Миничтерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования «Северо-Кавказский федеральный университет»

Методические указания для выполнения лабораторных работ
по дисциплине «**Генеративный искусственный интеллект**»

для студентов направление подготовки
09.03.02 Информационные системы и технологии
Направленность (профиль) Прикладное программирование в
интеллектуальных информационных системах

Квалификация выпускника
Бакалавр

Ставрополь, 2026

Лабораторная работа №1

ОЦЕНКА ПАРАМЕТРОВ И ПРОВЕРКА ГИПОТЕЗ В ЛИНЕЙНЫХ МОДЕЛЯХ С КАЧЕСТВЕННЫМИ ФАКТОРАМИ

1. Методические указания. Теоретические основы анализа линейных моделей с качественными факторами

Рассмотрим модель наблюдения вида

$$y = X\theta + \varepsilon, \qquad (1)$$

где $y-(n\times 1)-$ наблюдаемый в эксперименте отклик, измеряемый в количественной шкале; $X-(n\times p)-$ матрица наблюдения, порождаемая факторами $x_1,\ldots,x_k$, $\theta-(p\times 1)-$ вектор неизвестных параметров, подлежащих оцениванию из эксперимента; ε – аддитивная случайная составляющая модели наблюдения, обычное предположение относительно которой состоит в том, что $\varepsilon\sim N(0,\sigma^2 I_n)$. Рассмотрим случай, когда факторы $x_1,\ldots,x_k$ являются качественными, измеренными в номинальной шкале.

Как правило, номинальные качественные факторы $x_1,\ldots,x_k$ в модели (1) кодируют набором псевдопеременных. Число этих псевдопеременных для каждого фактора определяется числом его уровней варьирования в эксперименте. Все псевдопеременные могут принимать только два возможных значения: 1 или 0 – в зависимости от того находился ли в данном эксперименте фактор на заданном уровне или нет. Если предположить, что в эксперименте факторы $x_1,\ldots,x_k$ варьируются соответственно на $s_1,\ldots,s_k$ уровнях, то в линейной модели (1) без учета взаимодействий число неизвестных параметров определяется как $p=1+s_1+\cdots+s_k$.

Часто модель (1) записывают в другом, исторически более раннем виде, например,

$$y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij},\ i=1,\ldots,I;\ j=1,\ldots,J, \qquad (2)$$

где μ – среднее, α_i, β_j – эффекты уровней соответственно первого и второго факторов. В данном случае в модели (2) для двух факторов необходимо оценить параметры $\mu,\ \alpha_1,\ldots,\alpha_I,\ \beta_1,\ldots,\beta_J$.

Общеизвестно, что модели вида (1) или (2) относятся к моделям неполного ранга. Оценки параметров в этих моделях, полученные, например, по методу наименьших квадратов, в общем случае будут смещенными. Данное обстоятельство послужило одной из причин развития теории функций, допускающих оценку и использования методов псевдообращения при анализе моделей (1).

На наш взгляд принципиальным моментом является понимание того факта, что дефект ранга в модели (1) может иметь двойную природу. Назовем дефект ранга ***внутренним***, если он является следствием способа кодировки

качественных факторов в модели (1) и ***внешним***, если он является следствием ущербности реализованного плана эксперимента. Если нам удается тем или иным способом ликвидировать внутренний дефект ранга, то мы оказываемся в ситуации, ничем принципиально не отличающейся от задачи анализа линейных регрессионных моделей. Практика показывает, что внешний дефект ранга как в регрессионных так и в дисперсионных моделях встречается достаточно редко и является результатом очень неудачного реализованного плана эксперимента.

Рассмотрим методику редуцирования моделей (1), (2) к модели полного ранга. Запишем модель (2) для случая *k* факторов в виде

$$y_{ij\ldots l} = \mu + \alpha_i + \beta_j + \ldots + \gamma_l + \varepsilon_{ij\ldots l},$$
$$i = 1,\ldots,I;\ j = 1,\ldots,J;\ldots;\ l = 1,\ldots,L,\ I = s_1,\ J = s_2,\ldots,\ L = s_k. \qquad (3)$$

Пусть для модели (3) реализован некоторый план эксперимента с матрицей наблюдения *X*. Предположим также, что данный план не вносит дополнительно в модель внешний дефект ранга. В состав столбцов матрицы *X* войдут столбцы, порождаемые всеми псевдопеременными для факторов $x_1,\ldots,x_k$ и свободный член. Для удобства будем обозначать их также как неизвестные параметры с подчеркиванием снизу: $\underline{\mu},\ \underline{\alpha_1},\ldots,\underline{\alpha_I},\quad \underline{\beta_1},\ldots,\underline{\beta_J},\ldots,\underline{\gamma_1},\ldots,\underline{\gamma_L}$, имея ввиду

что это *N* - мерные вектора, состоящие из 0 и 1, а $\underline{\mu}$ – вектор из 1.

Утверждение 1. В линейной модели (3) для *k* факторов без взаимодействий внутренний дефект ранга матрицы плана равен *k*.

Утверждение 1 устанавливает факт, что среди столбцов матрицы *X* присутствует ро $\underline{\alpha_I},\ \underline{\beta_J},\ldots,\underline{\gamma_L}$. В модели (3) в связи с ее внутренним дефектом ранга несмещенно будут оцениваться только $r = p - k = rg(X)$ линейно независимых функций, допускающих оценку (ФДО), образующих базис ФДО. Вид ФДО, входящих в базис, устанавливает следующее утверждение.

Утверждение 2. В линейной модели (3) с линейно зависимыми от других столбцами $\underline{\alpha_I},\ \underline{\beta_J},\ldots,\underline{\gamma_L}$ базис ФДО составляют функции

$$\overline{\mu} = \mu + \alpha_I + \beta_J + \cdots + \gamma_L,\ \overline{\alpha_i} = \alpha_i - \alpha_I,\ i = 1,\ldots,I-1,$$
$$\overline{\beta_j} = \beta_j - \beta_J,\ j = 1,\ldots,J-1,\ldots,\ \overline{\gamma_l} = \gamma_l - \gamma_L,\ l = 1,\ldots,L-1. \qquad (4)$$

Доказательство. Пусть матрица наблюдения *X* разбита на две $X = (X_1, X_2)$, где в X_1 вошли все линейно независимые столбцы $\underline{\mu},\underline{\alpha_1},\ldots,\underline{\alpha_{I-1}},\ \underline{\beta_1},\ldots,\underline{\beta_{J-1}},\ldots,\underline{\gamma_1},\ldots,\underline{\gamma_{L-1}}$, а в X_2 оставшиеся линейно зависимые от них столбцы $\underline{\alpha_I},\ \underline{\beta_J},\ldots,\underline{\gamma_L}$. Тогда $X^T X = \begin{pmatrix} X_1^T X_1 & X_1^T X_2 \\ X_2^T X_1 & X_2^T X_2 \end{pmatrix}$.

Поскольку $X_1^T X_1$ полного ранга *r*, то для $X^T X$ существует рефлексивная

обратная $(X^T X)_r^-$ вида: $(X^T X)_r^- = \begin{pmatrix} (X_1^T X_1)^{-1} & 0 \\ 0 & 0 \end{pmatrix}$. Известно, что матрица $H = (X^T X)^- X^T X$ имеет ранг r и ненулевые ее строки образуют базис ФДО. В нашем случае имеем

$$H = (X^T X)_r^- X^T X = \begin{pmatrix} I_r & \tilde{A} \\ 0 & 0 \end{pmatrix}, \tag{5}$$

где $I_r -$ единичная матрица размера r, $\tilde{A} = (X_1^T X_1)^{-1} X_1^T X_2$. Матрица $\tilde{A}$ есть решение матричного уравнения $X_2 = X_1 \tilde{A}$, которое определяет как столбцы X_2 линейно выражаются через X_1.

Редуцирование модели (1) к модели полного ранга можно проводить через факторизацию матрицы $X = X_1 A$, где X_1 – матрица полного строчного ранга:

$$y = X\theta + \varepsilon = X_1 A\theta + \varepsilon = X_1 \bar{\theta} + \varepsilon. \tag{6}$$

В (6) матрица A задает вид базиса ФДО и в соответствии с (5) имеет вид $A = (I_r, \tilde{A})$.

Обратимся теперь к задачам оценивания ФДО в рассматриваемых линейных моделях. Пусть как и раньше матрица X разбита на две $X = (X_1, X_2)$, где в X_1 вышли все r линейно независимые столбцы из X. Общее решение нормальных уравнений для получения оценок для θ по методу наименьших квадратов может быть записано как

$$\hat{\theta} = (X^T X)^- X^T y + (H - I)z, \tag{7}$$

где z – произвольный $(p \times 1)$ вектор, $H = (X^T X)^- X^T X$. С учетом того, что матрицу $X^T X$ и рефлексивную обратную для нее можно представить в блочном виде, решение (7) запишется как

$$\hat{\theta} = \begin{pmatrix} (X_1^T X_1)^{-1} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & \tilde{A} \\ 0 & -I_{p-r} \end{pmatrix} \cdot z = \begin{pmatrix} \hat{\bar{\theta}} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & \tilde{A} \\ 0 & -I_{p-r} \end{pmatrix} \cdot z, \tag{8}$$

где $\bar{\theta} = A\theta$ из (6). Таким образом, $\hat{\theta}_i$, $i = 1, \ldots, r$ являются несмещенными оценками для ФДО $\bar{\theta} = A\theta$. Для вычисления оценок для $\bar{\theta}$ нам достаточно рассматривать только подматрицу $X_1^T X_1$. Если положить в (7) $z = 0$, то получаем одно из решений $\hat{\theta} = (\hat{\bar{\theta}}, 0)^T$. Это решение автоматически обеспечивает выполнение ограничений на параметры $\theta_{r+i} = 0$, $i = 1, \ldots, p - r$. Что бы найти оценки для ФДО $\bar{\theta} = A\theta$ достаточно применить теорему Гаусса-Маркова для модели (6): $\hat{\bar{\theta}} = (X_1^T X_1)^{-1} X_1^T y$. Оценки параметров,

соответствующие исключенным из модели линейно зависимым столбцам (регрессорам), автоматически считаются равными нулю.

Анализ линейных моделей с качественными факторами в основном сводится к выявлению значимости или незначимости эффектов различных уровней одного и того же фактора и к проверки значимости влияния на отклик в целом того или иного фактора. Рассмотрим формулировку этих гипотез на примере. Пусть исследуется влияние двух факторов с тремя и двумя уровнями варьирования соответственно. Модель наблюдения имеет вид

$$y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}, \quad i = 1,2,\ 3; j = 1,2.$$

Базис ФДО составят следующие параметрические функции

$$\overline{\mu} = \mu + \alpha_3 + \beta_2,\ \overline{\alpha_i} = \alpha_i - \alpha_3,\ i = 1,2,\ \overline{\beta_1} = \beta_1 - \beta_2.$$

Гипотеза о не значимости эффектов первого и третьего уровней первого фактора формулируется как $H : \alpha_1 = \alpha_3$. Поскольку в редуцированной модели на месте параметра α_1 оценивается сравнение (ФДО) $\overline{\alpha}_1 = \alpha_1 - \alpha_3$, то для проверки этой гипотезы достаточно проверить $H : \overline{\alpha}_1 = 0$. Для проверки гипотезы $H : \alpha_2 = \alpha_3$ достаточно проверить $H : \overline{\alpha}_2 = 0$. Для проверки гипотезы $H : \alpha_1 = \alpha_2$ достаточно проверить гипотезу $H : \overline{\alpha}_1 - \overline{\alpha}_2 = 0$. Не значимость в целом первого фактора может быть проверена по гипотезе $H : \begin{cases} \overline{\alpha}_1 = 0 \\ \overline{\alpha}_2 = 0 \end{cases}$.

2. Перечень вопросов к разработке

1. По имеющимся данным (см. варианты заданий) сформировать матрицу наблюдений *X*, постулировать модель дисперсионного анализа с главными эффектами (без взаимодействий уровней факторов).
2. Провести редукцию модели к модели полного ранга, определить базис ФДО.
3. По методу МНК- оценивания провести оценивание ФДО в редуцированной модели. Проверить гипотезы о не значимости различий в эффектах уровней для каждого фактора и фактора в целом.
4. Отчет должен содержать постановочную часть, решения по редукции модели, компьютерный листинг, результаты расчетов по проверке гипотез, статистические выводы.

3. Варианты заданий

1.

Уровни Фактора 1	Уровни фактора 2			
	B_1	B_2	B_3	B_4
A_1	3,1 2,9	4,1 3,9	2,1 1,9	2,1 1,9
A_2	3,9 4,1	4,9 5,1	2,9 3,1	3,0 3,0
A_3	2,1 1,9	3,1 2,9	1,1 0,9	1,0 1,0

2.

Уровни фактора 1	Уровни фактора 2				
	B_1	B_2	B_3	B_4	B_5
A_1	5,1 4,95	4,0 4,1	3,1 3,0	5,9 6,1	4,1 4,0
A_2	4,9 5,05	4,09 3,95	2,9 3,05	6,06 5,95	3,95 4,05
A_3	3,1 2,95	2,0 1,95	1,05 0,96	4,06 3,99	2,03 1,96

3.

Уровни фактора 2	Уровни фактора 1			
	A_1	A_2	A_3	A_2
B_1	5,1 4,95	6,1 6,05	3,1 3,0	6,15 6,0
B_2	5,4 5,6	6,4 6,55	3,4 3,56	6,4 6,57

4.

Уровни фактора 1	Уровни фактора 2			
	B_1	B_2	B_3	B_4
A_1	7,14 6,9	4,11 3,95	8,13 7,97	4,07 3,99
A_2	3,1 2,91	0,0 0,01	4,03 3,98	0,02 0,0
A_3	7,09 6,92	4,11 3,96	8,12 8,01	4,07 4,0

5.

	C_1		C_2		C_3	
	A_1	A_2	A_1	A_2	A_1	A_2
B_1	10,21 9,95	10,15 9,87	8,91 9,12	8,99 9,13	8,04 7,89	7,91 8,11
B_2	9,08 8,92	8,84 9,12	7,81 8,13	8,03 8,01	6,91 7,14	7,17 6,89

6.

