0. Генерация и визуализация исходных данных, основы классификации и аппроксимации

Цели работы:

- научиться формировать выборки, состоящие из обучающих и тестовых примеров для решения типовых задач классификации, аппроксимации;

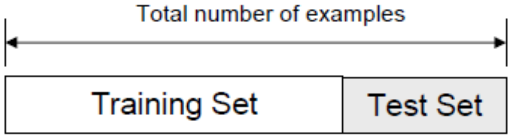
- овладеть навыками визуализации данных на плоскости при решении задач классификации и аппроксимации;

- научиться рассчитывать основные показатели качества распознавания и представлять полученные результаты в табличной и графической формах.

Теоретические сведения

Оценивание качества классификации

В процессе синтеза системы распознавания возникает задача оценки качества для выбора наилучших признаков, алгоритма классификации, подстройки различных параметров. В общем случае для обучения классификатора, выбора признаков приходится использовать обучающие примеры, составляющие обучающую выборку. Проверяется же результат классификации всегда на другой – тестовой выборке. Потому что при обучении и проверке на одной и той же выборке не избежать проблемы переобучения (overfitting), когда система запоминает примеры, а не учится их обобщать. Общий подход можно показан на рисунке



Здесь вся выборка делится на 2 части – обучающую и тестовую. Обучающая подвыборка используется для обучения, а тестовая – для оценивания ошибки обученного классификатора.

Но такой способ имеет недостатки:

- в задачах, когда набор исходных данных очень небольшой, не всегда есть возможность использовать («жалко») часть выборки только для тестирования системы;

- при неудачном разделении выборки оценка ошибки может отличаться от истинного значения.

Ограничения можно обойти с помощью различных методов повторной выборки (resampling) за счет удорожания вычислительной сложности:

- кросс-валидация;

- бутстрэп-оценки.

Кроссвалидация

Исходная выборка *K* раз случайно разбивается на подвыборки:

- в каждом разбиении из выборки выбирается некоторое фиксированное количество примеров (без возвращения, т.е. тестовые примеры должны отличаться) и получается две подвыборки – обучающая (исходная минус убранные примеры) и тестовая, состоящая из убранных примеров;

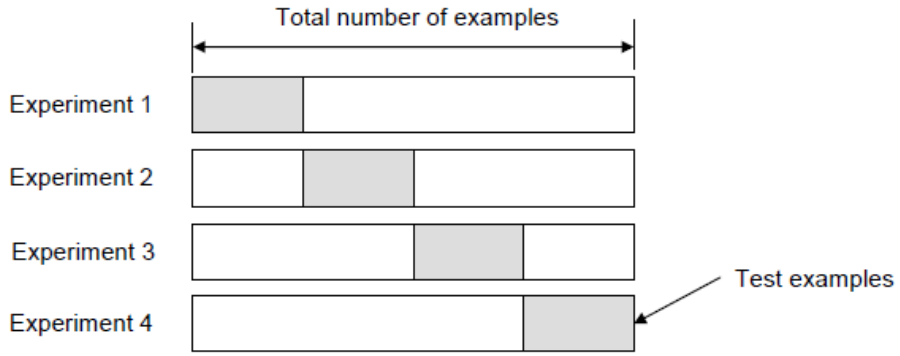
- для каждого i-го разбиения классификатор переобучается заново на обучающей подвыборке и затем вычисляется ошибка e_i на тестовой подвыборке.

Оценка ошибки получается как среднее по всем ошибкам e_i, намного точнее чем в случае простого разделения выборки:

