Домашнее задание №8

Deadline: 20.05.2017 23:59:59

ФИВТ, АПТ, Курс по машинному обучению, Весна 2017, Модуль Unspervised Learning,

Alexey Romanenko, alexromsput@gmail.com

Organization Info

Дополнительный материал для выполнения дз:

- Воронцов К. В. Математические методы обучения по прецедентам. 2012. http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf (разделы 5.2 и 7.1)
- Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Springer: Data Mining, Inference, and Prediction. — 2nd ed. — Springer-Verlag. 2009. — 746 p.http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf (глава 14)

Оформление дз:

- Присылайте выполненное задание на почту `ml.course.mipt@gmail.com`
- Укажите тему письма в следующем формате `ML2017_fall <номер_группы> <фамилия>`, к примеру -- `ML2017_fall 496 ivanov`
- Выполненное дз сохраните в файл `<фамилия>_<группа>_task<номер задания>.ipnb`, к примеру -- `ML2017_496_task1.ipnb`

Вопросы:

- Присылайте вопросы на почту `ml.course.mipt@gmail.com`
- Укажите тему письма в следующем формате `ML2016_fall Question <Содержание вопроса>`

- **PS1**: Используются автоматические фильтры, и просто не найдем ваше дз, если вы не аккуратно его подпишите.
- **PS2**: Дедлайн жесткий, в том числе помтоу что это ДЗ последнее в курсе.

Контрольные вопросы (0 % - для самоконтроля)

Ответе на вопросы своими словами (загугленный материал надо пересказать), ответ обоснуйте (напишите и ОБЪЯСНИТЕ формулки если потребуется), если не выходит, то вернитесь к лекции дополнительным материалам:

Вопрос 1: В чём заключается проблема мультиколлинеарности?

Вопрос 2: Какие проблемы при обучении алгоритмов возникают из-за большой разамерности

пространства признаков?

Вопрос 3: В чем суть проклятия размерности?

Вопрос 4: Какая связь между решением задачи PCA и SVD-разложение матрицы регрессии?

Вопрос 5: Почему в tSNE расстояние между парамми объектов измеряется "по-стьюденту" и как это помогает решить проблему "скрученности" (crowding problem)?

Вопрос 6: На какой идее базируются алгоритмы аггломеративной кластеризации? Напишите формулу Ланса-Вильма

Вопрос 7: Какие два шага выделяют в алгоритме кластеризации k-means?

Вопрос 8: В чём отличия (основные упрощения) k-means от EM-алгоритма кластеризации?

Вопрос 9 Какой принцип работы графовых алгоритмов кластеризации?

Вопрос 10 В чем некорректность постановки задачи кластеризации?

PS: Если проверяющий не понял ответ на большинство вопросов, то будет пичалька. Пишите так, чтобы можно было разобраться.

Вопросы по теории (30%)

Задача 1 Ответьте на вопросы:

1) Как можно не прибегая к визуализации понять, что кластерная структура у данного облака точек отсутствует? 2) Какие из алгоритмов кластеризации могут выделять кластеры с ленточной структурой? 3) Какие алгоритмы кластеризации чувствительны к шуму и перемычкам? 4) Каким образом приближают «центр кластера» в нелинейных пространствах? 5) Каким образом можно определять число кластеров?

1. Можно выбрать метрику сохранности информации при кластеризации, запустить какой-нибудь алгоритм кластеризации с заданным числом искомых кластеров и посмотреть на график этой метрики в зависимости от количества кластеров. Если он будет линейным с одинаковым k, то, скорее всего, в данных отсутствует кластерная структура.
2. DBSCAN, поиск связных компонент, AgglomerativeClustering. Эти алгоритмы не опираются на выпуклость кластеризованных множеств.
3. поиск связных компонент, кратчайший незамкнутый путь - к шуму.
4. Используют различные функции расстояний.
5. По графику качества выбрать переломный момент, после которого возрастание качества замедляется. Либо использовать визуализацию.

Задача 2 Даны пять точек на числовой оси $X = (1; 5; 7; 8; 8)$, число кластеров равно 2. Рассчитайте ответ алгоритма K-means (финальные центры кластеров), если начальные центры кластеров c1 = 1, c2 = 10.

1. c1 = {1, 5} c2 = {7, 8, 8} c1.c = 3 c2.c = 7.66

The End.

Задача 3 Докажите, что the k-means всегда сходится.

Введем фукнкцию $E(C) := \sum_{x \in X} ||x - c(x)||^2$, где $c(x)$ — текущий центр кластера x, а C — текущее разбиение на кластеры.

1) Заметим, что центры кластеров целиком определяются объектами и текущим разбиением на кластеры.

2) Заметим, что разбиений на кластеры конечное число.

3) Заметим, что шаг 3 не увеличивает $E(C)$ (потому что для каждого x мы не увеличили $||x - (c)||^2$, а шаг 4 ее либо уменьшает, либо не изменяет, но в таком случае мы не поменяли y_i (Потому что в центре тяжести достигается строгий глобальный минимум 2-нормы))

Из 1) и 3) следует, что мы не можем второй раз прийти в одно и то же состояние (зациклится) и не остановиться (потому что E(C) уменьшается).

Из 2) следует, что рано или поздно мы зациклимся

Значит, мы когда-нибудь остановимся, то есть сойдемся

Задача 4 Для сжатия размерности пространства алгоритм PCA применяется датасету с количеством признаков $D = 100$. Наблюдается следующий спектр собственных значений матрицы объектов-признаков.

Ответье на вопросы

- 1) Высокая ли эффективная размерность пространства признаков (intrinsic dimensionality) (насколько она близка к 100)? Высокая. почти нигде нет переломного момента, после которого спект резко падает.
- 2) Можно ли перевести датасет с помощью PCA в пространство меньшей размерности с минимальными потерями точности? Если да, то чему примерно будет равна размернось Нельзя

Практическое задание 1 (30%)

Реализуйте PCA

In [2]:

```
import numpy as np
import pylab as plt
import sklearn as sk
import matplotlib.pyplot as plt
%matplotlib inline
'''
Performs the Principal Coponent analysis of the Matrix F
Matrix must be n * l dimensions
where n is # features
l is # samples
'''

def PCA(F, varRetained = 0.95, show = False):
```

```
    # Input
    #     F - initaial matrix
    # Compute Covariance Matrix Sigma
    # Input
    (n, l) = F.shape
    Sigma = 1.0 / l * np.dot(F, np.transpose(F))
#     print Sigma
    # Compute eigenvectors and eigenvalues of Sigma by SVD
    # U, V - matrix, d - array: Sigma = U * np.diag(d) * V
    U, d, V = np.linalg.svd(Sigma)

    # compute the value m: number of minumum features that retains the give
n variance varRetaine
    dTot = np.sum(d)
    var_i = np.array([np.sum(d[: i + 1]) / \
                dTot * 100.0 for i in range(n)])
#     print(var_i)
    for i,v in enumerate(var_i):
        if v > varRetained * 100:
            break
    m = i
    print '{:,.2f} variance retained in {} dimensions'.format( varRetained,
m)

    # plot the variance plot
    if show:
        plt.plot(var_i)
        plt.xlabel('Number of Features')
        plt.ylabel(' Percentage Variance retained')
        plt.title('PCA $\% \sigma^2 $ vs # features')
        plt.show()

    # compute the reduced dimensional features by projection
    U_reduced = U[:, :m]
#     print(np.diag(d))
    G = (np.dot(U, np.diag(d)))[:, :m]

    return G, U_reduced
```

In [2]:

```
# Примените алгоритм к данным MNIST
from sklearn.model_selection import train_test_split

from sklearn.datasets import load_digits
X, y = load_digits(return_X_y=True)
X_train, X_test, Y_train, Y_test = train_test_split(X, y, train_size=0.7)
```

In [3]:

```
##############################################################
# PCA of training set
print 'Performing PCA - Principal COmponent Analysis'

Z, U_reduced = PCA(X.T, varRetained = 0.95, show = False)
```

```
Performing PCA - Principal COmponent Analysis
0.95 variance retained in 15 dimensions
```

