму при случайном выборе (из некоторого фиксированного диапазона) начальной точки. Для метода моделирования отжига определить, какая начальная температура обеспечивает сходимость к глобальному минимуму. Для исследования следует брать функции с разным числом минимумов.

4. Проанализировать полученные результаты. Определить предпочтительность использования каждого из методов при поиске

минимума функции с одним и многими минимумами, опираясь на скорость сходимости и на легкость нахождения глобального минимума. Сделать выводы по работе.

## Литература

1. Назаров, А.В. **Нейросетевые алгоритмы прогнозирования и оптимизации систем** / А.В. Назаров, А.И. Лоскутов. – СПб.: Наука и Техника, 2003. – С. 237-238.

## Вопросы для самопроверки:

1. Какое условие сходимости метода градиентного спуска к глобальному минимуму?
2. Каков смысл параметра температуры в методе моделирования отжига?
3. На что влияет выбор начальной температуры в методе моделирования отжига?
4. Пусть в методе моделирования отжига используется закон уменьшения температуры  $T_i = T_0 / (1 + i)$  и пусть  $T_0 = 100$ . Сколько требуется итераций, чтобы обеспечить точность нахождения минимума  $10^{-4}$ .
5. Какое основное отличие методов моделирования отжига и градиентного спуска?
6. Как можно было бы совместно использовать два этих метода?

## Вариант 2

### Задание по работе:

1. Изучить теоретическую часть работы.
2. Реализовать генетический алгоритм поиска минимума.
3. Для функций двух видов: вогнутой и с вторичными минимумами применить генетические алгоритмы и оценить скорость сходимости и возможность нахождения глобального минимума в зависимости от размера начальной популяции и скорости мутаций.

## Теоретическая часть

### Генетические алгоритмы и эволюционные стратегии поиска

Генетические алгоритмы (ГА) предназначены для нахождения экстремумов функций от произвольных объектов, используя при этом приемы, заимствованные у естественной эволюции. Сами объекты трактуются как некоторые организмы, а оптимизируемая функция – как приспособленность организмов, или фитнес-функция. Множество

возможных объектов взаимно однозначно отображается на некоторое подмножество множества битовых строк (обычно фиксированной длины). Эти строки трактуются как хромосомы (геномы).

Особенность генетических алгоритмов заключается в том, что они работают с битовыми строками, не опираясь на структуру исходных объектов, что позволяет применять ГА без модификации для любых объектов. Единственно, что требуется для такого применения, – это перекодирование объектов в геномы.

Эволюционные стратегии отличаются от генетических алгоритмов лишь в том, что в первых используются структурированные описания объектов, то есть такие описания, элементы которых имеют вполне четкий смысл в той предметной области, к которой относятся данные объекты.

Двумя основными механизмами эволюции, наиболее часто моделируемыми в генетических алгоритмах и эволюционных стратегиях, являются скрещивание и мутации. При этом схема эволюционных методов поиска выглядит следующим образом.

1. Сгенерировать начальную популяцию (случайную совокупность объектов).
2. Выбрать родительские пары.
3. Для каждой родительской пары с использованием оператора скрещивания породить потомство.
4. В хромосомы порожденного потомства внести случайные искажения оператором мутации.
5. Произвести отбор особей из популяции по значению их фитнес-функции.
6. Повторять шаги 2-4, пока не выполнится критерий остановки.

Рассмотрим каждый из шагов чуть подробнее.

1. Генерация начальной популяции обычно производится равномерно по пространству генов (или по пространству описаний объектов). Размер популяции – установочный параметр.

2. Выбор родительских пар может осуществляться различными способами. Выбор родителей осуществляется в два этапа: выбор первого родителя и формирование пары. При выборе одного родителя обычно используется один из следующих способов:

- с равной вероятностью выбирается любая особь из имеющейся популяции;
- особь выбирается случайно с вероятностью, пропорциональной значению фитнес-функции; то есть в этом случае значение фитнес-функции сказывается не только на том, какие особи останутся в популяции в результате отбора, но и на то, сколько потомства они произведут.



Выбор второго родителя осуществляется по одному из следующих критериев:

- независимо от уже выбранного родителя (то есть второй родитель выбирается абсолютно так же, как и первый);
- на основе ближнего родства;
- на основе дальнего родства.

В последних двух случаях выбор одного родителя влияет на выбор другого родителя: с большей вероятностью формируются пары, состоящие из особей, которые больше похожи друг на друга (то есть ближе находятся в пространстве геномов или описаний объектов) в случае использования ближнего родства и меньше похожи – в случае дальнего родства. В генетических алгоритмах в качестве меры близости обычно используется расстояние Хемминга.

3. Оператор скрещивания – это оператор, который определяет, как из хромосом родителей формировать хромосомы их потомства. Часто применяется следующий оператор скрещивания: хромосомы делятся в некоторой случайной точке и обмениваются этими участками (то есть, все, что идет до этой точки, берется от одного родителя, а все, что после, – от другого). Это одноточечный кроссинговер. В многоточечном кроссинговере таких участков обмена больше.

При равномерном скрещивании каждый бит хромосомы берется от случайного родителя.

4. Мутации обычно осуществляются как случайная замена одного бита хромосомы. Скорость мутаций выражается в том, как часто они осуществляются. Это управляемый параметр, который влияет на скорость сходимости и вероятность попадания в локальный экстремум.

5. Отбор особей в новую популяцию чаще всего осуществляется одной из двух стратегий:

- пропорциональный отбор, при котором вероятность того, что некая особь останется в следующей популяции, пропорциональна значению фитнес-функции этой особи;
- элитный отбор, при котором из популяции отбираются лучшие по значению фитнес-функции особи, и они детерминированным образом переходят в следующую популяцию.

Формирование новой популяции может осуществляться как на основе потомков и родителей, так и на основе только потомков в зависимости от конкретной реализации.

6. Основные критерии останова базируются либо на числе сменившихся поколений (количестве выполненных итераций), либо на некотором условии стабильности популяции. Число поколений не является адаптивным по отношению к виду фитнес-функции, поэтому используется обычно в качестве не основного критерия. Проверка стабильности популяции в общем виде, как правило, требует значительных

вычислений, поэтому чаще используется проверка того, что максимальное по популяции значение фитнес-функции перестает заметно расти от поколения к поколению. Все эти критерии останова соответствуют критериям останова в методе градиентного спуска, но учитывают ту специфику генетических алгоритмов, что в них на каждой итерации одновременно рассматривается не одно, а несколько решений.

### Особенности реализации генетических операторов в эволюционных стратегиях

Рассмотрим особенности реализации генетических операторов в эволюционных стратегиях на примере объектов, описаниями которых являются двухкомпонентные векторы:  $(x, y)$ .

1. Генерация начальной популяции может осуществляться путем выбора случайных векторов из области  $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ , где величины  $x_{\min}, x_{\max}, y_{\min}, y_{\max}$  задают ожидаемые минимальные и максимальные значения переменных  $x$  и  $y$  искомого положения экстремума фитнес-функции.

2. При выборе родителей особенность эволюционных стратегий выражается в способе задания меры родства. В данном случае, мерой родства двух особей  $(x_1, y_1)$  и  $(x_2, y_2)$  может служить евклидово расстояние:  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ .

3. Результатом скрещивания двух особей в рассматриваемом случае будет являться особь  $((\epsilon x_1 + (1 - \epsilon)x_2, \epsilon y_1 + (1 - \epsilon)y_2))$ , где  $\epsilon \in [0, 1]$  – случайная величина.

4. Результатом мутации для особи  $(x, y)$  будет являться особь  $(x + \delta_x, y + \delta_y)$ , где  $\delta_x, \delta_y$  – случайные величины. Их распределение вероятностей может быть выбрано гауссовым или, для простоты программной реализации, равномерным в некотором интервале. Дисперсия этих величин определяет скорость мутаций.

5 и 6. Операторы отбора и критерии останова в эволюционных стратегиях не имеют особых отличий от тех, которые используются в генетических алгоритмах.

