

Государственное образовательное учреждение высшего профессионального образования  
Санкт-Петербургский национальный исследовательский университет  
Информационных Технологий, Механики и Оптики  
Факультет инфокоммуникационных технологий

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

По Мультимедиа Технонолиям

На тему: «Квантование»

Проверил(а): Хлопотов М.В.

\_\_\_\_\_

Дата: «\_\_» \_\_\_\_\_ 201\_\_ г.

Оценка: \_\_\_\_\_

Работу выполнил(а):

Никончук А.П.  
Студент(ка) группы К3242  
Дневного отделения

\_\_\_\_\_

Цель: сравнить результаты вариаций квантования

Задачи:

1. реализовать равномерный скалярный квантователь
2. реализовать неравномерный скалярный квантователь построенный по алгоритму Ллойда-Макса
3. реализовать векторный квантователь с параметрами кодовой книги – 8 слов по 2 элемента
4. реализовать аналог алгоритма k-means
5. визуализировать результат
6. сравнить результаты квантователем среднеквадратичной ошибкой и энтропией

Выполнение работы:

При этом значение сигнала может округляться либо до ближайшего уровня, либо до меньшего или большего из ближайших уровней в зависимости от способа кодирования. Такое квантование называется скалярным. Существует также векторное квантование — разбиение пространства возможных значений векторной величины на конечное число областей и замена этих значений идентификатором одной из этих областей. Далее последовательно будет рассмотрено каждый из них.

Выбирается 100 значений нормального распределения, мат. ожидание 0, дисперсия 1.

Для генерации 100 случайных чисел из нормального распределения используется функция `np.random.normal(mu, sigma, n)`

`loc` (float или `array_like` из `floats`) - среднее («центр») распределения.

`scale` (float или `array_like` из `floats`) - стандартное отклонение (распространение или «ширина») распределения

`size` (int или кортеж из int, необязательно) - форма выхода. Если заданная форма, например `(m, n, k)`, тогда, образцы взяты. Если размер (по умолчанию), возвращается одно значение, если и оба являются скалярами. В противном случае образцы составляются. `(m, n, k)m * n * k`

Возвращает: `ndarray` или скаляр с отобранными образцами из параметризованного нормального распределения.

К основным характеристикам законов распределения относятся: наиболее вероятное значение измеряемой величины под названием математическое ожидание (`mean`); мера рассеивания случайной величины вокруг математического ожидания под названием среднеквадратическое отклонение (`std`). Исходя из определения полагаем, что оно выходит в качестве значения `mu`, исключая варианты малой плотности распределения при других значениях дисперсии. Так или иначе в задании значений нормального распределения в `python` применяется выбранная процедура, а `mu` определяется перманентно мат. ожиданием.

Квадрат стандартного отклонения дает дисперсию, и хотя в заданных параметрах используем 1, следует вычислять `sigma` как корень из переданной дисперсии.

$(\mu = 0, \sigma = 1)$

```

10 def generate_sequence_N(mu, dispersion, n):
11     # sigma - стандартное отклонение. Квадрат стандартного отклонения,
12     # sigma ^ 2 , называется дисперсией.
13     sigma = np.sqrt(dispersion)
14     # нужно мат ожидание 0 дисперсия 1
15     # loc, mu - среднее значение, центр распределения
16     # scale - стандартное отклонение (распространение или «ширина») распределения.
17     #distribution = np.random.normal(loc=5.0,scale=1.0,size=100)
18     distribution = np.random.normal(mu, sigma, n)
19     print(distribution)
20
21     print(abs(mu - np.mean(distribution)))
22     print(abs(sigma - np.std(distribution, ddof=1)) <= 1)
23     return distribution
24
25 def get_histogram(mu, sigma, distribution):
26     count, bins, ignored = plt.hist(distribution, 30, density=True)
27     plt.plot(bins, 1/(sigma * np.sqrt(2 * np.pi)) * \
28             np.exp( - (bins - mu)**2 / (2 * sigma**2) ), \
29             linewidth=2, color='r')
30     plt.show()
31     return 1