$$e = \frac{1}{K}\sum_{i=1}^{K} e_i$$

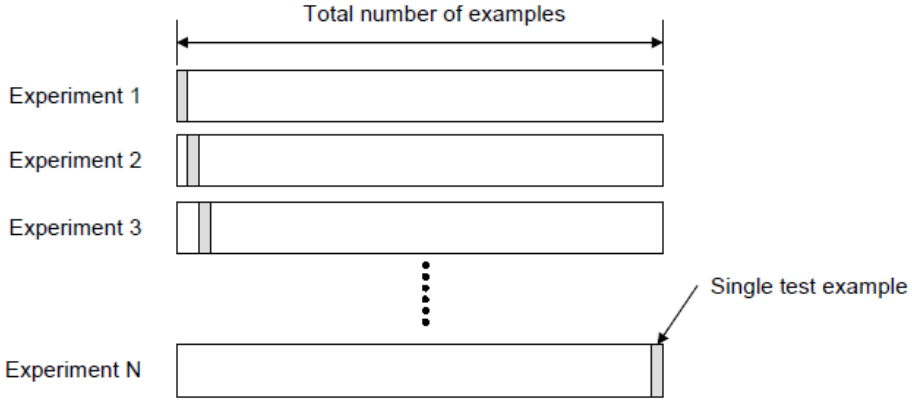
Довольно популярным вариантом является т.н. кросс-валидация по K блокам (K-Fold). Вся выборка делится изначально на K блоков, далее в каждом из K экспериментов

(K-1) блоков используется в качестве обучающей подвыборки, а 1 блок – в качестве тестовой. На рисунке показан пример для K=4.



K-Fold кросс-валидация похожа на случайную подвыборку. Преимуществом данного метода является то, что все примеры в выборке одновременно используются и для тестирования, и для обучения.

Иногда применяется и т.н. поэлементная кросс-валидация (Leave-one-out) Это частный случаем K-fold кросс-валидации, когда K = N-1. Всего таким образом получается N экспериментов, в каждом из которых тестовую выборку представляет всего лишь один пример (последовательно проходят все примеры из выборки).



Вычисление ошибок классификации, матрица неточностей

При тестировании системы классификации наиболее полную информацию содержит в себе матрица неточностей (confusion matrix).

$$C_m = \begin{bmatrix} n_{11} & n_{12} & \dots & n_{1m} \\ n_{21} & n_{22} & \dots & n_{2m} \\ \dots & \dots & \dots & \dots \\ n_{m1} & n_{m2} & \dots & n_{mm} \end{bmatrix}$$

Общее количество тестовых примеров равно сумме всех элементов матрицы C_m:

$$N = \sum_{i,j} n_{ij} .$$

Строки матрицы соответствуют истинным классам входных примеров, а столбцы – классам, с которыми эти примеры соотнес классификатор. Каждая классификация примера i класса как j-го добавляет единицу в соответствующий элемент n_{ij} матрицы. В результате после подачи всех тестовых примеров в n_{ij} содержится количество тестовых примеров класса i, которые были распознаны как класс j.

На диагонали матрицы неточностей – число правильно распознанных тестовых примеров соответствующего класса, а вне диагонали – число различного типа ошибок.

Среднюю вероятность ошибки можно посчитать, разделив сумму внедиагональных элементов на сумму всех элементов, аналогично среднюю вероятность правильного

распознавания можно посчитать, поделив сумму всех диагональных элементов на сумму всех элементов:

$$\overline{P_e} = \frac{\sum_{i,j:j\neq i}^{c} n_{ij}}{\sum_{i,j}^{c} n_{ij}},\ \overline{P_{np}} = \frac{\sum_{i}^{c} n_{ii}}{\sum_{i,j}^{c} n_{ij}} = 1 - \overline{P_e}.$$

С помощью матрицы неточностей можно посчитать т.н. ошибки первого и второго рода. Ошибка первого рода (ложная тревога, ложное срабатывание, ложноположительное срабатывание, риск производителя, уровень значимости критерия) – мера того как часто экземпляры других классов были приняты за этот класс) вычисляется по соответствующему столбцу матрицы:

$$\overline{e_1(i)} = \frac{\sum_{j:j\neq i}^{c} n_{ji}}{\sum_{j=1}^{c} n_{ji}}$$

Ошибка второго рода (пропуск события, риск покупателя, ложноотрицательное срабатывание, мощность критерия) – мера того, как часто данный класс был принят за другие классы вычисляется по соответствующей строке матрицы:

$$\overline{e_2(i)} = \frac{\sum_{j:j\neq i}^{c} n_{ij}}{\sum_{j=1}^{c} n_{ij}}$$