## **Экспериментальная часть**

В данной работе проводят анализ зависимости характеристик работы генетических алгоритмов или эволюционных стратегий (скорости их сходимости и возможности обнаружения глобального экстремума фитнес-функции) от установочных параметров – размера начальной популяции и скорости мутаций, а также от вида оптимизируемой функции.

Для этого необходимо выполнить следующую последовательность действий.

1. Выполнить реализацию генетического алгоритма или эволюционной стратегии и описать ее детали.

2. Для нескольких выпуклых (обладающих единственным максимумом) функций построить графики зависимости  $\sqrt{(x_m - x^*)^2 + (y_m - y^*)^2}$  (где  $(x^*, y^*)$  – истинное положение глобального экстремума, а  $(x_m, y_m)$  – максимальное по текущей популяции значение фитнес-функции) от номера популяции  $i$  при различных значениях скорости мутации и размера начальной популяции. Следует обратить внимание на выбор диапазона  $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ . Графики необходимо построить для двух случаев: когда точка  $(x^*, y^*)$  попадает в этот диапазон и когда она находится вне него.

3. Для нескольких функций с многими локальными экстремумами провести те же исследования, что и для функций с единственным экстремумом.

4. Проанализировать полученные результаты. Определить, какое именно влияние оказывает скорость мутаций на функционирование алгоритма поиска. Определить, может ли большая скорость мутаций быть заменена большим размером популяции. Установить различия в характере работы алгоритма поиска для функций с единственным и многими экстремумами. Сделать выводы по работе.

## Литература

1. Назаров, А.В. **Нейросетевые алгоритмы прогнозирования и оптимизации систем** / А.В. Назаров, А.И. Лоскутов. – СПб.: Наука и Техника, 2003. – С. 254-281.
2. Люгер, Д.Ф. **Искусственный интеллект: стратегии и методы решения сложных проблем** / Д.Ф. Люгер. – 4-е изд.: Пер. с англ. – М.: Изд. дом “Вильямс”, 2003. – С. 483-516.

## Вопросы для самопроверки:

1. В чем основное отличие генетических алгоритмов от эволюционных стратегий?
2. Из каких основных шагов состоят генетические алгоритмы?
3. Как влияет скорость мутаций на скорость сходимости алгоритмов поиска?
4. При использовании какого типа родства: ближнего или дальнего, – скорость сходимости алгоритма будет больше?

5. Если при использовании эволюционных стратегий область задания начальной популяции  $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$  не содержит искомого экстремума, то благодаря какому из генетических операторов этот экстремум все же может быть найден? Или при таких условиях он не может быть найден в принципе?
6. Благодаря какому генетическому оператору эволюционные методы существенно отличаются от градиентного спуска, и в чем именно заключается это отличие?

## 2. Методы построения ассоциативных сетей

Цель работы – ознакомиться с методами извлечения ассоциативных связей между понятиями на основе психофизических экспериментов, а также с методами построения баз знаний, включающих понятия и взаимосвязи между ними, которые описаны в рамках логических представлений. Данная работа имеет два варианта выполнения.

### Вариант 1

#### Задание по работе:

1. Изучить теоретическую часть работы.
2. Реализовать программу проведения тестирования в некоторой предметной области для выявления ассоциативных связей между понятиями.
3. Провести тестирование группы людей. Усреднить и проанализировать интервалы времени, необходимые испытуемым для установления наличия или отсутствия ошибок в утверждениях, связывающих два понятия, на основе чего определить длину пути в ассоциативной сети между соответствующими узлами. Восстановить ассоциативную сеть.

### Теоретическая часть

#### Описание методики построения ассоциативных сетей

Очень давно было замечено, что человек имеет склонность к ассоциированию. Это послужило отправной точкой для создания ассоцианистских теорий, в которых смысл некоторого понятия определяется через его ассоциативные связи с другими понятиями, которые в совокупности образуют своего рода сеть. Понятия являются основой нашего знания о мире, поэтому такую ассоциативную сеть можно рассматривать в качестве представления знаний. Ассоциативные связи возникают на основе опыта и выражают эмпирические отношения между признаками или поведением объектов.

Например, на основании опыта мы ассоциируем понятие «снег» с другими понятиями, такими как «зима», «холод», «белый», «лед» и т.д. Наши знания о снеге и истинность утверждений типа «снег белый» выражаются в виде сети ассоциаций. Если такая сеть реально используется человеком, то можно ли узнать, какова она?

Для ответа на этот вопрос психологами проводится следующий эксперимент. Людям задают вопросы из некоторой области, например, об особенностях и поведении птиц. Эти вопросы выбираются очень простыми: «Может ли голубь летать?», «Может ли канарейка петь?», «Является ли ворона птицей?».

При этом проверяется, конечно же, не правильность ответов на вопросы. Вместо этого оценивается время, требуемое для ответа. Все вопросы составляются так, что в них фигурируют два понятия. Наличие непосредственной ассоциации между этими понятиями должно было сказываться на времени ответа.

Как оказывается, времена ответов действительно различаются. Так, например, для ответа на вопрос «Может ли канарейка летать?» требуется больше времени, чем для ответа на вопрос: «Может ли канарейка петь?».

Коллинс и Квиллиан, ставившие этот эксперимент впервые, объясняли большее время ответа на некоторый вопрос тем, что участвующие в вопросе понятия, хотя и расположены достаточно близко в ассоциативной сети, не являются непосредственно связанными. При этом оказалось, что свойства объектов запоминаются людьми на наиболее абстрактном уровне.

Вместо того чтобы запоминать каждое свойство для каждой птицы (канарейки летают, вороны летают, голуби летают), люди хранят информацию о том, что канарейки – птицы, а птицы, как правило, способны летать. Поскольку поют не все птицы, то способность к пению – более частное свойство, которое запоминается на менее абстрактном уровне, чем способность летать, поэтому и ответ на вопрос «Может ли канарейка петь?» требует меньше времени, так же, как и ответ на вопрос «Является ли канарейка желтой?». Таким образом, быстрее всего вспоминаются самые конкретные свойства объектов. Как оказалось, и исключения из правил также запоминаются на самом нижнем, детальном уровне. Примером может служить вопрос «Может ли страус летать?», который для ответа требует меньше времени, чем тот же вопрос о канарейке или другой «нормальной» птице.

На рис. 1 представлен пример фрагмента ассоциативной сети, которая могла бы быть построена в результате такого эксперимента. В рамках данной работы предполагается, что в областях, относящихся к здравому смыслу, ассоциативные связи у разных людей устанавливаются сходным образом, поэтому для получения надежных значений времен ответов можно и нужно проводить усреднение по некоторой группе людей. При этом выпор сферы понятий для тестирования нужно выбирать таким

образом, чтобы ответы на составленные вопросы были очевидными для всех испытуемых. При малом числе испытуемых может использоваться повторное тестирование одного и того же человека, но необходимо, чтобы были выполнены следующие условия: база вопросов является достаточно большой (порядка 100 вопросов) и время перед повторным тестированием должно быть значительным (не менее суток).

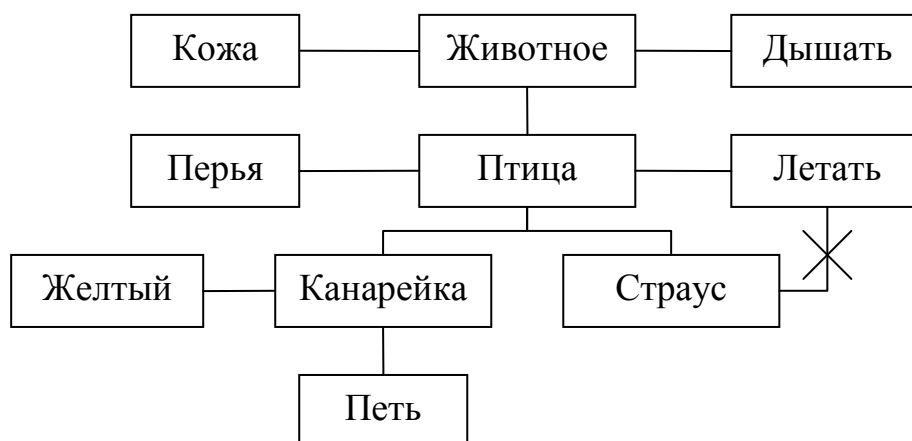


Рис. 1. Пример фрагмента ассоциативной сети

### Экспериментальная часть

В данной работе производят построение ассоциативной сети для некоторой ограниченной системы общеизвестных понятий на основе времени ответа человека на вопросы, связывающие пары понятий. Для этого необходимо выполнить следующую последовательность действий.