In [4]:

```
print Z
print U_reduced
```

```
[[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    0.00000000e+00   0.00000000e+00   0.00000000e+00]
 [ -1.54488948e+01  -3.10607025e+00   1.61146465e+00  -2.60319317e+00
    2.01100765e+00  -9.87497834e-01   6.26867673e-01  -4.31849803e-01
    8.24296592e-01  -8.03695500e-01   4.06519978e-01  -2.44323977e-01
   -1.37001215e+00   4.73831836e-01  -3.02828618e-01]
 [ -2.69518610e+02  -4.01097776e+01   7.32497095e+00  -1.74114581e+01
    1.83490771e+01  -6.05705316e+00   3.52884908e+00  -2.79240905e+00
    9.25282128e+00  -3.31234727e+00  -1.29583417e+00  -3.39274459e+00
   -6.32707466e+00   2.87901177e+00  -1.27824278e+00]
 [ -6.14649482e+02  -2.45925849e+01   1.42525281e-01  -1.72615881e+01
    2.09876415e+01   4.05524837e+00  -4.09857782e+00   1.84387204e+00
    7.14770402e+00   8.32651323e-01   9.32887104e-02  -3.07925393e+00
    2.98921283e+00  -3.87517228e+00   2.74876072e+00]
 [ -6.14615254e+02  -6.16647957e+00   7.70669818e+00   2.04655450e+01
    4.66743084e+00   5.14829571e-01  -3.46337477e+00  -2.90820682e+00
   -4.01224682e+00  -7.59036051e+00   8.56646552e+00   5.10070438e+00
   -2.01359459e+00  -3.79186723e+00   7.32072298e+00]
 [ -2.97452192e+02  -1.73953051e+01   1.90725423e+01   3.79840544e+01
    1.03032745e+01  -1.18550563e+01  -5.09990220e+00  -3.50443276e+00
   -1.39118579e+01  -1.03873064e+01   9.46247762e+00   4.01453257e-01
   -2.68428758e+00   5.05359929e+00   1.55121948e+00]
 [ -6.79444533e+01  -1.50037462e+00   1.02909054e+01   1.64221994e+01
    7.27423154e+00  -6.52640838e+00  -3.21599297e+00   8.77351588e-01
   -4.29629499e+00  -2.15556794e+00   4.85184804e+00  -4.88677181e+00
   -2.36845387e-01   4.04762374e+00  -2.03484322e+00]
 [ -6.08439967e+00   4.08167759e-01   1.33468708e+00   2.34818356e+00
    9.03406310e-01  -2.70025689e-01  -5.80907300e-01   6.23729231e-01
   -2.92256031e-01   1.07788989e-01   7.52290501e-01  -1.35792891e+00
    8.08565803e-02   3.51204699e-01  -3.96048787e-01]
 [ -3.08155344e-01  -5.76991764e-02   2.48799277e-02  -5.43487482e-02
    7.46345438e-03  -7.89225757e-03   6.09675181e-03  -1.27035203e-02
   -2.20864174e-02  -1.07614032e-02  -6.97922944e-03   4.30307206e-02
   -2.98859409e-02  -1.70667636e-02   1.22466414e-02]
 [ -1.03256422e+02  -2.14015295e+01   3.12527737e+00  -1.10741339e+01
    7.67647078e+00  -3.71279496e+00   3.71007599e-01  -8.80685588e-02
    2.34730997e+00  -7.82858418e-01   8.94587632e-01   3.54768008e+00
   -5.03159233e+00   3.92189365e-01  -5.27595244e-01]
 [ -5.39980099e+02  -4.39557750e+01  -1.11351305e+01  -1.09611705e+01
    3.23660945e+01  -3.77155483e+00  -1.61193619e+00   1.09304768e+00
    1.19769066e+00   2.03298837e+00  -2.73336020e+00   2.28516164e+00
    1.57552606e+00   6.16042397e-01  -2.02433289e+00]
 [ -6.23236165e+02   2.62867167e+01  -4.41646877e-01  -7.70143289e+00
    5.09512329e+00  -2.86346676e-01   3.20862731e+00   7.41016418e+00
   -9.72483358e-01   3.05282462e+00   5.03330010e+00  -3.81797231e+00
    5.20511894e+00  -2.60124387e+00  -1.67424476e+00]
 [ -5.37508806e+02  -8.65303495e+00   1.37661896e+01   1.67674655e+00
   -1.40638706e+01   8.48683895e+00   1.44731484e+01  -1.71782515e+00
    3.39611080e+00   2.05221922e+00   9.14956153e+00   6.48662943e+00
   -2.15009736e+00  -4.13679187e+00  -5.32852904e+00]
 [ -4.26754141e+02  -3.91373542e+01   7.77972320e+00   4.09709738e+01
    4.29647649e+00   1.41777975e+01   3.21576257e+00  -3.69722762e+00
   -1.53398220e+01  -4.38308500e+00   2.96998226e-01   2.70498326e+00
    1.26806043e+00  -1.21988860e+00   1.57181350e-02]
 [ -9.33128829e+01  -2.66783703e+00   9.74685592e+00   2.26375398e+01
```

```
[ -1.xxxxxxxe+02  -2.xxxxxxxe+00  -1.xxxxxxxe+00   2.xxxxxxxe+01
   4.74563560e+00  -2.44504035e+00  -2.69475173e+00   1.82756347e+00
  -6.27819614e+00  -2.98721912e+00   1.64493149e+00  -7.22927912e+00
   2.95153658e-02   2.56199009e+00  -2.14219398e+00]
[ -5.00981142e+00   8.04329177e-01   5.78305113e-01   2.12116007e+00
   2.20617196e-01  -6.24773353e-01  -2.69578171e-01   4.97858306e-01
   1.64547721e-03   8.38781153e-02   4.23848122e-01  -1.14271374e+00
  -1.42265153e-01  -4.39792315e-03  -2.36690501e-01]
[ -1.36346858e-01  -8.88465072e-03   6.67359597e-03  -3.03355742e-02
   1.08813024e-02  -1.24288502e-02   4.05287074e-03   3.15784422e-03
  -9.66704898e-03   1.78208987e-02   1.32502875e-02   2.49950056e-02
  -3.55005183e-03  -1.85599334e-02  -2.44850389e-04]
[ -1.34899263e+02  -1.42776773e+01  -6.25218865e+00  -5.67092897e+00
   1.28057331e+01  -7.04822307e+00   1.02186752e+00   2.65118060e+00
  -2.87055884e+00   4.19638229e+00   3.80926590e-01   7.94579672e+00
  -3.70026412e+00  -4.28526979e-01  -1.17038474e+00]
[ -5.16899802e+02   1.47465358e+01  -3.35352569e+01   1.01679977e+00
   2.37252798e+01  -1.60622655e+01   3.04550219e+00   4.76909005e+00
  -8.68540199e+00   1.21929326e+01  -4.04746339e+00   3.91565891e+00
   5.41418739e+00   4.17508534e+00  -2.43567704e+00]
[ -3.66874904e+02   3.84106545e+01   7.00517889e+00  -9.28527449e+00
  -2.69020976e+01  -1.08003408e+01   1.61285390e+01  -4.23494749e+00
  -1.30585957e+00   5.05161542e+00   4.66840616e+00  -7.74700329e+00
   2.01091265e+00   1.80077286e+00   8.69980595e-01]
[ -3.72756265e+02  -3.10778943e+01   3.57672164e+01  -1.26595816e+01
  -2.72005329e+01   8.59962415e+00   1.34747261e+01  -6.89101634e+00
   7.35529809e+00   6.07340458e+00   1.33955302e+00   8.45256794e+00
  -3.13871976e-01   2.85420654e+00  -2.35317665e-01]
[ -4.07457527e+02  -2.94463253e+01   9.64025389e-02   4.49177671e+01
  -4.25845097e+00   2.44830959e+01   3.63022549e+00   1.05364639e+00
  -8.37327974e+00   6.66370419e+00  -8.83415035e+00  -7.22697853e-01
   2.64124954e-01  -2.16998605e+00  -9.76352154e-01]
[ -9.01402193e+01   5.13303345e+00  -4.27869786e+00   2.11114796e+01
   2.46701217e+00   1.21157691e+00   6.91314449e-01   2.15241692e+00
  -1.12292780e+00   5.96956963e-01  -1.62067362e+00  -6.97006361e+00
  -2.55830874e+00  -1.42235808e+00  -1.35840735e+00]
[ -2.38821766e+00   7.58501015e-01   4.63437491e-02   8.91014320e-01
  -1.59351627e-01  -3.99972345e-01  -1.48050521e-01   7.35286849e-02
   2.36324127e-01   6.82692477e-02  -5.96353706e-02  -4.18285481e-01
  -2.08393600e-01  -2.27873274e-01  -6.81365895e-02]
[ -5.54465849e-02   1.76005585e-02   9.22708562e-03  -7.24049910e-03
   8.93513569e-03  -1.09662864e-02   2.65964515e-03  -1.01536638e-03
  -1.71856543e-03   1.38883981e-02   4.63760588e-03   1.07694029e-02
   1.89270142e-03  -8.49387549e-03   2.70507916e-04]
[ -1.26867827e+02   1.14790652e+01  -1.27389395e+01   6.58617895e+00
   1.02378217e+01  -8.01738344e+00   1.59506663e+00   2.08459747e+00
  -1.92291939e-01   3.39825473e+00  -7.62076847e-01   6.71803720e+00
  -2.05375242e+00  -5.42030635e-01   9.62188537e-01]
[ -4.71597943e+02   4.53305520e+01  -3.15941754e+01   1.16949823e+01
   1.29879568e+01  -2.25637178e+01   7.34281060e-01  -7.20787365e+00
  -4.69495541e+00   1.06471313e+01  -2.05271481e+00   5.18982768e+00
  -2.94472686e+00   1.44994543e+00  -3.86180044e-01]
[ -4.63164960e+02  -6.67644569e+00   2.13123462e+01  -8.74654842e+00
  -1.81932676e+01  -2.21411951e+01   1.79418603e+00  -2.38909296e+01
  -3.62585540e+00   5.82022874e+00   4.25030674e-01  -7.51270566e+00
   9.54857449e-01  -1.13873018e+00   1.46049491e+00]
[ -5.19363742e+02  -3.78089769e+01   4.14375391e+01   2.40765022e+00
  -2.06309853e+01  -2.34750987e+00  -6.60681205e+00  -7.48169130e+00
   3.95212620e+00   8.14860598e+00  -5.17358853e+00   3.01753430e+00
   4.57509814e+00   5.16524555e+00   3.73514574e+00]
[ -3.89978721e+02  -7.80888876e+00  -8.48327582e+00   5.10924585e+01
```

```
  -7.18270195e+00   1.14910201e+01  -4.26877562e+00   2.72267193e+00
   6.40791322e+00   1.38050354e+01  -3.03746482e+00  -6.33019266e+00
  -5.29016774e+00   8.16403672e-01  -1.63419985e+00]
[ -1.16353149e+02   9.19820003e+00  -1.05508196e+01   2.12024177e+01
   5.79846945e+00   6.76928800e+00   1.41509766e+00  -1.51212337e+00
   5.01156754e+00   4.55560014e+00   8.89093609e-01  -4.57137128e+00
  -5.50715461e+00  -3.32408702e+00  -2.70844738e-01]
[ -1.12343504e-01   3.82013445e-02  -6.36143261e-03   3.11112731e-02