	C_1		C_2	
	A_1	A_2	A_1	A_2
B_1	8,26 7,89	5,16 5,01	7,06 6,81	3,9 4,11
B_2	7,95 8,12	4,91 5,13	7,06 6,81	4,07 3,91
B_3	10,01 9,89	7,07 7,00	9,01 8,99	6,06 6,01

7.

Уровни Фактора 1	Уровни фактора 2			
	B_1	B_2	B_3	B_4
A_1	3,1 2,9 3,2	4,0	2,1 1,9	2,1 1,9
A_2	3,9 4,1	4,9 5,1	2,9 3,1 3,2	3,0 3,0
A_3	2,1 1,9 1,95	3,1 2,9	1,0	1,0 1,0 0,95

8.

Уровни фактора 1	Уровни фактора 2				
	B_1	B_2	B_3	B_4	B_5
A_1	5,1 4,95 5,0	4,0 4,1	3,1 3,0 2,9	5,9 6,1	4,1 4,0 3,9
A_2	4,9 5,05	4,09 3,95	2,9 3,05	6,06 5,95	3,95 4,05
A_3	3,1 2,95	2,0 1,95 2,1	1,05 0,96	4,06 3,99 4,0	2,03 1,96

Примечание. В ячейках таблиц 1 - 6 располагаются значения отклика, полученные в двух параллельных наблюдениях. В вариантах 7, 8 число наблюдений в ячейке от 1 до 3. Пересечение соответствующих строки и столбца определяет условия эксперимента. Например, в табл.1 ячейка на пересечении строки A_1 и столбца B_3 обозначает, что в данном эксперименте первый фактор находился на первом уровне, а второй фактор на третьем. В вариантах 5, 6 предполагается действие трех факторов.

4. Контрольные вопросы

1. Основы теории обобщенных обратных матриц. Типы обобщенных обратных матриц и их свойства.
2. Свойства псевдообратных матриц.
3. Вычисление обобщенных обратных матриц: алгоритм, основанный на обычном обращении матриц, алгоритм, основанный на использовании элементарных операторов.
4. Вычисление псевдообратных матриц: алгоритм Гревилля.
5. Решение СЛАУ с использованием обобщенного обращения.
6. Параметрические функции, допускающие оценку (ФДО). Определение и оценивание ФДО.
7. Дисперсия оценок ФДО, свойства остаточной суммы квадратов.
8. Распределения основных статистик. Проверка линейных гипотез.

9. Базис ФДО для линейной модели ДА.
10. Базис ФДО для модели ДА с взаимодействиями уровней факторов.
11. Базис ФДО для линейной модели с порядковыми переменными.
12. Оценивание параметров и ФДО в редуцированной модели.

Литература

1. Попов А.А. Конспект лекций по дисперсионному анализу. Электронный вариант. – Новосибирск, 2015.
2. Попов А.А. Конструирование линейных регрессионных моделей с разнотипными переменными. Уч. пособ., НГТУ. – Новосибирск, 1999.

Лабораторная работа №2

УСТОЙЧИВЫЕ МЕТОДЫ ОЦЕНИВАНИЯ ПАРАМЕТРОВ РЕГРЕССИИ

1. Цель работы

Изучить методы устойчивого оценивания параметров регрессии.

2. Методические указания

2.1. МНК и устойчивость

В условиях нормальной гипотезы метод наименьших квадратов является оптимальным. Отметим характерную особенность нормального распределения – основная масса распределения сосредоточена на конечном интервале $(-3\sigma, +3\sigma)$. Вне этого интервала находится лишь 0,27% распределения. Другими словами, нормальное распределение имеет «легкие хвосты».

Таким образом, принимая гипотезу нормальности, мы автоматически предполагаем, что основная масса отклонений сосредоточена на некотором интервале. Вероятность большого отклонения при этом весьма мала. В реальной ситуации эта гипотеза является чересчур жесткой: предполагаемая модель редко является абсолютно точно специфицированной; в наборе данных могут присутствовать *выбросы* – грубые ошибки. Выбросы могут быть результатом нарушения условия эксперимента, неправильного измерения, засорения данных и т.п. Поэтому необходимо предположить, что отклонения с большей вероятностью могут принимать и большие значения. Это заставляет нас отказаться от распределения с легкими хвостами (в частности, от нормального распределения) и перейти к распределениям с тяжелыми хвостами.

Оценки, ориентированные на распределения с легкими хвостами (в частности, оценка МНК), в новой ситуации оказываются далекими от эффективных. В распределениях с тяжелыми хвостами более эффективными будут менее чувствительные оценки, а именно такие, которые не меняют резко своих значений при возникновении больших отклонений (выбросов). Такие оценки называют *устойчивыми* или *робастными*.

Если наблюдения не засорены, т.е. вероятность больших отклонений мала, устойчивые оценки будут менее эффективны, зато если наблюдения содержат выбросы, то эти оценки будут малочувствительны к ним, а потому более удовлетворительными. Таким образом, переходя к распределениям с более тяжелыми хвостами, мы теряем в эффективности, но приобретаем в надежности. Соответствующие методы будут менее чувствительны к ошибкам спецификации отклонений регрессии.

Пример. Рассмотрим простейший случай – оценивание модели с одним свободным членом:

$$y = \theta + e.$$

Пусть имеется следующий набор данных:

0.96 1.01 0.97 1.02 1.04 1.00 10.52.

Последнее число, очевидно, является выбросом.

МНК-оценка неизвестного параметра – среднее арифметическое элементов выборки – равна 2.36 и, таким образом, сильно отличается от истинного значения, равного 1.

Более устойчивой оценкой является *усеченное среднее*, которое вычисляется следующим образом: отбрасывается некоторое количество минимальных и максимальных наблюдений в выборке. На основе оставшихся наблюдений находится среднее арифметическое.

В нашем случае усеченное среднее (при отбрасывании одного минимального и одного максимального наблюдений) равно 1.008.

Рассмотрим другую устойчивую оценку – *медиану*.

Напомним, что медиана последовательности есть величина, по левую и по правую стороны от которой лежит одинаковое количество элементов. Вычисляется медиана так: элементы последовательности упорядочиваются по их величине. Если количество элементов в последовательности нечетное, то величина медианы равна значению среднего элемента упорядоченной последовательности. Если количество элементов в последовательности четное, то медиана определяется как среднее арифметическое двух средних элементов упорядоченной последовательности.

Медиана нашей последовательности равна 1.01.

Пример. На графике приведены наблюдения (отмечены ромбиками), полученные по линейной модели $y = 12 + x + e$, где e – ошибка, имеющая нормальное распределение, однако наблюдения в точках $x = 0,9$ и $x = 1$ заменены выбросами – значениями 11. Пунктирная линия соответствует МНК-оценке отклика, при этом МНК-оценки параметров равны 11.82 и 0.49. Сплошная линия соответствует некоторой устойчивой оценке отклика, для

которой оценки параметров равны 11.99 и 0.94.

Как видно из графика, МНК-оценка чувствительна к наличию выбросов – линия МНК-оценки отклика отклоняется в сторону наблюдений-выбросов. Линия устойчивой оценки отклика наблюдения-выбросы игнорирует.

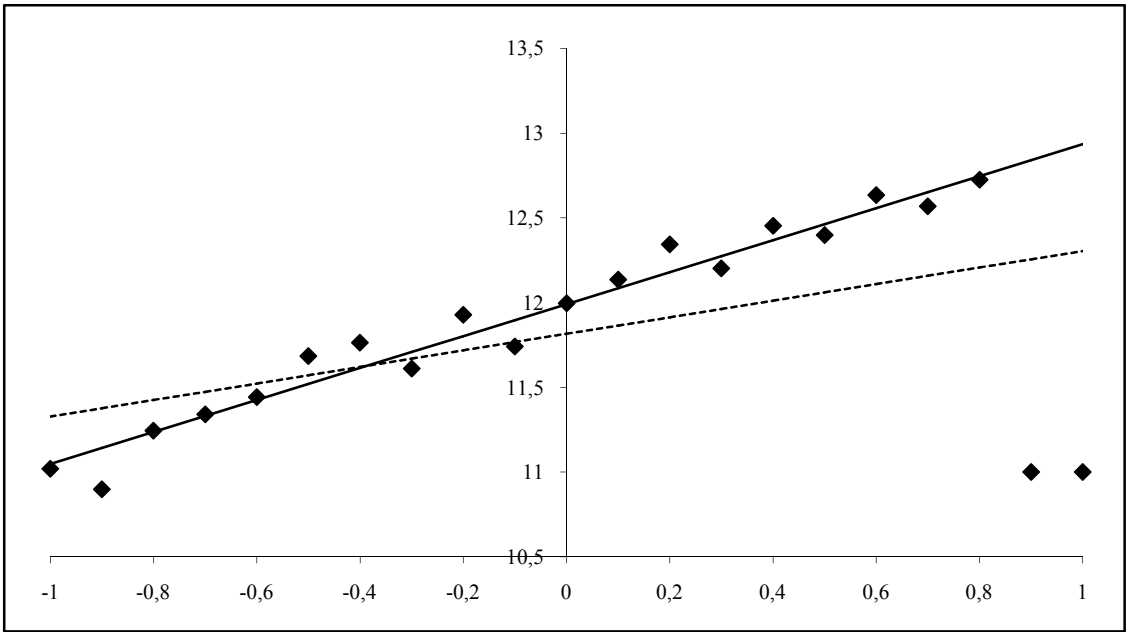

Рис. 1

2.2. М-оценки

Рассмотрим регрессионную модель вида

$$y = f^T(x)\theta + e = \sum_{l=1}^{m} f_l(x)\theta_l + e,$$

где y – значение зависимой переменной; $f^T(x) = (f_1(x),...,f_m(x))$ – заданная вектор-функция от независимой векторной переменной x; $\theta = (\theta_1,...,\theta_m)^T$ – вектор неизвестных параметров, которые необходимо определить по результатам экспериментов (измерений); e – ошибка.

Предположим, что ошибки наблюдений имеют засоренное нормальное распределение, т.е. предполагается, что бóльшая часть наблюдений имеет нормальное распределение, а ряд наблюдений имеют другое (засоряющее) распределение. В качестве засоряющего может рассматриваться, например, нормальное распределение со значительно большей дисперсией по отношению к дисперсии ошибок основной части выборки.

М-оценки вектора параметров θ находятся путем минимизации функции вида

$$Q = \sum_{i=1}^{n} \rho\left(\frac{r_i(\theta)}{\sigma}\right), \qquad (1)$$

где $\rho(z)$ – функция потерь, $r_i(\theta) = y_i - f^T(x_i)\theta$ – остаток i-го измерения, $\sigma > 0$ – корень квадратный из дисперсии.

Путем выбора подходящей функции потерь можно обеспечить робастность оценки.

Пример. Приведем примеры М-оценок:

– оценка по методу наименьших квадратов $\rho(z) = z^2$;

– оценка по методу наименьших модулей $\rho(z) = |z|$;

– L_p-оценка $\rho(z) = |z|^p$.

Необходимые условия минимума функции (1) получают, приравнивая к нулю частные производные по параметрам θ_l:

$$\sum_{i=1}^{n} \psi\left(\frac{r_i(\hat{\theta})}{\sigma}\right) f_l(x_i) = 0, l = 1,...,m, \tag{2}$$

где $\psi(z) = \rho'(z)$. Система уравнений (2) дает альтернативное определение М-оценки (с точностью до константного сомножителя).

В данной постановке задачи параметр σ предполагается известным. Если же это не так, то параметр σ приходится оценивать, используя некоторое уравнение

$$\sum_{i=1}^{n} \chi\left(r_i(\hat{\theta}), \hat{\sigma}\right) = 0. \tag{3}$$

При этом функции $\psi(z)$ и $\chi(z, \sigma)$ не обязаны быть связанными с одной и той же функцией $\rho(z)$.

Так, на практике часто предпочитают в качестве устойчивой оценки σ для нормального распределения использовать медиану абсолютных отклонений (MAD-оценку)

$$\hat{\sigma} = \underset{i}{\text{med}} \left| r_i(\hat{\theta}) \right| / 0.67449, \tag{4}$$

где $|\bullet|$ – модуль; деление на константу обеспечивает асимптотическую несмещенность оценки при незасоренном нормальном распределении.

М-оценку в можно рассматривать как оценку по методу максимального правдоподобия (ММП). Действительно, пусть ошибки e_i имеют функцию плотности $\frac{1}{\sigma} f\left(\frac{e}{\sigma}\right)$. Тогда в предположении независимости и одинаковой распределенности наблюдений в выборке функция правдоподобия равна $\prod_{i=1}^{n} \frac{1}{\sigma} f\left(\frac{r_i(\theta)}{\sigma}\right)$. Напомним, что ММП-оценка находится в результате минимизации логарифма функции правдоподобия, взятого со знаком минус.

В результате, обозначив $\rho(z) = \ln \sigma - \ln f(z)$, приходим к задаче

минимизации функции (1), а обозначив $\psi(z) = -\dfrac{f'(z)}{f(z)}$ и $\chi\left(z,\sigma\right) = \psi(z/\sigma)z/\sigma - 1$ – к задаче поиска корня системы уравнений (2)–(3).

2.3. Робастные оценки

В предположении, что ошибки наблюдений имеют засоренное нормальное распределение, разработано большое число робастных оценок параметров регрессии, относящихся к классу М-оценок со специально подобранной функцией $\rho(z)$. Часто такая функция является квадратичной при малых значениях z, а при больших значениях z является «менее возрастающей», чем z^2.

Робастные функции потерь обычно имеют положительный параметр c, значение которого часто неизвестно, и стоит вопрос его выбора.