1. Реализовать программу тестирования, которая выбирает из базы случайные, но неповторяющиеся вопросы и определяет с высокой точностью (порядка 0.1 секунды) время ответа. Ввод ответа рекомендуется осуществлять по факту нажатия одной из двух заранее определенных клавиш. Не рекомендуется использовать способы ввода, требующие перемещения мышки или ввода слов с нажатием Enter, поскольку это существенно снижает точность оценки времени. Показ очередного вопроса рекомендуется осуществлять только при готовности человека (о чем он может сообщать программе путем нажатия некоторой третьей клавиши). Рекомендуется, чтобы программа автоматически записывала результаты ответов в лог-файл.

2. Составить базу вопросов. Для этого следует выбрать область со сложными (иерархическими) отношениями между понятиями, которые предположительно описываются ассоциативной сетью, подобной представленной на рис. 1. Однако сами понятия должны быть общеизвестными, а ответы на вопросы – не требующими привлечения дополнительных знаний.

3. Провести тестирование группы людей. При этом требуется установить предпочтительную продолжительность тестирования одного человека. Для этого необходимо определить, спустя какой промежуток времени начинает систематически увеличиваться время ответов или число ошибок. При тестировании каждого человека требуется объяснить ему условия эксперимента и продемонстрировать их на нескольких вопросах, которые затем не учитываются при анализе результатов.

4. Произвести усреднение времен ответов на вопросы по группе людей и вычислить дисперсии времен для каждого вопроса. Установить достоверность различия средних времен ответов на разные вопросы.

5. Построить ассоциативную сеть, если времена ответов на вопросы достоверно различаются. Построение сети рекомендуется начинать с пар понятий, которым соответствуют наименьшие времена ответов. Эти понятия должны быть непосредственно связаны в сети.

6. Проанализировать полученные результаты. Если построение ассоциативной сети осуществлено удачно, то проанализировать ее структуру, в противном случае определить возможные причины неудачи. Сделать выводы по работе.

## Литература

1. Люгер, Д.Ф. **Искусственный интеллект: стратегии и методы решения сложных проблем** / Д.Ф. Люгер. – 4-е изд.: Пер. с англ. – М.: Изд. дом “Вильямс”, 2003. – С. 226-230.

## Вопросы для самопроверки:

1. На основе какого принципа формируются тестовые вопросы, используемые при построении ассоциативной сети?
2. Какой критерий используется при определении «ассоциативного расстояния» между понятиями?
3. Почему не все пары понятий, имеющих отношение друг к другу, в ассоциативной сети связаны непосредственно?
4. Насколько корректно производить усреднение результатов (времени ответов), полученным при тестировании разных людей?
5. Какие группы понятий предпочтительнее использовать при построении ассоциативной сети и почему? Если требуется построить ассоциативную сеть для специальных понятий, как следует подбирать тестируемых людей? Что означает различие структуры ассоциативной сети, построенной по разным группам людей?
6. Какие ограничения имеет представление знаний в форме ассоциативных сетей, и как эти ограничения могут быть сняты в рамках других представлений?

## Вариант 2

### Задание по работе:

1. Изучить теоретическую часть работы.
2. Выбрать систему понятий из некоторой предметной области и описать взаимосвязи между этими понятиями.
3. Реализовать программу на языке Пролог, в которой установленные взаимосвязи описываются логическими формулами в исчислении предикатов в виде общих правил и частных фактов. Определить возможности интерпретатора Пролога по ответу на вопросы из данной области.

### Теоретическая часть

#### Краткое описание основ языка Пролог

Программа на языке Пролог преимущественно состоит из списка логических утверждений, которые можно разделить на факты и общие правила. Эти утверждения состояются из имен предикатов, переменных и их значений, которые представляют собой символьную строку, начинающуюся с буквы, а также совокупности логических операций. В конце каждого утверждения ставится точка ".". Переменные обязательно начинаются с прописной буквы, поэтому все значения должны начинаться со строчной буквы. Используемые в Прологе обозначения логических операций представлены в табл. 1.

Табл. 1. Логические операции в Прологе

Имя операции	Логическая операция	Обозначение в Прологе
и	$\wedge$	,
или	$\vee$	;
если	$\leftarrow$	:-
не	$\neg$	not

Рассмотрим некоторые примеры утверждений на языке Пролог. Ниже приведен пример списка фактов, не содержащих переменных.

man(serg).

man(alex).

father(serg, alex).

mother(kat, serg).

Эти факты говорят, что предикаты "man", "father", "mother" истинны при указанных значениях аргументов. Пролог работает в рамках предположения о замкнутости базы знаний. Это означает то, что все



предикаты будут ложны для всех значений переменных, для которых не было указано обратное.

Общие правила в Прологе записываются так же, как и факты, но в них присутствуют переменные. При этом подразумевается, что общее правило верно при всех значениях всех входящих в него переменных. Иными словами, правило

$$p_1(X) :- p_2(X, Y).$$

означает логическое выражение

$$(\forall X, Y) p_2(X, Y) \Rightarrow p_1(X).$$

Рассмотрим некоторые примеры общих фактов, записанных на языке Пролог, и приведем для них соответствующие формулы в логике предикатов.

$father(X, Y) :- man(X), parent(X, Y).$

$$(\forall X, Y) man(X) \wedge parent(X, Y) \Rightarrow father(X, Y)$$

Если некто  $X$  является мужчиной и родителем  $Y$ , то  $X$  – отец  $Y$ .

$parent(X, Y) :- mother(X, Y); father(X, Y).$

$$(\forall X, Y) mother(X, Y) \vee father(X, Y) \Rightarrow parent(X, Y)$$

$X$  является родителем  $Y$ , если  $X$  – отец или мать  $Y$ .

$grandfather(X, Y) :- father(X, Z), parent(Z, Y).$

$$(\forall X, Y) (grandfather(X, Y) \Leftarrow (\exists Z) father(X, Z) \wedge parent(Z, Y))$$

$X$  является дедушкой  $Y$  только тогда, когда  $X$  является отцом (некоторого) родителя  $Y$ .

За исполнение программы на языке Пролог отвечает интерпретатор, который интерактивно взаимодействует с пользователем, отвечая на его запросы. Запрос к интерпретатору также представляется в виде логического выражения, истинность которого требуется установить. Рассмотрим следующую простую программу.

$mother(X, Y) :- woman(X), parent(X, Y).$

$father(X, Y) :- man(X), parent(X, Y).$

$grandfather(X, Y) :- father(X, Z), parent(Z, Y).$

$man(serg).$

$man(alex).$

$woman(kat).$

$parent(serg, victor).$

$parent(kat, victor).$

$parent(victor, alex).$

И посмотрим на следующие запросы, начинающиеся с приглашения "?-", на которые интерпретатор возвращает некоторый ответ.

?-  $father(serg, victor).$

Yes

?-  $grandfather(serg, alex).$

Yes  
?- mother(kat, alex).  
No  
?- father(victor, alex).  
No

Как видно, для обработки этих запросов интерпретатору необходимо выполнить логический вывод, а не просто обратиться к базе правил. Также видно, как действует предположение о замкнутости базы знаний: поскольку в базе нет фактов об истинности `parent(kat, alex)` и `man(victor)`, то последние два запроса возвращают "Нет".

В Прологе существуют также и запросы в виде выражений с переменными. Посмотрим, как работают эти запросы.