  -5.78613246e-03  -1.21993609e-02   7.11564605e-04  -1.97171832e-03
   1.10163544e-02   1.21043819e-02  -1.43644195e-03  -1.98496171e-02
  -7.11321600e-03  -8.43112296e-03  -2.02550156e-03]
[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00  -4.36192176e-27   0.00000000e+00]
[ -1.19714939e+02   2.86129170e+01  -1.44803185e+01   8.03402971e+00
   5.55563342e+00   2.59331660e+00   2.02051620e+00   4.22936609e-03
   6.18362028e+00  -8.12924049e-01  -1.36248527e+00   3.46525025e+00
  -2.31807171e+00   1.19586585e+00   2.82277318e+00]
[ -3.97913887e+02   6.58273370e+01  -1.48755214e+01   5.79855498e+00
   1.08410957e+01   4.78386540e+00  -6.89777049e+00  -1.32797402e+01
   3.72889686e+00  -2.81506649e+00  -1.09134952e+00   1.64453531e+00
  -5.99182654e+00  -2.54500686e+00  -2.19488867e-02]
[ -4.75865776e+02   2.90793436e+01   4.31241711e+01  -1.90860734e+01
   1.98728796e+00  -6.28137116e-01  -1.86387885e+01  -1.36028733e+01
  -3.15540319e+00  -3.79675760e+00  -4.85702297e+00  -1.18497285e+00
  -2.19164215e+00  -6.95225558e+00  -5.61868299e+00]
[ -5.37292094e+02   1.49103830e+01   4.55625964e+01  -1.11880485e+00
  -1.24566294e+01  -5.06556905e+00  -2.00217114e+01   9.57697356e+00
   2.36624223e+00  -8.16279273e+00  -7.57636905e+00   3.18446345e+00
   3.09155254e+00   1.29154099e+00  -9.23595403e-01]
[ -4.50504659e+02   6.58913330e+00  -2.77225508e+01   3.78514974e+01
  -1.53000677e+01  -1.09551754e+00  -1.70068540e+01   3.10791115e+00
   1.74406612e+01  -3.96995506e-01   4.97803135e+00  -3.52431480e-01
   2.08268407e+00   4.75615420e+00  -4.52996952e-01]
[ -1.48672881e+02   3.86734126e+00  -2.10225655e+01   1.37574102e+01
   1.33943985e+00   7.34047674e+00  -3.37253481e+00  -1.58940188e+00
   7.63753581e+00   4.27417857e+00   4.94448153e+00   2.65065809e-01
  -6.23745129e-01  -9.64043462e-01   1.75482796e+00]
[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   5.78023733e-35]
[ -4.48335032e-01   2.30571466e-01  -4.70474652e-02   4.62425430e-02
  -4.99844180e-02   2.79131778e-02   7.21061748e-02   1.68845057e-02
   6.67815079e-02  -4.05548732e-02   9.32554722e-03  -2.20025074e-03
  -4.06384633e-02   5.62372598e-02   4.63745570e-02]
[ -8.11245742e+01   1.91238973e+01  -8.38564766e+00   3.64829281e+00
   8.29918389e-01   6.78306971e+00   4.17603932e+00  -7.13354130e-01
   3.98766134e+00  -3.46371426e+00  -7.44807617e-01   1.27155303e+00
  -1.00619395e+00   4.26681438e+00   3.15558086e+00]
[ -3.61006045e+02   5.40628705e+01  -2.22395153e+01  -1.59636455e+01
   6.39836741e+00   2.70834649e+01   3.78851818e+00  -1.04947488e+01
  -2.42297187e+00  -7.66956491e+00  -3.51590134e+00   1.74373578e-02
   1.49248935e+00   4.81847121e+00   2.48107274e+00]
[ -3.77451729e+02   4.40596212e+01   4.31017502e+01  -2.37766273e+01
   6.48353970e+00   1.58274908e+01  -3.89936726e+00   3.36544128e+00
  -3.50231384e-01   3.06317975e+00   3.49146291e+00   2.66956846e+00
  -1.82309672e+00   3.70157948e+00  -3.20067114e+00]
[ -3.98767917e+02   3.72531962e+01   4.87382369e+01   1.26652770e+01
```

```
  -1.88944560e+00  -7.41081264e+00   7.26688740e+00   1.77405469e+01
  -3.13253533e-01  -4.02689577e+00  -6.22260832e+00  -4.78562024e-01
  -3.47729538e+00   1.78904476e+00   2.68849309e+00]
[ -4.30475096e+02   2.06954357e+00  -4.01832177e+01   2.22870723e+01
  -1.81611472e+01  -8.97803742e+00   1.45870206e+00   1.14606471e-01
   6.31746671e+00  -1.43348631e+01  -2.16826321e+00   1.67778010e+00
   4.64085405e+00   2.71810764e+00  -4.57708449e+00]
[ -1.77926670e+02  -6.61820599e+00  -3.60186319e+01  -9.19966767e+00
  -8.26661738e+00   3.01143747e+00  -1.20872163e+01  -1.76753117e+00
   2.83478941e+00   1.74391367e+00   6.26929507e+00   6.73879908e-01
   2.09821110e+00  -6.80682472e-01   1.53180175e+00]
[ -1.47500254e+00   2.88023598e-01  -2.24551191e-01  -4.06840096e-01
  -2.31779371e-01   4.39362486e-02  -3.71810874e-01  -5.64023689e-03
  -1.67466536e-02   1.77514430e-01   1.64055154e-01  -1.71176512e-01
   1.98450227e-02  -3.90303382e-02   9.62060128e-02]
[ -3.68166637e-01   1.23955191e-01  -4.72133058e-02   1.95917888e-02
  -4.00039940e-02   5.07940086e-02   9.57038004e-02  -6.27435128e-03
   1.62971477e-02  -3.28022921e-02   4.85759338e-02  -3.15853811e-02
  -1.67884284e-02   3.73013005e-02   2.38505939e-02]
[ -3.66332193e+01  -1.51023224e+00  -1.83365935e+00  -2.51643689e+00
   1.18385031e+00   1.65914646e+00   3.17995687e+00  -2.02335964e+00
   6.81768359e-01  -1.84564713e+00  -4.11874676e-01  -1.22766936e+00
  -6.49342386e-01   2.25297808e+00   7.75351314e-01]
[ -3.93491616e+02  -1.01738113e+01  -2.63462760e+01  -3.18496292e+01
   1.20109829e+01   1.72912536e+01   1.00845344e+01  -9.07748741e+00
  -4.98636491e+00  -4.35881200e+00  -3.07882669e+00  -4.87544975e+00
   6.95976627e-01   6.38973808e+00   6.40218106e-01]
[ -4.98425256e+02   1.63610056e+01   1.83252425e+01  -2.29512603e+01
   1.43552477e+01   1.31715947e+01   4.14956400e-02   1.02925038e+01
  -3.08290291e+00   4.75854347e+00   9.46786501e+00  -1.78019621e+00
   4.65659801e+00   4.59762202e-01  -3.06316185e+00]
[ -4.92703033e+02   1.89505163e+01   1.48743466e+01   8.99057851e+00
  -1.81886354e+00  -9.47709733e+00   2.28490194e+01   1.41697476e+01
   5.23433125e+00  -5.84439464e+00  -1.14205676e+00  -1.93085263e+00
  -2.07697889e+00  -2.15956130e+00   6.90594730e-01]
[ -4.58940414e+02  -2.48009217e+01  -4.80952090e+01  -9.99419671e+00
  -2.13465069e+01  -3.91833033e+00   6.67246974e+00   1.12407377e+00
  -1.15353886e+00  -1.05974707e+01  -2.51729824e+00   5.08560800e-01
  -2.39377917e-01  -2.65045160e+00  -7.26419573e+00]
[ -1.91937185e+02  -1.13768716e+01  -2.49684328e+01  -3.09423648e+01
  -1.41323270e+01   1.62644922e+00  -1.50349671e+01   8.25902657e+00
  -2.96574754e+00   3.09470242e+00   3.52348597e+00  -1.47341955e+00
  -2.27878377e+00   1.10515552e+00   6.78624802e-02]
[ -1.04803211e+01   1.66781533e-01  -1.25223305e-01  -3.08851567e+00
  -1.66003547e+00  -4.43854551e-01  -5.90399362e-01   2.10743539e+00
  -2.88885000e-01   6.63830883e-01   1.00597470e-01  -9.68937102e-01
  -4.61101470e-01  -1.31329007e-01   2.56349208e-01]
[ -2.83736251e-02   1.68790337e-03   5.42680815e-03   2.01529220e-03
  -1.11835205e-03   6.40089075e-03   6.35481986e-03   3.11627915e-03
  -8.50033289e-03  -1.09078108e-03   2.16400928e-03   7.92610548e-04
   1.25313866e-03   3.81200303e-03  -4.74571061e-04]
[ -1.41711732e+01  -2.52741087e+00   1.61156044e+00  -2.49934736e+00
   2.11110533e+00  -8.77509691e-01   7.58110761e-01  -3.69862520e-01
   7.12161581e-01  -8.27109243e-01   3.20098609e-01  -4.13721855e-01
  -1.18140617e+00   6.88517771e-01  -3.97887099e-01]
[ -2.87549608e+02  -4.23117997e+01   1.07724012e+01  -1.69547789e+01
   2.09735490e+01  -5.23320532e+00   2.62655605e+00  -4.45053169e+00
   9.94528432e+00  -4.57755774e+00   4.68644139e-03  -3.25660110e+00
  -6.66687291e+00   3.66919932e+00  -1.32602395e+00]
[ -6.27465508e+02  -2.55335229e+01  -4.17835083e+00  -1.24097687e+01
```

```
    2.23235329e+01  -2.10201301e+00  -2.00449618e+00   4.93395378e+00
    4.87949206e+00  -8.93726861e-01  -1.33253817e+00  -4.59217988e+00
    5.13269705e+00  -4.91550732e+00   4.72050194e+00]
 [ -6.13736352e+02  -1.87193142e+00  -3.32877054e+01  -1.76014901e+01
   -2.13581003e+01  -4.04491320e+00   3.87599692e+00   3.61518164e+00
   -3.85142842e+00  -3.16779513e+00  -3.40748098e+00   1.33697984e-01
   -1.52714965e+00  -4.79275707e+00   6.02697545e+00]
 [ -3.50321448e+02  -1.61241112e+01  -2.99508348e+01  -3.16613330e+01
   -3.06887323e+01   2.64743537e+00  -9.81265629e+00   7.30493928e+00
   -1.26616198e+01   1.66051553e+00  -2.86635610e-01  -3.09233819e-01
   -7.10602045e+00   1.66633759e+00   3.80756740e-01]
 [ -1.05013006e+02  -6.59091287e+00  -3.51400667e+00  -2.35229835e+01
   -1.03034598e+01   1.48972859e+00  -5.61328966e+00   1.26999693e+01
   -5.71622370e+00   4.44277693e+00   2.46563857e-01  -2.62873161e+00
   -6.37916049e+00   2.38432870e+00   1.68313933e+00]
 [ -1.80936560e+01  -2.06005919e+00   1.04699897e+00  -4.94768446e+00
   -2.57470633e+00  -1.45745676e+00   4.60313690e-01   5.55779357e+00
   -3.41042086e-01   1.23576377e+00  -8.19158094e-01  -1.64001673e+00
   -1.39595303e+00  -1.06021920e-01   7.44192358e-01]]
