Кластеризация документов с помощью нейронных сетей[1]

Габдрахманова Н.Т.,
кандидат технических наук, доцент кафедры высшей математики Российского университета дружбы народов (РУДН), Москва

Аннотация

В работе рассматривается задача автоматизации кластеризации документов, классификации документов и динамической классификации документов (ситуационная задача). Предлагается метод кластеризации с использованием локального коэффициента кластеризации для графов. Алгоритм кластеризации основан на структурном анализе графа. Представление текста в виде графа позволяет определить дискретный аналог кривизны Риччи на метрическом пространстве, как это сделано в работах Олливье. Для решения задачи классификации документов с помощью нейронных сетей предложены регуляризаторы на основе введенных понятий.

Ключевые слова: лексема, кластер, локальный коэффициент кластеризации, нейронная сеть, локальный коэффициент кривизны.

Введение

В работе рассматривается решение задачи кластеризации документов с использованием нейронных сетей, неориентированных графов, оценок кривизны Риччи на графе. Кластеризация документов — это разбиение документов по группам (видам) на основе признаков содержания, формы составления и др., так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Задача кластеризации относится к статистической обработке, а также к широкому классу задач без учителя.

Классификация документов заключается в отнесении документа к одной из нескольких категорий на основании содержания документа. Задача кластеризации отличается от задачи классификации прежде всего тем, что в этих задачах не заданы категории (классы). Кластерный анализ — многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы. Задача кластеризации встречается в самых различных областях. Разработано достаточно много хороших алгоритмов кластеризации, среди них: иерархическая кластеризация, эвристические графовые алгоритмы, статистические методы (метод к-средних) [1]. Выбор

[1] Работа выполнена при поддержке гранта РФФИ № 16-08-00568.

алгоритма кластеризации определяется постановкой задачи и свойств объектов, к которым он применяется. Решению задачи кластеризации документов посвящено множество работ, например [2]. Однако, задача не перестает быть актуальной. Причина этого, необходимость обработки большого объема данных, возросшие требования к качеству и скорости решения задачи с одной стороны, и с другой стороны, разработка новых информационных технологий, создание сверхскоростных вычислительных машин и развитие новых разделов прикладной математики.

Постановка задачи

Имеется совокупность n объектов D = $\{d_1,d_2,...,d_n\}$, отличающихся друг от друга рядом признаков. Требуется разбить данное множество на k непересекающихся подмножеств (кластеров), каждая из которых максимально однородна. При этом каждому объекту $d_i \in D$ приписывается метка (номер) кластера yi.

Алгоритм кластеризации — это функция $f : D \to Y$, которая любому объекту $d_i \in D$ ставит в соответствие номер кластера $y_j \in Y$. Множество меток Y в некоторых случаях известно заранее, однако чаще ставится задача определения оптимального числа кластеров, с точки зрения того или иного критерия качества. Для построения алгоритма кластеризации требуется ввести функцию расстояния между объектами ρ(x, x′) и функцию качества разбиения F(.). Решение задачи кластеризации принципиально неоднозначно. Во-первых, не существует однозначно наилучшего критерия качества кластеризации. Во-вторых, число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием. В-третьих, результат кластеризации существенно зависит от метрики ρ, выбор которой, как правило, также субъективен и определяется экспертом.

Заметим, что нейросетевые технологии позволяют быстро и эффективно решать задачи кластеризации. Главное преимущество нейронных сетей в том, что они хорошо приспособлены для параллельных вычислений. И, конечно, важно, что они обучаемы.

Предобработка данных обучающего массива нейросети

Прежде чем использовать нейронную сеть, необходимо ее обучить (настроить). Качество построенной нейросети зависит от качества данных обучающего массива и от того, насколько верно мы подобрали ее структуру. Предобработка данных это преобразование статистического набора данных. Большой опыт нейросетевого моделирования различных объектов показал, что предобработка исходных данных, на которых обучается нейросеть, может существенно повысить качество построения нейросетевой модели. Ниже рассматриваются основные шаги предобработки текста.