```

Рисунок 1. Вариант генерации выборки нормального распределения и построения гистограммы с линией тренда

```

[ 1.3158006 -2.45677821  0.54352045 -0.14106508  0.23513193 -0.36292628
 0.90243361 -0.14536643 -1.46801483  0.44261825  0.38843632 -1.82063799
 1.31543399  0.69356634  0.16041174  0.98493171 -0.13551327  1.23541863
 0.95210511 -0.06585809 -0.82611768 -0.67309931 -0.76080306 -1.23371526
-1.09988864 -1.49680864  0.3667013 -1.25905782  0.30013024  0.66157705
-0.34055059  0.15382166 -0.66424259  1.33198847  1.71607822  1.48916607
 1.0117308  1.38634016  0.86760575 -0.96628115 -1.06085691 -0.83529262
-0.93367317 -0.11108838  0.16953682 -1.14657612  0.79002362  1.15747936
-1.83179412 -0.45135497  0.3635317 -1.57040541  2.56240298  2.03667232
 1.05787374 -0.7516366  0.18537683  0.13468368  0.09211  0.45813666
-0.28300264 -0.44608182  0.0939373  0.68842533 -0.22890885 -1.71786003
 0.63062916  0.0607422  0.15275705  1.37494718  2.03064677  0.1326493
 1.12692353 -0.79661524 -0.1688151 -0.15467217  0.7131292 -0.29994256
-0.56412924 -0.04145376 -0.43046424 -0.63252451 -1.37327877 -2.0999412
-0.40279749 -0.61871218  0.10206605 -0.01062984 -2.23605642 -1.61036094
-1.48024384  0.81023954  1.0326031  1.39715117  0.7382808 -0.28619269
-0.10121803  0.37099522  0.92983144  0.76886753]
0.000242932122360493
True

```

Рисунок 2. Полученная выборка

Последние две строчки результата вызова функции генерации распределения показывают среднее значение и проверку истинности соответствия дисперсии единице.

