

ИСПОЛЬЗОВАНИЕ КЛАСТЕРИЗАЦИИ ДЛЯ АНАЛИЗА ТЕКСТУР ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ

Трефилова А.И.

Научный руководитель: Аксенов С.В., к.т.н., доцент
Томский политехнический университет
E-mail: alyona.trefilova@gmail.com

Введение

Для того, чтобы сделать процесс анализа текстур графического изображения легче, используется упрощение его представления. Это достигается с помощью сегментации исходного изображения. Для сегментации изображений обычно используются несколько методов и алгоритмов. В данной статье будет рассмотрена сегментация, основанная на кластеризации изображений.

Кластеризация изображений

Кластеризация используется для разбиения графического изображения на отдельные области с целью определения текстур на этих изображениях. Задача кластеризации относится к классу задач обучения без учителя.

Кластерный анализ – это совокупность математических методов, предназначенных для формирования относительно «отдаленных» друг от друга групп «близких» между собой объектов по информации о расстояниях или связях (мерах близости) между ними. [1]

Наиболее распространенным методом кластеризации является метод k-средних. Метод k-средних представляет собой итеративный метод для разбиения изображения на k число кластеров. Общая схема данного алгоритма выглядит следующим образом:

1. Выбирается k число кластеров и их центры.
2. Предполагаемый центр кластера сдвигается к тому кластеру, который ближе всех других к нему и центром которого он будет являться.

Использование сети Кохонена и самоорганизующейся карты Кохонена

Сети Кохонена и самоорганизующиеся карты Кохонена являются разновидностью нейросетевых алгоритмов. Особенность данного алгоритма в том, что при обучении используется метод обучения без учителя. Это означает, что результат обучения зависит только от структуры входных данных. Для обучения такой сети применяются механизмы конкуренции. На вход сети подается некоторый вектор x , и побеждает тот нейрон, вектор весов которого в самой меньшей степени отличается от входного вектора. В качестве меры расстояния используется Евклидово расстояние, которое рассчитывается по следующей формуле:

$$d(x, w_i) = \|x - w_i\| = \sqrt{\sum_{j=1}^n (x_j - w_{ij})^2},$$

где $d(x, w)$ – расстояние между векторами x и w , n – количество нейронов, j – номер нейрона-победителя.

При применении данного алгоритма схожие в исходном пространстве вектора оказываются рядом и на полученной карте.

Идея алгоритма обучения состоит в том, что сначала векторы входных сигналов x обрабатываются по одному, затем для каждого из них находится ближайший кодовый вектор («вектор-победитель») $W_{j(x)}$. В первую очередь, самоорганизующиеся карты Кохонена служат для визуализации и первоначального анализа данных. Каждая точка данных отображается соответствующим вектором из решетки

Алгоритм самоорганизующихся карт Кохонена представляет собой вариант кластеризации многомерных векторов. Его важным признаком является то, что все нейроны в нем структурированы и упорядочены чаще всего в двумерную сетку. Благодаря этому самоорганизующиеся карты Кохонена можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью.

Сеть Кохонена обучается методом последовательных приближений. В процессе обучения таких сетей на входы подаются данные, но сеть при этом подстраивается не под эталонное значение выхода, а под закономерности во входных данных. Начинается обучение с выбранного случайным образом выходного расположения центров.

В процессе последовательной подачи на вход сети обучающих примеров определяется наиболее схожий нейрон (тот, у которого скалярное произведение весов и поданного на вход вектора минимально). Этот нейрон объявляется нейроном-победителем и является центром при подстройке весов у соседних нейронов. Такое правило обучения предполагает обучение по механизму конкуренции с учетом расстояния нейронов от «нейрона-победителя».

Обучение при этом заключается в подстройке весов для наибольшего совпадения с входными данными.

Входные сигналы последовательно предъявляются сети, при этом желаемые выходные сигналы не определяются. После предъявления доста-

точного числа входных векторов синаптические веса сети становятся способны определить кластеры. Веса организуются так, что топологически близкие узлы чувствительны к похожим входным сигналам. [5]

В результате работы алгоритма центр кластера устанавливается в определенной позиции, объединяющей примеры, для которых данный нейрон является "победителем". В результате обучения сети необходимо определить меру соседства нейронов, то есть окрестность нейрона-победителя. Этой окрестностью являются несколько нейронов, которые окружают нейрон-победитель.