В случае двух классов как правило один из классов означает событие (некоторую ситуацию, которую нужно выявить из общего множества – выявлена болезнь, определен пешеход, распознан голос, буква), а другой класс – отсутствие события (человек здоров, ни каких объектов не обнаружено). Класс, который не соотносится с событием, часто называют нулевой или отрицательного гипотезой H_0, а класс – событие – с положительной гипотезой H_1. В этом случае результат классификации описывается четверкой вероятностей:

- специфичность (specificity) – вероятность правильно обнаружить отстутствие события;
- чувствительность (sensitivity) – вероятность правильно обнаружить событие;
- ошибка первого рода – вероятность ложного обнаружения события;
- ошибка второго рода – вероятность пропуска события.

Матрица неточностей принимает вид

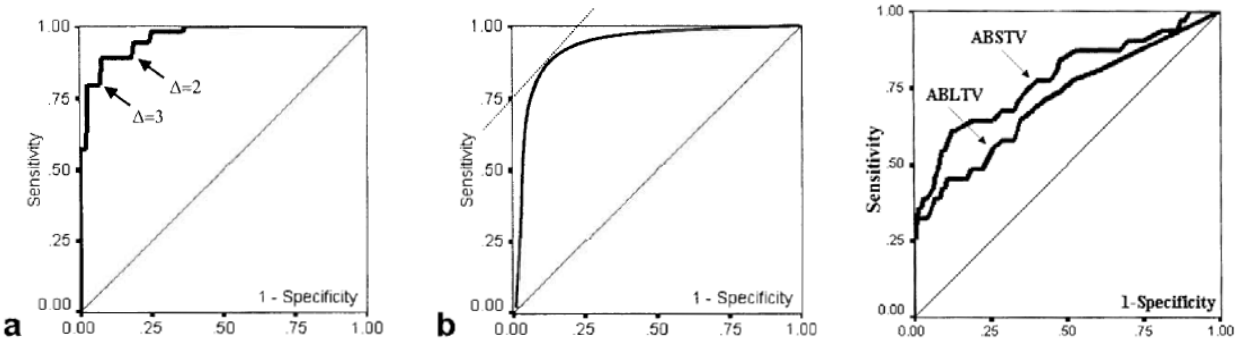
$$C_m = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix},$$

где первый класс соотносится с отсутствием события, а второй – с событием. Тогда специфичность равна n_{11}/N, чувствительность n_{22}/N, ошибка первого рода n_{12}/N, ошибка второго рода n_{21}/N.

ROC-кривая

При выборе параметров, сравнении различных конфигураций систем распознавания большую популярность получила т.н. ROC-кривая (ROC-curve, receiver operating characteristic). Предположим, что решается задача классификации 2 классов и подбирается некоторое значение параметра (пороговое значение отношения правдоподобия). Тогда для различных значений параметра можно построить график зависимости вероятности обнаружения события (чувствительности) как функцию от вероятности ложного срабатывания. Данный график носит название ROC-кривая. Изначально он использовался в электротехнике для выбора наилучшего порога отделения полезного сигнала от шума. По

этому графику можно наблюдать за зависимостью между ошибками первого и второго рода при изменении варьируемого параметра.



ROC кривая всегда находится выше прямой l_1, проходящей через точки (0,0) и (1,1). Кривая является неубывающей. Чем ROC-кривая выше l_1, тем меньше ошибка и лучше система распознавания (рис. справа). Если кривая совпадает с прямой l_1, это означает, что система распознавания непригодна, т.к. она принимает решения наугад с вероятностью правильного распознавания 50% (как при бросании монеты).

Используемые функции Matlab

Построение графиков	
plot	построение графиков 2D
surf	построение поверхностей
contour	построение контуров
xlabel, ylabel, zlabel	подписи к осям
title	подпись к графику
subplot	разделение окна на несколько частей для построения нескольких графиков
grid	координатная сетка
hold	построение нескольких зависимостей на одном графике
axis	задание рабочей области построения графика
meshgrid	генерация сетки значений для построения 3D-графиков
imshow	отображение изображений
Другие функции	
rand	генерация случайных величин
inpolygon	определение попадания точек внутрь полигона
imread	чтение изображений из файлов

Практические задания

Задание 1

Придумайте собственные варианты исходных данных в соответствие с правилами, изложенными ниже.