Швейцарским математиком П. Хьюбером предложена следующая функция (см. рисунок):

$$\rho(z) = \begin{cases} \dfrac{1}{2}z^2, \text{ если } |z| < c \\ c|z| - \dfrac{1}{2}c^2, \text{ если } |z| \geq c \end{cases}$$

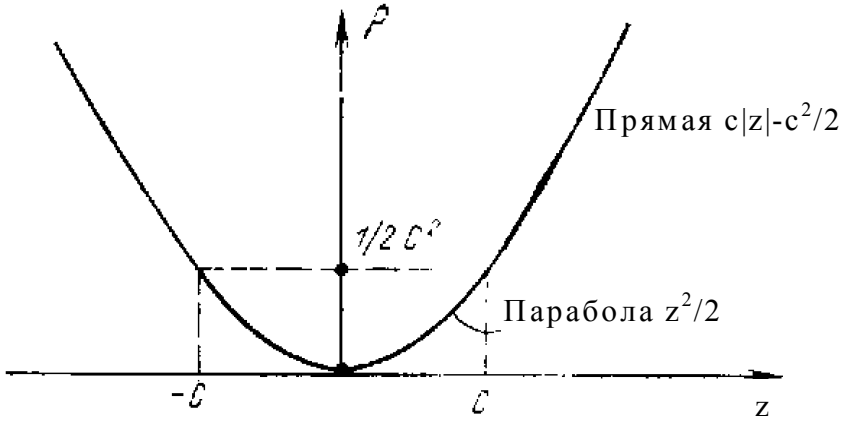

Рис. 2

Здесь вклад в сумму остатков, значения которых меньше по абсолютной величине некоторого порогового значения $c>0$, измеряется в квадратах отклонений (на рис. этим значениям соответствует интервал $[-c, c]$); для наблюдений, для которых абсолютные величины остатков больше c, вклад измеряется в более умеренных единицах – пропорционально модулю отклонения (на рисунке этим значениям соответствует интервал $(-\infty, c)$ и $(c, +\infty)$. Таким образом подход Хьюбера сочетает в себе МНК и метод

наименьших модулей.

Оценка Хьюбера является устойчивой в том случае, когда засоряющее распределение симметрично, в случае же асимметричных отклонений от нормального распределения она недостаточно устойчива. В последнем случае необходимо использовать функции, при увеличении аргумента растущие медленнее, чем функция Хьюбера.

Основная часть множества оценок, устойчивых к асимметричным отклонениям, имеют горизонтальную асимптоту.

Простейшей из оценок, устойчивых к асимметричным отклонениям, является обобщение усеченного среднего с функцией

$$\rho(z) = \begin{cases} z^2, \text{ если } |z| < c \\ c^2, \text{ если } |z| \geq c \end{cases}$$

Фактически, наблюдения, для которых абсолютные величины остатков больше порогового значения c, учитываются в минимизируемой функции вне зависимости от их значения.

2.4. Итеративный МНК

Для поиска значений М-оценок можно применять общие методы оптимизации: градиентный, метод Ньютона, метод сопряженных градиентов и т.д. Однако существует более простой метод минимизации – итеративный МНК (ИМНК), опирающийся на обобщенный (взвешенный) МНК. Идея его заключается в следующем. Рассмотрим систему уравнений (2):

$$\sum_{i=1}^{n} \psi\left(\frac{r_i(\hat{\theta})}{\sigma}\right) f_l(x_i) = 0, l = 1,\ldots,m,$$

где $\psi(z) = \rho'(z)$.

Перепишем ее левую часть в следующем виде:

$$\sum_{i=1}^{n} \frac{\psi\left(r_i(\hat{\theta})/\sigma\right)}{r_i(\hat{\theta})/\sigma} f_l(x_i) \frac{r_i(\hat{\theta})}{\sigma} = \sum_{i=1}^{n} \frac{1}{\sigma} \frac{\psi\left(r_i(\hat{\theta})/\sigma\right)}{r_i(\hat{\theta})/\sigma} f_l(x_i)\left(y_i - f_l(x_i)\hat{\theta}\right) =$$

$$= \sum_{i=1}^{n} \frac{1}{\sigma} w\left(r_i(\hat{\theta})/\sigma\right) f_l(x_i)\left(y_i - f_l(x_i)\hat{\theta}\right),\ l = 1,\ldots,m,$$

где $w(z) = \psi(z)/z$.

Переходя к матричным обозначениям и опуская константный сомножитель $1/\sigma$, мы можем переписать полученную систему следующим образом:

$$X^T W(\hat{\theta})\left(y - X\hat{\theta}\right) = 0,$$

$$\left[X^T W(\hat{\theta}) X\right]\hat{\theta} = X^T W(\hat{\theta}) y,$$

или

$$\hat{\theta} = \left[X^T W(\hat{\theta}) X\right]^{-1} X^T W(\hat{\theta}) y,$$

где $W(\hat{\theta})$ – диагональная матрица, i-й диагональный элемент которой равен $w\left(r_i(\hat{\theta}) \big/ \sigma\right)$, X – матрица значений регрессоров.

Таким образом, данная система уравнений соответствует схеме взвешенного МНК с весами, зависящими от $\hat{\theta}$.

В результате решение исходной задачи может быть найдено по следующей итерационной схеме:

$$\left[X^T W(\hat{\theta}^{(k-1)}) X\right]\hat{\theta}^{(k)} = X^T W(\hat{\theta}^{(k-1)}) y, \qquad (5)$$

где k – номер итерации. Решение СЛАУ (5) может быть записано в явном виде

$$\hat{\theta}^{(k)} = \left[X^T W(\hat{\theta}^{(k-1)}) X\right]^{-1} X^T W(\hat{\theta}^{(k-1)}) y.$$

Веса на каждой итерации «оцениваются» на основе оценки вектора параметров, полученной из предыдущей итерации. Окончательный вес, полученный на последней итерации, указывает, принадлежит ли данное наблюдение к выбросам.

В случае одновременного оценивания вектора параметров θ и параметра масштаба σ следует дополнить каждую итерацию ИМНК вычислением оценки σ, например, по формуле (4).

Укажем *достаточные условия сходимости* ИМНК:

1) $\rho(z) = \rho(-z)$,

2) $w(z) = \psi(z)/z$ – невозрастающая функция при $z \geq 0$.

Если функция потерь является выпуклой, то функция (1) имеет единственный минимум и в качестве начального приближения $\hat{\theta}$ можно взять МНК-оценку, что приводит к весам $w(z) = 1$ на начальной итерации. В качестве примера назовем L_p-оценку при $p > 1$, которой соответствует строго выпуклая функция потерь.

При невыпуклой функции потерь может иметься несколько локальных минимумов, соответствующих нескольким решениям системы (2) (по этой причине определения М-оценки через формулы (1) и (2) не эквивалентны). В силу этого выбор начального приближения здесь особенно важен. Начальное приближение из области, далекой от глобального экстремума, может привести к таким оценкам, которые соответствуют локальному минимуму и не обладают требуемыми устойчивыми свойствами. Не годятся, в частности, обычные МНК-оценки из-за их чувствительности к грубым ошибкам. В качестве начальных приближений нужно выбирать некоторые устойчивые оценки.

Алгоритм ИМНК

1. Выбрать начальное приближение $\hat{\theta}^{(1)}$, $\hat{\sigma}^{(1)}$, $k=1$.

2. Вычислить остатки $r_i(\hat{\theta}^{(k)}) = y_i - f^T(x_i)\hat{\theta}^{(k)}$. Вычислить веса по формуле $w\left(\frac{r_i(\hat{\theta}^{(k)})}{\hat{\sigma}^{(k)}}\right) = \psi\left(\frac{r_i(\hat{\theta}^{(k)})}{\hat{\sigma}^{(k)}}\right) \Big/ \frac{r_i(\hat{\theta}^{(k)})}{\hat{\sigma}^{(k)}}$.

3. Положить $k = k+1$. Вычислить очередное приближение $\hat{\theta}^{(k)}$ путем решения СЛАУ (5) каким-либо подходящим численным методом, затем вычислить $\hat{\sigma}^{(k)}$ по формуле (4).

4. Проверить выполнение условия останова $\max\limits_i \left|\frac{\hat{\theta}_i^{(k)} - \hat{\theta}_i^{(k-1)}}{\hat{\theta}_i^{(k-1)}}\right| < \delta$, где δ – малая величина, и если оно выполняется – остановиться. В противном случае перейти на шаг 2.

2.5. Метод псевдонаблюдений

В этом методе начальное значение вектора параметров модели регрессии $\hat{\theta}^{(1)}$, значение оценок отклика $\hat{y}_i^{(1)}$, остатков $r_i(\hat{\theta}^{(1)}) = y_i - f^T(x_i)\hat{\theta}^{(1)}$ вычисляем по МНК. Далее на последующих итерациях также применяем метод наименьших квадратов, но используя уже псевдонаблюдения $\hat{y}_i^{(2)} = \hat{y}_i^{(1)} + r_i^*(\hat{\theta}^{(1)})$, где остатки $r_i^*(\hat{\theta}^{(1)})$ вычисляются исходя из используемой функции потерь.

Проиллюстрируем данный подход на примере функции потерь Хьюбера:

$$\rho(r) = \begin{cases} \frac{1}{2}\left|\frac{r}{\sigma}\right|^2, & \left|\frac{r}{\sigma}\right| \le c \\ c\left|\frac{r}{\sigma}\right| - \frac{c^2}{2}, & \left|\frac{r}{\sigma}\right| > c \end{cases}$$

Или освобождаясь от нормирующего множителя ½:

$$\rho(r) = \begin{cases} \left|\frac{r}{\sigma}\right|^2, & \left|\frac{r}{\sigma}\right| \le c \\ 2c\left|\frac{r}{\sigma}\right| - c^2, & \left|\frac{r}{\sigma}\right| > c \end{cases}$$

Здесь σ – робастная оценка стандартного отклонения, посчитанная, например, через медиану абсолютных отклонений. Применительно к функции потерь Хьюбера элементы вектора псевдонаблюдений вычисляются как

$$y_i^{(2)} = \begin{cases} y_i, & \left| r_i(\hat{\theta}^{(1)}) \right| \le c\sigma \\ \hat{y}_i^{(1)} - r_i^*(\hat{\theta}^{(1)}), & r_i(\hat{\theta}^{(1)}) < -c\sigma \\ \hat{y}_i^{(1)} + r_i^*(\hat{\theta}^{(1)}), & r_i(\hat{\theta}^{(1)}) > c\sigma \end{cases},$$

где скорректированные остатки вычисляются как квадратный корень из значения функции потерь Хьюбера для линейной зоны: $r_i^*(\hat{\theta}^{(1)}) = \sqrt{2cs\left| r_i(\hat{\theta}^{(1)}) \right| - c^2\sigma^2}$. В приведенных выше выражениях $\hat{y}_i^{(1)}$ это значение отклика, рассчитанное по текущей модели. После каждой итерации вычисления оценок модель изменяется, меняются соответственно остатки и вектор псевдонаблюдений. Останов процедуры осуществляется после стабилизации процесса с использованием правила из п. 4 алгоритма ИМНК.

3. Содержание работы

1. Изучить методы устойчивого оценивания параметров регрессии и метод поиска значений оценок.

2. Выбрать полиномиальную невысокого порядка (квадратичную, кубическую) модель зависимости отклика *y* от одного фактора *x*. По данной модели сгенерировать экспериментальные данные, содержащие выбросы. Выбросы можно смоделировать, увеличив в несколько раз (в несколько десятков раз) величину ошибки в нескольких точках выборки. Проконтролировать наличие выбросов в выборке визуально. Сформировать несколько выборок с различной степенью засорения. Степень засорения варьировать от 1 до 25% с шагом в 4-5%.

3. Разработать программу, реализующую поиск М-оценок параметров итерационным МНК или по методу псевдонаблюдений для функции потерь, указанной в варианте.

4. Выбрать несколько значений параметра функции потерь (включая указанные в варианте) и найти значения М-оценок для каждого из них. В качестве начального приближения оценок в алгоритме итерационного МНК можно взять истинные значения параметров или можно произвести поиск глобального оптимума путем многократного применения итерационного МНК из различных начальных приближений. Вычислить МНК-оценку. Сравнить качество всех полученных оценок по величинам $(\hat{\theta} - \theta)^T(\hat{\theta} - \theta)$, $MSE = (\frac{1}{n}\sum_{i=1}^{n}(\hat{y}(x_i) - r(x_i))^2)^{1/2}$, где $r(x_i)$ - незашумленный выход объекта. Выбрать значение параметра функции потерь, дающее наилучшее качество при заданной степени засорения.

5. Оформить отчет, включающий в себя

- постановку задачи;
- полученный набор данных и значения ошибок наблюдений;

– оценки параметров и результаты их сравнения;
  1) для наилучшей из оценок: значения весов наблюдений w на последней итерации, значения $y, \hat{y}, y - \hat{y}$, оценку параметра σ;
– графики, отражающие качество оценивания (один из двух вариантов):
  1) график зависимостей от фактора x измеренных значений y и прогнозных значений $\hat{y}$ для МНК-оценки и для наилучшей из М-оценок (пример графика см. на рис. 2.1).
  2) два графика – график наблюдений и график зависимостей от фактора x истинных значений отклика и прогнозных значений $\hat{y}$ для МНК-оценки и для наилучшей из М-оценок
– текст программы.

5. Защитить лабораторную работу.

4. Варианты заданий

Варианты заданий перечислены в таблице на последней странице. В последнем столбце указаны некоторые возможные значения параметра c функции $\rho(z)$. В качестве алгоритма вычисления М-оценок возможно использовать ИМНК либо метод псевдонаблюдений.

5. Контрольные вопросы

1. МНК и устойчивость
2. М-оценки
3. Робастные оценки
4. Итеративный МНК
5. Метод псевдонаблюдений
6. Виды функций потерь

Литература

1. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Исследование зависимостей. – М.: Финансы и статистика, 1985.
2. Вучков И., Бояджиева Л., Солаков Е. Прикладной линейный регрессионный анализ. – М.: Финансы и статистика, 1987.
3. Демиденко Е.З. Линейная и нелинейная регрессии. – М.: Финансы и статистика, 1981.
4. Дрейпер Н., Смит Г. Прикладной регрессионный анализ, 3-е изд. – М.: Издательский дом «Вильямс», 2007.
5. Хьюбер Дж.П. Робастность в статистике. – М.: Мир, 1984.