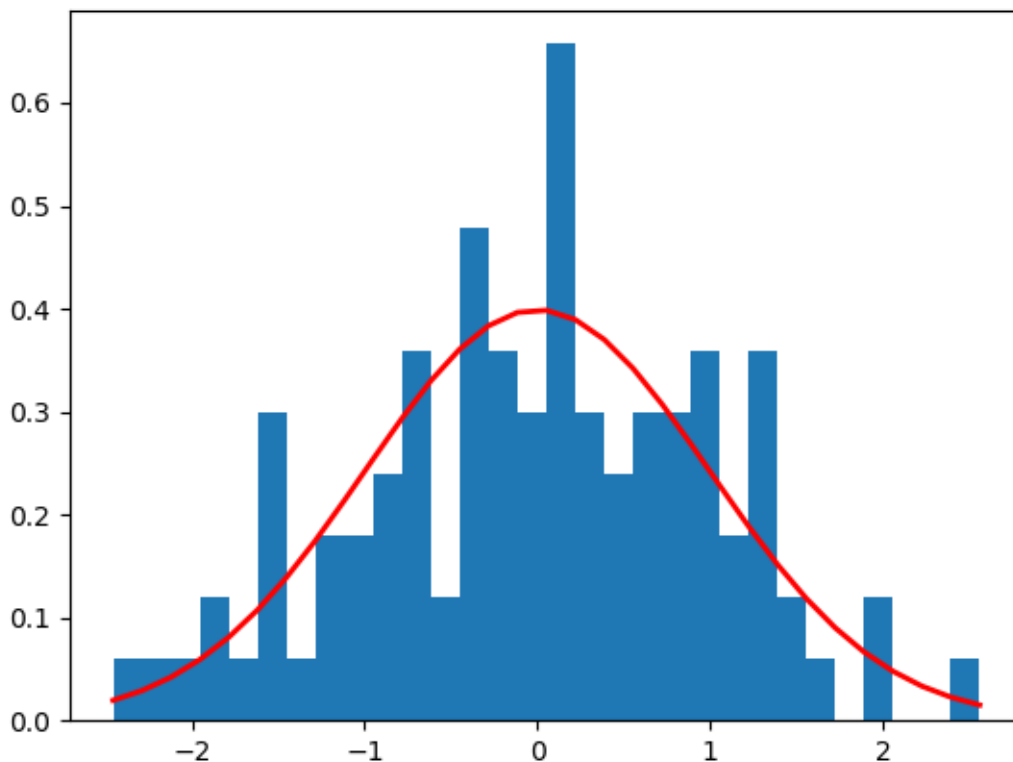


Рисунок 3. Гистограмма распределения с линией тренда выборки

Квантуется скалярным квантователем с переменной скоростью. Количество квантов от 7 до 9 на выбор.

Необходимо рассчитать оптимальный скалярный квантователь, который минимизирует некоторую функцию ошибки квантования  $q = \tilde{x} - x$ , где  $\tilde{x}$  - квантованное значение  $x$ . оптимальный квантователь минимизирует  $D$  (искажение, возникающее за счёт квантования сигнальных уровней) путём оптимального выбора выходных уровней и входного диапазона для каждого выходного уровня. Эту оптимизационную проблему рассматривали Ллойд (1982) и Макс (1960), и полученный оптимальный квантователь назван квантователем Ллойда-Макса. Необходимые условия для минимальных искажений можно получить дифференцированием  $D$  по  $\{x_k\}$  и  $\{\tilde{x}_k\}$ . Результат такой оптимизации выражается двумя уравнениями:

$$f(\tilde{x}_k - x_k) = f(\tilde{x}_{k+1} - x_k), k = 1, 2, \dots, L-1 \quad (1)$$

$$\int_{x_{k-1}}^{x_k} f'(\tilde{x}_k - x) p(x) dx = 0, k = 1, 2, \dots, L \quad (2)$$

Как частный случай рассмотрим минимизацию среднеквадратических значений искажений. В этом случае  $f(x) = x^2$ , и, следовательно, из (1) следует:

$$x_k = \frac{1}{2}(\tilde{x}_k + \tilde{x}_{k+1}), k=1, 2, \dots, L-1 \quad (3)$$

что является среднеарифметическим  $\tilde{x}_k$  и  $\tilde{x}_{k+1}$ . Соответствующие уравнения, определяющие  $\{\tilde{x}_k\}$ ,

$$\int_{x_{k-1}}^{x_k} (\tilde{x}_k - x) p(x) dx = 0, k=1, 2, \dots, L-1 \quad (4)$$

## Равномерное квантование

Обычно равномерное квантование используется тогда, когда о случайной величине  $X$  известно лишь то, что она попадает в некоторый диапазон  $X \in [x_{\min}; x_{\max}]$  и (или) необходимо обеспечить простейшую реализацию квантователя. Для удобства выберем в качестве диапазона возможных значений случайной величины непрерывного типа полуинтервал  $X \in [x_{\min}; x_{\max})$ .

При равномерном квантовании диапазон разбивается на  $N$  равных интервалов длины  $q = (x_{\max} - x_{\min}) / N$ :  $\Delta_j = [x_{\min} + jq; x_{\min} + (j+1)q), j=0, \dots, N-1$ . В качестве уровней квантования  $\{d_j\}_{j=0}^{N-1}$  выбираются середины интервалов:  $X \in [x_{\min}; x_{\max}]$

Правило квантования имеет простой вид: если  $X \in \Delta_j$ , то  $Q(X) = d_j$ .

```

67 def uniform_quantization(arr, L):
68     res = []
69     xmin, xmax, N = min(arr), max(arr), len(arr)
70     q = (xmax - xmin) / (L-1)
71     for j in range(N):
72         # arr[j] = arr[j: j+q]
73         a = (arr[j] - xmin)/q
74         res.append(xmin + (a+0.5)*q)
75     return res

```

Рисунок 4. Вариант кода преобразования равномерного скалярного квантователя

## Неравномерный скалярный квантователь построенный по алгоритму Ллойда-Макса

Вновь предположим, что случайная величина непрерывного типа  $X$  принимает значения из некоторого диапазона  $X \in [x_{\min}; x_{\max})$ . Разобьем интервал на  $N$  неравных частей

$\Delta = [t_j; t_{j+1}), j=0, \dots, N-1; t_0 = x_{\min}, t_N = x_{\max}$ . Внутри каждого полуинтервала дельта расположим точку уровня квантования  $d_j$ .