Прежде всего, необходимо разработать способы формализации текста. Согласно [2], при построении тематической модели нет смысла различать

формы (склонения, спряжения) одного и того же слова. Это приведёт к неоправданному разрастанию словаря, дроблению статистики, увеличению ресурсоёмкости и снижению качества модели. Лемматизация — это приведение каждого слова в документе к его нормальной форме. Например, в русском языке нормальными формами считаются: для существительных именительный падеж, единственное число. Далее будем полагать, что словарь W получен в результате предварительной обработки всех документов коллекции D. Элементы словаря $v \in V$ будем называть лексемами. В статистических методах обработки используется допущение, что всегда имеется шум. В наших объектах будем полагать, что стоп-слова являются шумом. Стоп-слова это слова, встречающиеся во многих текстах различной тематики, бесполезны для тематического моделирования. Такие слова могут быть отброшены. К ним относятся предлоги, союзы, числительные, местоимения, некоторые глаголы, прилагательные и наречия. Число таких слов обычно варьируется в пределах нескольких сотен. Их отбрасывание почти не влияет на длину словаря, но может приводить к заметному сокращению длины некоторых текстов. Слова, встречающиеся в длинном документе слишком редко, например, только один раз, также можно отбрасывать, полагая, что данное слово не характеризует тематику данного документа.

Каждый документ может быть записан в виде графа. Вершинами графа являются лексемы, вершины соединены ребром, если они входят в одно общее предложение. Построим алгоритм кластеризации на основе анализа топологической структуры графа документа. В графах наиболее очевидным свойством является количество связей между вершинами и наличие треугольных отношений. Назначим каждому ребру длину 1, тогда три вершины, соединенные ребрами длинной 1, образуют треугольник. Например, в двудольных графах нет треугольников, в полном графе каждая тройка вершин образует треугольник. Чем больше треугольников у этих двух соседних вершин содержатся, тем больше перекрытие их окрестностей. Это предполагает аналогию с понятием кривизны Риччи в Римановой геометрии.

Локальный коэффициент кластеризации и кривизна Риччи

Кривизна Риччи является фундаментальным понятием в Римановой геометрии. Тензор Риччи, точно так же как метрический тензор, есть симметричная билинейная форма на касательном пространстве риманова многообразия. Тензор Риччи, задаёт способ измерения кривизны многообразия, то есть степени отличия геометрии многообразия от геометрии плоского евклидова пространства.

Одним из математических инструментов, позволяющим исследовать структуру связей вершин графа является понятие кривизны Риччи на дискретных пространствах. Понятие кривизны Риччи обобщены на метрические пространства в работах [3–6] и др. Среди введенных определений кривизны Риччи, в упомянутой выше литературе, на дискретных пространствах особенно хорошо работает кривизна Олливье.

Дадим определение локальной кривизны Риччи, которые дает Yann Olliver

Определение 1. Пусть (X,d) — польское метрическое пространство, снабженное борелевской сигма алгеброй. Случайное блуждание m на X есть семейство вероятностных мер $m_x(.)$ на X для любого $x \in X$, удовлетворяющее двум следующим предположениям: (i) мера mx зависит от точки $x \in X$; (ii) каждая мера m_x имеет конечный первый момент.

Определение 2. Пусть (X,d) — метрическое пространство, и пусть μ_1 и μ_2 две вероятностные меры на X. Вводится метрика: расстояние между μ_1 и μ_2 есть:

$$\tau(\mu_1, \mu_2) := inf_{\varepsilon \in \Pi} \int_{(x,y)\in X\times X} d(x,y)d\varepsilon(x,y) \text{ , где} \quad (1)$$

$\Pi = \Pi(\mu_1, \mu_2)$ –это множество мер на $X \times X$ проецируемое на μ1 и μ2 .
$d(x,y)$ — это стоимость транспортировки единичной массы из x в y.
Пусть $x, y \in X$ две различные точки. Локальная кривизна между парой точек x, y пространства определяется формулой

$$k(x,y) := 1 - \frac{\tau(m_x, m_y)}{d(x,y)}. \quad (2)$$

Локальная кривизна будет использована в работе ниже.
Исследования показывают, что кривизна, как правило, чрезвычайно низка в случайных графах. Кластеры высокой кривизны имеют весьма неслучайную структуру. В геометрии кривизна (интуитивно, мера, количественно определяющая отклонение геометрического объект из плоского) играет центральную роль.
Для решения задачи кластеризации в работе используется формула Watts-Strogatz [7] :

curv(A)=t/(v(v-1)/2),

здесь v — числа вершин и t — число треугольников, которые образованы ребрами графа, содержащими вершину A. Данная функция — функция двух переменных. Заметим, что величина v (v-1) / 2-максимальное количество треугольников, которое можно составить с помощью всех вершин графа, следовательно, curv(A) лежит между 0 и 1. На рис. 3 примеры графов и кривизны. curv(A) — локальный коэффициент кластеризации.

