
Перекрёстная проверка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Разбейте множество из N объектов, каждый из которых принадлежит к одному из M классов, на K частей. Каждый объект должен попасть ровно в одну часть так, чтобы размеры частей, а также распределение классов по этим частям было сбалансировано. Формально, пусть $cnt(x, c)$ — число объектов с классом c попавших в часть x , тогда должно выполняться $\forall x, y, c : |cnt(x, c) - cnt(y, c)| \leq 1$ и $\forall x, y : |\sum_c cnt(x, c) - \sum_c cnt(y, c)| \leq 1$.

Формат входных данных

Первая строка: три целых числа N, M, K ($1 \leq N \leq 10^5, 1 \leq M, K \leq N$) — число объектов, классов и частей.

Вторая строка: N целых чисел C_i ($1 \leq C_i \leq M$) — класс i -го объекта.

Формат выходных данных

Выведите K строк. Каждая строка x начинается с целого числа S — размера части x . Далее идут S целых чисел — номера объектов попавших в часть x . Объекты нумеруются с единицы.

Пример

стандартный ввод	стандартный вывод
10 4 3	4 1 4 9 10
1 2 3 4 1 2 3 1 2 1	3 2 3 5
	3 6 7 8

Замечание

В первой части содержится четыре объекта, два из них первого класса, один второго и один четвёртого. Во второй и третьей части по три объекта первых трёх классов.

F-мера

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В результате эксперимента по классификации на K классов была получена матрица неточностей (Confusion matrix) CM , где $CM[c, t]$ — число объектов класса c , которые были классифицированы как t . Посчитайте по данной матрице неточностей средневзвешенную по классам макро и микро F-меру.

Формат входных данных

Первая строка содержит целое число K — число классов ($1 \leq K \leq 20$). Далее идёт K строк — описание матрицы неточностей. Каждая строка s содержит K целых чисел — s -я строка матрицы неточностей. $\forall c, t : 0 \leq CM[c, t] \leq 100$ и $\exists c, t : CM[c, t] \geq 1$.

Формат выходных данных

Выведите два вещественных числа с плавающей точкой — взвешенно усреднённую по классам макро и микро F-меру. Абсолютная погрешность ответа не должна превышать 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
2 0 1 1 3	0.6 0.6
3 3 1 1 3 1 1 1 3 1	0.326860841 0.316666667

Замечание

В первом примере классы распределены как 1:4. Точность (precision), полнота (recall) и F-мера первого класса равны 0, а второго 0.75. При этом средняя точность, полнота и F-мера равны 0.6.

Непараметрическая регрессия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте алгоритм непараметрической регрессии, который бы поддерживал различные функции расстояний, ядер и окон. Описание ядер можно найти здесь: <https://en.wikipedia.org/w/index.php?oldid=911077090>

Формат входных данных

Первая строка содержит два целых числа N и M — число объектов и признаков ($1 \leq N \leq 100$, $1 \leq M \leq 10$).

Далее идёт N строк — описание набора данных. Каждая строка i содержит $M + 1$ целое число $d_{i,j}$ ($-100 \leq d_{i,j} \leq 100$) — описание i -го объекта. Первые M из этих чисел признаки i -го объекта, а последнее — его целевое значение.

Следующая строка описывает объект запроса q . Она состоит из M целых чисел $d_{q,j}$ ($-100 \leq d_{q,j} \leq 100$) — признаки объекта q .

Далее идут три строки состоящих из строчных латинских букв.

Первая из них — название используемой функции расстояния: *manhattan*, *euclidean*, *chebyshev*.

Вторая — название функции ядра: *uniform*, *triangular*, *epanechnikov*, *quartic*, *triweight*, *tricube*, *gaussian*, *cosine*, *logistic*, *sigmoid*.

Третья — название типа используемого окна: *fixed* — окно фиксированной ширины, *variable* — окно переменной ширины.