Правило квантования имеет тот же вид, что и ранее. Однако теперь параметры квантования – пороги  $\{t_j\}_{j=1}^{N-1}$  и уровни  $\{d_j\}_{j=0}^{N-1}$  – будем искать с точки зрения минимизации

величины математического ожидания квадрата разницы проквантованной и исходной выборки:

$$E = M((\tilde{X} - X)^2)$$

Алгоритм действий по обработке сигнала следующий.

Сначала формируется сигнал в цифровом виде с разрядностью  $\text{sqr}(L)$ , затем отсчеты одного блока сигнала подаются на анализирующее устройство, которое находит и запоминает максимальный по модулю отсчет данного блока. Затем вычисляется шаг квантования по формуле  $q = 2M/L$ , где  $M$  – максимальный по модулю отсчет блока сигнала,  $L$  – также количество уровней квантования.

После нахождения шага квантования  $q$  данный шаг запоминается и все следующие отсчеты анализируемого блока сигнала квантуются с новым шагом, затем обработанные отсчеты подаются на цифроаналоговый преобразователь.

Вычисление ошибки квантования (D)

```
84     E = []
85     dictionary_a = {}
86     dictionary_r = {}
87     for i in range(N):
88         E.append(abs(res[i]-arr[i]))
89     print('ошибка квантования', (max(E)+min(E))/2)
```

Рисунок 5. Вариант кода вычисления простейшего вычисления ошибки квантования

Вычисление энтропии

```
85     dictionary_a = {}
86     dictionary_r = {}
87     for i in range(N):
88         if arr[i] in dictionary_a:
89             dictionary_a[arr[i]] = dictionary_a.get(arr[i])+1
90         else:
91             dictionary_a[arr[i]] = 1
92         if res[i] in dictionary_r:
93             dictionary_r[res[i]] = dictionary_r.get(res[i])+1
94         else:
95             dictionary_r[res[i]] = 1
96         E.append(abs(res[i]-arr[i]))
97     ea = 0
98     er = 0
99     for i in dictionary_a:
100         ea += (math.fabs(dictionary_a.get(i) / N) * np.log2(dictionary_a.get(i) / N))
101     for i in dictionary_r:
102         er += (math.fabs(dictionary_r.get(i) / N) * np.log2(dictionary_r.get(i) / N))
103     print('энтропия', ea, '-', er, '=', ea - er)
```

Рисунок 6. Вариант вычисления энтропии

Вычисление среднеквадратичной ошибки

```
68 def mse(res1, res2):  
69     e = 0  
70     for i in range(len(res1)):  
71         e += (res2[i]-res1[i])**2  
72     e = (e / (len(res1) - 1))**(1/2)  
73     return e
```

Рисунок 7. Вариант функции вычисления среднеквадратичной ошибки

```
96 print('среднеквадратическая ошибка квантования msi', q*q/12, mse(arr, res))
```

Рисунок 8. Вариант кода вычисления среднеквадратичной ошибки

В результате сравнения выполнения равномерного и неравномерного (ллойда-макса) скалярных преобразований вышли значения с отличной от нуля энтропией и уменьшенной среднеквадратичной ошибкой. Уменьшение mse свидетельствует о верности применения критерия (скорее, чем алгоритма засчет поставленного только необходимого условия, а не порядка действий) Ллойда-Макса .