?- mother(kat, X).  
X = victor  
Yes  
?- grandfather(X, alex).  
X = serg  
Yes  
?- grandfather(X, Y).  
X = serg, Y = alex  
Yes  
?- grandfather(X, kat).  
No  
?- man(X).  
X = serg  
Yes

Здесь можно отметить следующее. В подобного рода запросах предполагается квантор существования для входящих в него переменных. Выражение истинно, если найдется хотя бы одно подходящее значение X (не имеет значения, как именно обозначать переменные в запросе). В действительности, при нахождении в базе первого значения, для которого введенное выражение истинно, интерпретатор спрашивает пользователя, продолжать ли поиск. Поиск продолжается при нажатии команды ";" (или) до тех пор, пока не будет дан ответ "Нет" в связи с исчерпанием всех возможностей, например,

?- man(X).  
X = serg ;  
X = alex ;  
No

Более сложные выражения в запросах также допустимы, например,

?- mother(kat, X), father(Y, X).  
X = victor, Y = serg  
Yes

Этот запрос позволяет определить, кто является отцом сына kat.

Следует обратить внимание на некорректность работы программы, содержащей противоречия вида

$\text{something}(X) :- \text{not}(\text{something}(X)).$

а также циклические определения вида

$\text{man}(X) :- \text{not}(\text{woman}(X)).$

$\text{woman}(X) :- \text{not}(\text{man}(X)).$

### Экспериментальная часть

В данной работе производят построение простейшей базы знаний в рамках логического представления с использованием языка Пролог и устанавливают возможности и ограничения: 1) логических представлений при описании понятий и взаимосвязей между ними; 2) интерпретатора отвечать на запросы. Для этого необходимо выполнить следующую последовательность действий.

1. Выбрать систему взаимосвязанных понятий, подобную системе родственных отношений, но отличную от нее.

2. Формализовать отношения между понятиями в форме логических выражений в рамках исчисления предикатов первого порядка.

3. Реализовать логические выражения как общие правила в программе на языке Пролог; дополнить программу совокупностью частных фактов. Полученная в результате программа является основой отчета по лабораторной работе.

4. Определить возможности созданной базы знаний по ответу на запросы, требующие логического вывода (т.е. ответы на которые не содержатся в базе в явном виде). Установить запросы, на которые ответ интерпретатора не соответствует ожидаемому (уделить внимание неявному предположению замкнутости базы знаний).

5. Проанализировать полученные результаты. Сделать выводы по работе: сформулировать эмпирически обоснованные преимущества и недостатки логических представлений для систем взаимосвязанных понятий.

### Литература

1. Люгер, Д.Ф. **Искусственный интеллект: стратегии и методы решения сложных проблем** / Д.Ф. Люгер. – 4-е изд.: Пер. с англ. – М.: Изд. дом “Вильямс”, 2003. – С. 609-684.
2. Джексон, П. **Введение в экспертные системы: учеб. пособие** / П. Джексон. – Пер. с англ. – М.: Изд. дом “Вильямс”, 2001. – С. 177-200.

### Вопросы для самопроверки:

1. К какому типу представлений знаний можно отнести язык Пролог?
2. Из каких базовых элементов состоит программа на языке Пролог?
3. Что означает высказывание, согласно которому программирование на языке Пролог является декларативным?
4. Имеет ли значение порядок, в котором в программе на Прологе заданы правила и факты?
5. Как реализуется предположение о замкнутости базы знаний в языке Пролог? Какие следствия из этого можно сделать?
6. В чем заключается функция интерпретатора языка Пролог?
7. Какие понятия или взаимосвязи между понятиями затруднительно представить на языке Пролог? Какие преимущества и недостатки Пролог имеет по сравнению с другими языками программирования и с другими представлениями знаний?

### **3. Исследование влияния параметров обучающей выборки на вероятность распознавания новых образов**

Цель работы – ознакомиться с методами распознавания образов и освоить их применение в различных условиях, определяемых характером обучающей выборки: ее размером, наличием в ней выбросов, степенью перекрытия классов. Данная работа имеет два варианта выполнения.

#### **Вариант 1**

##### **Задание по работе:**

1. Изучить теоретическую часть работы.
2. Реализовать методы эталонных образов и ближайшего соседа.
3. Путем варьирования обучающей выборки определить влияние следующих факторов на вероятности правильного распознавания: наличие в обучающей выборке выбросов, размер обучающей выборки, форма областей, занимаемых классами в пространстве признаков.

##### **Теоретическая часть**

###### *Постановка задачи дискриминантного распознавания образов*

В распознавании образов рассматривается вопрос о разделении объектов на классы. При этом требуется по некоторому описанию объекта (такое описание называется образом) определить его отношение к тому или иному классу. В зависимости от способа описания объектов различают дискриминантные, логические и синтаксические методы распознавания. В

данной лабораторной работе изучаются простейшие дискриминантные методы.

В дискриминантном распознавании образов объекты представляются в виде векторов. Каждый из компонентов этих векторов рассматривается как признак с вещественным значением, а сам вектор называется вектором признаков.

Решающим правилом называется правило, которое по вектору признаков, описывающих объект, позволяет определить соответствующий данному объекту класс. Задача распознавания образов заключается в восстановлении решающего правила по совокупности примеров – обучающей выборке, состоящей из векторов признаков, для каждого из которых задан также соответствующий класс.

Итак, пусть обучающая выборка состоит из  $M$  пар  $\mathbf{x}_i, \alpha_i$ , где  $\mathbf{x}_i$  – образ объекта (вектор признаков), а  $\alpha_i$  обозначает класс, к которому данный объект принадлежит. На основе этой обучающей выборки требуется построить решающее правило  $\varphi(\mathbf{x})$ , которое для произвольного образа  $\mathbf{x}$  будет определять номер наиболее подходящего класса.

В данной работе рассматривается случай двух классов.

#### Метод эталонных образов

Метод эталонных образов – это один из эвристических методов построения решающих правил. В основу этого метода положена идея, которая заключается в том, что некоторая совокупность объектов, объединенных в отдельный класс, может быть представлена одним или несколькими эталонными объектами. Эти эталонные объекты являются наиболее типичными представителями класса. Типичность эталонного объекта означает, что он в среднем максимально похож на все объекты класса.

Поскольку сходство двух объектов может трактоваться как величина, противоположная расстоянию между ними, то эталон – это объект, для которого минимально среднее расстояние до других объектов.

Пусть в обучающей выборке первому классу соответствует  $M_1$  элементов  $\mathbf{x}_{1,i}$ , а второму классу –  $M_2$  элементов  $\mathbf{x}_{2,i}$ . Тогда эталонные образы для каждого из классов могут быть определены как

$$\mathbf{x}_{0,1} = \frac{1}{M_1} \sum_{i=1}^{M_1} \mathbf{x}_{1,i}, \quad \mathbf{x}_{0,2} = \frac{1}{M_2} \sum_{i=1}^{M_2} \mathbf{x}_{2,i}. \quad (8)$$

Классы, однако, могут обладать разными свойствами. Простейшим свойством является характерный размер класса, который вычисляется как

$$r_1 = \sqrt{\frac{1}{M_1} \sum_{i=1}^{M_1} |\mathbf{x}_{0,1} - \mathbf{x}_{1,i}|^2}, \quad r_2 = \sqrt{\frac{1}{M_2} \sum_{i=1}^{M_2} |\mathbf{x}_{0,2} - \mathbf{x}_{2,i}|^2}. \quad (9)$$

Тогда для классификации нового образа  $\mathbf{x}$  используется следующая решающая функция:

$$k(\mathbf{x}) = \frac{|\mathbf{x} - \mathbf{x}_{0,1}|}{r_1} - \frac{|\mathbf{x} - \mathbf{x}_{0,2}|}{r_2}. \quad (10)$$

Если значение этой функции отрицательное, то образ относится к первому классу, в противном случае – ко второму. Разделяющая поверхность для двух классов задается уравнением  $k(\mathbf{x}) = 0$ .

