

Уменьшение количества цветов изображения

Данное задание основано на материалах лекций по методу k-средних (K-Means). Обратите внимание, что задание не является обязательным.

Вы научитесь:

- использовать алгоритм K-Means
- работать с задачами обучения без учителя
- работать с изображениям в Python

Введение

Самый распространенный тип задач машинного обучения — это задачи обучения с учителем. В них имеется обучающая выборка, для каждого объекта которой есть ответ, и нужно научиться предсказывать эти ответы для новых объектов. В такой постановке можно строго определить критерии качества.

Если же имеются лишь объекты, а ответов для них нет, то все равно можно пытаться найти в данных некую структуру. Задачи, которые ищут закономерности в неразмеченных выборках, называют задачами обучения без учителя. Типичный пример такой задачи — кластеризация, где требуется найти группы похожих объектов.

Кластеризация может использоваться для самых разных целей. В этом задании мы попробуем группировать схожие пиксели на изображении. Такой подход позволяет переходить к суперпиксельному пред-

ставлению изображений, которое является более компактным и лучше подходит для решения ряда задач компьютерного зрения.

Реализация в sklearn

Алгоритм KMeans реализован в классе `sklearn.cluster.KMeans`. Так как это один из примеров `unsupervised`-задачи, для обучения достаточно передать только матрицу объектов.

В качестве метрики будем использовать PSNR — адаптация метрики MSE для задачи нахождения сходства изображений.

Для работы с изображениями мы рекомендуем воспользоваться пакетом `scikit-image`. Чтобы загрузить изображение, необходимо выполнить следующую команду:

```
from skimage.io import imread
image = imread('parrots_4.jpg')
```

После этих действий переменная `image` будет содержать изображение в виде `numpy`-массива размера $n * m * 3$, где n и m соответствуют размерам изображения, а 3 соответствует формату представления RGB.

Инструкция по выполнению

1. Загрузите картинку `parrots.jpg`. Преобразуйте изображение, приведя все значения в интервал от 0 до 1. Для этого можно воспользоваться функцией `img_as_float` из модуля `skimage`.
2. Создайте матрицу объекты-признаки: характеризуйте каждый пиксель тремя координатами - значениями интенсивности в пространстве RGB.
3. Запустите алгоритм K-Means с параметрами `init='k-means++'` и `random_state=241`. После выделения кластеров все пиксели, отнесенные в один кластер, попробуйте заполнить двумя способами: медианным и средним цветом по кластеру.

4. Измерьте качество получившейся сегментации с помощью метрики PSNR. Эту метрику нужно реализовать самостоятельно (см. определение).

Ответ на каждое задание — текстовый файл, содержащий ответ в первой строчке. Обратите внимание, что отправляемые файлы не должны содержать пустую строку в конце. Данный нюанс является ограничением платформы Coursera. Мы работаем над тем, чтобы убрать это ограничение.