равномерное квантование

ошибка квантования 0.3585129421428571

энтропия -6.6438561897747395 - -6.6438561897747395 = 0.0

среднеквадратическая ошибка квантования msi 0.042843843227975874 0.36031906411340936

[1.674313542142857, -2.0982652678571427, 0.9020333921428572, 0.21744786214285705, 0.5936448721428573, -0.004413337857142707, 1.2609465521428573, 0.21314651214285707, -1.109501887857143, 0.8011311921428574, 0.7469492621428571, -1.462125047857143, 1.6739469321428575, 1.0520792821428566, 0.5189246821428566, 1.3434446521428574, 0.2229996721428571, 1.5939315721428575, 1.3106180521428574, 0.2926548521428573, -0.46760473785714285, -0.3145863678571428, -0.40229011785714297, -0.8752023178571429, -0.7413756978571429, -1.1382956978571428, 0.7252142421428567, -0.900544877857143, 0.6586431821428573, 1.0200899921428572, 0.017962352142857352, 0.5123346021428574, -0.30572964785714296, 1.6905014121428574, 2.0745911621428568, 1.8476790121428572, 1.3702437421428573, 1.7448531021428573, 1.2261186921428573, -0.6077682078571427, -0.702343967857143, -0.4767796778571429, -0.575160227857143, 0.2474245621428568, 0.5286049762142857, -0.788063177857143, 1.1485365621428572, 1.5159923021428572, -1.4732811778571429, -0.0928420278571429, 0.7220446421428566, -1.211892467857143, 2.9209159221428567, 2.3951852621428578, 1.416386682142857, -0.3931236578571431, 0.5438897721428568, 0.493196622142857, 0.45062294214285714, 0.8166496021428573, 0.07551030214285737, -0.08756887785714307, 0.4524502421428567, 1.046938272142857, 0.1296040921428574, -1.359347087857143, 0.9891421021428575, 0.4192551421428572, 0.5112699921428572, 1.7334601221428572, 2.3891597121428574, 0.4911622421428574, 1.4854364721428568, -0.4381022978571427, 0.189697842142857, 0.20384077214285723, 1.0716421421428568, 0.05857038214285737, -0.20561629785714297, 0.3170591821428572, -0.07195129785714283, -0.27401156785714287, -1.0147658278571425, -1.7414282578571427, -0.04428454785714253, -0.26019923785714294, 0.46057899214285714, 0.3478831021428568, -1.877543477857143, -1.2518479978571428, -1.1217308978571428, 1.1687524821428572, 1.391116042142857, 1.7556641121428576, 1.0967937421428573, 0.07232025214285676, 0.2572949121428567, 0.729508162142857, 1.2883443821428573, 1.1273804721428573]

неравномерное квантование ллойда-макса

ошибка квантования 0.14460196000000003

энтропия -6.6438561897747395 - -5.543074189428577 = -1.1007820003461628

среднеквадратическая ошибка квантования msi 0.018332912674214816 0.11234777703490778

[1.228389105, -2.45677821, 0.3670037075, 0.0, 0.4551594975, -0.4551594975, 0.910318995, 0.0, -1.3654784925, 0.4551594975, 0.4551594975, -1.82063799, 1.31543399, 0.617709315, 0.3088546575, 0.9265639725000001, 0.0, 1.23541863, 1.12260648, 0.0, -0.74840432, -0.74840432, -0.74840432, -1.12260648, -1.12260648, -1.49680864, 0.3329971175, -1.287058665, 0.429019555, 0.85803911, -0.429019555, 0.0, -0.85803911, 1.287058665, 1.71607822, 1.48916607, 1.0397551200000001, 1.38634016, 0.85993209, -0.85993209, -1.15747936, -0.91589706, -0.91589706, 0.0, 0.0, -1.28120149, 0.640600745, 1.28120149, -1.921802235, -0.640600745, 0.640600745, -1.28120149, 2.56240298, 2.03667232, 1.05787374, -0.7516366, 0.1721063325, 0.1721063325, 0.0, 0.4294650075, -0.4294650075, -0.4294650075, 0.0, 0.5076616925, 0.0, -1.5229850775, 0.5076616925, 0.0, 0.0, 1.5229850775, 2.03064677, 0.0, 1.12692353, -0.79661524, -0.1782823, 0.0, 0.5249853, -0.5249853, 0.0, -0.5249853, -0.559014105, -1.11802821, -2.23605642, -0.559014105, -0.559014105, 0.0, 0.0, -2.23605642, -1.61036094, -1.48024384, 0.698575585, 1.0478633775000001, 1.39715117, 0.6973735799999999, -0.23245786, 0.0, 0.46491572, 0.92983144, 0.76886753]