#### Метод ближайшего соседа

Другой широко распространенный эвристический метод распознавания – метод ближайшего соседа (или его обобщение – метод  $k$ -ближайших соседей).

Идея этого метода крайне проста: новый образ относится к тому классу, к которому он ближе. При этом расстояние от образа до класса определяется как расстояние от образа до ближайшего элемента класса.

Тогда на основе обучающей выборки  $\mathbf{x}_i, \alpha_i, i=1, \dots, M$ , может быть построено следующее решающее правило:

$$\Phi(\mathbf{x}) = \alpha_{\arg \min_i |\mathbf{x} - \mathbf{x}_i|}. \quad (11)$$

В соответствии с данным решающим правилом просматривается вся обучающая выборка, в ней находится образ, расположенный наиболее близко к данному и устанавливается, к какому классу он принадлежит (это известно, поскольку он находится в обучающей выборке). Этот класс и приписывается новому образу.

### **Экспериментальная часть**

В данной работе проводят сравнительный анализ метода эталонных образов и метода ближайшего соседа. При этом основными характеристиками являются вероятность распознавания и скорость работы в зависимости от параметров обучающей выборки: ее размеров, наличия в ней выбросов, формы областей, занимаемых классами. Для этого необходимо выполнить следующую последовательность действий.

1. Выполнить реализацию методов эталонных образов и ближайшего соседа.

2. Сформировать выборку образов, которую разделить на обучающую и тестовую часть. Образы из тестовой выборки не участвуют в обучении, а используются затем для определения процента правильных ответов, даваемых тем или иным методом. Разделение на обучающую и тестовую выборку желательно производить случайным образом. Размер тестовой выборки должен быть достаточно большим (по крайней мере, несколько десятков элементов).

3. Варьируя обучающую выборку, определить проценты правильного распознавания каждого из методов. Обучающую выборку следует изменять следующим образом: добавлять и исключать из нее элементы, чтобы менялся размер выборки, вносить в обучающую выборку ошибки (для нескольких элементов указывать неправильный класс).

4. Для определения скорости работы каждого из методов следует варьировать размер обучающей выборки. Для формирования обучающих выборок больших размеров допустимо многократно дублировать содержимое исходной обучающей выборки. Для определения времени классификации нового образа следует многократно (в цикле) вызывать классифицирующую процедуру и оценивать время для такого многократного вызова, после чего делить полученное общее время на число вызовов.

5. Проанализировать полученные результаты. Определить, как влияют ошибки в обучающей выборке на каждый из методов, при каких размерах обучающей выборки у какого из методов больше процент правильного распознавания (и при какой форме областей, занимаемых классами), как влияет размер обучающей выборки на время классификации нового образа в каждом из методов. Сделать выводы по работе.

### Литература

1. **Потапов, А.С. Распознавание образов и машинное восприятие: общий подход на основе принципа минимальной длины описания / А.С. Потапов. – СПб.: Политехника, 2007. – С. 135-138, 152-155.**
2. **Ту, Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес – М.: Мир, 1978. – С. 90-98.**

### Вопросы для самопроверки:

1. К какому типу методов распознавания относятся методы ближайшего соседа и эталонных образов, и что, на ваш взгляд, это означает?
2. Какова форма разделяющей поверхности в методе эталонных образов?
3. Какова форма разделяющей поверхности в методе ближайшего соседа?
4. Работа какого из этих двух методов будет в большей степени нарушена, если пространство признаков сильно растянуть в направлении одного из признаков, оставив остальные направления неизменными?
5. В каком из двух методов время классификации нового образа зависит от размера обучающей выборки?
6. Какой из двух методов более чувствителен к ошибкам в обучающей выборке?

### Вариант 2

### Задание по работе:

1. Изучить теоретическую часть работы.
2. Реализовать метод решающих функций.
3. Путем варьирования обучающей выборки определить влияние следующих факторов на вероятности правильного распознавания: наличие в обучающей выборке выбросов, размер обучающей выборки, форма областей, занимаемых классами в пространстве признаков.

### Теоретическая часть

Постановку задачи дискриминантного распознавания образов см. в теоретической части первого варианта выполнения работы. Здесь будет рассмотрен только сам метод решающих функций.

*Решающей функцией*  $k(\mathbf{x})$  для двух классов  $a_1, a_2 \in A$  называется такая функция  $k: X \rightarrow R$ , что  $k(\mathbf{x}) > 0$ , если образ  $\mathbf{x}$  принадлежит классу  $a_1$ , и  $k(\mathbf{x}) < 0$ , если образ  $\mathbf{x}$  принадлежит классу  $a_2$ .

Обычно рассматриваются не произвольные решающие функции, а лишь функции, относящиеся к некоторому параметрическому семейству, элементы которого  $k(\mathbf{x}, \mathbf{w})$  определяются вектором параметров  $\mathbf{w} = (w_1, \dots, w_n)$ , где  $n$  – количество параметров. Выбор конкретных значений параметров  $w_i$  соответствует выбору решающей функции. Тогда решение задачи распознавания образов сводится к определению оптимальных значений параметров по обучающей выборке.

Мы рассматриваем задачу построения решающих функций для случая двух классов  $a_1$  и  $a_2$ . Как и раньше, в задаче распознавания имеются исходные данные:  $D = ((\mathbf{x}_1, A_1), (\mathbf{x}_2, A_2), \dots, (\mathbf{x}_M, A_M))$  – обучающая выборка из  $M$  элементов, в которой  $\mathbf{x}_i \in R^N$  – образы, представленные  $N$ -компонентными векторами признаков, а  $A_i \in \{a_1, a_2\}$  – соответствующие им классы.

Важное параметрическое семейство составляют *линейные решающие функции*. Поиск линейных решающих функций проще как с теоретической, так и с практической точки зрения, а классификаторы, построенные на их основе, являются также и наиболее эффективными по отношению к требуемым вычислительным ресурсам. Линейные решающие функции задаются следующим образом:

$$k(\mathbf{x}, \mathbf{w}) = w_1 x_1 + w_2 x_2 + \dots + w_N x_N + w_{N+1} = \mathbf{w} \mathbf{x}', \quad (12)$$

где  $\mathbf{x}' = (x_1, x_2, \dots, x_N, 1)^T$  – дополненный вектор признаков, а  $\mathbf{w} = (w_1, w_2, \dots, w_{N+1})$  – вектор весов, который требуется определить по



обучающей выборке, исходя из условий:  $\mathbf{w}\mathbf{x}'_i > 0$ , если  $A_i = a_1$  и  $\mathbf{w}\mathbf{x}'_i < 0$ , если  $A_i = a_2$ . Разделяющая поверхность, задаваемая уравнением  $\mathbf{w}\mathbf{x}'_i = 0$ , в данном случае будет гиперплоскостью.

Чтобы представить эти условия единообразно, обычно пользуются следующим приемом. Пусть  $z_i = 1$ , если  $A_i = a_1$ , и  $z_i = -1$ , если  $A_i = a_2$ , тогда ограничения на вектор параметров будут следующие:

$$(\forall i) z_i \mathbf{w}\mathbf{x}'_i > 0. \quad (13)$$

В зависимости от критерия качества и метода поиска параметров, максимизирующих этот критерий, могут быть построены различные процедуры нахождения линейных решающих функций.

Одной из идей здесь является применение классического метода наименьших квадратов (МНК). Необходимо, чтобы решающая функция правильно классифицировала образы обучающей выборки. Это можно выразить в виде следующего условия:  $\kappa(\mathbf{x}_i, \mathbf{w}) = z_i$ . Тогда задача распознавания сводится к задаче аппроксимации, для которой в рамках МНК можно записать следующую целевую функцию:

$$L = \frac{1}{M} \sum_{i=1}^M (\kappa(\mathbf{x}_i, \mathbf{w}) - z_i)^2. \quad (14)$$

Удобство линейных решающих функций в том, что для их нахождения используется линейный МНК, имеющий эффективное решение. Для определения значений параметров, при которых достигается минимум критерия (14), продифференцируем его и получим систему линейных уравнений:

$$\frac{\partial L}{\partial w_k} = \frac{2}{M} \sum_{i=1}^M \left( \sum_{j=1}^{N+1} w_j x_{i,j} - z_i \right) x_k = 0, \quad (15)$$

которую не представляет сложности решить.