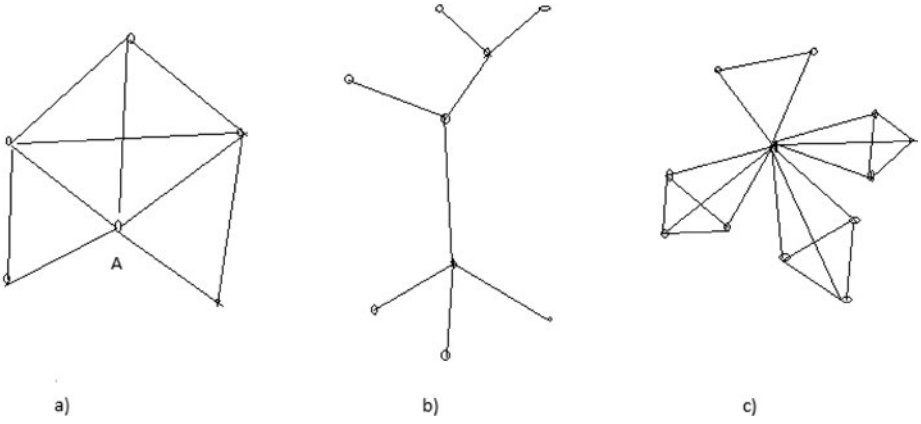

Рис 1. a) the node n has V = 5 neighbors and T = 5 triangles, thus curvature (n) = 1/2, b) tree, each node has a curvature of 0
c) node is a hub with curvature ≈ 1 / v

Алгоритмы построения моделей

Рассмотрим алгоритмы решения задач классификации и кластеризации на основе введенных понятий.

А) *Задача классификации документов.*

Пусть дано множество документов D. Необходимо построить нейросетевую модель для классификации модели.

Решение. Будем полагать, что классификация для документов множества D уже решена. Нам нужно построить нейросетевую модель, которая осуществляла бы классификацию новых документов автоматически.

Алгоритм решения задачи.

1. Записать предложения документов в виде лексем.
2. Построить граф G соответствующий документу. Вершины графа -лексемы. Лексемы соединены ребром, если они принадлежат общему предложению.
3. Вычислить локальные коэффициенты всех вершин графа curv(v).
4. Задать порог h. Удалить из графа вершины, для которых выполнено curv(v)<h. Полученный граф назовем сужением графа G и обозначим G*.
5. Для каждого документа записать кортежи $<A_i, y_j>$, $i = \overline{1,n}$, $j = \overline{1,m}$.
6. Вычислить весовые коэффициенты W нейросети: F(<A,y>)=W.

Использование локального коэффициента кластеризации позволяет найти вершины, которые наиболее важны (они имеют много связей с другими вершинами) в документе. Такая предобработка входных данных позволяет более четко найти области кластеров, т.е. показатель curv(v) выступает в роли регулизатора.

Б) Задача кластеризации документов.

Пусть дано множество документов D. Необходимо определить число кластеров.

Решение. Пусть $D = \{di, i=1,2,...,k\}$ множество документов. Пусть $V = \{V_i, i=1,2,..,n\}$ множество лексем D. Каждой лексеме присвоим код (номер). Поставим в соответствие каждому документу d_i граф G_i*. Построим объединение графов G. В результате, получим неориентированный граф соответствующий множеству документов D:

$G(V,E,w): |V|=n, |E|=m$

$w: E \to \mathbb{R}^+$ — весовая функция

V — множество вершин графа

E — множество ребер графа

k — число кластеров, $k \in Z$.

Задача кластеризации разбить множество V на k непересекающихся подмножеств $V = \cup_{i=1}^{k} V_i$, $V_i \cap V_j = \emptyset$ если $i \neq j$.

Критерий качества разбиения:

$$f(G) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \sum_{\substack{v_1 \in V_i \\ v_2 \in V_j \\ i \neq j}} n(v_1, v_2) \to \min \quad (1)$$

Весовую функцию всех ребер будем считать единицей.

Пример такого графа приведен на рис 2.

Согласно критерию (1), мы хотим разбить вершины на такие подмножества, что количество ребер между кластерами было минимально (другими словами — нахождение минимального сечения)

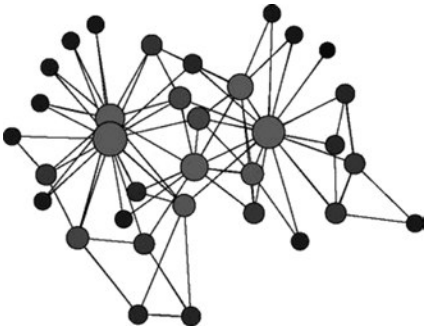

Рис. 2. Граф документов двух различных классов.