Последняя строка содержит параметр окна: целое число h ($0 \leq h \leq 100$) — радиус окна фиксированной ширины, либо целое число K ($1 \leq K < N$) — число соседей учитываемое для окна переменной ширины.

Формат выходных данных

Выведите одно вещественное число с плавающей точкой — результат запроса.

Примеры

стандартный ввод	стандартный вывод
3 2 0 2 1 1 1 0 2 0 1 0 0 euclidean uniform fixed 2	0.0000000000
3 2 0 2 1 1 1 0 2 0 1 0 0 euclidean gaussian variable 2	0.6090086848

Замечание

В случае неопределённости, когда в окно не попало ни одного объекта, требуется вывести значе-

ние по умолчанию для задачи регрессии — среднее значение целевой переменной по всем объектам из обучающей выборке.

Линейная регрессия

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Найдите уравнения прямой аппроксимирующей положение объектов из заданного набора данных.

Формат входных данных

Первая строка содержит два целых числа N ($1 \leq N \leq 10^4$) — число объектов в обучающем множестве, и M ($1 \leq M \leq \min(N, 1000)$) — число признаков у объектов исключая зависимую переменную.

Следующие N строк содержат описание объектов. i -я из этих строк содержит описание i -го объекта, $M + 1$ целых чисел. Первые M из этих чисел: $X_{i,j}$ ($|X_{i,j}| \leq 10^9$) — признаки i -го объекта, а последнее Y_i ($|Y_i| \leq 10^9$) — значение его зависимой переменной.

Формат выходных данных

Выведите $M + 1$ вещественных чисел с плавающей точкой A_j — коэффициенты прямой из уравнения $Y = A_0 \cdot X_0 + A_1 \cdot X_1 + \dots + A_{M-1} \cdot X_{M-1} + A_M$

Система оценки

Пусть $Score = 100 \cdot \frac{B-S}{B-J}$, где S — SMAPE вашего решения, J — SMAPE решения эталона с запасом $\approx 1\%$, B — SMAPE наивного решения с запасом $\approx 2\%$.

Тогда Verdict = $\begin{cases} \text{Ok} & \text{Score} \geq 100 \\ \text{PartiallyCorrect} & 0 \leq \text{Score} < 100 \\ \text{WrongAnswer} & \text{Score} < 0 \end{cases}$

Примеры

стандартный ввод	стандартный вывод
2 1 2015 2045 2016 2076	31.0 -60420.0
4 1 1 0 1 2 2 2 2 4	2.0 -1.0

Замечание

Не стоит “дудосить” тестирующую систему для подбора оптимальных параметров алгоритма! Их следует настраивать локально используя следующие наборы данных: <https://drive.google.com/file/d/1D2xJ6ujn4qR73suNJ64DGosfU1b-xmgD>

Эти наборы данных отличаются от тех, на которых будет тестироваться ваше решение, но они получены тем же самым методом генерации. Каждый набор данных начинается с целого положительного числа M ($1 \leq M \leq 1000$) — число признаков. Далее следуют два множества объектов: тренировочное и тестовое. Каждое множество начинается с целого положительного числа N_t ($1 \leq N_t \leq 10^4$) — число объектов в множестве. Далее следуют N_t объектов в формате, который соответствует формату задачи на codeforces.

Метод опорных векторов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Найдите коэффициенты λ_i опорных векторов и сдвиг b , для классификации по формуле $class(x) = sign(\sum_i y_i \cdot \lambda_i \cdot k(x, x_i) + b)$, где x — это векторное описание запрашиваемого объекта, а k — функция ядра.

Формат входных данных

В первой строке находится целое число N ($1 \leq N \leq 100$) — число объектов в обучающем множестве.

Следующие N строк содержат описание объектов по одному объекту на строке. i -й объект описывается $N + 1$ целым числом: первые N из них $K_{i,j}$ ($|K_{i,j}| \leq 10^9$) — значение функции ядра между i -м и j -м объектом, последнее Y_i ($Y_i = \pm 1$) — класс i -го объекта.