[[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
    0.00000000e+00   0.00000000e+00   0.00000000e+00]
 [ -5.77192878e-03  -1.73619371e-02   9.85740003e-03  -1.84048382e-02
    1.99513790e-02  -1.42232213e-02   1.09749979e-02  -8.50458120e-03
    1.89535667e-02  -2.00303310e-02   1.39722270e-02  -8.73734257e-03
   -5.71260055e-02   2.19374814e-02  -1.57656455e-02]
 [ -1.00696020e-01  -2.24200800e-01   4.48071690e-02  -1.23100765e-01
    1.82042764e-01  -8.72415154e-02   6.17819566e-02  -5.49919656e-02
    2.12755902e-01  -8.25529225e-02  -4.45382518e-02  -1.21328951e-01
   -2.63822844e-01   1.33292578e-01  -6.65469554e-02]
 [ -2.29641867e-01  -1.37464667e-01   8.71833404e-04  -1.22041169e-01
    2.08220187e-01   5.84089331e-02  -7.17565845e-02   3.63120681e-02
    1.64351626e-01   2.07519908e-02   3.20636404e-03  -1.10118118e-01
    1.24642536e-01  -1.79412849e-01   1.43104002e-01]
 [ -2.29629079e-01  -3.44686443e-02   4.71422113e-02   1.44693467e-01
    4.63059807e-02   7.41524150e-03  -6.06356534e-02  -5.72724148e-02
   -9.22560987e-02  -1.89172931e-01   2.94432272e-01   1.82407810e-01
   -8.39617486e-02  -1.75555989e-01   3.81126209e-01]
 [ -1.11132408e-01  -9.72341798e-02   1.16667579e-01   2.68551096e-01
    1.02219668e-01  -1.70751857e-01  -8.92874500e-02  -6.90141172e-02
   -3.19884043e-01  -2.58880617e-01   3.25228506e-01   1.43564896e-02
   -1.11927932e-01   2.33971700e-01   8.07584715e-02]
 [ -2.53850228e-02  -8.38661323e-03   6.29499204e-02   1.16106606e-01
    7.21682736e-02  -9.40017765e-02  -5.63045722e-02   1.72780160e-02
   -9.87873958e-02  -5.37227587e-02   1.66759633e-01  -1.74757303e-01
   -9.87584737e-03   1.87397012e-01  -1.05936543e-01]
 [ -2.27321903e-03   2.28152694e-03   8.16433952e-03   1.66018946e-02
    8.96277131e-03  -3.88925929e-03  -1.01703385e-02   1.22833352e-02
   -6.72002558e-03   2.68640192e-03   2.58564750e-02  -4.85613004e-02
    3.37151276e-03   1.62600863e-02  -2.06188068e-02]
 [ -1.15131258e-04  -3.22519902e-04   1.52191610e-04  -3.84251131e-04
    7.40455696e-05  -1.13674504e-04   1.06739973e-04  -2.50175220e-04
   -5.07846799e-04  -2.68204150e-04  -2.39878440e-04   1.53883443e-03
   -1.24616736e-03  -7.90157560e-04   6.37575828e-04]
 [ -3.85780810e-02  -1.19627690e-01   1.91174589e-02  -7.82952433e-02
    7.61589235e-02  -5.34764761e-02   6.49548191e-03  -1.73436737e-03
    5.39731651e-02  -1.95110129e-02   3.07472748e-02   1.26869645e-01
   -2.09804541e-01   1.81575956e-02  -2.74672837e-02]
 [ -2.01744314e-01  -2.45698693e-01  -6.81140823e-02  -7.74965809e-02
```

```
    3.21106793e-01  -5.43228119e-02  -2.82212612e-02   2.15258005e-02
    2.75392498e-02   5.06677342e-02  -9.39465003e-02   8.17203464e-02
    6.56954101e-02   2.85215504e-02  -1.05389361e-01]
[ -2.32849975e-01   1.46934320e-01  -2.70157334e-03  -5.44499073e-02
    5.05491542e-02  -4.12433527e-03   5.61756167e-02   1.45931160e-01
   -2.23609176e-02   7.60848951e-02   1.72996200e-01  -1.36535646e-01
    2.17040157e-01  -1.20432471e-01  -8.71633252e-02]
[ -2.00821003e-01  -4.83676918e-02   8.42083867e-02   1.18547672e-02
   -1.39528864e-01   1.22238434e-01   2.53391236e-01  -3.38297792e-02
    7.80888981e-02   5.11470207e-02   3.14473476e-01   2.31970289e-01
   -8.96535650e-02  -1.91525321e-01  -2.77410042e-01]
[ -1.59441471e-01  -2.18765265e-01   4.75889085e-02   2.89668917e-01
    4.26257109e-02   2.04206981e-01   5.63005385e-02  -7.28108993e-02
   -3.52718114e-01  -1.09238690e-01   1.02079279e-02   9.67337127e-02
    5.28748792e-02  -5.64784412e-02   8.18306225e-04]
[ -3.48630321e-02  -1.49123539e-02   5.96219458e-02   1.60049690e-01
    4.70818569e-02  -3.52166341e-02  -4.71788480e-02   3.59908973e-02
   -1.44358487e-01  -7.44498233e-02   5.65368430e-02  -2.58528405e-01
    1.23071533e-03   1.18615098e-01  -1.11525361e-01]
[ -1.87173744e-03   4.49594228e-03   3.53751778e-03   1.49968156e-02
    2.18876208e-03  -8.99879408e-03  -4.71968806e-03   9.80451156e-03
    3.78354859e-05   2.09047633e-03   1.45678011e-02  -4.08649267e-02
   -5.93209331e-03  -2.03615185e-04  -1.23224105e-02]
[ -5.09411428e-05  -4.96623497e-05   4.08226797e-05  -2.14475569e-04
    1.07954332e-04  -1.79016380e-04   7.09563595e-05   6.21886184e-05
   -2.22280499e-04   4.44146446e-04   4.55416791e-04   8.93853846e-04
   -1.48028089e-04  -8.59288383e-04  -1.27472247e-05]
[ -5.04003004e-02  -7.98076397e-02  -3.82449126e-02  -4.00940398e-02
    1.27046774e-01  -1.01517627e-01   1.78905284e-02   5.22107004e-02
   -6.60045533e-02   1.04585538e-01   1.30925737e-02   2.84152005e-01
   -1.54291557e-01  -1.98399557e-02  -6.09317275e-02]
[ -1.93121184e-01   8.24284086e-02  -2.05136640e-01   7.18887695e-03
    2.35380531e-01  -2.31349528e-01   5.33196744e-02   9.39194907e-02
   -1.99708876e-01   3.03881851e-01  -1.39112664e-01   1.40029045e-01
    2.25757778e-01   1.93298233e-01  -1.26804464e-01]
[ -1.37069729e-01   2.14703247e-01   4.28509870e-02  -6.56478272e-02
   -2.66898013e-01  -1.55560481e-01   2.82373281e-01  -8.34004196e-02
   -3.00264452e-02   1.25900331e-01   1.60454674e-01  -2.77042894e-01
    8.38499186e-02   8.33722387e-02   4.52923033e-02]
[ -1.39267090e-01  -1.73715467e-01   2.18789634e-01  -8.95045185e-02
   -2.69858815e-01   1.23862912e-01   2.35911177e-01  -1.35707386e-01
    1.69124966e-01   1.51366163e-01   4.60408832e-02   3.02274802e-01
   -1.30876593e-02   1.32144145e-01  -1.22509388e-02]
[ -1.52231979e-01  -1.64595520e-01   5.89698565e-04   3.17573144e-01
   -4.22484565e-02   3.52637220e-01   6.35568223e-02   2.07498561e-02
   -1.92532054e-01   1.66078074e-01  -3.03632689e-01  -2.58446133e-02
    1.10133356e-02  -1.00466083e-01  -5.08301428e-02]
[ -3.36776795e-02   2.86920117e-02  -2.61729828e-02   1.49260290e-01
    2.44754389e-02   1.74506980e-02   1.21033114e-02   4.23883590e-02
   -2.58201808e-02   1.48778307e-02  -5.57030920e-02  -2.49258522e-01
   -1.06674936e-01  -6.58523792e-02  -7.07204251e-02]
[ -8.92272390e-04   4.23977755e-03   2.83486749e-04   6.29956111e-03
   -1.58094113e-03  -5.76091914e-03  -2.59202099e-03   1.44802815e-03
    5.43394834e-03   1.70145986e-03  -2.04968754e-03  -1.49584317e-02
   -8.68948053e-03  -1.05500840e-02  -3.54727806e-03]
[ -2.07156398e-05   9.83814804e-05   5.64424881e-05  -5.11910589e-05
    8.86462458e-05  -1.57950643e-04   4.65642133e-05  -1.99959934e-05
   -3.95160489e-05   3.46137576e-04   1.59396058e-04   3.85127827e-04
    7.89208121e-05  -3.93249716e-04   1.40829883e-05]
[ -4.73996408e-02   6.41642953e-02  -7.79246525e-02   4.65649495e-02
```

```
   1.01570305e-01  -1.15476728e-01   2.79259144e-02   4.10527650e-02
  -4.42148872e-03   8.46939758e-02  -2.61928350e-02   2.40245731e-01
  -8.56362273e-02  -2.50949515e-02   5.00927668e-02]
[ -1.76195759e-01   2.53383256e-01  -1.93262959e-01   8.26847045e-02
   1.28854631e-01  -3.24991854e-01   1.28555570e-02  -1.41947377e-01
  -1.07954044e-01   2.65356176e-01  -7.05524914e-02   1.85594975e-01
  -1.22787585e-01   6.71296192e-02  -2.01050274e-02]
[ -1.73045076e-01  -3.73191914e-02   1.30368558e-01  -6.18389795e-02
  -1.80496965e-01  -3.18906135e-01   3.14120329e-02  -4.70493095e-01
  -8.33715592e-02   1.45056316e-01   1.46084458e-02  -2.68664106e-01
   3.98151154e-02  -5.27209658e-02   7.60352341e-02]
[ -1.94041747e-01  -2.11340062e-01   2.53475247e-01   1.70223299e-02
  -2.04681771e-01  -3.38118740e-02  -1.15669944e-01  -1.47339771e-01
   9.08737076e-02   2.03085964e-01  -1.77817960e-01   1.07910943e-01
   1.90769900e-01   2.39140702e-01   1.94456468e-01]
[ -1.45701647e-01  -4.36491852e-02  -5.18925708e-02   3.61228835e-01
  -7.12602007e-02   1.65508537e-01  -7.47363529e-02   5.36186060e-02
   1.47341153e-01   3.44059945e-01  -1.04398677e-01  -2.26375907e-01
  -2.20586475e-01   3.77978830e-02  -8.50785357e-02]
[ -4.34712062e-02   5.14149899e-02  -6.45398268e-02   1.49903232e-01
   5.75271116e-02   9.75000430e-02   2.47750755e-02  -2.97788164e-02
   1.15234104e-01   1.13538248e-01   3.05584434e-02  -1.63478171e-01
  -2.29634273e-01  -1.53898686e-01  -1.41005237e-02]
[ -4.19731452e-05   2.13533271e-04  -3.89131627e-05   2.19959839e-04
  -5.74047153e-05  -1.75710978e-04   1.24578446e-05  -3.88297932e-05
   2.53305921e-04   3.01674922e-04  -4.93709881e-05  -7.09848074e-04