Не любые два набора точек в  $R^N$  разделяются гиперплоскостью, задаваемой линейной решающей функцией, а значит, не все образы в таких наборах могут быть корректно классифицированы с помощью линейной решающей функции. Это является платой за простоту и вычислительную эффективность линейных методов. Нелинейные методы сложны и вычислительно трудоемки. К счастью, существует стандартный прием, позволяющий расширять процедуры построения линейных решающих функций на нелинейные функции. Этот прием заключается в том, что вводятся *обобщенные решающие функции* вида

$$\kappa(\mathbf{x}, \mathbf{w}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \dots + w_n f_n(\mathbf{x}), f_i : R^N \rightarrow R. \quad (16)$$

В частности, несложно получить линейные решающие функции, используя  $n = N + 1$  и  $f_i(\mathbf{x}) = x'_i$ .

Функции  $f_i(\mathbf{x})$  предполагаются известными заранее, то есть имеется возможность однозначно получить их значения для любого вектора  $\mathbf{x}$ . Тогда решающие функции вида (16) оказываются линейными по неизвестным параметрам  $w_i$ , и несложно убедиться, что любой метод, предназначенный для нахождения параметров линейных решающих функций, также будет работать и для обобщенных решающих функций.

Один из стандартных способов задания обобщенных решающих функций – это представление их в виде многочленов (при этом, обычно используются ортонормированные системы функций, например, многочлены Лежандра или Эрмита). При этом, однако, количество параметров в обобщенной решающей функции перестает быть фиксированным.

На практике при полуавтоматическом распознавании образов варианты обобщенной решающей функции можно задавать вручную, подбирая наиболее подходящие дополнительные признаки  $f_i(\mathbf{x})$ .

### Экспериментальная часть

В данной работе проводят анализ метода обобщенных решающих функций при задании дополнительных признаков вручную. При этом основной характеристикой является вероятность распознавания в зависимости от параметров обучающей выборки: ее размеров, наличия в ней выбросов, формы областей, занимаемых классами, – и в зависимости от привлекаемых дополнительных признаков. Для этого необходимо выполнить следующую последовательность действий.

1. Выполнить реализацию метода обобщенных решающих функций.
2. Сформировать выборку образов, которую разделить на обучающую и тестовую часть. Образы из тестовой выборки не участвуют в обучении, а используются затем для определения процента правильных ответов, даваемых при тех или иных дополнительных признаках. Разделение на обучающую и тестовую выборку желательно производить случайным образом. Размер тестовой выборки должен быть достаточно большим (по крайней мере, несколько десятков элементов).
3. Варьируя обучающую выборку, определить проценты правильного распознавания для различных дополнительных признаков (рекомендуется использовать линейные и квадратичные решающие функции). Обучающую выборку следует изменять следующим образом: добавлять и исключать из нее элементы, чтобы менялся размер выборки, вносить в обучающую выборку ошибки (для нескольких элементов указывать неправильный класс).
4. Для нелинейно разделимых классов образов осуществить перебор дополнительных признаков и найти минимальное количество признаков,

при которых корректное разделение находится методом обобщенных решающих функций.

5. Проанализировать полученные результаты. Определить, как влияют ошибки в обучающей выборке на метод обобщенных решающих функций, при каких размерах обучающей выборки и при каких дополнительных признаках больше процент правильного распознавания (и при какой форме областей, занимаемых классами). Сделать выводы по работе.

### **Литература**

1. **Потапов, А.С. Распознавание образов и машинное восприятие: общий подход на основе принципа минимальной длины описания** / А.С. Потапов. – СПб.: Политехника, 2007. – С. 144-152.
2. **Ту, Дж. Принципы распознавания образов** / Дж. Ту, Р. Гонсалес – М.: Мир, 1978. – С. 53-87.

### **Вопросы для самопроверки:**

1. В чем заключается основная идея метода решающих функций?
2. В каком смысле метод обобщенных решающих функций можно назвать линейным, а в каком – нелинейным? Почему для этого метода верны обе эти характеристики?
3. Какая проблема возникает в методе обобщенных решающих функций, когда количество дополнительных признаков может быть произвольным? Как эта проблема может решаться?
4. Любой ли формы разделяющие поверхности могут быть построены в методе обобщенных решающих функций?
5. Как влияют выбросы в обучающей выборке на качество строящейся решающей функции? Можно ли модифицировать метод решающих функций так, чтобы он был менее чувствителен к выбросам?

### **4. Изучение методов анализа пространства признаков**

Цель работы – ознакомиться с методами анализа пространства признаков в рамках задач кластеризации и выбора признаков, а также освоить их применение в различных условиях, определяемых характером распределения образов обучающей выборки. Данная работа имеет два варианта выполнения.

#### **Вариант 1**

#### **Задание по работе:**

1. Изучить теоретическую часть работы.
2. Реализовать метод  $k$  внутригрупповых средних.
3. Путем варьирования взаимного расположения и формы кластеров, образуемых образцами обучающей выборки, определить ограничения метода кластеризации.

### Теоретическая часть

#### Постановка задачи выделения кластеров в пространстве признаков

В задаче распознавания без учителя машинной системе предоставляется лишь совокупность образов  $\mathbf{x}_i \in X, i=1, \dots, M$ . При этом на основе этих образов система должна сформировать некое множество классов  $A = \{a_1, a_2, \dots, a_N\}$  и построить решающее правило  $\phi: X \rightarrow A$ .

Поскольку решающее правило относит каждый образ из обучающей выборки к одному из классов, то задача, по сути, сводится к тому, чтобы объединить образы обучающей выборки в группы (на основе которых и формируются классы). Такое объединение называется *группированием*. Здесь возникает вопрос: на каком основании какие-то образы следует относить к одной группе, а какие-то – к другой?

Один из интуитивно очевидных ответов на этот вопрос заключается в том, что объединяться должны похожие друг на друга образы. Степень сходства определяется расстоянием в пространстве признаков. Выбор метрики, однако, во многом произволен, хотя чаще всего используют евклидово расстояние. Если в классы объединяются наиболее близко расположенные друг к другу образы, то задача группирования превращается в задачу *кластеризации*, то есть в задачу поиска кластеров (областей, содержащих компактно расположенные группы образов).

#### Алгоритм $k$ внутригрупповых средних

Алгоритм  $k$  внутригрупповых средних (или кратко алгоритм  $k$  средних) требует задания числа кластеров, исторически обозначаемых через  $k$ . Здесь для обозначения числа классов будет использоваться переменная  $d$ , а через  $A = \{a_1, \dots, a_d\}$  будет обозначаться множество классов. Алгоритм состоит из следующих шагов:

1. Каждому из  $d$  кластеров произвольным образом назначаются их центры (или эталонные образы)  $\mathbf{x}_{0,i}$ . Часто в качестве этих центров выступают первые  $d$  образов обучающей выборки  $\mathbf{x}_{0,i} = \mathbf{x}_i, i = 1, \dots, d$ .
2. Каждый образ выборки относится к тому классу, расстояние до центра которого минимально:

$$A_i = \arg \min_{a_j \in A} (s(\mathbf{x}_i, \mathbf{x}_{0,j})), i = 1, \dots, M, \quad (17)$$

где  $s(\mathbf{x}, \mathbf{x}')$  – функция расстояния, в качестве которой может использоваться как евклидово расстояние, так и другие метрики, например, полезным может быть нормированное евклидово расстояние:

$$s(\mathbf{x}_i, \mathbf{x}_{0,j}) = \frac{\|\mathbf{x}_i - \mathbf{x}_{0,j}\|}{r_j}, \quad (18)$$

где  $r_j$  – размер  $j$ -го кластера. Этот размер вычисляется как внутриклассовое расстояние (среднеквадратичное расстояние от образов класса до его центра).