Исходные данные 1 – Крестики-нолики

Разделите таблицу 4х4 на крестики “X” и нолики “O” так, чтобы классы “0” и “X” были линейно неразделимы, а количество примеров каждого класса было одинаковым и равным 8. Например, так

X	0	0	0
X	X	X	X
0	0	X	0
0	X	X	0

Исходные данные 2 - Логическая функция 5 переменных

Придумайте логическую функцию (ЛФ) 5 переменных так, чтобы множество ее выходных значений 0 и 1 было линейно неразделимым. Как вариант, придумайте n = 5-6 чисел от 0 до 31. На двоичных представлениях этих чисел ЛФ будет принимать значение 1. На всех остальных значениях ЛФ будет принимать значение 0.

Например, пусть на значениях 1,4,6,12,15,27 функция принимает значение 1. Тогда таблица истинности выглядит следующим образом, как показано на табл. ниже.

X	x_1	x_2	x_3	x_4	x_5	Y
0	0	0	0	0	0	**0**
1	0	0	0	0	1	**1**
2	0	0	0	1	0	**0**
3	0	0	0	1	1	**0**
4	0	0	1	0	0	**1**
5	0	0	1	0	1	**0**
6	0	0	1	1	0	**1**
7	0	0	1	1	1	**0**
8	0	1	0	0	0	**0**
9	0	1	0	0	1	**0**
10	0	1	0	1	0	**0**
11	0	1	0	1	1	**0**
12	0	1	1	0	0	**1**
13	0	1	1	0	1	**0**
14	0	1	1	1	0	**0**
15	0	1	1	1	1	**1**

X	x_1	x_2	x_3	x_4	x_5	Y
16	1	0	0	0	0	**0**
17	1	0	0	0	1	**0**
18	1	0	0	1	0	**0**
19	1	0	0	1	1	**0**
20	1	0	1	0	0	**0**
21	1	0	1	0	1	**0**
22	1	0	1	1	0	**0**
23	1	0	1	1	1	**0**
24	1	1	0	0	0	**0**
25	1	1	0	0	1	**0**
26	1	1	0	1	0	**0**
27	1	1	0	1	1	**1**
28	1	1	1	0	0	**0**
29	1	1	1	0	1	**0**
30	1	1	1	1	0	**0**
31	1	1	1	1	1	**0**

Исходные данные 3 – Разбиение плоскости на 2 класса.

Прямоугольный участок плоскости с помощью отрезков прямых линий разбейте на два класса так, чтобы

- хотя бы один из классов состоял из нескольких непересекающихся частей;
- классы были линейно неразделимы.

Приведите графический эскиз полученного разбиения плоскости.

Исходные данные 4 – Разбиение плоскости на *n* классов

Прямоугольный участок плоскости с помощью отрезков прямых линий разбейте на 7-8 классов так, чтобы

- хотя бы два класса состояли из нескольких непересекающихся частей;
- несколько пар классов были линейно неразделимы.

Приведите графический эскиз полученного разбиения плоскости.

Исходные данные 5 – Непрерывная функция одной переменной

Определите функцию одной переменной в некотором интервале входных значений. Функция должна иметь умеренную сложность:

- как минимум 15-20 минимумов и максимумов;
- наличие колебаний различной частоты в различных диапазонах входных значений;
- наличие нескольких изломов.

Приведите графический эскиз полученной функции.

Исходные данные 8 – Многомерные образы

Сформируйте по $N = 100\text{-}200$ тысяч примеров для $M = 5\text{-}10$ различных классов, представленных в форме изображений невысокой размерности (10x10 или 20x20). Изображения могут быть ЧБ или цветными (буквы, символы, лица, объекты произвольного типа). Можно воспользоваться имеющимися в интернете базами данных.