Таблица. Виды функций потерь

№	Название	$\rho(z)$	$w(z)$	Параметр
1	Хьюбера	$\begin{cases}\frac{1}{2}z^2, \lvert z\rvert < c \\ c\lvert z\rvert - c^2/2, \lvert z\rvert \geq c\end{cases}$	$\begin{cases}1, \lvert z\rvert < c \\ c/\lvert z\rvert, \lvert z\rvert \geq c\end{cases}$	1.345 1.5 3
2	Эндрюса	$\begin{cases}c(1-\cos\frac{z}{c}), \lvert z\rvert < \pi c \\ 2c, \lvert z\rvert \geq \pi c\end{cases}$	$\begin{cases}\frac{1}{z}\sin\frac{z}{c}, \lvert z\rvert < \pi c \\ 0, \lvert z\rvert \geq \pi c\end{cases}$	1.339 1.5 1.91 2.1
3	Рамсея	$\frac{1-(1+c\lvert z\rvert)\exp\{-c\lvert z\rvert\}}{c^2}$	$\exp\{-c\lvert z\rvert\}$	0.3 1/3 0.5
4	биквад-ратная Тьюки	$\begin{cases}\frac{z^6}{6c^4} - \frac{z^4}{2c^2} + \frac{z^2}{2}, \lvert z\rvert < c \\ \frac{c^2}{6}, \lvert z\rvert \geq c\end{cases}$	$\begin{cases}\left[1-\left(\frac{z}{c}\right)^2\right]^2, \lvert z\rvert < c \\ 0, \lvert z\rvert \geq c\end{cases}$	4.685 6
5	Меррилла-Швеппе	$\begin{cases}z^2/2, \ \lvert z\rvert < c \\ 2\sqrt{c^3\lvert z\rvert} - 3c^2/2, \ \lvert z\rvert \geq c\end{cases}$	$\begin{cases}1, \ \lvert z\rvert < c \\ \sqrt{(c/\lvert z\rvert)^3}, \ \lvert z\rvert \geq c\end{cases}$	1.5 3
6	Демиденко	$\frac{z^2}{c+\lvert z\rvert}$	$\frac{2c+\lvert z\rvert}{(c+\lvert z\rvert)^2}$	3
7	Демиденко (с горизон-тальной асимптотой)	$\frac{z^2}{c+z^2}$	$\frac{1}{(c+z^2)^2}$	3 6
8	Мешалкина	$-\exp\left\{-\frac{cz^2}{2}\right\}$	$\exp\left\{-\frac{cz^2}{2}\right\}$	1/9 0.5 1
9	Смита	$\begin{cases}\frac{z^2}{2} - \frac{z^4}{4c^2}, \ \lvert z\rvert < c \\ \frac{c^2}{4}, \ \lvert z\rvert \geq c\end{cases}$	$\begin{cases}1 - \frac{z^2}{c^2}, \ \lvert z\rvert < c \\ 0, \ \lvert z\rvert \geq c\end{cases}$	3 6
10	Бернулли	$\begin{cases}1-\left[1-\frac{z^2}{c^2}\right]^{\frac{3}{2}}, \ \lvert z\rvert < c \\ 0, \ \lvert z\rvert \geq c\end{cases}$	$\begin{cases}\frac{3}{c^2}\left[1-\frac{z^2}{c^2}\right]^{\frac{1}{2}}, \ \lvert z\rvert < c \\ 0, \ \lvert z\rvert \geq c\end{cases}$	3 6

Лабораторная работа №3

Построение регрессионных моделей на основе концепции нечетких систем «Такаги-Сугено

1. Цель работы

Построить регрессионную зависимость с использованием моделей типа «Такаги-Сугено».

2. Методические указания

Пусть имеется база правил

$$ЕСЛИ\ \ (x_1 \in A_{1i})\ и\ (x_2 \in A_{2i})\ и \ldots и\ (x_k \in A_{ki})\ \ ТО\ \ y = \eta^i(x),\ \ i = 1,\ \ldots,\ M, \qquad (1)$$

где A_{ji} – нечеткое подмножество для переменной x_j с функцией принадлежности $\mu_{A_{ji}}(x_j)$; M – число правил, $\eta^i(x)$ – функция, определяющая локальную зависимость отклика y от набора регрессоров $x = (x_1, \ldots, x_k)^T$, например, линейная модель: $\eta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_k x_k$.

Прогнозное значение для отклика y определяется обычно по методу центра масс:

$$\hat{y}(x) = \frac{\sum_{i=1}^{M} \mu_i \eta^i(x)}{\sum_{i=1}^{M} \mu_i}, \qquad \text{где}\ \ \mu_i = \prod_{j=1}^{k} \mu_{A_{ji}}(x_j). \qquad (2)$$

Модель вида (1)-(2) называется FLR (Fuzzy Logic Regression) регрессионной моделью [2].

В случае ***симметричной треугольной функции*** $\mu(x)$ расчетная формула выглядит следующим образом:

$$\mu(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{c-a}, & a < x \le c \\ \dfrac{b-x}{b-c}, & c < x < b \\ 0, & x \ge b. \end{cases}$$

В случае ***трапециевидной функции*** μ_i:

$$\mu(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{c-a}, & a < x < c \\ 1, & c \le x \le d \\ \dfrac{b-x}{b-d}, & d < x < b \\ 0, & x \ge b. \end{cases}$$

В целях избежание необходимости нормировки в (2) будем считать, что функции принадлежности обладают тем свойством, что в любой точке x выполняется условие

$$\sum_{i=1}^{M} \mu_{A_i}(x) = 1. \qquad (3)$$

Таким образом, там, где соседние μ_i и μ_{i+1} имеют ненулевые значения всегда выполняется равенства $\mu_{i+1}=1-\mu_i$. На рисунках 1-2 приведены примеры используемых функций принадлежности $\mu_{A_i}(x)$:

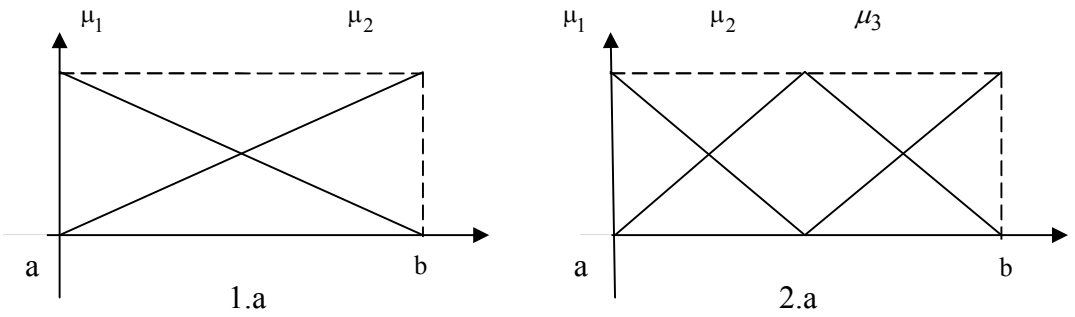

Рисунок 1. Примеры треугольных функций принадлежности (1.а – 2 ФП, 1.б - 3 ФП)

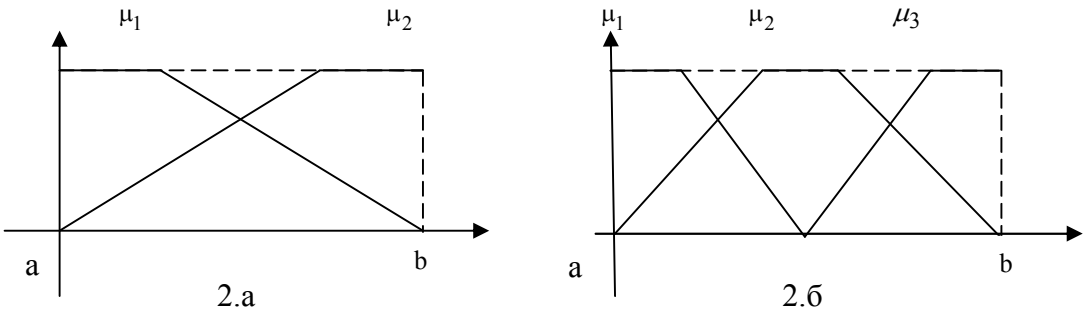

Рисунок 2. Примеры трапециевидных функций принадлежности (2.а – 2 ФП, 2.б - 3 ФП)

Для одномерного случая функции $\eta^i(x)$ приобретают вид

$$\eta^i(x) = \theta_0^i + \theta_1^i x, \quad i = 1, ..., M.$$

Будем считать, что регрессия y по x подчиняется следующему уравнению наблюдения

$$y_j = \sum_{i=1}^{M} (\theta_0^i + \theta_1^i x_j)\mu_{A_i}(x) + e, \quad j = 1, \ ..., \ N \quad , \qquad (4)$$

где e - случайная величина, центрированная и с конечной дисперсией.

Оценивать неизвестные параметры, входящие в (4) будем по глобальному методу наименьших квадратов (МНК). Во избежание возможных проблем с линейной зависимостью столбцов матрицы наблюдений будем использовать **трапецевидные функции принадлежности**. Для трапециевидных функций следующие регрессоры:

$$\mu_{A_1}(x),\ldots,\ \mu_{A_M}(x),\ x\mu_{A_1}(x),\ldots,\ x\mu_{A_M}(x). \qquad (5)$$

Ограничимся рассмотрением вариантов, когда область определения по каждой независимой переменной задана отрезком -1.+1, а число нечетких партиций равно 2 или 3. Для случая двух партиций функции принадлежности будут иметь, например, вид:

$\mu_1(x) = \{1, x \le -0.5; 1/2 - x, -0.5 \le x \le 0.5; 0, \text{x} > 0.5\ \}$, $\mu_2(x) = 1 - \mu_1(x)$,

а для случая трех партиций -

$\mu_1(x) = 1, x \le -0.75, \mu_1(x) = -1/2 - 2x, -0.75 \le x \le -0.25;\ \mu_1(x) = 0, x > -0.25,$

$\mu_2(x) = 0, x < -0.75, \mu_2(x) = 3/2 + 2x, -0.75 \le x \le -0.25;\ \mu_2(x) = 1, -0.25 < x \le 0.25,$

$\mu_2(x) = 3/2 - 2x, 0.25 < x \le 0.75, \mu_2(x) = 0, x > 0.75$

$\mu_3(x) = 0, x \le 0.25, \mu_3(x) = 2x - 0.5, 0.25 \le x \le 0.75,\ \mu_3(x) = 1, x > 0.75$

3. Содержание работы

1. Ознакомиться с понятиями нечеткого множества, функции принадлежности, методологией построения регрессионных моделей в рамках концепции нечетких систем;
2. Провести моделирование экспериментальных данных в соответствии с вариантом задания;
3. По полученным экспериментальным данным построить регрессионную модель с использованием разбиения областей определения входных факторов на нечеткие партиции.
4. Оформить отчет, включающий в себя постановку задачи, полученные результаты и выводы;
5. Защитить лабораторную работу.

4.Варианты заданий

1. Моделируемая зависимость одномерная, близкая к кусочно линейной. При восстановлении регрессионной зависимости использовать варианты разбиения области определения входной переменной на две и три партиции. В качестве локальной модели $\eta^i(x)$ использовать линейную модель. Выбрать оптимальную результирующую модель;
2. Моделируемая зависимость одномерная, близкая к параболической, но левая ее ветка имеет форму близкую к прямой линии. При восстановлении регрессионной зависимости использовать варианты разбиения области определения входной переменной на две и три

партиции. В качестве локальной модели $\eta^i(x)$ использовать квадратичную модель. Выбрать оптимальную результирующую модель;

3. Моделируемая зависимость одномерная, близкая к сигмоидальной. При восстановлении регрессионной зависимости использовать варианты разбиения области определения входной переменной на две и три партиции. В качестве локальной модели $\eta^i(x)$ использовать линейную модель или квадратичную. Выбрать оптимальную результирующую модель;
4. Моделируемая зависимость двумерная, близкая к кусочно-линейной. При восстановлении регрессионной зависимости использовать варианты разбиения области определения входных переменных на две и три партиции. В качестве локальной модели $\eta^i(x)$ использовать линейную модель. Выбрать оптимальную результирующую модель;
5. Моделируемая зависимость двумерная колокообразного вида, но в отрицательном квадранте поверхность отклика близка к линейной, а в положительном к квадратичной. При восстановлении регрессионной зависимости использовать варианты разбиения области определения входной переменной на две и три партиции. В качестве локальной модели $\eta^i(x)$ использовать квадратичную модель. Выбрать оптимальную результирующую модель;

5.Контрольные вопросы:

1. Функция принадлежности. Виды;
2. В чем состоит фаззификация входных переменных и дефаззификация выходной переменной;
3. Глобальный и локальный вариант МНК для оценивания параметров модели Такаги-Сугено;
4. Алгоритмы нечеткой кластеризации, используемые при построении нечетких регрессионных моделей.

Литература

1. Пегат А. Нечеткое моделирование и управление / А.Пегат ; пер. с англ. – 2-е изд. – М. : БИНОМ. Лаборатория знаний, 2013. – 798 с.
2. Попов А. А. Регрессионное моделирование на основе нечетких правил / А. А. Попов // Сборник научных трудов НГТУ. – 2000. – № 2 (19). – С. 49–57.

Лабораторная работа №4

Построение регрессионных моделей по методу LS SVM

1. Цель работы

Получить практические навыки по построению регрессионных зависимостей с использованием метода опорных векторов с квадратичной функцией потер.

2. Методические указания

Одним из методов непараметрического оценивания является *метод опорных векторов с квадратичной функцией потерь (LS-SVM)*, принадлежащий к классу ядерных. Этот метод представляет собой модификацию алгоритма опорных векторов (SVM).

Изначально SVM использовался в задачах классификации, но в 1995 году был обобщен В. Вапником для оценивания действительных функций. Позднее, Дж. Сайкенсом было предложено расширение SVM с использованием квадратичной функции потерь, которое и получило название LS-SVM.

Введем следующие обозначения:

$X \in \mathsf{X} \subset \mathbb{R}^d$ – действительный случайный входной вектор;

$Y \in \mathsf{Y} \subset \mathbb{R}$ – действительная случайная переменная отклика;

$\Psi \in \mathbb{R}^{nf}$ – определяет пространство функций высокой размерности.