Сначала к окрестности принадлежит большое число нейронов, далее ее размер постепенно уменьшается. Сеть формирует топологическую структуру, в которой похожие примеры образуют группы примеров, близко находящиеся на топологической карте.

Предположим, у нас есть набор точек и вектор точек, которые должны стать центрами кластеров. Алгоритм должен найти кластеры, являющиеся скоплением точек, путем установления весовых векторов в центры найденных кластеров.

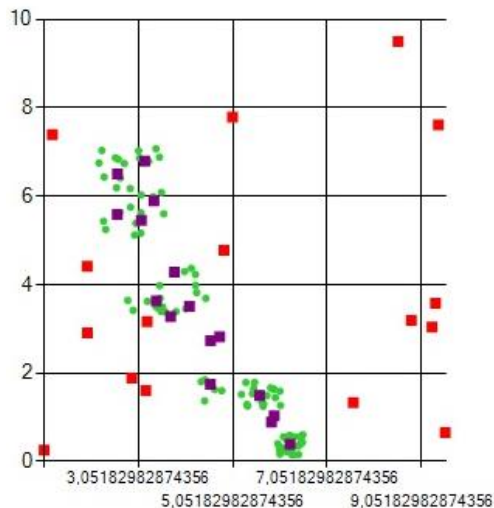


Рис. 28. Результат до и после обучения самоорганизующейся карты Кохонена

Сначала задается некоторый набор точек, которые, в свою очередь, будут представлять определенное количество кластеров, центрами которых должны стать весовые вектора. Их координаты также поступают на вход. Далее определяется число итераций, каждая из которых состоит из двух операций. Сначала выбирается подходящий весовой вектор, а затем обновляется его положение. Для каждой точки кластера рассчитывается Евклидово расстояние между этой точкой и всеми весовыми векторами в сети Кохонена. Тот весовой вектор, расстояние от которого наименьшее до выбранного случайным образом обучающего вектора, является вектором-победителем. Далее для

этот победитель обучается по следующему правилу: $w(t+1) = w(t) + speed * (d(t) - w(t))$, где $w(t+1)$ – новый весовой вектор, $w(t)$ – значение предыдущего весового вектора, $d(t)$ – обучающий вектор, $speed$ – это скорость обучения. В нашем случае скорость обучения выбрана 0.7. В конечном итоге весовые вектора передвигаются к центрам своих кластеров.

На рис. 1 представлен пример работы алгоритма кластеризации с использованием самоорганизующихся карт Кохонена. Изначальный набор векторов, представляющих кластеры, изображен зеленым цветом. Весовые вектора до обучения выделены зеленым цветом, а после обучения – фиолетовым.

Для каждого из трех компонентов цветовой модели RGB (красный, зеленый, синий) определяются по 4 характеристики:

- изменение значения характеристики;
- рябь;
- волнистость;
- изменение значения характеристики и рябь.

Например, возьмем волнистость и изменение значения характеристики для красного цвета и рябь для синего цвета. В результате запуска программы получим следующий результат.

Заключение

Использование кластеризации для задачи анализа текстур является способом повышения эффективности данного анализа. Кластеризация, основанная на работе алгоритма самоорганизующихся карт Кохонена, позволяет получить увеличение скорости программы, нацеленной на анализ текстур графических изображений.

Литература

1. Королев М.А. Статистический словарь. М.: Финансы и статистика. 1989 г.-623с.
2. Мандель И.Д. Кластерный анализ. М.: Финансы и статистика. 1988 г.-176с.
3. Самоорганизующиеся карты Кохонена – математический аппарат [Электронный ресурс]. – Режим доступа: <http://www.basegroup.ru/library/analysis/clusterization/som/>, свободный.
4. Нейронная сеть Кохонена [Электронный ресурс]. – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Самоорганизующаяся_карта_Кохонена#cite_note-0, свободный.
5. Аверченко В.И., Казаков Ю.М. Автоматизация проектирования технологических процессов. Учебное пособие для вузов. М.: Флинта. 2011 г.-229с.