  -2.96602929e-04  -3.90344398e-04  -1.05450204e-04]
[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00  -2.01948392e-28   0.00000000e+00]
[ -4.47272192e-02   1.59937035e-01  -8.85767445e-02   5.68014004e-02
   5.51179145e-02   3.73523006e-02   3.53745488e-02   8.32905030e-05
   1.42183846e-01  -2.02603322e-02  -4.68290723e-02   1.23921847e-01
  -9.66576661e-02   5.53662348e-02   1.46957185e-01]
[ -1.48666338e-01   3.67953714e-01  -9.09942176e-02   4.09963686e-02
   1.07555438e-01   6.89034183e-02  -1.20763951e-01  -2.61522936e-01
   8.57408559e-02  -7.01593000e-02  -3.75100464e-02   5.88107174e-02
  -2.49843854e-01  -1.17828808e-01  -1.14268713e-03]
[ -1.77790283e-01   1.62544210e-01   2.63792449e-01  -1.34940464e-01
   1.97160539e-02  -9.04724334e-03  -3.26321923e-01  -2.67886519e-01
  -7.25541575e-02  -9.46257775e-02  -1.66937497e-01  -4.23761673e-02
  -9.13858767e-02  -3.21875749e-01  -2.92515828e-01]
[ -2.00740037e-01   8.33442616e-02   2.78708404e-01  -7.91006312e-03
  -1.23583286e-01  -7.29608787e-02  -3.50533694e-01   1.88602955e-01
   5.44084865e-02  -2.03439537e-01  -2.60402327e-01   1.13880547e-01
   1.28909840e-01   5.97958056e-02  -4.80835589e-02]
[ -1.68315005e-01   3.68311431e-02  -1.69580061e-01   2.67613904e-01
  -1.51793280e-01  -1.57790609e-02  -2.97750539e-01   6.12052675e-02
   4.01024022e-01  -9.89423409e-03   1.71096595e-01  -1.26034072e-02
   8.68426031e-02   2.20200578e-01  -2.35836011e-02]
[ -5.55463219e-02   2.16171980e-02  -1.28595956e-01   9.72662777e-02
   1.32886974e-02   1.05727042e-01  -5.90452565e-02  -3.13006911e-02
   1.75614634e-01   1.06524438e-01   1.69943476e-01   9.47909741e-03
  -2.60085778e-02  -4.46333147e-02   9.13585895e-02]
[  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00
   0.00000000e+00   0.00000000e+00   3.00926554e-36]
[ -1.67504402e-04   1.28882059e-03  -2.87791411e-04   3.26939443e-04
  -4.95899688e-04   4.02041697e-04   1.26241175e-03   3.32512943e-04
```

```
  -4.95899688e-04   4.02041697e-04   1.26241175e-03   3.32512943e-04
   1.53554895e-03  -1.01074043e-03   3.20522163e-04  -7.86838225e-05
  -1.69452006e-03   2.60367444e-03   2.41431880e-03]
[ -3.03093051e-02   1.06896456e-01  -5.12953751e-02   2.57937981e-02
   8.23369135e-03   9.76985451e-02   7.31127555e-02  -1.40483522e-02
   9.16907894e-02  -8.63254097e-02  -2.55992858e-02   4.54723870e-02
  -4.19557161e-02   1.97545107e-01   1.64283579e-01]
[ -1.34877039e-01   3.02194117e-01  -1.36040092e-01  -1.12864584e-01
   6.34787506e-02   3.90091098e-01   6.63281597e-02  -2.06677049e-01
  -5.57129064e-02  -1.91146926e-01  -1.20842700e-01   6.23582549e-04
   6.22329917e-02   2.23085732e-01   1.29167822e-01]
[ -1.41021382e-01   2.46279160e-01   2.63655299e-01  -1.68103153e-01
   6.43237522e-02   2.27967998e-01  -6.82688700e-02   6.62769056e-02
  -8.05308910e-03   7.63429738e-02   1.20002743e-01   9.54672334e-02
  -7.60184742e-02   1.71375844e-01  -1.66631036e-01]
[ -1.48985416e-01   2.08233426e-01   2.98133936e-01   8.95447861e-02
  -1.87453515e-02  -1.06740111e-01   1.27226331e-01   3.49371287e-01
  -7.20283425e-03  -1.00361462e-01  -2.13873120e-01  -1.71139992e-02
  -1.44994331e-01   8.28292508e-02   1.39966391e-01]
[ -1.60831673e-01   1.15680852e-02  -2.45802507e-01   1.57571849e-01
  -1.80178295e-01  -1.29313310e-01   2.55384873e-02   2.25698849e-03
   1.45261460e-01  -3.57264726e-01  -7.45239284e-02   5.99995943e-02
   1.93511754e-01   1.25843033e-01  -2.38288877e-01]
[ -6.64759571e-02  -3.69936501e-02  -2.20327553e-01  -6.50425783e-02
  -8.20138184e-02   4.33746183e-02  -2.11619102e-01  -3.48086586e-02
   6.51820844e-02   4.34631873e-02   2.15477759e-01   2.40988202e-02
   8.74900405e-02  -3.15142587e-02   7.97475596e-02]
[ -5.51082114e-04   1.60995959e-03  -1.37358950e-03  -2.87640051e-03
  -2.29950297e-03   6.32826691e-04  -6.50954540e-03  -1.11075314e-04
  -3.85066271e-04   4.42415416e-03   5.63863025e-03  -6.12149427e-03
   8.27486729e-04  -1.80702783e-03   5.00860817e-03]
[ -1.37552339e-04   6.92869783e-04  -2.88805865e-04   1.38515923e-04
  -3.96883048e-04   7.31601023e-04   1.67555141e-03  -1.23563168e-04
   3.74730502e-04  -8.17524510e-04   1.66957102e-03  -1.12953423e-03
  -7.00034560e-04   1.72697679e-03   1.24169245e-03]
[ -1.36866964e-02  -8.44171416e-03  -1.12165748e-02  -1.77914626e-02
   1.17450802e-02   2.38971737e-02   5.56736638e-02  -3.98467852e-02
   1.56763259e-02  -4.59986688e-02  -1.41562697e-02  -4.39030499e-02
  -2.70759181e-02   1.04308451e-01   4.03657818e-02]
[ -1.47014114e-01  -5.68683443e-02  -1.61161328e-01  -2.25180091e-01
   1.19161989e-01   2.49051004e-01   1.76556789e-01  -1.78766386e-01
  -1.14654605e-01  -1.08633739e-01  -1.05820298e-01  -1.74352412e-01
   2.90204467e-02   2.95832296e-01   3.33305741e-02]
[ -1.86218828e-01   9.14527770e-02   1.12096313e-01  -1.62267725e-01
   1.42419640e-01   1.89714346e-01   7.26492342e-04   2.02694163e-01
  -7.08871133e-02   1.18596161e-01   3.25413673e-01  -6.36621273e-02
   1.94168237e-01   2.12860850e-02  -1.59472127e-01]
[ -1.84080924e-01   1.05927312e-01   9.09870317e-02   6.35642969e-02
  -1.80451008e-02  -1.36501416e-01   4.00033295e-01   2.79050189e-01
   1.20356250e-01  -1.45658598e-01  -3.92528710e-02  -6.90497965e-02
  -8.66047120e-02  -9.99834374e-02   3.59532456e-02]
[ -1.71466725e-01  -1.38629203e-01  -2.94200506e-01  -7.06599788e-02
  -2.11780521e-01  -5.64368625e-02   1.16819458e-01   2.21368090e-02
  -2.65240398e-02  -2.64118496e-01  -8.65203785e-02   1.81867944e-02
  -9.98144737e-03  -1.22710692e-01  -3.78183327e-01]
[ -7.17104866e-02  -6.35930658e-02  -1.52733000e-01  -2.18765640e-01
  -1.40208026e-01   2.34262258e-02  -2.63227376e-01   1.62648128e-01
  -6.81932867e-02   7.71286064e-02   1.21103386e-01  -5.26913957e-02
  -9.50194594e-02   5.11665255e-02   3.53300758e-03]
[ -3.91559838e-03   9.32255312e-04  -7.65996455e-04  -2.18361173e-02
  -1.64693540e-02  -6.39296744e-03  -1.03365224e-02   4.15025206e-02
```

```
  -6.64251333e-03   1.65445151e-02   3.45756854e-03  -3.46504485e-02
  -1.92267529e-02  -6.08027450e-03   1.33458679e-02]
[ -1.06007935e-05   9.43483882e-06   3.31960238e-05   1.42483191e-05
  -1.10952664e-05   9.21939093e-05   1.11258145e-04   6.13700617e-05
  -1.95453466e-04  -2.71853035e-05   7.43777195e-05   2.83447821e-05
   5.22526796e-05   1.76488237e-04  -2.47067769e-05]
[ -5.29455366e-03  -1.41274167e-02   9.85798597e-03  -1.76706379e-02
   2.09444566e-02  -1.26390298e-02   1.32727598e-02  -7.28384222e-03
   1.63751763e-02  -2.06138667e-02   1.10018958e-02  -1.47952306e-02
  -4.92616181e-02   3.18770176e-02  -2.07145117e-02]
[ -1.07432660e-01  -2.36509398e-01   6.58952515e-02  -1.19871997e-01
   2.08080374e-01  -7.53753930e-02   4.59848999e-02  -8.76460005e-02
   2.28678138e-01  -1.14085493e-01   1.61074551e-04  -1.16460283e-01
  -2.77991563e-01   1.69876706e-01  -6.90345043e-02]
[ -2.34430118e-01  -1.42724209e-01  -2.55591556e-02  -8.77383165e-02
   2.21473680e-01  -3.02759106e-02  -3.50940754e-02   9.71662143e-02
   1.12197210e-01  -2.22741634e-02  -4.57997806e-02  -1.64222313e-01
   2.14020350e-01  -2.27578313e-01   2.45755373e-01]
[ -2.29300708e-01  -1.04634966e-02  -2.03622356e-01  -1.24444310e-01
  -2.11895541e-01  -5.82600728e-02   6.78597096e-02   7.11951367e-02
  -8.85583007e-02  -7.89502804e-02  -1.17116256e-01   4.78121347e-03
  -6.36782378e-02  -2.21895219e-01   3.13772056e-01]
[ -1.30885120e-01  -9.01286131e-02  -1.83210572e-01  -2.23848818e-01
  -3.04465539e-01   3.81317892e-02  -1.71796835e-01   1.43858927e-01
  -2.91136538e-01   4.13846735e-02  -9.85176134e-03  -1.10586028e-02
  -2.96302893e-01   7.71481507e-02   1.98226832e-02]
[ -3.92343661e-02  -3.68410903e-02  -2.14953332e-02  -1.66309866e-01
  -1.02221506e-01   2.14569983e-02  -9.82756728e-02   2.50105290e-01
  -1.31436704e-01   1.10726379e-01   8.47448184e-03  -9.40068545e-02
  -2.65994690e-01   1.10389726e-01   8.76263880e-02]
[ -6.76004954e-03  -1.15150706e-02   6.40453868e-03  -3.49806282e-02
  -2.55438818e-02  -2.09921777e-02   8.05902428e-03   1.09451727e-01
  -7.84179381e-03   3.07986761e-02  -2.81547364e-02  -5.86491272e-02
  -5.82076739e-02  -4.90860622e-03   3.87436067e-02]]
```