оригинальный массив

[1.3158006, -2.45677821, 0.54352045, -0.14106508, 0.23513193, -0.36292628, 0.90243361, -0.14536643, -1.46801483, 0.44261825, 0.38843632, -1.82063799, 1.31543399, 0.69356634, 0.16041174, 0.98493171, -0.13551327, 1.23541863, 0.95210511, -0.06585809, -0.82611768, -0.67309931, -0.76080306, -1.23371526, -1.09988864, -1.49680864, 0.3667013, -1.25905782, 0.30013024, 0.66157705, -0.34055059, 0.15382166, -0.66424259, 1.33198847, 1.71607822, 1.48916607, 1.0117308, 1.38634016, 0.86760575, -0.96628115, -1.06085691, -0.83529262, -0.93367317, -0.11108838, 0.16953682, -1.14657612, 0.79002362, 1.15747936, -1.83179412, -0.45135497, 0.3635317, -1.57040541, 2.56240298, 2.03667232, 1.05787374, -0.7516366, 0.18537683, 0.13468368, 0.09211, 0.45813666, -0.28300264, -0.44608182, 0.0939373, 0.68842533, -0.22890885, -1.71786003, 0.63062916, 0.0607422, 0.15275705, 1.37494718, 2.03064677, 0.1326493, 1.12692353, -0.79661524, -0.1688151, -0.15467217, 0.7131292, -0.29994256, -0.56412924, -0.04145376, -0.43046424, -0.63252451, -1.37327877, -2.0999412, -0.40279749, -0.61871218, 0.10206605, -0.01062984, -2.23605642, -1.61036094, -1.48024384, 0.81023954, 1.0326031, 1.39715117, 0.7382808, -0.28619269, -0.10121803, 0.37099522, 0.92983144, 0.76886753]



## Векторное квантование

В системах цифровой обработки изображений стремятся уменьшить число уровней и порогов квантования, т.к. от их количества зависит длина двоичного кодового слова, которым представляются проквантованные отсчеты в ЭВМ.

Процесс квантования отсчетов дискретного сигнала можно рассматривать как преобразование случайной величины непрерывного типа  $x$  в случайную величину дискретного типа  $\tilde{x} = Q(x)$ . Трактуя уровни квантования как символы некоторого алфавита  $\{d_0, \dots, d_{N-1}\}$ , цифровой сигнал можно считать “словом”, составленным из этого алфавита, “буквы” которого представляют собой проквантованные отсчеты дискретного сигнала.

Составляется кодовая книга 8 слов по 2 элемента. Оптимальная кодовая книга строится при помощи алгоритма Линде-Бузо-Грея (Linde-Buzo-Gray algorithm) – обобщение алгоритма Ллойда.

Задаётся количество кластеров, которое увеличивается с каждым шагом в два раза. Выбор числа уровней квантования 8 оказалось удобным.

На каждом шаге цикла происходит условное разделение на словарь (книга).

Вычисляется новый центр кластеров, выделенных по уровню квантования и случайному разделению с вычислением дистанции (ищется центр масс).

Векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике.

Предыдущие шаги, за исключением первого (случайного выбора начальных точек), повторяются, пока алгоритм не сойдётся.

Алгоритм завершается, когда на какой-то итерации не происходит изменения внутрикластерного расстояния. Это происходит за конечное число итераций, так как количество возможных разбиений конечного множества конечно, а на каждом шаге суммарное квадратичное отклонение  $V$  уменьшается, поэтому заикливание невозможно.

Алгоритм не был до конца реализован в связи с неопределенностью по определению словаря (кодовой книгой) и сложностью вычислений как вспомогательных значений, а также

### Выводы:

В результате выполнения дополнительной лабораторной работы изучены вариации квантования, изучены возможности визуализации графиков, в том числе делений на кластеры Воронова, отчего только больше подчеркивается выделение алгоритмов визуализации в отдельные задачи. А также произведен обзор средств вычислений и манипуляций со статическими данными.