Далее идёт строка содержащая целое число C ($1 \leq C \leq 10^5$) — ограничение на коэффициенты λ_i .

Формат выходных данных

Выведите $N + 1$ число с плавающей точкой: первые N чисел — коэффициенты λ_i ($0 \leq \lambda_i \leq C$, $\sum \lambda_i \cdot Y_i = 0$) соответствующие объектам из тренировочного множества, последнее число b ($|b| \leq 10^{12}$) — коэффициент сдвига.

Система оценки

Пусть $Score = 100 \cdot \frac{F-B}{J-B}$, где F — F_1 -мера вашего решения, J — F_1 -мера решения эталона с запасом $\approx 1\%$, B — F_1 -мера наивного решения с запасом $\approx 2\%$.

$$\text{Тогда } Verdict = \begin{cases} Ok & Score \geq 100 \\ PartiallyCorrect & 0 \leq Score < 100 \\ WrongAnswer & Score < 0 \end{cases}$$

Пример

стандартный ввод	стандартный вывод
6	0.0
5 4 6 9 11 10 -1	0.0
4 5 6 9 10 11 -1	1.0
6 6 8 12 14 14 -1	1.0
9 9 12 18 21 21 1	0.0
11 10 14 21 25 24 1	0.0
10 11 14 21 24 25 1	-5.0
1	

Наивный байесовский классификатор

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Реализуйте наивный байесовский классификатор.

Априорные вероятности классов оцениваются обыкновенным частотным методом.

Для оценки вероятности встречи слов в каждом классе используется модель Бернулли с аддитивным сглаживанием (сглаживание Лапласа) $p(x) = \frac{\text{count}(x) + \alpha}{\sum_{y \in Q} \text{count}(y) + \alpha \cdot |Q|}$, где x — рассматриваемое событие, а Q — множество всех событий.

Каждое слово это отдельный признак с двумя возможными событиями встретилось / не встретилось.

Формат входных данных

В первой строке содержится целое положительное число K ($1 \leq K \leq 10$) — число классов.

Во второй строке содержится K целых положительных чисел λ_C ($1 \leq \lambda_C \leq 10$) — штрафы за ошибки классификации сообщений соответствующих классов.

В третьей строке содержится целое положительное число α ($1 \leq \alpha \leq 10$) — интенсивность аддитивного сглаживания.

Следующая строка содержит целое положительное число N ($1 \leq N \leq 200$) — число сообщений в обучающей выборке.

Следующие N строк содержат описания соответствующих сообщений из обучающей выборки. Каждое сообщение в ней начинается с целого положительного числа C_i ($1 \leq C_i \leq K$) — класса к которому относится i -е сообщение. Далее следует целое положительное число L_i ($1 \leq L_i \leq 10^4$) — число слов в i -м сообщении. Затем следует содержание сообщения — L_i слов состоящих из маленьких латинских букв.

Далее в отдельной строке содержится целое положительное число M ($1 \leq M \leq 200$) — число сообщений в проверочной выборке.

Следующие M строк содержат описания соответствующих сообщений из проверочной выборки. Каждое сообщение в ней начинается с целого положительного числа L_j ($1 \leq L_j \leq 10^4$) — число слов в j -м сообщении. Затем следует содержание сообщения — L_j слов состоящих из маленьких латинских букв.

Гарантируется, что сумма длин всех сообщений в обучающей и проверочной выборках меньше чем $2 \cdot 10^6$.

Формат выходных данных

Выведите M строк — результаты мягкой классификации оптимального наивного байесовского классификатора соответствующих сообщений из проверочной выборки.

Каждый j -й результат мягкой классификации должен содержать K чисел p_C — вероятности того, что j -е сообщение относится к классу C .