Ключевая составляющая метода опорных векторов заключается в следующем: он отображает случайный входной вектор в пространство признаков высокой размерности Ψ через некоторое нелинейное отображение $\varphi : \mathsf{X} \to \Psi$. В этом пространстве мы рассмотрим класс линейных функций

$$\mathcal{F}_\Psi = \left\{ f : f(x) = \omega^T \varphi(x) + b : \varphi : \mathsf{X} \to \Psi, \omega \in \mathbb{R}^{nf}, b \in \mathbb{R} \right\}. \quad (1)$$

Однако даже если линейная функция в пространстве признаков (1) хорошо обобщает и теоретически может быть найдена, проблема работы с пространством функций высокой размерности остается. Отметим, что для построения линейной функции (1) в пространстве признаков Ψ отсутствует необходимость рассмотрения пространства признаков в явном виде. Достаточно заменить скалярное произведение в пространстве признаков $\varphi(x_k)^T \varphi(x_l)$ соответствующим ядром $K(x_k, x_l)$, удовлетворяющим *условиям Мерсера*, так называемый «kernel trick».

Теорема Мерсера.

Пусть $K \in L^2(C)$, $g \in L^2(C)$, где C является компактным подмножеством $\mathbb{R}^d$ и $K(t,z)$ описывает скалярное произведение в некотором пространстве признаков. Чтобы гарантировать, что непрерывная симметричная функция *K* имеет представление

$$K(t,z)=\sum_{k=1}^{\infty}\alpha_k\phi_k(t)\phi_k(z)$$

с положительными коэффициентами $\alpha_k>0$, (т.е. $K(t,z)$ описывает скалярное произведение в некотором пространстве признаков), необходимо и достаточно, чтобы выполнялось условие

$$\int_C\int_C K(t,z)g(t)g(z)\partial t\partial z\geq 0$$

действительное для всех $g\in L^2(C)$.

Предположим, что $\omega=\sum_{k=1}^{n}\beta_k\varphi(x_k)$ и на основе *теоремы Мерсера*, класс линейных функций в пространстве признаков (1) имеет следующее эквивалентное представление во входном пространстве X :

$$\mathcal{F}_X=\left\{f: f(x)=\sum_{k=1}^{n}\beta_k K(x,x_k)+b : b\in\mathbb{R},\beta_k\in\mathbb{R}\right\}, \tag{2}$$

где x_k – векторы и $K(x,x_k)$ – функция, удовлетворяющая *условию Мерсера*.

Существует ряд ядер, которые могут быть использованы в моделях метода опорных векторов:

Линейные: $K(x,z)=x^Tz+c$.

Полиномиальное: $K(x,z)=(ax^Tz+c)^d$.

Радиально базисная функция (RBF): $K(x,z)=\exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$.

Сигмоидальная («нейронная») функция: $K(x,z)=tanh\left(ax^T\cdot z+c\right)$.

Рассмотрим теперь случай, когда в описании функции присутствует шум. Дана обучающая выборка $\mathcal{D}_n=\{(x_k,y_k): x_k\in X, y_k\in Y; k=1,\ldots,n\}$ объема *n* независимых наблюдений с неизвестным распределением F_{XY}, согласно

$$y_k=m(x_k)+e_k, k=1,\ldots,n, \tag{3}$$

где $e_k\in\mathbb{R}$ будем считать независимо и одинаково распределенной ошибкой с $E[e_k\mid X=x_k]=0$ и $Var[e_k]=\sigma^2<\infty$, $m(x)\in\mathcal{F}_\Psi$ – неизвестная действительная гладкая функция и $E[y_k\mid x=x_k]=m(x_k)$. Нашей целью является поиск параметров ω и b исходного пространства, которые минимизируют эмпирический функционал риска

$$\mathcal{R}_{emp}(\omega,b)=\frac{1}{n}\sum_{k=1}^{n}\left((\omega^T\varphi(x_k)+b)-y_k\right)^2 \tag{4}$$

с ограничением $\|\omega\|_2\leq a, a\in\mathbb{R}_+$. Задачу оптимизации нахождения вектора ω и $b\in\mathbb{R}$ можно свести к решению следующей задачи оптимизации

$$\min_{\omega,b,e}\mathcal{J}(\omega,e)=\frac{1}{2}\omega^T\omega+\frac{1}{2}\gamma\sum_{k=1}^{n}e_k^2, \tag{5}$$

такой, что

$$y_k = \omega^T \varphi(x_k) + b + e_k, k = 1,\ldots,n .$$

Стоит заметить, что относительная значимость слагаемых функция затрат $\mathcal{J}$ определяется положительной *действительной константой* γ. В случае зашумленных данных может возникнуть ситуация *переобучения* (нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке). Можно избежать переобучения, принимая меньшее значение γ.

Для решения задачи оптимизации в двойственном пространстве, определим функционал Лагранжа

$$\mathcal{L}(\omega, b, e; \alpha) = \mathcal{J}(\omega, e) - \sum_{k=1}^{n} \alpha_k (\omega^T \varphi(x_k) + b + e_k - y_k), \tag{6}$$

с лагранжевыми множителями $\alpha_k \in \mathbb{R}$.

Условия оптимальности задаются следующим образом:

$$\begin{cases} \dfrac{d\mathcal{L}}{d\omega} = 0 \to \omega = \sum_{k=1}^{n} \alpha_k \varphi(x_k) \\ \dfrac{d\mathcal{L}}{db} = 0 \to \sum_{k=1}^{n} \alpha_k = 0 \\ \dfrac{d\mathcal{L}}{de_k} = 0 \to \alpha_k = \gamma e_k, \qquad k = 1,\ldots,\mathrm{n} \\ \dfrac{d\mathcal{L}}{da_k} = 0 \to \omega^T \varphi(x_k) + b + e_k = y_k, k = 1,\ldots,\mathrm{n} \end{cases} \tag{7}$$

После исключения ω и e, получаем решение

$$\begin{bmatrix} 0 & 1_n^T \\ 1_n & \Omega + \dfrac{1}{\gamma}\mathrm{I}_n \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{8}$$

где $y = (y_1,\ldots,y_n)^T, 1_n = (1,\ldots,1)^T, \alpha = (\alpha_1,\ldots,\alpha_n)$ и $\Omega_{kl} = \varphi(x_k)^T \varphi(x_l)$ для $k,l = 1,\ldots,\mathrm{n}$. Согласно теореме Мерсера, результирующая LS-SVM модель для оценки функции имеет вид

$$\hat{m}_n(x) = \sum_{k=1}^{n} \hat{\alpha}_k K(x, x_k) + \hat{b}, \tag{9}$$

где $\hat{\alpha}, \hat{b}$ являются решением (8)

$$\hat{b} = \frac{1_n^T \left(\Omega + \dfrac{1}{\gamma}\mathrm{I}_n\right)^{-1} y}{1_n^T \left(\Omega + \dfrac{1}{\gamma}\mathrm{I}_n\right)^{-1} 1_n}.$$

$$\hat{\alpha} = \left(\Omega + \frac{1}{\gamma} \mathrm{I}_n \right)^{-1} (y - 1_n \hat{b}).$$

Как и во многих других алгоритмах, в методе опорных векторов с квадратичной функцией потерь на точность итоговой модели оказывает влияние *ряд задаваемых параметров*. К таким параметрам можно отнести:

- тип ядерной функции, каждая из которых так же обладает собственным набором параметров;
- коэффициент γ.

Использование автоматизированного метода подбора параметров позволяет значительно упростить этот процесс настройки и повысить качество работы алгоритма. Порядок, в котором происходит настройка параметров, так же оказывает влияние на качество работы алгоритма.

При решении задачи выбора «наилучшей» модели регрессии из числа построенных при различных наборах задаваемых внутренних параметров, возникает необходимость использования того или иного *критерия качества*.

Одним из критериев, который часто используется при построении регрессионных зависимостей, является *процедура скользящего контроля*, представляющая собой стандартную методику тестирования и сравнения алгоритмов классификации, регрессии и прогнозирования.

Скользящий контроль или *кросс-проверка (cross-validation, CV)* – процедура эмпирического оценивания обобщающей способности алгоритмов, обучаемых по прецедентам.

Фиксируется некоторое множество разбиений исходной выборки на две подвыборки: *обучающую* и *контрольную*. Для каждого разбиения выполняется настройка алгоритма по обучающей подвыборке, затем оценивается его средняя ошибка на объектах контрольной подвыборки. Оценкой скользящего контроля называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.

Перекрестная проверка имеет два основных преимущества перед применением одного множества для обучения и одного для тестирования модели:

- распределение классов оказывается более равномерным, что улучшает качество обучения;
- если выборка независима, то средняя ошибка скользящего контроля даёт несмещённую оценку вероятности ошибки. Это выгодно отличает её от средней ошибки на обучающей выборке, которая может оказаться смещённой (оптимистически заниженной) оценкой вероятности ошибки, что связано с явлением переобучения.

На практике скользящий контроль чаще всего применяется либо для выбора одной модели алгоритмов из нескольких (model selection), либо для оптимизации небольшого числа параметров, определяющих структуру алгоритма, таких, как степень полинома, параметр регуляризации или количество нейронов на скрытом уровне нейронной сети.

Рассматривается задача обучения с учителем.

Пусть X – множество описаний объектов, Y – множество допустимых ответов. Задана конечная выборка прецедентов $X^l = (x_i, y_i)_{i=1}^l \subset X \times Y$.

Задан алгоритм обучения – отображение μ, которое произвольной конечной выборке прецедентов X^m ставит в соответствие функцию (алгоритм) $\alpha : X \to Y$.

Качество алгоритма α оценивается по произвольной выборке прецедентов X^m с помощью функционала качества $Q(\alpha, X^m)$. Для процедуры скользящего контроля не важно, как именно вычисляется этот функционал. Как правило, он аддитивен по объектам выборки:

$$Q(\alpha, X^m) = \frac{1}{m} \sum_{x_i \in X^m} L(\alpha(x_i), y_i),$$

где $L(\alpha(x_i) y_i)$ неотрицательная функция потерь, возвращающая величину ошибки ответа алгоритма $\alpha(x_i)$ при правильном ответе y_i.

Процедура скользящего контроля

Выборка X^l разбивается N различными способами на две непересекающиеся подвыборки: $X^l = X_n^m \cup X_n^k$, где X_n^m – обучающая подвыборка длины m, X_n^k – контрольная подвыборка длины $k = l - m$, где $n = \overline{1, N}$ – номер разбиения.

Для каждого разбиения n строится алгоритм $\alpha_n = \mu(X_n^m)$ и вычисляется значение функционала качества $Q_n = Q(\alpha_n, X_n^k)$. Среднее арифметическое значений Q_n по всем разбиениям называется *оценкой скользящего контроля*:

$$CV(\mu, X^l) = \frac{1}{N} \sum_{n=1}^{N} Q(\mu(X_n^m) X_n^k),$$

Оценка скользящего контроля предполагает, что алгоритм обучения μ уже задан. Она ничего не говорит о том, какими свойствами должны обладать «хорошие» алгоритмы обучения, и как их строить.

Возможны различные варианты скользящего контроля, отличающиеся видами функционала качества и способами разбиения выборки.

Контроль по отдельным объектам

Контроль по отдельным объектам (leave-one-out CV, LOO) является частным случаем полного скользящего контроля при $k = 1$, соответственно, $N = l$. Это самый распространённый вариант скользящего контроля.

Преимущества LOO в том, что каждый объект ровно один раз участвует в контроле, а длина обучающих подвыборок лишь на единицу меньше длины полной выборки.

Недостатком LOO является большая ресурсоёмкость, задачу обучения приходится решать l раз, что сопряжено со значительными вычислительными затратами. Одним из вариантов ускорения является использование формулы *быстрого контроля по отдельным объектам (fast leave-one-out, fast LOO),* которая была введена для LS-LVM классификаторов.

3. Содержание работы

1. Ознакомиться с теоретическими основами метода опорных векторов в модификации, связанной с использованием квадратичной функции потерь (LS SVM).
2. Провести моделирование экспериментальных данных в соответствии с вариантом задания.
3. По полученным экспериментальным данным построить регрессионную модель по методу LS SVM.
4. Оформить отчет, включающий в себя постановку задачи, полученные результаты и выводы.
5. Защитить лабораторную работу.

4. Варианты заданий

6. Моделируемая зависимость одномерная: $u(x) = x + 4e^{-2x^2} / \sqrt{2\pi}$. Интервал варьирования фактора $x \in [-4,+4]$. Использовать гауссово ядро.
7. Моделируемая зависимость одномерная: $u(x) = 2x + 10e^{-(x-4)^2/0.25}$. Интервал варьирования фактора $x \in [0,+8]$. Использовать гауссово ядро.
8. Моделируемая зависимость одномерная: $u(x) = 1.5x - 3e^{-(x-4.5)^2/0.15}$. Интервал варьирования фактора $x \in [2,10]$. Использовать гауссово ядро.
9. Моделируемая зависимость одномерная: $u(x) = x - 3e^{-(x)^2/0.15} + e^{-(x-1)^2/0.1}$. Интервал варьирования фактора $x \in [-1,2]$. Использовать гауссово ядро.
10. Моделируемая зависимость одномерная: $u(x) = x + 4e^{-2x^2} / \sqrt{2\pi}$. Интервал варьирования фактора $x \in [-4,+4]$. Использовать полиномиальное ядро.
11. Моделируемая зависимость одномерная: $u(x) = 2x + 10e^{-(x-4)^2/0.25}$. Интервал варьирования фактора $x \in [0,+8]$. Использовать полиномиальное ядро.
12. Моделируемая зависимость одномерная: $u(x) = 1.5x - 3e^{-(x-4.5)^2/0.15}$. Интервал варьирования фактора $x \in [2,10]$. Использовать полиномиальное ядро.
13. Моделируемая зависимость одномерная: $u(x) = x - 3e^{-(x)^2/0.15} + e^{-(x-1)^2/0.1}$. Интервал варьирования фактора $x \in [-1,2]$. Использовать полиномиальное ядро.

5. Контрольные вопросы:

1. Целевая функция LS SVM в исходном пространстве;
2. Целевая функция LS SVM в двойственном пространстве;
3. Условия оптимальности и решение по LS SVM;
4. Виды ядерных функций.

Литература

1. К. В. Воронцов Математические методы обучения по прецедентам (теория обучения машин) http://www.ccas.ru/voron voron@ccas.ru
2. Мерков, Александр Борисович. Распознавание образов: введение в методы статистического обучения / А.Б. Мерков ; Рос. акад. наук, Ин-т систем. анализа .— Москва : УРСС = URSS, 2010 .— 254 с.
3. Johan A.K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, Joos Vandewalle Least Square Support Vector Machines / - World Scientific, New Jersey-London-Singapore-Hong Kong, 2002. – 290 p.
4. Wiki-портал http://www.machinelearning.ru

Лабораторная работа №5

Классификация по методу SVM

1. Цель работы

Получить практические навыки по решению задачи классификации с использованием метода опорных векторов.