Практическое задание 2 (40%)

Изучение алгоритмов кластеризации на разных выборках

Кластеризация цифр с помощью dbscan

На данных из sklearn.datasets.load_digits примените алгоритмы кластеризации (знания о метках классов при кластеризации использовать нельзя):

- dbscan запускайте при различных параметрах eps и minsamples, для всех экспериментов можете выбрать одну метрику (вспомните семинар про метрические алгоритмы);
- Используя метки классов цифр, оцените качество различных кластеризаций при помощи Adjusted Mutual Information и Adjusted Rand Index.
- визуалируйте изображения тех цифр, которые соответствуют core_points;
- визуалируйте изображения тех цифр, которые соответствуют выбросам;
- сделайте выводы и применимости алгоритмов.

Уменьшение палитры изображения

- для картинки нужно уменьшить число цветов в палитре; для этого нужно выделить кластеры в пространстве RGB, объекты соответствуют пикселам изображения; после выделения кластеров, все пикселы, отнесенные в один кластер, заполняются одним цветом; этот цвет может быть центроидом соответствующего кластера, медианным цветом по кластеру.
- Попробуйте различные алгоритмы кластеризации:

    -- KMeans
    -- MeanShift
    -- AgglomerativeClustering

  Рассмотрите число кластеров K = 2, 3, 10, 20
- Для различных кластеризаций оцените и сравните потери от уменьшения цветов при помощи метрики SSIM. Какой способ оказался лучшим?

In [5]:

```
mnist = sk.datasets.load_digits()
```

In [8]:

```
from sklearn import cluster
```

In [7]:

```
digits = np.reshape(mnist.images, (mnist.images.shape[0], -1))
```

In [8]:

```
dbscan = cluster.DBSCAN(eps=21, min_samples=3, metric='euclidean',
algorithm='auto', leaf_size=30, p=None, n_jobs=1)
```

In [9]:

```
labels = dbscan.fit_predict(digits)
```

In [10]:

```
 sk.metrics.adjusted_mutual_info_score(labels, mnist.target)
```

Out[10]:

```
0.76524515153558503
```

In [11]:

```
sk.metrics.adjusted_rand_score(labels, mnist.target)
```

Out[11]:

```
0.59463144328200968
```

Попробуем найти наилучшие параметры

In [34]:

```
b_eps_i = 0
b_min_samples_i = 0
b_eps_r = 0
b_min_samples_r = 0
rand = 0
info = 0
for eps in np.linspace(3.5, 30., 100):
    for min_samples in np.arange(3, 20):
        dbscan = cluster.DBSCAN(eps=eps, min_samples=min_samples,
                                metric='euclidean', algorithm='auto',
leaf_size=30, p=None, n_jobs=4)
        labels = dbscan.fit_predict(digits)
        i = sk.metrics.adjusted_mutual_info_score(labels, mnist.target)
        r = sk.metrics.adjusted_rand_score(labels, mnist.target)
        if i > info:
            b_eps_i = eps
            b_min_samples_i = min_samples
            info = i
        if r > rand:
            b_eps_r = eps
            b_min_samples_r = min_samples
            rand = r
```

In [35]:

```
print(b_eps_i)
print(b_min_samples_i)
print(b_eps_r)
print(b_min_samples_r)
print(rand)
print(info)
```

```
21.1666666667
4
21.1666666667
4
0.673685734897
0.779628108702
```

In [36]:

```
dbscan = cluster.DBSCAN(eps=21.1666666667, min_samples=4,
                                metric='euclidean', algorithm='auto',
leaf_size=30, p=None, n_jobs=1)
labels = dbscan.fit_predict(digits)
dbscan.components_.shape
```

In [37]:

```
labels = dbscan.fit_predict(digits)
```

In [38]:

```
labels[:100]
```

Out[38]:

```
array([ 0,  1, -1,  2,  4, -1,  3,  6, -1, -1,  0,  1,  7,  2,  4,  5,  3,
        6,  1, -1,  0,  1,  7,  2,  4,  5,  3, -1, -1,  8,  0, -1,  5, -1,
        3, -1,  0, -1, -1,  8,  1,  4,  1,  6,  6,  2, -1,  1,  0,  0,  9,
        9,  6, -1, -1,  0,  1, -1,  3,  2,  2,  6,  2,  2,  4,  3,  3,  3,
```

```
        4, -1,  1,  5,  0,  8,  5, -1,  1, -1,  0,  0,  1,  6,  3,  2,  7,
        1, -1, -1,  3,  2,  1,  2,  8,  1,  6,  1, -1,  4,  2,  1])
```

In [39]:

```
dbscan.core_sample_indices_[:100]
```

Out[39]:

```
array([  0,   1,   3,   6,  10,  11,  13,  14,  15,  16,  17,  20,  21,
        22,  23,  24,  25,  26,  30,  32,  34,  36,  39,  40,  41,  42,
        44,  45,  47,  48,  49,  52,  55,  56,  58,  59,  60,  61,  62,
        63,  64,  65,  66,  67,  70,  71,  72,  73,  76,  78,  79,  80,
        81,  82,  83,  84,  85,  88,  89,  90,  91,  94,  97,  98,  99,
       100, 101, 102, 107, 108, 109, 111, 112, 114, 115, 117, 119, 124,
       126, 128, 130, 131, 132, 135, 136, 137, 139, 140, 141, 142, 143,
       144, 145, 146, 147, 149, 150, 151, 153, 154])
```