Пример

стандартный ввод	стандартный вывод
3	0.4869739479 0.1710086840 0.3420173681
1 1 1	0.1741935484 0.7340501792 0.0917562724
1	0.4869739479 0.1710086840 0.3420173681
4	0.4869739479 0.1710086840 0.3420173681
1 2 ant emu	0.4869739479 0.3420173681 0.1710086840
2 3 dog fish dog	
3 3 bird emu ant	
1 3 ant dog bird	
5	
2 emu emu	
5 emu dog fish dog fish	
5 fish emu ant cat cat	
2 emu cat	
1 cat	

Замечание

В примере условные вероятности выглядят следующим образом:

$p(w_x c_y)$	ant	bird	dog	emu	fish
c_1	3/4	1/2	1/2	1/2	1/4
c_2	1/3	1/3	2/3	1/3	2/3
c_3	2/3	2/3	1/3	2/3	1/3

Слово cat не рассматривается, так как оно ни разу не встретилось в обучающей выборке.

Для первого запроса $p(c_1|M) \cdot p(M) = \frac{2}{4} \cdot (1 - \frac{3}{4}) \cdot (1 - \frac{1}{2}) \cdot (1 - \frac{1}{2}) \cdot (\frac{1}{2}) \cdot (1 - \frac{1}{4})$ и $p(c_1|M) = \frac{3/256}{3/256+1/243+2/243}$

Дерево принятия решений

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Постройте дерево принятия решений.

Формат входных данных

Первая строка содержит три целых положительных числа M ($1 \leq M \leq 100$) — число признаков у объектов (исключая класс), K ($1 \leq K \leq 20$) — число классов и H ($1 \leq H \leq 10$) — максимальная глубина (в рёбрах) дерева принятия решений.

Вторая строка содержит целое положительное число N ($1 \leq N \leq 4000$) — число объектов в обучающей выборке.

Следующие N строк содержат описания объектов в обучающей выборке. В i -й из этих N строк перечислено $M+1$ целое число: первые M чисел $A_{i,j}$ ($|A_{i,j}| \leq 10^9$) — признаки i -го объекта, последнее число C_i ($1 \leq C_i \leq K$) — его класс.

Формат выходных данных

Выведите построенное дерево принятия решений.

В первой строке выведите целое положительное число S ($1 \leq S \leq 2^{11}$) — число вершин в дереве.

В следующих S строках выведите описание вершин дерева. В v -й из этих строк выведите описание v -й вершины:

- Если v -я вершина узел, выведите через пробел: заглавную латинскую букву 'Q', целое положительное число f_v ($1 \leq f_v \leq M$) — индекс признака по которому происходит проверка в данном узле, вещественное число с плавающей точкой b_v — константа с которой происходит сравнения для проверки, два целых положительных числа l_v и r_v ($v < l_v, r_v \leq S$) — индекс вершины дерева в которую следует перейти, если выполняется условие $A[f_v] < b_v$, и индекс вершины дерева в которую следует перейти, если условие не выполняется.
- Если v -я вершина лист, выведите через пробел: заглавную латинскую букву 'C' и целое положительное число D_v ($1 \leq D_v \leq K$) — класс объекта попавшего в данный лист.

Вершины нумеруются с единицы. Корнем дерева считается первая вершина.

Система оценки

Решение будет проверено на секретном наборе данных. На основании предсказанных и реальных классов вычисляется усреднённая по классам микро F_1 -мера.

Пусть $Score = 100 \cdot \frac{F-B}{J-B}$, где F — F_1 -мера вашего решения, J — F_1 -мера решения эталона с запасом $\approx 1\%$, B — F_1 -мера наивного решения с запасом $\approx 2\%$.

$$\text{Тогда } Verdict = \begin{cases} Ok & Score \geq 100 \\ PartiallyCorrect & 0 \leq Score < 100 \\ WrongAnswer & Score < 0 \end{cases}$$

Пример

стандартный ввод	стандартный вывод
2 4 2	7
8	Q 1 2.5 2 5
1 2 1	Q 2 2.5 3 4
2 1 1	C 1
3 1 2	C 4
4 2 2	Q 2 2.5 6 7
3 4 3	C 2
4 3 3	C 3
1 3 4	
2 4 4	