2. Методические указания

Объем тестовой части выборки выбирать равным 1/5 части объема общей выборки. Выбор объектов, относимых к тестовой части осуществлять по принципу случайности, обеспечивая необходимую пропорциональность наличие в ней представителей разных классов.

Настройку внутренних параметров алгоритма SVM осуществлять в ручном режиме, ориентируясь на значение ошибки на тестовой выборке.

Для решения задачи классификации по методу SVM использовать свободно распространяемое программное обеспечение. В случае использования метода LS SVM реализовать его самостоятельно, пользуясь соответствующими наработками для решения задачи восстановления зависимостей по одноименному методу.

Оптимальная разделяющая гиперплоскость в методе LS SVM находится через решение следующей задачи оптимизации:

$$\min_{w,b,e} J_P(w,e) = \frac{1}{2}\|w\|^2 + \gamma \frac{1}{2}\sum_{k=1}^{N} e_k^2 \qquad (1)$$

при условии

$$y_k\left[w^T\varphi(x_k)+b\right] = 1 - e_k, \quad k = \overline{1,N}. \qquad (2)$$

Классификатор в исходном пространстве имеет вид:

$$y(x) = sign\left[w^T\varphi(x)+b\right], \qquad (3)$$

где $\varphi(x)$ – некоторая заданная вектор-функция.

Формулировка задачи (1)-(3) приведена для случая двухклассовой классификации, где значения отклика кодируются {+1, -1}. К целевому значению добавляется переменная ошибки e, что подразумевает возможность

ошибочной классификации. По теореме Куна — Таккера задача (1)-(2) эквивалентна двойственной задаче поиска седловой точки функции Лагранжа:

$$\mathcal{L}(w,b,e;\alpha) = J_P(w,e) - \sum_{k=1}^{N} \alpha_k \left\{ y_k \left[w^T \varphi(x_k) + b \right] - 1 + e_k \right\} , \qquad (4)$$

где $\alpha -$ это коэффициенты Лагранжа, которые могут быть как положительным, так и отрицательными из-за ограничений в виде равенства.

Условия оптимальности:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w} = 0 \to w = \sum\limits_{k=1}^{N} \alpha_k y_k \varphi(x_k), \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \to \sum\limits_{k=1}^{N} \alpha_k y_k = 0, \\ \dfrac{\partial \mathcal{L}}{\partial e_k} = 0 \to \alpha_k = \gamma e_k, & k = \overline{1,N} \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \to y_k \left[w^T \varphi(x_k) + b \right] - 1 + e_k = 0, & k = \overline{1,N} \end{cases} \qquad (5)$$

После исключения w, e из системы уравнений (5) получаем СЛАУ:

$$\begin{bmatrix} 0 & y^T \\ y & \Omega + \dfrac{1}{\gamma} \mathrm{I}_N \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_N \end{bmatrix}, \qquad (6)$$

где $y = [y_1,...,y_N], \alpha = [\alpha_1,...,\alpha_N], 1_N = [1,...,1], \quad \Omega_{kl} = y_k y_l K(x_k,x_l), k,l = \overline{1,N}$, $K(x_k,x_l) = \varphi^T(x_k)\varphi(x_l), \quad k,l = \overline{1,N}$.

3. Содержание работы

1. Ознакомиться с теоретическими основами метода опорных векторов (SVM), а также метода LS SVM для решения задачи классификации.
2. Осуществить выбор прикладного набора данных двухклассовой классификации из репозитариев данных.
3. Сформировать обучающую и тестовую части выборки.
4. Провести классификацию данных с настройкой внутренних параметров алгоритма SVM по ошибке на тестовой части выборки.
5. Оформить отчет, включающий в себя постановку задачи, полученные результаты и выводы. Отчет должен также включать описание использованного набора данных, способа доступа к нему. Отдельным разделом в отчете должно идти описание используемого программного обеспечения, его инсталляцию, сценарий использования.
6. Защитить лабораторную работу.

4. Контрольные вопросы

1. Оптимальная разделяющая гиперплоскость.
2. Понятие зазора между классами (margin).
3. Случаи линейной разделимости и отсутствия линейной разделимости.
4. Связь с минимизацией регуляризованного эмпирического риска.
5. Кусочно-линейная функция потерь.
6. Задача квадратичного программирования и двойственная задача. Понятие опорных векторов.
7. Функция ядра (kernel functions), спрямляющее пространство. Способы конструктивного построения ядер. Примеры ядер.
8. Классификация с использованием LS SVM классификатора.

Литература

1. К. В. Воронцов Математические методы обучения по прецедентам (теория обучения машин) http://www.ccas.ru/voron voron@ccas.ru
2. Мерков, Александр Борисович. Распознавание образов: введение в методы статистического обучения / А.Б. Мерков ; Рос. акад. наук, Ин-т систем. анализа .— Москва : УРСС = URSS, 2010 .— 254 с.
3. Johan A.K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, Joos Vandewalle Least Square Support Vector Machines / - World Scientific, New Jersey-London-Singapore-Hong Kong, 2002. – 290 p.
4. Wiki-порталhttp://www.machinelearning.ru

Свободное ПО в области машинного обучения общего назначения

Weka 3• – Data Mining Software in Java (разработана командой специалистов Университета Вайкато, Новая Зеландия); http://www.cs.waikato.ac.nz/ml/weka/
Orange• – Data Mining Fruitful & Fun (пакет создан лабораторией искусственного интеллекта Университета Любляни, Словения); http://www.ailab.si/orange/
QuDA• – Data Miner's Discovery Environment (разработана в техническом Университете города Дармштадта, Германия); http://sourceforge.net/projects/quda/
Coron System• – платформа Data mining (разработана коллегами из группы Orpailleur в лаборатории LORIA Университета Нанси, Франция); http://coron.loria.fr/
Concept Explorer• – один из основных инструментов анализа формальных понятий (разработана в техническом Университете города Дармштадта, Германия);
Система для статистических вычислений R http://www.r-project.org/
Библиотека Scikit-learn (Python)http://scikit-learn.org/

Данные и идеи задач:
• https://www.kaggle.com Данные, использованные в соревнованиях по машинному обучению.
Много задач по разной тематике.
• http://cs229.stanford.edu/projects2013.html Отчеты студентов Стэнфорда (можно искать идеи задач, ссылки на данные).
• http://archive.ics.uci.edu/ml Опять же много наборов данных для различных прикладных задач

Free SVM Tools
http://www.support-vector-machines.org/SVM_soft.html - Ресурс по проблематике SVM.

The free tools listed below range from C++ libraries, to drag and drop graphical environments. Some are pure support vector machine (SVM) and others are data mining platforms which include SVM.
KNIME – broad functionality GUI based data mining tool with particularly easy to use SVM support.
LIBSVM – is a sophisticated SVM library containing C-classification, v-classification, epsilon-regression, and v-regression. It supports multi-class classification, weighted SVM for unbalanced data, cross validation and automatic model selection. Interfaces for Python, R, S+, MATLAB, Perl, Ruby and LABVIEW make it particularly popular.
mySVM – written in C++ and includes SVM classification and regression. Avaliable as source code and WIndows binaries. Kernels include linear, polynomial, radial basis function, neural(tanh) and anova.
Orange – is another GUI based data mining toolset that includes a SVM learner.
RapidMiner – is a sophisticated GUI based data mining toolset, and possibly the most widely used platform of its type. It includes and includes several SVM implementations (including LIBSVM).
SVM Light – is a very widely used classification and regression package and is distributed as C++ source. It is well known for its speed of execution and an efficient implementation of the leave-one-out (LOO) cross-validation method.
Weka – one of the best known collection of data mining tools contains an SVM implementation

Лабораторная работа №6

Поиск ассоциаций

1. Цель работы

Получить практические навыки по выявлению в наборах данных ассоциативных правил.

2. Методические указания

Ассоциативные правила позволяют находить закономерности между связанными событиями. Примером такого правила, служит утверждение, что покупатель, приобретающий 'Хлеб', приобретет и 'Молоко' с вероятностью 75%. Первый алгоритм поиска ассоциативных правил, называвшийся AIS (предложенный Agrawal, Imielinski and Swami) был разработан в 1993 году сотрудниками исследовательского центра IBM Almaden. В алгоритме AIS кандидаты множества наборов генерируются и подсчитываются «на лету», во время сканирования базы данных.

С этой пионерской работы возрос интерес к ассоциативным правилам; на середину 90-х годов прошлого века пришелся пик исследовательских работ в этой области, и с тех пор каждый год появлялось по несколько алгоритмов. Впервые задача поиска ассоциативных правил была предложена для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины (market basket analysis).

Пусть имеется база данных, состоящая из покупательских транзакций. Каждая транзакция – это набор товаров, купленных покупателем за один визит. Такую транзакцию еще называют рыночной корзиной.

Определение 1.

Пусть $I = \{i_1, i_2, \ldots, i_n\}$ – множество (набор) товаров, называемых элементами. Пусть D – множество транзакций, где каждая транзакция T – это набор элементов из $I, T \subseteq I$. Каждая транзакция представляет собой бинарный вектор, где $t[k] = 1$, если *ik* элемент присутствует в транзакции, иначе $t[k] = 0$. Мы говорим, что транзакция T содержит X, некоторый набор элементов из I, если $X \subseteq T$. Ассоциативным правилом называется импликация $X \Rightarrow Y$, где $X \subset I, Y \subset I$ и $X \cap Y = \varnothing$. Правило $X \Rightarrow Y$ имеет поддержку s (support), если s% транзакций из D, содержат $X \cap Y$, $\text{supp}(X \Rightarrow Y) = \text{supp}(X \cup Y)$. Достоверность правила показывает, какова вероятность того, что из X следует Y. Правило $X \Rightarrow Y$ справедливо с достоверностью *(confidence*) *c*, если *c%* транзакций из D, содержащих X, также содержат Y, $conf(X \Rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X)$.

Поддержка(*support*) – показывает, какой процент транзакций поддерживает данное правило. Так как правило, строится на основании набора, то, значит, правило $X \Rightarrow Y$ имеет поддержку, равную поддержке набора *F*, который составляют *X* и *Y*:

$$supp_{X \Rightarrow Y} = supp_F = \frac{|D_{F=X \Rightarrow Y}|}{|D|}. \quad (1)$$

Правила, построенные на основании одного и того же набора, имеют, одинаковую поддержку.

Достоверность(*confidence*) – показывает вероятность того, что из наличия в транзакции набора *X* следует наличие в ней набора *Y*.достоверность правила $X \Rightarrow Y$ является отношением числа транзакций, содержащих наборы *X* и *Y*, к числу транзакций, содержащих набор *X*:

$$conf_{\mathrm{X} \Rightarrow \mathrm{Y}} = \frac{|D_{F=\mathrm{X} \Rightarrow \mathrm{Y}}|}{|D_X|} = \frac{supp_{x \cup Y}}{supp_X}. \quad (2)$$

Алгоритмы поиска ассоциативных правил предназначены для нахождения всех правил $X \Rightarrow Y$, причем поддержка и достоверность этих правил должны быть выше некоторых наперед определенных порогов, называемых соответственно минимальной поддержкой (*minsupport*) и минимальной достоверностью (*minconfidence*).

Задача нахождения ассоциативных правил разбивается на две подзадачи:

1. нахождение всех наборов элементов, которые удовлетворяют порогу *minsupport*. Такие наборы элементов называются часто встречающимися;
2. генерация правил из наборов элементов, найденных согласно пункту 1 с достоверностью, удовлетворяющей порогу *minconfidence*.

Значения для параметров минимальная поддержка и минимальная достоверность выбираются таким образом, чтобы ограничить количество

найденных правил. Если поддержка имеет большое значение, то алгоритмы будут находить правила, хорошо известные аналитикам или настолько очевидные, что нет никакого смысла проводить такой анализ. С другой стороны, низкое значение поддержки ведет к генерации огромного количества правил, что, конечно, требует существенных вычислительных ресурсов. Тем не менее, большинство интересных правил находится именно при низком значении порога поддержки. Хотя слишком низкое значение поддержки ведет к генерации статистически необоснованных правил.

Поиск ассоциативных правил совсем не тривиальная задача, как может показаться на первый взгляд. Одна из проблем – алгоритмическая сложность при нахождении часто встречающих наборов элементов, т.к. с ростом числа элементов в I ($|I|$) экспоненциально растет число потенциальных наборов элементов.

Помимо описанных выше ассоциативных правил существуют косвенные ассоциативные правила, ассоциативные правила с отрицанием, временные ассоциативные правила для событий связанных во времени и другие.

Выявление частых наборов объектов требует разработки соответствующих эффективных алгоритмов. Одним из первых таких алгоритмов был алгоритм Apriori, разработанный С. Рамакришнан и Р. Агравалом. Он использует одно из свойств поддержки: поддержка любого набора объектов не может превышать минимальныой поддержки любого из его подмножеств.

Работа данного алгоритма состоит из нескольких этапов, каждый из i этапов состоит из следующих шагов: формирование кандидатов; подсчет кандидатов. Формирование кандидатов (*candidate generation*) – шаг, на котором алгоритм, сканируя базу данных, создает множество i-элементных кандидатов (i-номер этапа). На этом шаге поддержка кандидатов не рассчитывается. Подсчет кандидатов – это шаг, на котором вычисляется поддержка каждого i – элементного кандидата. На этом шаге осуществляется отсечение кандидатов, поддержка которых меньше минимума, установленного пользователем (*min_sup*). Оставшиеся i-элементные наборы называются часто встречающимися. Алгоритм Apriori уменьшает количество кандидатов, отсекая – априори - тех, которые заведомо не могут стать часто встречающимися, на основе информации об отсеченных кандидатах на предыдущих этапах работы алгоритма.