3. Центры кластеров пересчитываются, исходя из того, какие образы к каждому из них были отнесены:  $\mathbf{x}_{0,j} = \frac{1}{M_j} \sum_{(\forall i) A_i = a_j} \mathbf{x}_i$ , где  $M_j$  – количество образов, попавших в класс  $a_j$ . После пересчитываются радиусы кластеров:  $r_j^2 = \frac{1}{M_j} \sum_{(\forall i) A_i = a_j} \|\mathbf{x}_i - \mathbf{x}_{0,j}\|^2$ .
4. Шаги 2 и 3 повторяются, пока не будет достигнута сходимость, то есть пока классы не перестанут изменяться.

### Экспериментальная часть

В данной работе проводят анализ метода  $k$  внутригрупповых средних, используемого для распознавания без учителя (кластеризации). При этом требуется установить влияние формы и взаимного расположения кластеров на возможность их обнаружения данным методом, а также устойчивость результатов метода при выборе различных начальных центров кластеров. Для этого необходимо выполнить следующую последовательность действий.

1. Выполнить реализацию метода  $k$  внутригрупповых средних.
2. Сформировать различные обучающие выборки образов, варьирующиеся по форме кластеров (круглые, сильно вытянутые, неправильной формы типа «Г»), их относительным размерам (одинаковые или разные размеры кластеров) и близости расположения кластеров.
3. Для нескольких обучающих выборок определить различия в конечных результатах при использовании разных способов задания начальных центров кластеров: начальные центры формируются из близко расположенных образов, случайно выбранных образов, наиболее удаленно расположенных образов. Оценить количество итераций, требуемых методу для схождения, при разных способах задания начальных центров.

4. Установить различия в результатах кластеризации для нескольких обучающих выборок в зависимости от того, используется ли евклидово расстояние, или оно нормируется на размеры кластеров.

5. Определить характер формируемых кластеров в случаях, когда заданное значение  $k$  отличается (как в большую, так и в меньшую сторону) от действительного числа кластеров в обучающей выборке.

6. Проанализировать полученные результаты. Определить ограничения метода  $k$  средних. Сделать выводы по работе.

### **Литература**

1. **Потапов, А.С. Распознавание образов и машинное восприятие: общий подход на основе принципа минимальной длины описания / А.С. Потапов. – СПб.: Политехника, 2007. – С. 185-191.**
2. **Ту, Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес – М.: Мир, 1978. – С. 101-112.**

### **Вопросы для самопроверки:**

1. К какому типу методов распознавания относится метод  $k$  внутригрупповых средних?
2. Какое ограничение данного метода мешает утверждать, что метод является полностью автоматическим?
3. Классы какой формы строятся методом  $k$  внутригрупповых средних?
4. Какой способ задания начальных центров кластеров в данном методе предпочтительнее?
5. В чем различие метода  $k$  средних при нормировании евклидова расстояния на размеры кластеров и без нормирования?
6. Какие эффекты возникают, когда заданное значение  $k$  больше или меньше действительного числа кластеров?

### **Вариант 2**

#### **Задание по работе:**

1. Изучить теоретическую часть работы.
2. Реализовать метод оценивания плотности вероятностей на основе смесей.
3. Путем варьирования компонент смеси и их количества, определить ограничения метода на основе смесей.

#### **Теоретическая часть**

Часто возможна такая ситуация, что никаких предположений о виде плотности распределения сделать нельзя. В этом случае используют непараметрические методы оценивания. Однако и в этих методах все же необходимо делать некоторые априорные допущения, такие, как, например, непрерывность или симметрия плотности распределения вероятностей. Один из широко распространенных подходов к непараметрическому оцениванию заключается в представлении неизвестной плотности в виде линейной комбинации плотностей известного (параметрического) вида. Это т.н. *смеси*. Мы рассмотрим *конечные смеси*, которые представляются в виде:

$$p(x) = \sum_{i=1}^m p(x | w_i) P(w_i), \quad (19)$$

где  $m$  – число компонентов смеси. Чтобы подчеркнуть, что величины  $P(w_i)$  являются численными коэффициентами, мы будем использовать обозначение  $P_i = P(w_i)$ . Поскольку они имеют смысл вероятностей, то для них должны выполняться ограничения  $0 \leq P_i \leq 1$  и  $P_1 + \dots + P_m = 1$ . Поскольку нас интересует оценивание плотностей распределения вероятностей, как векторы параметров  $w_1, \dots, w_m$ , так и коэффициенты  $P_1, \dots, P_m$  являются неизвестными. В связи с этим необходимо писать:

$$p(x | w_1, \dots, w_m, P_1, \dots, P_m) = \sum_{i=1}^m P_i \cdot p(x | w_i), \quad (20)$$

Смеси полезны и как средство непараметрического оценивания плотностей распределения вероятности в отдельных классах. При этом работа со смесью может вестись абсолютно так же, как и с обычной параметрической плотностью. В частности, здесь оказывается применимым метод стохастической аппроксимации.

Один из способов использования смеси для оценивания плотностей заключается в разложении последних по базисным функциям. Если в уравнении (19) в качестве набора функций  $\{p(x | w_i)\}_{i=1}^m$  использовать полную систему функций (задаваемую, как правило, априори), то с помощью смеси можно будет аппроксимировать произвольную (непрерывную) плотность вероятностей. Это является замечательным свойством, когда априорные сведения о виде плотности вероятностей отсутствуют. Выбор набора базисных функций, казалось бы, не накладывает никаких ограничений на то, какие плотности могут быть восстановлены, коль скоро либо этот набор является полным, либо пространство, натянутое на него, гарантированно содержит искомую плотность.

Одной из наиболее популярных смесей является смесь нормальных плотностей.

В частном случае для аппроксимации плотности распределения элементов одного класса можно жестко задать параметры смеси следующим образом. Количество  $m$  компонентов смеси равно числу

эталонных образов  $M$ . Ковариационные матрицы всех компонентов являются единичными матрицами. Вектор средних  $\mathbf{x}_{0,i}$   $i$ -го компонента смеси равен  $i$ -му образу обучающей выборки  $\mathbf{x}_{0,i} = \mathbf{x}_i$ , а коэффициенты смеси  $P_i = 1/M$ . Иными словами, в каждую точку обучающей выборки «помещается» нормальное распределение с единичной ковариационной матрицей.

### Экспериментальная часть

В данной работе проводят исследование метода оценивания плотности вероятностей на основе смесей, используемого для оценки неизвестной плотности вероятностей. При этом требуется установить влияние типа входящих в смесь распределений и их количества на точность оценки. Для этого необходимо выполнить следующую последовательность действий.

1. Сгенерировать обучающую выборку на основе заданной функции распределения плотности вероятностей.

2. Реализовать метод, основанный на представлении искомого распределения в виде конечных смесей, с целью оценки плотности вероятности в произвольной точке.

3. Оценить при каких входящих типах распределений в смесь, обнаружение искомого распределения будет наилучшим.

4. Установить, как количество входящих в смесь компонентов влияет на обнаружение искомого распределения, и на быстродействие программы

5. Проанализировать полученные результаты. Определить ограничения метода конечных смесей. Сделать выводы по работе.

### Литература

1. **Потапов, А.С. Распознавание образов и машинное восприятие: общий подход на основе принципа минимальной длины описания / А.С. Потапов. – СПб.: Политехника, 2007. – С. 191-197.**

### Вопросы для самопроверки:

1. К каким методам оценивания относится подход на основе смесей?
2. Как вы понимаете метод конечных смесей? Какие допущения в нем используются?
3. Для каких способов обучения применим подход на основе смесей? Почему?
4. Какие вы знаете алгоритмы для оценки неизвестных параметров распределения компонент смеси?