Задание 2

Представьте исходные данные из задания 1 в форме некоторой выборки (P,T) и визуализируйте их.

Исходные данные 1 – Крестики-нолики

Сформируйте матрицу P размерности 2x16 (2-количество признаков, 16 – количество примеров). Можно принять, что все крестики и нолики находятся внутри квадрата с вершинами (0,0) и (1,1). Далее сформируйте матрицу T размерности 1x16 (1-количество выходов, 16 – количество примеров). Значения элементов T следует задать равными 0 или 1 в зависимости от того, к какому классу принадлежит пример.

Визуализируйте примеры на плоскости, построив примеры одного класса одним цветом и символом, а примеры другого класса – другим цветом и символом.

Исходные данные 2 - Логическая функция 5 переменных

Сформируйте матрицу P размерности 5x32 и матрицу T размерности 1x32 в соответствии с выбранной логической функцией 5 переменных.

Исходные данные 3 – Разбиение плоскости на 2 класса.

Шаг 1. Задайте координаты для прямоугольного участка плоскости и определите координаты всех точек, составляющих границы выбранного в задании 1 разбиения этого участка плоскости на классы.

Шаг 2. Напишите функцию в Matlab, которая будет принимать на вход матрицу P размерности [2 x N_{ex}] и выдавать на выходе матрицу-строку T размерности [1 x N_{ex}]. Элемент T_{1i} содержит номер класса, соответствующего i-му примеру $[P_{1i}\ P_{2i}]^{\mathrm{T}}$. Номер класса – 0 или 1.

Шаг 3. Сформируйте множество входных значений P в диапазоне рассматриваемого прямоугольного участка плоскости. Возможно два способа:

- регулярная решетка, когда берутся N_1 значений признака 1, N_2 значений признака 2 и строится массив из N_1xN_2 значений;

- набор случайно сгенерированных равномерно распределенных 2-мерных векторов.

Шаг 4. Далее с помощью своей функции определите номер класса T для значений P.

Шаг 5. Визуализируйте сформированную выборку на плоскости, построив примеры одного класса одним цветом и символом, а примеры другого класса – другим цветом и символом. Убедитесь визуально, что разбиение на классы выполнено корректно, сравнив полученный график с эскизом из задания 1.

Исходные данные 4 – Разбиение плоскости на *n* классов

Шаг 1. Задайте координаты для прямоугольного участка плоскости и определите координаты всех точек, составляющих границы выбранного в задании 1 разбиения этого участка плоскости на классы.

Шаг 2. Напишите функцию в Matlab, которая будет принимать на вход матрицу P размерности [2 x N_{ex}] и второй аргумент type, задающий формат выходных данных.

Если type равен 1, то функция должна выдавать на выходе матрицу-строку T размерности [1 x N_{ex}]. Элемент T_{1i} содержит номер класса, соответствующего i-му примеру $[P_{1i}\ P_{2i}]^{\mathrm{T}}$. Номер класса – 1, 2, 3, .., C, где C – количество классов.

Если type равен 2, то функция должна выдавать на выходе матрицу-строку T размерности [C x N_{ex}], где C – количество классов. Элемент T_{ji} равен 1, если i-й пример $[P_{1i}\ P_{2i}]^{\mathrm{T}}$ относится к j-му классу и равен 0, если нет

Шаг 3. Сформируйте множество входных значений P в диапазоне рассматриваемого прямоугольного участка плоскости.

Шаг 4. Далее с помощью своей функции определите номер класса T для значений P.

Шаг 5. Визуализируйте сформированную выборку на плоскости, построив примеры одного класса одним цветом и символом, а примеры другого класса – другим цветом и символом. Убедитесь визуально, что разбиение на классы выполнено корректно, сравнив полученный график с эскизом из задания 1.

Исходные данные 5 – Непрерывная функция одной переменной

Шаг 1 Выберите область возможных значений функции и задайте все ее основные параметры, ключевые координаты точек.