Отсечение кандидатов происходит на основе предположения о том, что у часто встречающегося набора товаров все подмножества должны быть часто встречающимися. Если в наборе находиться подмножество, которое на предыдущем этапе было определено как нечасто встречающееся, этот кандидат уже не включается в формирование и подсчет кандидатов. Алгоритм Apriori рассчитывает также поддержку наборов, которые не могут быть отсечены априори. Это встречается редко, их самих нельзя отнести к часто встречающимся, но все подмножества данных наборов являются часто встречающимися.

В зависимости от размера самого длинного часто встречающегося набора алгоритм Apriori сканирует базу данных определенное количество раз. Разновидности алгоритма Apriori, являющиеся его оптимизацией, предложены для сокращения количества сканирований базы данных, количества наборов кандидатов или того и другого. Были предложены следующие разновидности алгоритма Apriori: AprioriTID и AprioriHybrid.

AprioriTID. Интересная особенность этого алгоритма - то, что база данных не используется для подсчета поддержки кандидатов набора товаров после первого прохода. С этой целью используется кодирование кандидатов, выполненное на предыдущих подходах. В последующих проходах размер закодированных данных может быть намного меньше, чем база данных, и таким образом экономятся значительные ресурсы.

AprioriHybrid. Анализ времени работы алгоритмов Apriori и AprioriTID показывает, что в более ранних проходах Apriori добивается большего успеха, чем AprioriTID; однако AprioriTID работает лучше Apriori в более поздних проходах. Кроме того, они используют одну и ту же процедуру формирования наборов-кандидатов. Основанный на этом наблюдении, алгоритм AprioriHybrid предложен чтобы объединить лучшие свойства алгоритмов Apriori и AprioriTID. AprioriHybrid использует алгоритм Apriori в начальных проходах и переходит к алгоритму AprioriTID, когда ожидается, что закодированный набор первоначального множества в конце прохода будет соответствовать возможностям памяти. Однако, переключение от Apriori до AprioriTID требует дополнительных ресурсов.

Алгоритм DHP, также называемый хеширования (J. Park, M. Chen and P.Yu, 1995 год). В основе его работы – вероятностный подсчет наборов-кандидатов, осуществляемый для сокращения числа подсчитываемых кандидатов на каждом этапе выполнение алгоритма Apriori. Сокращение обеспечивается за счет того, что каждый из k -элементных наборов-кандидатов помимо шага сокращения проходит шаг хеширования. В алгоритме на k=1 этапе во время выбора кандидатов создается хэш-таблица. Каждая запись хеш-таблицы является счетчиком всех поддержек k -элементных наборов, которые соответствуют этой записи в хеш-таблице. Алгоритм использует эту информацию на этапе k для сокращения множества k-элементных наборов-кандидатов. После сокращения подмножества, как это происходит в Apriori, алгоритм может удалить набор-кандидат, если его значение в хеш-таблице меньше порогового значения.

Apriori - масштабируемый алгоритм поиска ассоциативных правил

Современные базы данных имеют очень большие размеры, достигающие гига - и терабайтов, и тенденцию к дальнейшему увеличению. И поэтому, для нахождения ассоциативных правил требуются эффективные масштабируемые алгоритмы, позволяющие решить задачу за приемлемое время. Для этого был выбран алгоритм Apriori.

Для того чтобы было возможно применить алгоритм, необходимо провести предобработку данных: во-первых, привести все данные к бинарному виду; во-вторых, изменить структуру данных.

В таблице 1 представлен обычный вид базы данных транзакций.

Таблица 1. Обычный вид базы данных транзакции

Номер транзакции	Наименование элемента	Количество
1001	A	2
1001	D	3
1001	E	1
1002	A	2
1002	F	1
1003	B	2
1003	A	2
1003	C	2
…	…	…

В таблице 2 представлен нормализованный вид базы данных транзакций:

Таблица 2. Нормализованный вид базы данных транзакций.

TID	A	B	C	D	E	F	G	H	I	K	…
1001	1	0	0	1	1	0	0	0	0	0	…
1002	1	0	0	0	0	1	0	0	0	0	…
1003	1	1	1	0	0	0	0	0	1	0	…

Количество столбцов в таблице равно количеству элементов, присутствующих в множестве транзакций D. Каждая запись соответствует транзакции, где в соответствующем столбце стоит 1, если элемент присутствует в транзакции, и 0 в противном случае. Заметим, что исходный вид таблицы может быть отличным от приведенного в таблице 1. Главное, чтобы данные были преобразованы к нормализованному виду, иначе алгоритм не применим.

Более того, как видно из таблицы 2, все элементы упорядочены в алфавитном порядке (если это числа, они должны быть упорядочены в числовом порядке).

Алгоритм Apriori работает в два этапа. На первом шаге необходимо найти часто встречающиеся наборы элементов, а затем, на втором, извлечь из них правила. Количество элементов в наборе будем называть размером набора, а набор, состоящий из k элементов, k -элементным набором.

Свойство анти-монотонности. Выявление часто встречающихся наборов элементов – операция, требующая много вычислительных ресурсов и, соответственно, времени. Примитивный подход к решению данной задачи –

простой перебор всех возможных наборов элементов. Это потребует $O(2^{|I|})$ операций, где $|I|$ – количество элементов. Априори использует одно из свойств поддержки, гласящее: поддержка любого набора элементов не может превышать минимальной поддержки любого из его подмножеств. Например, поддержка 3-элементного набора *{Хлеб, Масло, Молоко}* будет всегда меньше или равна поддержке 2-элементных наборов *{Хлеб, Масло}, {Хлеб, Молоко}, {Масло, Молоко}*. Дело в том, что любая транзакция, содержащая {Хлеб, Масло, Молоко}, также должна содержать *{Хлеб, Масло}, {Хлеб, Молоко}, {Масло, Молоко}*, причем обратное не верно. Это свойство носит название анти-монотонности и служит для снижения размерности пространства поиска. Не имей мы в наличии такого свойства, нахождение многоэлементных наборов было бы практически невыполнимой задачей в связи с экспоненциальным ростом вычислений. Свойству анти-монотонности можно дать и другую формулировку: с ростом размера набора элементов поддержка уменьшается, либо остается такой же. Из всего вышесказанного следует, что любой k-элементный набор будет часто встречающимся тогда и только тогда, когда все его $(k-1)$–элементные подмножества будут часто встречающимися.

Все возможные наборы элементов из I можно представить в виде решетки, начинающейся с пустого множества, затем на 1 уровне 1-элементные наборы, на 2-м – 2-элементные и т.д. На k уровне представлены k -элементные наборы, связанные со всеми своими *(k -1)*-элементными подмножествами.

Рассмотрим рисунок 1, иллюстрирующий набор элементов *I – {A, B, C, D}*. Предположим, что набор из элементов *{A, B}* имеет поддержку ниже заданного порога и, соответственно, не является часто встречающимся. Тогда, согласно свойству анти-монотонности, все его супермножества также не являются часто встречающимися и отбрасываются. Вся эта ветвь, начиная с *{A, B},* отмечена желтым фоном. Использование этой эвристики позволяет существенно сократить пространство поиска.

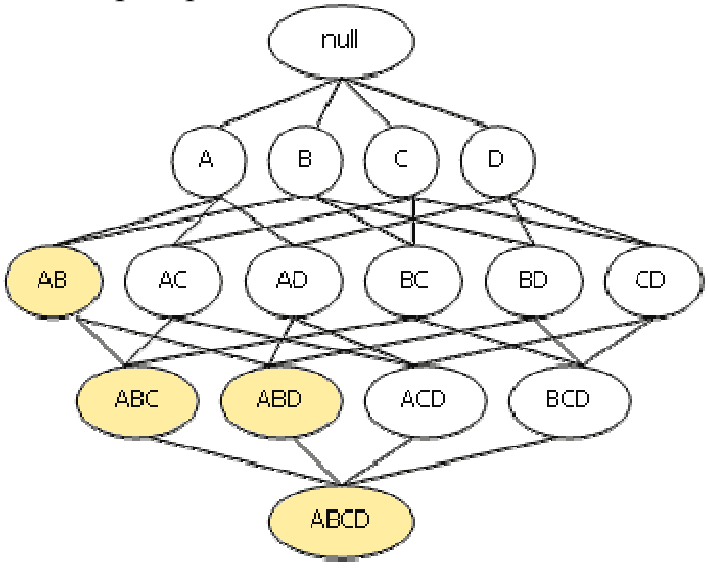

Рисунок 1. Набор элементов I – {A, B, C, D}.

На первом шаге алгоритма подсчитываются 1-элементные часто встречающиеся наборы. Для этого необходимо пройтись по всему набору данных и подсчитать для них поддержку, т.е. сколько раз встречается в базе.
Следующие шаги будут состоять из двух частей: генерации потенциально часто встречающихся наборов элементов (их называют кандидатами) и подсчета поддержки для кандидатов.
Описанный выше алгоритм можно записать в виде следующего псевдокода:

F1 = {часто встречающиеся 1-элементные наборы}
для (k=2; Fk-1 <> ∅; k++) {
Ck = Apriorigen(Fk-1) // генерация кандидатов
для всех транзакций t ∈T {
Ct = subset(Ck, t) // удаление избыточных правил
для всех кандидатов c ∈ Ct
c.count ++
}
Fk = { c ∈ Ct | c.count >= minsupport} // отбор кандидатов
Результат $\bigcup_k Fk$.

Опишем функцию генерации кандидатов. На это раз нет никакой необходимости вновь обращаться к базе данных. Для того чтобы получить *k* - элементные наборы, воспользуемся *(k -1)*-элементными наборами, которые были определены на предыдущем шаге и являются часто встречающимися.
Генерация кандидатов также будет состоять из двух шагов.
Объединение. Каждый кандидат *Ck* будет формироваться путем расширения часто встречающегося набора размера *(k -1)* добавлением элемента из другого *(k-1)*- элементного набора.
Приведем алгоритм этой функции Apriorigen в виде небольшого SQL-подобного запроса.

insert into Ck
select p.item1, p.item2, …, p.itemk-1, q.itemk-1
From Fk-1 p, Fk-1 q
where p.item1= q.item1, p.item2 = q.item2, … , p.itemk-2 = q.itemk-2, p.itemk-1 < q.itemk-1

Удаление избыточных правил. На основании свойства анти-монотонности, следует удалить все наборы $c \in Ck$ если хотя бы одно из его *(k-1)* подмножеств не является часто встречающимся.

После генерации кандидатов следующей задачей является подсчет поддержки для каждого кандидата. Очевидно, что количество кандидатов может быть очень большим и нужен эффективный способ подсчета. Самый тривиальный способ – сравнить каждую транзакцию с каждым кандидатом. Но это далеко не лучшее решение. Гораздо быстрее и эффективнее использовать подход, основанный на хранении кандидатов в хэш-дереве. Внутренние узлы дерева содержат хэш-таблицы с указателями на потомков, а листья – на кандидатов. Это дерево необходимо для быстрого подсчета поддержки для кандидатов.

Хэш-дерево строится каждый раз, когда формируются кандидаты. Первоначально дерево состоит только из корня, который является листом, и не содержит никаких кандидатов-наборов. Каждый раз, когда формируется новый кандидат, он заносится в корень дерева и так до тех пор, пока количество кандидатов в корне-листе не превысит некоего порога. Как только количество кандидатов становится больше порога, корень преобразуется в хэш-таблицу, т.е. становится внутренним узлом, и для него создаются потомки-листья. И все кандидаты распределяются по узлам-потомкам согласно хэш-значениям элементов, входящих в набор, и т.д. Каждый новый кандидат хэшируется на внутренних узлах, пока он не достигнет первого узла-листа, где он и будет храниться, пока количество наборов опять же не превысит порога.

Хэш-дерево с кандидатами-наборами построено, теперь, используя хэш-дерево, легко подсчитать поддержку для каждого кандидата. Для этого нужно 'пропустить' каждую транзакцию через дерево и увеличить счетчики для тех кандидатов, чьи элементы также содержатся и в транзакции, т.е. $Ck \cap Ti = Ck$. На корневом уровне хэш-функция применяется к каждому элементу из транзакции. Далее, на втором уровне, хэш-функция применяется ко вторым элементам и т.д. На k -уровне хэшируется k -элемент. И так до тех пор, пока не достигнем листа. Если кандидат, хранящийся в листе, является подмножеством рассматриваемой транзакции, тогда увеличиваем счетчик поддержки этого кандидата на единицу.

После того, как каждая транзакция из исходного набора данных 'пропущена' через дерево, можно проверить удовлетворяют ли значения поддержки кандидатов минимальному порогу. Кандидаты, для которых это условие выполняется, переносятся в разряд часто встречающихся. Кроме того, следует запомнить и поддержку набора, она нам пригодится при извлечении правил. Эти же действия применяются для нахождения $(k+1)$-элементных наборов и т.д.

После того как найдены все часто встречающиеся наборы элементов, можно приступить непосредственно к генерации правил.

Извлечение правил – менее трудоемкая задача. Во-первых, для подсчета достоверности правила достаточно знать поддержку самого набора и множества, лежащего в условии правила. Например, имеется часто встречающийся набор $\{A, B, C\}$ и требуется подсчитать достоверность для правила $AB \Rightarrow C$. Поддержка самого набора нам известна, но и его множество $\{A, B\}$, лежащее в условии правила, также является часто встречающимся в силу свойства анти-монотонности, и значит его поддержка нам известна. Тогда мы легко сможем подсчитать достоверность. Это избавляет нас от нежелательного просмотра базы транзакций, который потребовался бы в том случае, если бы это поддержка была неизвестна.

Чтобы извлечь правило из часто встречающегося набора F, следует найти все его непустые подмножества. И для каждого подмножества s мы сможем сформулировать правило $s \Rightarrow (F - s)$, если достоверность правила

$conf(s \Rightarrow (F - s)) = \text{supp}(F) / \text{supp}(s)$ не меньше порога *minconf* (минимальной достоверности).

Заметим, что числитель остается постоянным. Тогда достоверность имеет минимальное значение, если знаменатель имеет максимальное значение, а это происходит в том случае, когда в условии правила имеется набор, состоящий из одного элемента. Все супермножества данного множества имеют меньшую или равную поддержку и, соответственно, большее значение достоверности. Это свойство может быть использовано при извлечении правил. Если мы начнем извлекать правила, рассматривая сначала только один элемент в условии правила, и это правило имеет необходимую поддержку, тогда все правила, где в условии стоят супермножества этого элемента, также имеют значение достоверности выше заданного порога. Например, если правило $A \Rightarrow BCDE$ удовлетворяет минимальному порогу достоверности *minconf*, тогда $AB \Rightarrow CDE$ также удовлетворяет. Для того, чтобы извлечь все правила используется рекурсивная процедура. Важное замечание: любое правило, составленное из часто встречающегося набора, должно содержать все элементы набора. Например, если набор состоит из элементов *{A, B, C},* то правило $A \Rightarrow B$ не должно рассматриваться.