## Приложение 1. Шаблон оформления отчета по лабораторной работе

Министерство образования и науки Российской Федерации  
федеральное государственное бюджетное образовательное учреждение высшего  
профессионального образования

Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики  
Факультет Фотоники и оптоинформатики  
Кафедра Компьютерной фотоники и видеоинформатики

Дисциплина: Технологии искусственного интеллекта

# Отчет

по лабораторной работе № \_\_\_\_  
(Название работы)

Выполнил: (Фамилия,  
инициалы)  
Группа: (Номер группы)  
Преподаватель: (Фамилия,  
инициалы)

	Дата	Количество баллов	Подпись преподавателя
Допуск к работе (защита)			
Выполнение работы			
Отчет			

Санкт-Петербург  
2013 г.

**Цель работы:** *(формулировка цели работы).*

**Ход работы:**

*(По пунктам приводятся основные этапы выполнения работы и результаты, получаемые на каждом из этапов. Приветствуется краткое обсуждение результатов.)*

**Выводы:**

*(Выводы.)*

**Приложение**  
Листинг программы

*(Листинг программы).*

## ОГЛАВЛЕНИЕ

1. ИЗУЧЕНИЕ КЛАССИЧЕСКИХ МЕТОДОВ ПОИСКА – ГРАДИЕНТНОГО СПУСКА И МОДЕЛИРОВАНИЯ ОТЖИГА.....	3
2. МЕТОДЫ ПОСТРОЕНИЯ АССОЦИАТИВНЫХ СЕТЕЙ .....	12
3. ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПАРАМЕТРОВ ОБУЧАЮЩЕЙ ВЫБОРКИ НА ВЕРОЯТНОСТЬ РАСПОЗНАВАНИЯ НОВЫХ ОБРАЗОВ .....	20
4. ИЗУЧЕНИЕ МЕТОДОВ АНАЛИЗА ПРОСТРАНСТВА ПРИЗНАКОВ.....	27
ПРИЛОЖЕНИЕ 1. ШАБЛОН ОФОРМЛЕНИЯ ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ.....	33
КАФЕДРА КОМПЬЮТЕРНОЙ ФОТОНИКИ И ВИДЕОИНФОРМАТИКИ .....	<i>i</i>



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена Программа развития государственного образовательного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» на 2009–2018 годы.

---

## **КАФЕДРА КОМПЬЮТЕРНОЙ ФОТОНИКИ И ВИДЕОИНФОРМАТИКИ**

Достижения в оптической науке, технике и технологиях за последние годы способствовали появлению нового направления – фотоники. Этот термин охватывает область науки и техники, связанную с использованием светового излучения (или потока фотонов) в оптических элементах, устройствах и системах.

На рубеже XX – XXI веков электронные информационные технологии достигли фундаментальных и технических пределов производительности при продолжающемся росте потребительского спроса на скорость и объем обрабатываемой и передаваемой информации. Решение данной проблемы потребовало разработки нового поколения информационно – телекоммуникационных систем, основанных на технологиях фотоники. В фотонике появилось новое динамично развивающееся направление, определяющее прогресс мировой науки и техники, – «оптоинформатика». Под «оптоинформатикой» понимают область науки и техники, связанную с исследованием, разработкой, созданием и эксплуатацией новых материалов, технологий, приборов и устройств, направленных на передачу, прием, обработку, хранение и отображение информации.

Изучение фотоники основывается на знании принципов формирования, преобразования, анализа изображений, теории построения информационных систем. Интеграция фотоники и компьютерных технологий позволяет создавать методы, которые возможно реализовать исключительно средствами компьютерной фотоники, обеспечивая развитие технологий качественно нового уровня.

По многим направлениям фотоники и оптоинформатики Россия находится на уровне промышленно – развитых стран (интегральная оптика, системы приема, обработки и отображения информации и др.), а по некоторым – даже опережает. Приоритетными направлениями являются: волоконная оптика (работы академика Дианова Е.М. - ИОФ РАН), голография (академик Денисюк Ю.Н. – ГОИ им. С.И. Вавилова), полупроводниковые лазеры (академик Алферов Ж.И – ФТИ РАН им. А.Ф. Иоффе), полифункциональные оптические материалы (академик Петровский Г.Т. – ГОИ им. С.И. Вавилова) и др.

Ввиду большого научного и практического значения направления "Фотоника и оптоинформатика", а также спроса на него на потребительском рынке, в 2002 г. в СПбГУ ИТМО был организован факультет «Фотоники и оптоинформатики» под руководством доктора физ.-мат. наук, профессора С.А. Козлова. По инициативе профессорско-преподавательского состава, начиная с 2005 года, на факультете стала работать выпускающая кафедра «Компьютерной фотоники», которую возглавил доктор технических наук, профессор И.П. Гуров. В 2010 году название кафедры было уточнено как «Компьютерная фотоника и видеоинформатика», а в 2011 году на ней появилось второе направление подготовки – «Прикладная информатика».

История кафедры началась в 1946 году. На всех этапах развития результаты научных исследований, проводимых сотрудниками кафедры, неизменно использовались в учебном процессе. Совершенствовались направления подготовки студентов, изменялось название кафедры, но всегда кафедра гордилась своими выпускниками.

Выпускники кафедры занимают видное место в оптической науке: академик РАН Ю.Н. Денисюк, изобретатель трехмерной голографии; член-корр. РАЕН, профессор Н.Г. Бахшиев, известный специалист в области спектроскопии межмолекулярных взаимодействий; Заслуженный деятель науки и техники РСФСР, профессор Г.Н. Дульнев, крупный ученый в области теплофизики, долгие годы – ректор ЛИТМО; профессор И.М. Нагибина, исследования которой в области физической оптики получили широкое признание.

Одной из важнейших задач кафедры является организация учебного процесса и подготовка профессионалов в области компьютерной фотоники. Направление работы кафедры определяется развитием информационных технологий и компьютерных систем в области формирования, синтеза, обработки и анализа изображений на основе интеграции эффективных компьютерных систем с системами фотоники.

Проводимые исследования в области компьютерной обработки когерентных и некогерентных изображений обеспечивают решение научно-технических задач оптической томографии, цифровой голографии, синтеза, анализа, распознавания и классификации изображений.

В частности, в 2009 году на базе кафедры создан Научно-образовательный инновационный центр интеллектуальных систем компьютерного восприятия и управления для проведения исследований в области компьютерного зрения в интеллектуальных системах.

Научным консультантом работ кафедры в области компьютерной обработки изображений – иконики – является член-корреспондент РАН М.М. Мирошников.

Кафедра проводит работы в рамках международных научных проектов в сотрудничестве с ведущими зарубежными университетами, институтами и исследовательскими лабораториями Италии, Финляндии, Франции, Германии, Великобритании, Японии, США и других стран в области оптической когерентной томографии для биомедицинских исследований, цифровой голографии для исследования микро- и наноструктур, трехмерной фотографии микро- и макроскопических объектов, гиперспектральной обработки изображений.

В 2007 году СПбГУ ИТМО стал победителем в российском конкурсе на разработку Инновационной образовательной программы, в этом есть и заслуга преподавательского коллектива кафедры Компьютерной фотоники. Участие в реализации Национального проекта «Образование» позволила в 2007-2008 учебном году разработать инновационные программы подготовки бакалавров и магистров, оснастить учебные лаборатории современным оборудованием, подготовить и издать новые учебные пособия.

Алексей Сергеевич Потапов  
Щербаков Олег Викторович  
Жданов Иннокентий Николаевич

СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ:

Учебно-методическое пособие  
по лабораторному практикуму

В авторской редакции  
Дизайн и верстка

А.С. Потапов  
О.В. Щербаков  
И.Н. Жданов

Редакционно-издательский отдел Санкт-Петербургского государственного  
университета информационных технологий, механики и оптики

Зав. РИО

Н.Ф. Гусарова

Лицензия ИД № 00408 от 05.11.99

Подписано к печати \_\_\_\_\_

Заказ № \_\_\_\_\_

Тираж 100 экз.

Отпечатано на ризографе

**Редакционно-издательский отдел**  
Санкт-Петербургского национального ис-  
следовательского университета информа-  
ционных технологий, механики и оптики  
197101, Санкт-Петербург, Кронверкский пр., 49