Для решения задачи разработаны различные алгоритмы и их модификации. Например, алгоритм Кернигана-Лина и спектральная релаксация [8]. Ниже приведен алгоритм Кернигана-Лина для разбиения вершин на два кластера.

Алгоритм Кернигана-Лина
Пусть дан граф $G=V,E$.
1. Разделить V на $V1$ и $V2$ случайным образом.
2. Найти пару вершин $v1 \in V1$ и $v2 \in V2$, перемещение которых макс. уменьшит (мин. увеличит) размер сечения.
3. Повторить шаг 2, пока не останется не перемещенных вершин.
4. Выбрать конфигурацию с минимальным сечением.
5. Перейти на шаг 2.

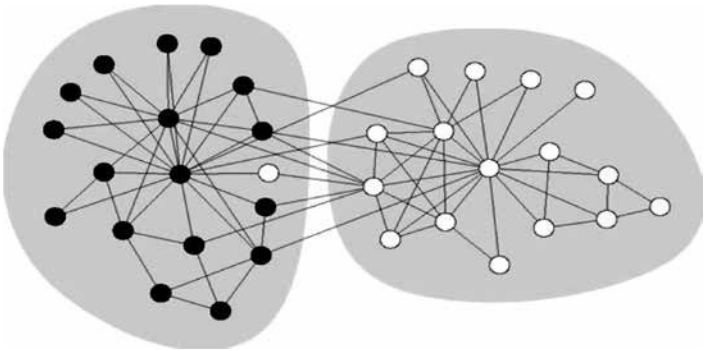

Рис.3. Алгоритм Кернигана-Лина.

В) Ситуационная задача (задача динамической классификации).

Имеется массив документов объединенных одной ситуацией и записанных в хронологической последовательности $D=\{d(t)\}$, t— принимает дискретные значения, t=1,...,n. Необходимо найти способы оценки моментов изменения ситуации в этой последовательности, или другими словами, осуществить динамическую классификацию последовательности документов.

Алгоритм решения задачи.

Поставим в соответствие каждому документу $d(t)$ граф $G^*(t)$, t=1,...,n..

На множестве вершин $G^*(t)$ определим вероятностное распределение. Это можно сделать аналогично тому, как это сделано в работах Олливье [4], или взять в качестве вероятностного распределения нормализованные значения локальных коэффициентов.

При переходе от t к (t+1) вероятностное распределение на множестве вершин графа изменяется.
Вычислить расстояния между мерами по формуле (1).
Вычислить локальные кривизны между парой точек по формуле (2).
Появление отрицательных значений кривизн ребер в графе $G^*(t)$ означает существенное изменение ситуации в момент времени t. Такой документ и является границей следующего кластера.

Числовые эксперименты решения задачи классификации

Приведем пример численных экспериментов решения задачи 1. Перенумеруем кластеры. Введем обозначения: y_i — номер кластера, $y_j \in Y$. Построим (например экспертно) отображение f: $D \to Y$. Всем документам множества D, чьи графы попали в один кластер поставим в соответствие номер этого кластера. Запишем матрицы смежности каждого документа.

Матрицы смежности графа документа d_i обозначим A_i. Согласно построению, поставим матрице смежности A_i тот же номер кластера y_j, что и документу. Кортежи $<A_i,y_i>$, $i=1,..,n$ — обучающий массив нейросети.

Пример. В качестве исходных данных взяты тексты различных классов. По выше описанному алгоритму были построены матрицы смежности. Фрагмент такой матрицы представлен в таблице1. В таблице x1–x8 — номера вершин, y— номер кластера(класса). Нейронная сеть построена с использованием обучающего массива, построенного по выше описанному способу. Затем, осуществлялась классификация документов не входящих в обучающий массив. В таблице 2 представлены результаты вычислений построенной нейросети. Приняты условные обозначения: MLP 8-6-1 означает многослойный персептрон, входных нейронов 8, в скрытом слое 6 нейронов и 1 нейрон в выходном слое.

В качестве оценки качества классификации принята величина:

$F= (1/n) \sum(y_i\text{-output } y_i)$, где y_i — желаемое значение номера кластера, output y_i — вычисленное значение номера кластера. Суммирование осуществляется по всем элементам тестового множества

Таблица 1

Фрагмент исходных данных

n	x1	x2	x3	x4	x5	x6	x7	x8	y
1	1	0	1	1	0	0	0	0	1
2	1	1	1	0	0	0	0	0	1
3	0	1	1	1	0	0	0	0	1
4	1	1	0	1	0	0	0	0	1
5	1	1	1	1	0	0	0	0	1
6	0	0	0	0	1	0	1	1	2
7	0	0	0	0	1	1	1	0	2
8	0	0	0	0	1	0	1	1	2
9	0	0	0	0	0	1	1	1	2
10	0	0	0	0	1	1	1	0	2

Таблица предсказанных значений для y (NS) Выборки: Обучающая						
	y - Целевая	**y - Выход - 1. MLP 8-6-1**	**y - Выход - 2. MLP 8-6-1**	**y - Выход - 3. MLP 8-4-1**	**y - Выход - 4. MLP 8-4-1**	**y - Выход - 5. MLP 8-4-1**
1	1,000	1,000	1,000	1,000	1,000	1,002
2	1,000	1,000	1,000	1,000	1,000	1,002
4	1,000	1,000	1,000	1,000	1,000	1,002
5	1,000	1,000	1,000	1,000	1,000	1,002
6	2,000	2,000	2,000	2,000	2,000	2,001
8	2,000	2,000	2,000	2,000	2,000	2,001
9	2,000	2,000	2,000	2,000	2,000	2,000
10	2,000	2,000	2,000	2,000	2,000	2,000

Выводы

В работе исследована возможность использования геометрических методов для решения задач кластеризации и классификации документов. Сформулированы идеи использования геометрического подхода для анализа ситуационных задач. Исследования в данном направлении только начаты. В дальнейшем, предполагается развить предложенные подходы на основе достигнутых в этой области результатов другими методами.

Литература

1. *Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д.* Прикладная статистика: Классификация и снижение размерности. М.: Финансы и статистика, 1989. 607 с.
2. *Воронцов К.В.* Вероятностное тематическое моделирование.
3. *Forman R.* Bochner's method for cell complexes and combinatorial Ricci curvature. Discrete Comput.Geom. 29, 323–374 (2003)
4. *Lin Y., Lu L., Yau S.T.:* Ricci curvature of graphs. Tohoku Math. J. 63, 605–627 (2011)
5 *Ollivier Y.* Ricci curvature of Markov chains on metric spaces. Journal of Functional Analysis. 256, 810–864 (2009).
6. *Ollivier Y.* Ricci curvature of metric spaces. C. R. Math. Acad. Sci. Paris. 345, 643–646 (2007).
7. *Lott J. & Villani C.* Ricci curvature for metric-measure spaces via optimal transport. Annals of Mathematics. 169, 903–991 (2009).
7. *Watts D.J. and Strogatz S.H.* Collective dynamics of 'small-world' networks Nature 1998, 393:440–442
8. *Zachary W.W.* An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33, 452–473 (1977).

CLUSTERING DOCUMENTS USING THE NEURAL NETWORKS

Gabdrakhmanova N.T.,
candidate of technical Sciences, associate Professor, Department of higher mathematics, peoples ' friendship University of Russia (RUDN), Moscow

Abstract

A new algorithm for clustering documents based on neural networks, weighted graphs, and adjacency matrices is proposed. Neural networks derive their power from a parallel processing method and the ability to self-learn. The construction of a weighted graph for the document assumes the solution of the task of formalizing the object of modeling.

The following clustering algorithm is proposed. Suppose we have N documents. We use these documents to get the training array of our neural network. Let each document already be divided into lexemes. A lexeme is a unit of the vocabulary of a language. A lexeme is the totality of the forms of a single word. For each document a weighted graph is constructed according to the following rule: the vertices of the graph are lexemes; the vertices of the graph are connected by an edge if the lexemes meet in the same sentence; the weight of the edge is the relative frequency of the lexemes in the text. In the tasks of clustering, we call the connective words in the text the "noise", i.e. such words as "so", "however", etc. In order to smooth "noise" we use filtering. We set an unspecified limit h, remove all edges with weight less than h. Base on the constructed weighted graph, we write the adjacency matrix Ai, where i is the document number. To every adjacency matrix Ai we associate the class of the document Yi. We obtain the tuples <Ai, Yi>, i = 1,2, ... N for training the neural network. After training the neural network, it can be used to cluster documents. At the input of the neural network, the adjacency matrix of the document is fed, at the output — the document class number.

In the future, it is proposed to develop the proposed clustering approach using the methods of modern geometry.

Keywords: lexeme, cluster, weighted graph, adjacency matrix, neural network.