In [40]:

```
dbscan.components_.shape
```

Out[40]:

```
(1303, 64)
```

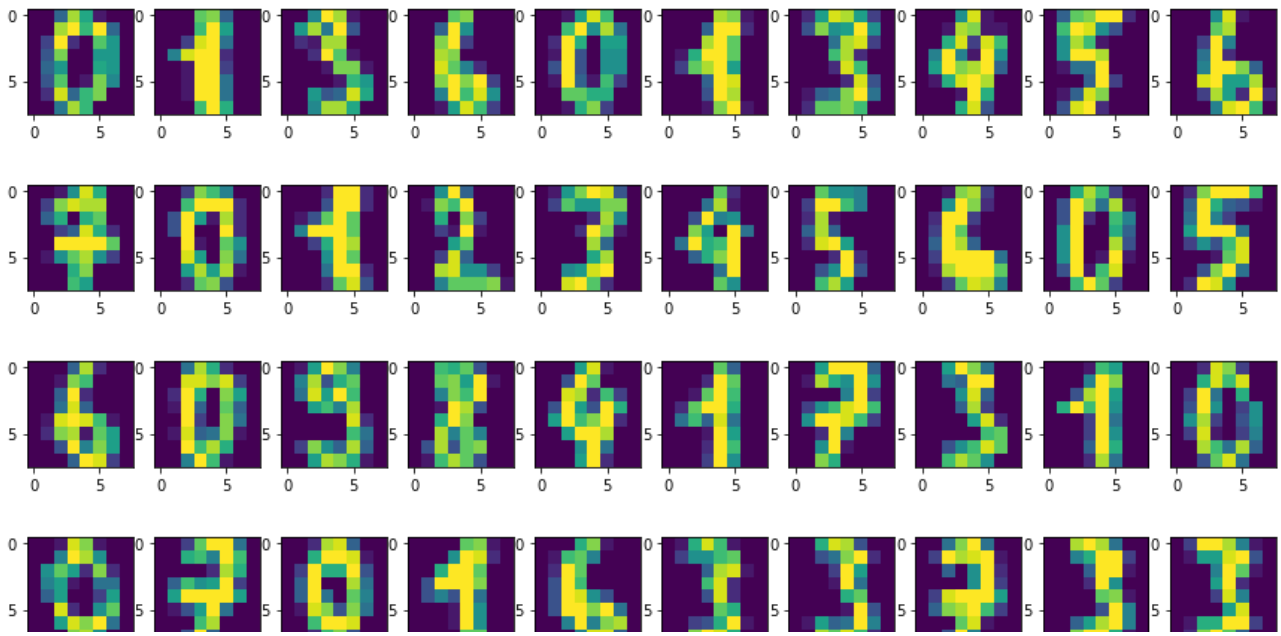
Это очень много! Фактически кластеризация не удалась. Посмотрим на центры.

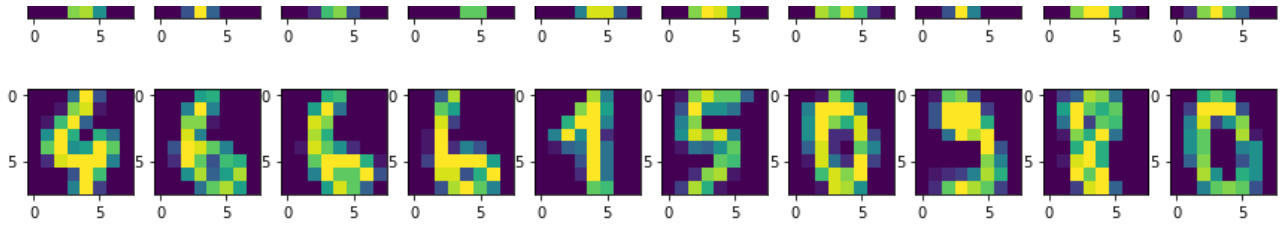
In [41]:

```
def draw(images):
    plt.figure(figsize=(15, 10))
#     print(len(images))
    for i, image in enumerate(images):
        plt.subplot(i // 10 + 1, 10, i+1)
        plt.imshow(image)
#         plt.axis('off')
```

In [42]:

```
for i in range(5):
    draw(mnist.images[dbscan.core_sample_indices_[i * 10:(i + 1) * 10]])
```



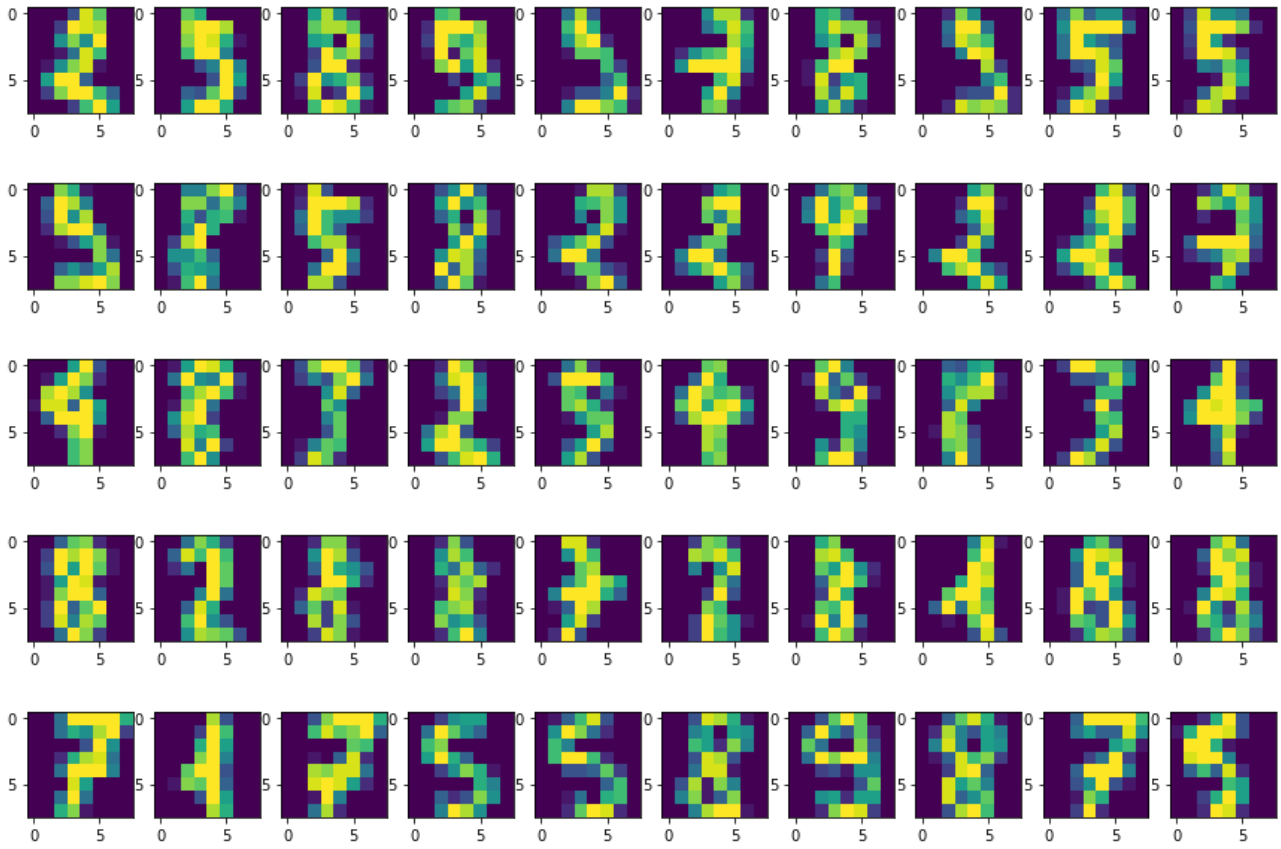


Вон там две почти одинаковые тройки подряд идут ;(

А это выбросы

In [43]:

```
for i in range(5):
    draw(mnist.images[labels == -1][i * 10:(i + 1) * 10])
```



С выбросами более понятно. Они действительно мало на что похожи.

Ну и ради интереса попробуем совсем другие параметры.

In [91]:

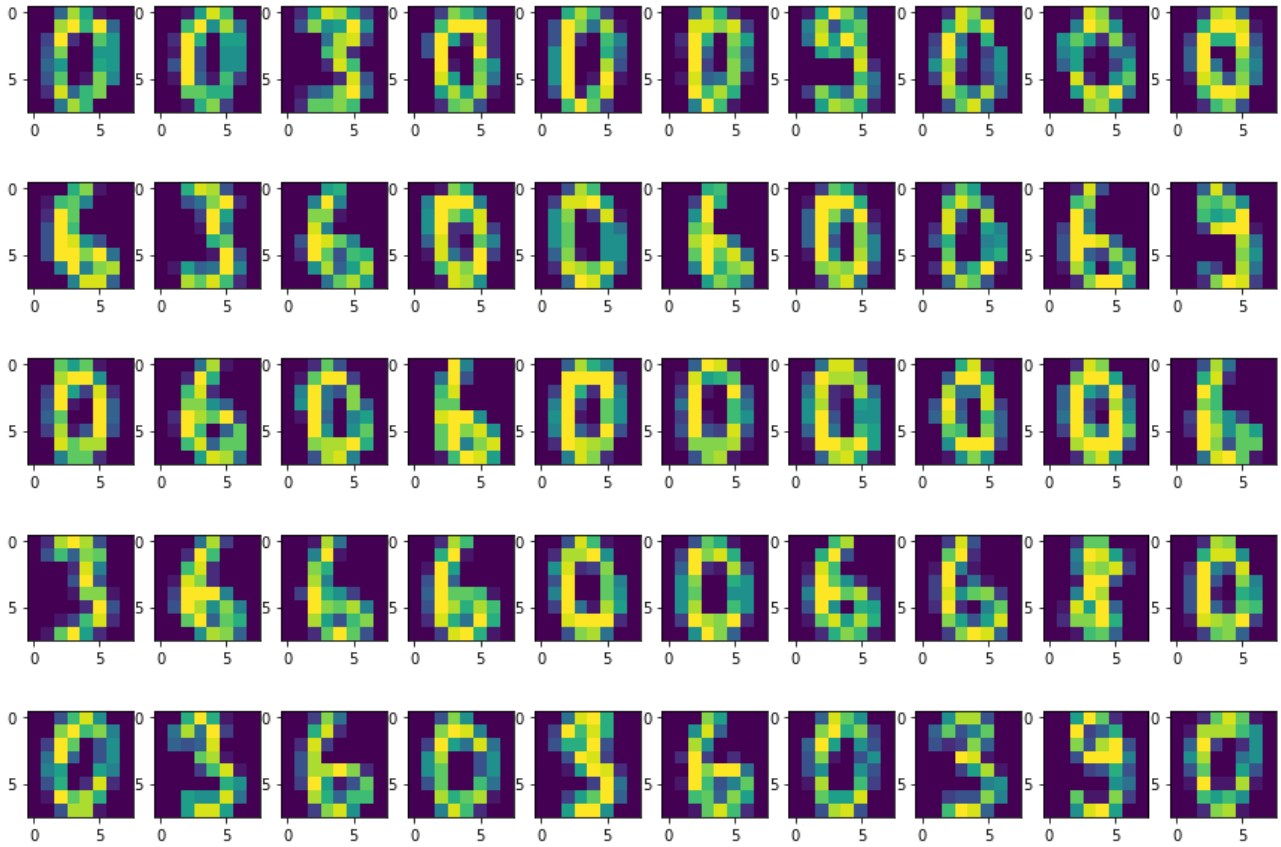
```
dbscan = cluster.DBSCAN(eps=31, min_samples=100,
                                 metric='euclidean', algorithm='auto',
leaf_size=30, p=None, n_jobs=1)
labels = dbscan.fit_predict(digits)
clusters = np.reshape(dbscan.components_, (dbscan.components_.shape[0], 8, 8
))
dbscan.components_.shape
```

Out[91]:

```
(325, 64)
```

In [92]:

```
for i in range(5):
    draw(clusters[i * 10:(i + 1) * 10])
```



Я вообще не понимаю, что происходит. Одни нули, но все почему-то в разных кластерах.
Вывод: алгоритм dbscan не подходит для этой задачи.