3. Содержание работы

1. Ознакомиться с постановкой задач поиска ассоциаций, алгоритмами их построения.
2. Провести генерацию набора данных с транзакциями в соответствии с вариантом задания.
3. Реализовать алгоритм Apriori поиска ассоциативных правил.
4. Осуществить решение задачи поиска ассоциативных правил в ранее созданном наборе данных. Минимальный уровень поддержки задается пользователем и в процессе тестирования приложения варьируется. Оценить качество работы приложения.
5. Оформить отчет, включающий в себя постановку задачи, полученные результаты и выводы.
6. Защитить лабораторную работу.

4. Варианты заданий

1. Общее количество объектов 100. Общее число транзакций не менее 100000. При генерации набора данных использовать ассоциативные правила разного уровня поддержки. Число элементов в частных наборах используемых правил равно 2, 3, 4. Число правил с большим уровнем поддержки 3, со средним уровнем 5, с небольшим уровнем 10.
2. Общее количество объектов 500. Общее число транзакций не менее 1000000. При генерации набора данных использовать ассоциативные правила разного уровня поддержки. Число элементов в частных наборах

используемых правил равно 2, 3, 4. Число правил с большим уровнем поддержки 5, со средним уровнем 10, с небольшим уровнем 20.

3. Общее количество объектов 1000. Общее число транзакций не менее 1000000. При генерации набора данных использовать ассоциативные правила разного уровня поддержки. Число элементов в частных наборах используемых правил равно 2, 3, 4, 5. Число правил с большим уровнем поддержки 5, со средним уровнем 15, с небольшим уровнем 20.
4. Общее количество объектов 200. Общее число транзакций не менее 100000. При генерации набора данных использовать ассоциативные правила разного уровня поддержки. Число элементов в частных наборах используемых правил равно 2, 3, 4. Число правил с большим уровнем поддержки 5, со средним уровнем 10, с небольшим уровнем 20.
5. Общее количество объектов 700. Общее число транзакций не менее 1000000. При генерации набора данных использовать ассоциативные правила разного уровня поддержки. Число элементов в частных наборах используемых правил равно 2, 3, 4. Число правил с большим уровнем поддержки 7, со средним уровнем 15, с небольшим уровнем 20.
6. Общее количество объектов 2000. Общее число транзакций не менее 1000000. При генерации набора данных использовать ассоциативные правила разного уровня поддержки. Число элементов в частных наборах используемых правил равно 2, 3, 4, 5. Число правил с большим уровнем поддержки 10, со средним уровнем 10, с небольшим уровнем 10.

5. Контрольные вопросы

1. Задача поиска ассоциативных правил;
2. Понятия поддержки, достоверности;
3. Алгоритм Apriori и его разновидности.

Список литературы

1. Zaki M. J. Scalable. Data Mining for Rule» -University of Rochester New York, 1998.
2. M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In 3rd Intl. Conf. on Knowledge Discovery and Data Mining, August 1997.
3. Константин Буров. Обнаружение знаний в хранилищах данных. Открытые системы, №05-06/1999.
4. С.Д. Коровкин, И.А. Левенец, И.Д. Ратманова. Решение проблемы комплексного оперативного анализа информации хранилищ данных – СУБД, №5-6,1997. - 47-51с.

5. H. Edelstein. Интеллектуальные средства анализа, интерпретации и представления данных в информационных хранилищах. №16, 1996.- 32-35с.
6. M. Holsheimer, M. Kersten, H Mannila and H. Toivonen. A perspective on databases and data mining. – CS-R9531, 1995.
7. B. Moxon. Defining Data Mining. //DBMS Data Warehouse Supplement, 1996.
8. В.А. Дюк. Data Mining –интеллектуальный анализ данных //Санкт-Петербургский институт информатики и автоматизации РАН
9. DATA MINING. Что такое Data Mining? //Центр Статистических Технологий, 2003.
10. Чубукова И.А. Data Mining : учебное пособие / И. А. Чубукова**.** М. : Интернет-Университет информационных технологий: Бином. Лаборатория знаний, 2006.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания

по организации и проведению самостоятельной работы
по дисциплине
«Генеративный искусственный интеллект»
для студентов направления подготовки
09.03.02 «Информационные системы и технологии»
направленность (профиль)
«Прикладное программирование в интеллектуальных информационных системах»

Ставрополь, 2025

1. Общие положения

Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

− формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);

− написание докладов;

− подготовка к семинарам, практическим и лабораторным работам, их оформление;

− составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);

− выполнение учебно-исследовательских работ, проектная деятельность;

− подготовка практических разработок и рекомендаций по решению проблемной ситуации;

− выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;

− компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;

− выполнение курсовых работ (проектов) в рамках дисциплин;

− выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине «Генеративный искусственный интеллект» направлена на формирование следующих **компетенций**:

Код, формулировка компетенции	Код, формулировка индикатора
ПК-9 Способен проводить научно-исследовательскую работу на основе анализа	ПК-$9_{ИД-9.1}$ – Проводит анализ отечественного и зарубежного опыта по тематике исследований

передового опыта в сферах проектирования и разработки информационных систем, в том числе интеллектуальных	ПК-9$_{ИД-9.2}$ – Составляет обзоры, рефераты, отчеты, готовит научные публикации и доклады на научных конференциях и семинарах по тематике своих исследований
	ПК-9$_{ИД-9.3}$ – Реализует вычислительные эксперименты в рамках исследований с использованием передовых технологий и методик в области интеллектуальных информационных систем
ПК-4 Способен анализировать предметную область и обосновывать эффективность применения интеллектуальных моделей различного типа	ПК-4$_{ИД-4.1}$ – Демонстрирует знания в области теории функционирования обучаемых алгоритмов и интеллектуальных моделей
	ПК-4$_{ИД-4.2}$ – Демонстрирует знание стандартов оценки качества систем искусственного интеллекта
	ПК-4$_{ИД-4.3}$ – Осуществляет обоснованный отбор моделей искусственного интеллекта для решения профессиональных задач

2. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и специальных компетенций будущего бакалавра по соответствующему направлению подготовки

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;

● использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

3. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контактная работа с преподавателем	Всего
6 семестр					
ПК-4, ПК-9	Самостоятельное изучение литературы и источников	Собеседование	2	2	4
ПК-4, ПК-9	Подготовка к лабораторным занятиям	Защита ЛР	36	14	40
Итого за 6 семестр			38	16	44
Итого			38	16	44

4. Порядок выполнения самостоятельной работы студентом

4.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют ***четыре основные установки в чтении научного текста***:

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;
2. Выделите главное, составьте план;
3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;
4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной

последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

4.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

4.4. Методические рекомендации по написанию научных текстов (докладов, докладовов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т, п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Доклад - это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Доклад не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в доклад е должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки

а студентом.

Выполнение доклада начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы - изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания доклада. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста доклада предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки доклад сдается на кафедру для его оценивания руководителем.

Требования к написанию доклада

Написание 1 доклада является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема доклада может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласоваться с преподавателем.

Доклад должен быть написан научным языком.

Объем доклада должен составлять 20-25 стр.

Структура доклада:

● Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.

● Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.

● Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса

● Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.

● Приложение (при необходимости).

Требования к оформлению:

● текст с одной стороны листа;
● шрифт Times New Roman;
● кегль шрифта 14;
● межстрочное расстояние 1,5;
● поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
● доклад должен быть представлен в сброшюрованном виде.

Порядок защиты доклада:

Защита доклада проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту доклада отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите доклада приветствуется использование мультимедиа-презентации.

Оценка доклада

Доклад оценивается по следующим критериям:

• соблюдение требований к его оформлению;

• необходимость и достаточность для раскрытия темы приведенной в тексте доклада информации;

• умение студента свободно излагать основные идеи, отраженные в докладе;

• способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

4.5. Методические рекомендации по выполнению исследовательских проектов

Исследовательская проектная работа – это групповая работа, для выполнения которой необходим выбор и приложение научной методики к поставленной задаче, получение собственного теоретического или экспериментального материала, на основании которого необходимо провести анализ и сделать выводы об исследуемом явлении. Выполнение проекта – это всегда коллективная, творческая практическая работа, предназначенная для получения определенного продукта или научно-технического результата. Такая работа подразумевает четкое, однозначное формирование поставленной задачи, определение сроков выполнения намеченного, определение требований к разрабатываемому объекту.

Выполнение 1 группового проекта является обязательным условием выполнения самостоятельной работы по любой дисциплине профессионального цикла. Тема проектного задания может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласоваться с преподавателем.

Требования по выполнению и оформлению проекта

При выполнении проекта приветствуется работа в группе (2-3 человека). Проект – это исследовательская работа, в ходе которой студенты должны продемонстрировать владение навыками научного исследования, умения проводить анализ, обобщать информацию, делать выводы, предлагать свои решения проблемы, рассматриваемой в проекте.

При подготовке материалов проекта студенты должны продемонстрировать владение современными методами компьютерной обработки данных.

Критерии оценки работы участника проекта.

Для каждого из участников проекта оцениваются:

• профессиональные теоретические знания в соответствующей области;

• умение работать со справочной и научной литературой, осуществлять поиск необходимой информации в Интернет;

• умение работать с техническими средствами;

• умение пользоваться соответствующими выполняемому проекту информационными технологиями;

• умение готовить материалы проекта для презентации: составлять и редактировать тексты, формировать презентацию проекта;

• умение работать в команде;

• умение публично представлять результаты собственной деятельности;
• коммуникабельность, инициативность, творческие способности.

Критерии выставления оценки участникам проекта

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
«Отлично»	Работа выполнена на высоком профессиональном уровне. Представленный материал в основном фактически верен, допускаются негрубые фактические неточности. Студент свободно отвечает на вопросы, связанные с проектом.	Технические средства и информационные технологии освоены и использованы для реализации проекта полностью	Студент проявил инициативу, творческий подход, способность к выполнению сложных заданий, навыки работы в коллективе, организационные способности.	Проект представлен полностью и в срок.
«Хорошо»	Работа выполнена на достаточно высоком профессиональном уровне. Допущено до 4–5 фактических ошибок. Студент отвечает на вопросы, связанные с проектом, но недостаточно полно.	Обнаруживаются некоторые ошибки в использовании соответствующих технических средств и информационных технологий	Студент достаточно полно, но без инициативы и творческих находок выполнил возложенные на него задачи.	Проект представлен достаточно полно и в срок, но с некоторыми недоработками.
«Удовлетворительно»	Уровень недостаточно высок. Допущено до 8 фактических ошибок. Студент может ответить лишь на некоторые из заданных вопросов, связанных с проектом.	Обнаруживает недостаточное владение навыками работы с техническими средствами и соответствующими информационными технологиями	Студент выполнил большую часть возложенной на него работы.	Проект сдан со значительным опозданием (более недели) и не полностью
«Неудовлетворительно»	Работа не выполнена или выполнена на низком уровне.	Навыков работы с техническими средствами нет,	Студент практически не работал, не	Проект не сдан.

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
	Допущено более 8 фактических ошибок. Ответы на связанные с проектом вопросы обнаруживают непонимание предмета и отсутствие ориентации в материале проекта.	информационные технологии не освоены	выполнил свои задачи или выполнил лишь отдельные не существенные поручения в групповом проекте.	

Студенты должны: защитить проект в режиме презентации, предъявить файлы выполненного проекта, уметь рассказать о технологиях, использованных ими при выполнении проекта, дать оценку работы каждого члена группы (*если проект групповой*).

4.6. Методические рекомендации по подготовке к экзаменам и зачетам

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами интервал 3-4 дня. Не следует думать, что 3-4 дня достаточно для успешной подготовки к экзаменам.

В эти 3-4 дня нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня; сон не менее 8 часов в сутки, занятия заканчиваются не позднее, чем за 2-3 часа до сна. Оптимальное время занятий - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неутомительные занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая-либо лекция, необходимо во время ее восстановить (переписать ее на кафедре), обдумать, снять возникшие вопросы для того, чтобы запоминание материала было осознанным. В-третьих, при подготовке к экзаменам у студента должен быть хороший

учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных сигналов.

Вначале следует просмотреть весь материал по сдаваемой дисциплине, отметить для себя трудные вопросы. Обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения, используя при этом листы опорных сигналов.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Контроль самостоятельной работы студентов

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка доклада, оценка презентации, оценка участия в круглом столе, оценка выполнения проекта.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

Список литературы для выполнения СРС

Перечень основной литературы:

1. Генеративное глубокое обучение. Как не мы рисуем картины, пишем романы. и музыку. 2-е межд изд. — Астана: «Спринт Бук», 2024. — 448 с.: ил. ISBN 978-601-08-3729-4

2. Кэвин П. Мэрфи. Вероятностное машинное обучение. Дополнительные темы: основания, вывод / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2024. – 770 с.: ил

3. Флах, Петер Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. Учебник / Петер Флах. - М.: ДМК Пресс, 2019. - 400 с.

Перечень дополнительной литературы:

1. Хапке Х., Нельсон К.Разработка конвейеров машинного обучения. Автоматизация жизненных / циклов модели с помощью TensorFlow / пер. с анг. Н. Б. Желновой. – М.: ДМК Пресс, 2021. – 346 с.: ил.

Методическая литература:

1. Методические рекомендации для самостоятельной работы студентов по дисциплине «Генеративный искусственный интеллект»
2. Методические указания к лабораторным работам по дисциплине «Генеративный искусственный интеллект»

Интернет-ресурсы:

1. http://el.ncfu.ru/ – система управления обучением ФГАОУ ВО СКФУ. Дистанционная поддержка дисциплины «Генеративный искусственный интеллект».
2. http://www.un.org - Сайт ООН Информационно-коммуникационные технологии.
3. http://www.intuit.ru – Интернет-Университет Компьютерных технологий.