Шаг 2. Напишите функцию в Matlab, которая будет принимать на вход матрицу-строку P размерности [1 x N_{ex}] и выдавать на выходе матрицу-строку T размерности [1 x N_{ex}]. Элемент T_{1i} содержит значение функции, соответствующего входному значению i-го примеру P_{1i}.

Шаг 3. Сформируйте множество входных значений P в диапазоне возможных значений функции.

Шаг 4. Далее с помощью своей функции определите соответствующие значения T.

Шаг 5. Визуализируйте сформированную выборку, построив зависимость $T=f(P)$.

Убедитесь визуально, что функция строится корректно, сравнив полученный график с эскизом из задания 1.

Исходные данные 8 – Многомерные образы

Шаг 1. Визуализируйте сформированные образы. Для этого можно использовать функции imshow и subplot.

Шаг 2. Сгенерируйте образы, зашумленные относительно исходных. Варьируйте степень зашумления. Приведите примеры образов, имеющих различную степень зашумления.

Шаг 3. Сгенерируйте образы, имеющие различные геометрические искажения (масштаб, смещение, поворот). Приведите примеры образов, имеющих различные искажения.

Шаг 4. Сгенерируйте образы, являющиеся некоторой частью от исходных (например, половина изображения видна, а половина нет). Используйте различные варианты заслонения образов. Приведите примеры образов, имеющих различную степень и форму заслонения.

Задание 3

Исходные данные типа 3 – определение качества классификации, 2 класса

Шаг 1. Сформируйте выборку (P,T) объемом N, как в задании 2.

Шаг 2. Проинвертируйте метки классов для k (5, 10, 20) % случайно взятых примеров. Далее интерпретируйте эти данные, как ответ Y некоторого распознающего устройства (нейронной сети).

Шаг 3. На основании желаемых T и реальных Y ответов определите основные показатели качества распознавания:

- матрицу неточностей;
- среднюю вероятность ошибки и среднюю вероятность правильного распознавания;
- ошибки первого и второго рода, чувствительность, специфичность.

Убедитесь, что средняя ошибка совпадает со значением k.

Исходные данные типа 4 – определение качества классификации, N классов

Шаг 1. Сформируйте выборку (P,T) объемом N, как в задании 2.

Шаг 2. Измените метки классов на случайные другие (равномерно распределенные) для k (5, 10, 20) % случайно взятых примеров. Далее интерпретируйте эти данные, как ответ Y некоторого распознающего устройства (нейронной сети).

Шаг 3. На основании желаемых T и реальных Y ответов определите основные показатели качества распознавания:

- матрицу неточностей;
- среднюю вероятность ошибки и среднюю вероятность правильного распознавания;
- ошибки первого и второго рода для каждого класса.

Убедитесь, что средняя ошибка совпадает со значением k.

Исходные данные типа 5 – определение качества аппроксимации

Шаг 1. Сформируйте выборку (P,T) объемом N, как в задании 2.

Шаг 2. Добавьте к значениям T равномерный шум различной амплитуды (5, 10, 20 % от максимального значения). Далее интерпретируйте полученный сигнал, как ответ Y некоторого распознающего устройства (нейронной сети).

Шаг 3. На основании желаемых T и реальных Y ответов определите основные показатели качества распознавания:

- среднюю абсолютную ошибку;
- среднюю относительную ошибку;
- максимальную по модулю ошибку.

Сравните полученные значения ошибок и убедитесь, что они соответствуют исходному уровню шума.

Задание 4

Исходные данные типа 3 – кросс-валидация

Шаг 1. Сформируйте выборку (P,T) объемом N, как в задании 2.

Шаг 2. Разделите выборку на обучающую и тестовую, выбрав случайно k% примеров как тестовые, а остальные – как обучающие.

Шаг 3. Выполните визуализацию, как в задании 2, при этом отобразив тестовые и обучающие примеры разными символами.

Шаг 4. Примените K-fold кроссвалидацию (k=4,8) к исходной выборке. Для этого перемешайте выборку, разделите ее на k частей и далее сформируйте разбиения всей выборки на подвыборки так, чтобы одна часть была тестовой, а все остальные – обучающие. Визуализируйте полученные разбиения подходящим способом.