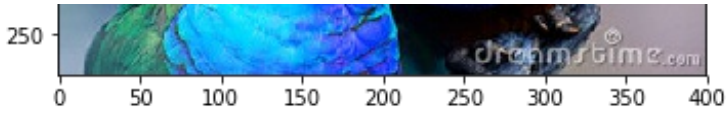
3.

In [36]:

```
from skimage import io
import cv2
img = cv2.imread('img.jpg')
plt.imshow(img)
img.shape
```

Out[36]:

```
(275, 400, 3)
```



In [38]:

```
kmeans = cluster.KMeans()
```

In [47]:

```
from skimage import measure
import scipy.misc
```

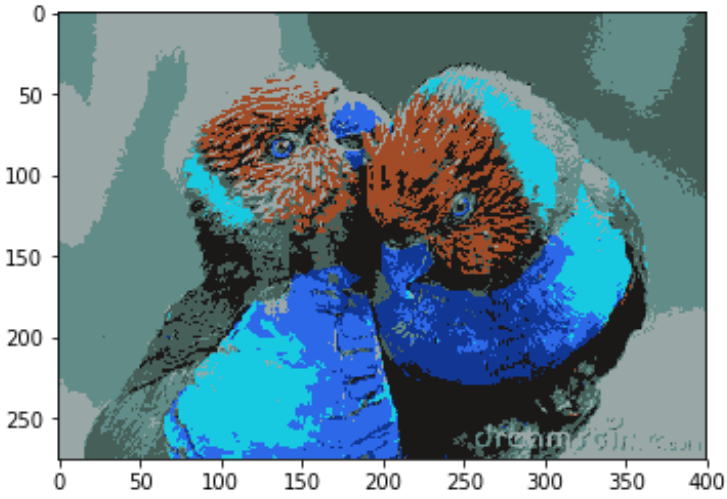
In [51]:

```
img = np.reshape(img, (-1, 3))
labels = kmeans.fit_predict(img)
values = kmeans.cluster_centers_
img_processed = list(map(lambda (i, v): values[labels[i]], enumerate(img)))
# print(img)
# img_processed =
np.zeros((img_processed.shape[0],img_processed.shape[1],3), np.uint8)
img_processed = np.reshape(img_processed, (275, 400, 3))
# img_processed = scipy.misc.toimage(img_processed)
img_processed = img_processed.astype(img.dtype)
img = np.reshape(img, (275, 400, 3))
print(measure.compare_ssim(img_processed, img, multichannel=True))
plt.imshow(img_processed)
```

```
0.655606723764
```

Out[51]:

```
<matplotlib.image.AxesImage at 0x7ff947dbfa90>
```



In [53]:

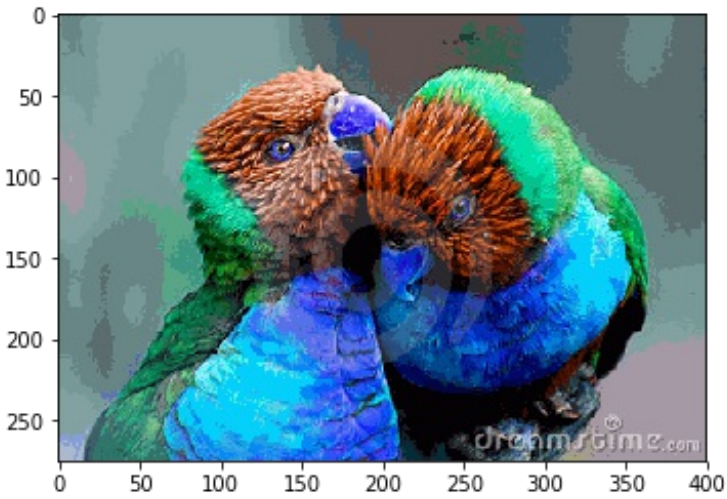
```
msh = cluster.MeanShift(bandwidth=15)
img = np.reshape(img, (-1, 3))
labels = msh.fit_predict(img)
values = msh.cluster_centers_
img_processed = list(map(lambda (i, v): values[labels[i]], enumerate(img)))
# print(img)
# img_processed =
np.zeros((img_processed.shape[0],img_processed.shape[1],3), np.uint8)
```

```
img_processed = np.reshape(img_processed, (275, 400, 3))
# img_processed = scipy.misc.toimage(img_processed)
img_processed = img_processed.astype(img.dtype)
img = np.reshape(img, (275, 400, 3))
print(measure.compare_ssim(img_processed, img, multichannel=True))
plt.imshow(img_processed)
```

```
0.852643312418
```

Out[53]:

```
<matplotlib.image.AxesImage at 0x7ff8eebcf790>
```



In [54]:

```
msh.cluster_centers_.shape
```

Out[54]:

```
(398, 3)
```

In [84]:

```
from sklearn import cluster
from sklearn.neighbors import kneighbors_graph
img = np.reshape(img, (-1, 3))
knn_graph = kneighbors_graph(img, 30, include_self=False)
agg = sk.cluster.AgglomerativeClustering(linkage='ward',
                                         connectivity=knn_graph,
                                         n_clusters=8)
labels = agg.fit_predict(img)
clusters = [img[labels == i] for i in range(agg.n_components_)]
values = []
print(clusters[0][:][2])
for i in range(agg.n_components_):
    values.append(np.mean(clusters[i], axis=0))
print(values)
img_processed = list(map(lambda (i, v): values[labels[i]], enumerate(img)))
img_processed = np.reshape(img_processed, (275, 400, 3))
img_processed = img_processed.astype(img.dtype)
img = np.reshape(img, (275, 400, 3))
print(measure.compare_ssim(img_processed, img, multichannel=True))
plt.imshow(img_processed)
```

```
/usr/local/lib/python2.7/dist-packages/sklearn/cluster/hierarchical.py:193:
UserWarning: the number of connected components of the connectivity matrix
is 11 > 1. Completing it to avoid stopping the tree early.
```

```
  connectivity, n_components = _fix_connectivity(X, connectivity)
```

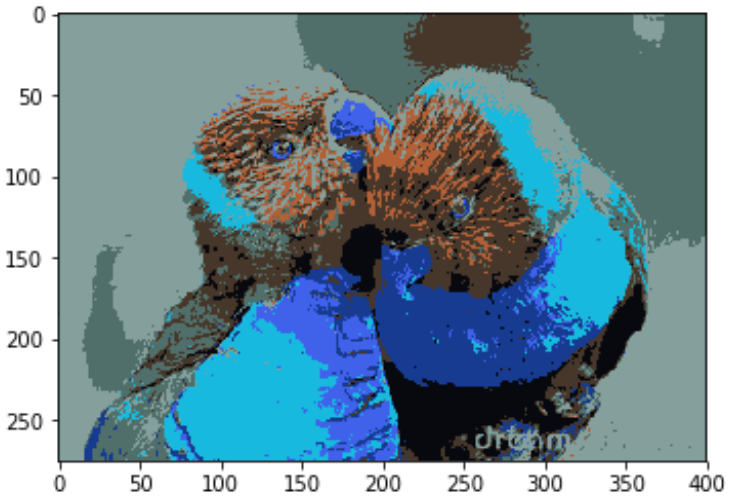
```
[113 147 137]
[array([ 133.90430058,  160.63140344,  156.82967286]), array([
23.53779275,  186.35717083,  222.78499256]), array([ 71.40072202,
55.44958742,  42.49207065]), array([  81.45904437,  111.90964613,
106.7829621 ]), array([  22.99145779,   59.37495785,  145.74710577]), array
([ 179.5278335 ,   96.62462387,   55.67728185]), array([  64.04131535,   97
.64333895,  234.03920742]), array([  8.75362319,   9.12432509,
15.56393862]), array([ nan,  nan,  nan]), array([ nan,  nan,  nan]), array(
[ nan,  nan,  nan])]
```

```
/usr/local/lib/python2.7/dist-packages/numpy/core/fromnumeric.py:2889: Runt
imeWarning: Mean of empty slice.
  out=out, **kwargs)
/usr/local/lib/python2.7/dist-packages/numpy/core/_methods.py:73: RuntimeWa
rning: invalid value encountered in true_divide
  ret, rcount, out=ret, casting='unsafe', subok=False)
```

```
0.646406433364
```

Out[84]:

```
<matplotlib.image.AxesImage at 0x7ff8ef06de10>
```



С кластеризацией на 8 цветов Kmeans и AgglomerativeClustering сработали почти одинаково по качеству, но на самом деле по-разному (видно визуально). MeanShift сработал лучше, но просто потому, что он кластеризовал на большее количество цветов (398 вместо 8!) Работал он вечность

In [ ]: