5.3. Обзор алгоритмов кластеризации числовых пространств данных

Задача кластеризации – частный случай задачи обучения без учителя, которая сводится к разбиению имеющегося множества объектов данных на подмножества таким образом, что элементы одного подмножества существенно отличались по некоторому набору свойств от элементов всех других подмножеств. Объект данных обычно рассматривается как точка в многомерном метрическом пространстве, каждому измерению которого соответствует некоторое свойство (атрибут) объекта, а метрика – есть функция от значений данных свойств. От типов измерений этого пространства, которые могут быть как числовыми, так и категориальными, зависит выбор алгоритма кластеризации данных и используемая метрика. Этот выбор продиктован различиями в природе разных типов атрибутов.

Приведён краткий обзор методов кластеризации числовых пространств данных. Он поможет сориентироваться в многообразии современных алгоритмов кластеризации и получить о них общее представление. Материал не претендует на полноту изложения, напротив, описание алгоритмов максимально упрощено. Для более подробного изучения того или иного алгоритма рекомендуется использовать научную работу, в которой он был представлен.

Методы разбиения

Наиболее известные представители этого семейства методов – алгоритмы *k*-means [1] и *k*-medoids [2]. Они принимают входной параметр *k* и разбивают пространство данных на *k* кластеров таких, что между объектами одного кластера сходство максимально, а между объектами разных кластеров минимально. Сходство измеряется по отношению к некоторому центру кластера как дистанция от рассматриваемого объекта до центра. Основное различие между этими методами заключается в способе определения центра кластера.

В алгоритме *k*-means сходство рассматривается по отношению к центру масс кластера – среднему значению координат объектов кластера в пространстве данных. Сначала произвольно выбираются *k* объектов, каждый из которых является прототипом кластера и представляет его центр масс. Затем для каждого из оставшихся объектов выполняется присоединение к тому кластеру, с которым сходство больше. После этого центр масс каждого кластера вычисляется заново. Для каждого полученного разбиения рассчитывается некоторая оценочная функция, значения которой на каждом шаге образуют сходящейся ряд. Процесс продолжается до тех пор, пока указанный ряд не сойдётся к своему предельному значению. Иными словами, перемещение объектов из кластера в кластер заканчивается тогда, когда с каждой итерацией кластеры будут оставаться неизменными. Минимизация оценочной функции позволяет сделать результирующие кластеры настолько компактными и раздельными, насколько это возможно.

Метод *k*-means хорошо работает, когда кластеры представляют собой значительно разделённые между собой компактные «облака». Он эффективен для обработки больших объёмов данных, однако не применим для обнаружения кластеров невыпуклой формы или сильно различающегося размера. Более того, метод очень чувствителен к шуму и обособленным точкам пространства, поскольку даже малое количество таких точек может существенно влиять на вычисление центра масс кластера.

Чтобы сократить влияние шума и обособленных точек пространства на результат кластеризации, алгоритм *k*-medoids, в отличие от *k*-means, использует для представления центра кластера не центр масс, а представительный объект – один из объектов кластера. Как и в методе k-means, сначала произвольным образом выбирается *k* представительных объектов. Каждый из оставшихся объектов объединяется в кластер с ближайшим представительным объектом. Затем итеративно для каждого представительного объекта производится его замена произвольным непредставительным объектом пространства данных. Процесс замены

продолжается до тех пор, пока улучшается качество результирующих кластеров. Качество кластеризации определяется суммой отклонений между каждым объектом и представительным объектом соответствующего кластера, которую метод стремится минимизировать.То есть, итерации продолжаются до тех пор, пока в каждом кластере его представительный объект не станет медоидом – наиболее близким к центру кластера объектом. Алгоритм плохо масштабируем для обработки больших объёмов данных, но эту проблему решает дополняющий метод k-medoids алгоритм CLARANS [3]. Для кластеризации многомерных пространств на основе CLARANS построен алгоритм PROCLUS [4].

Иерархические методы

Общая идея методов данной группы заключается в последовательной иерархической декомпозиции множества объектов. В зависимости от направления построения иерархии различают дивизимный и агломеративный методы. В случае агломеративного метода (снизу вверх) процесс декомпозиции начитается с того, что каждый объект представляет собой самостоятельный кластер. Затем на каждой итерации пары близлежащих кластеров последовательно объединяются в общий кластер. Итерации продолжаются до тех пор, пока все объекты не будут объединены в один кластер или пока не выполнится некоторое условие остановки. Дивизимный метод (сверху вниз) напротив, подразумевает, что на начальном этапе все объекты объединены в единый кластер. На каждой итерации он разделяется на более мелкие до тех пор, пока каждый объект не окажется в отдельном кластере или не будет выполнено условие остановки. В качестве условия остановки можно использовать пороговое число кластеров, которое необходимо получить, однако обычно используется пороговое значение расстояния между кластерами.

Основная проблема иерархических методов заключается в сложности определения условия остановки таким образом, чтобы выделить «естественные» кластеры и в то же время не допустить их разбиения. Еще

одна проблема иерархических методов кластеризации заключается в выборе точки разделения или слияния кластеров. Этот выбор критичен, поскольку после разделения или слияния кластеров на каждом последующем шаге метод будет оперировать только вновь образованными кластерами, поэтому неверный выбор точки слияния или разделения на каком-либо шаге может привести к некачественной кластеризации. Кроме того, иерархические методы не могут быть применены к большим наборам данных, потому как решение о разделении или слиянии кластеров требует анализа большого количества объектов и кластеров, что ведёт к большой вычислительной сложности метода. Примерами алгоритмов, основанных на иерархическом методе являются BIRCH [5] и CHAMELEON [6].

Плотностные методы

Кластеры рассматриваются как регионы пространства данных с высокой плотностью объектов, которые разделены регионами с низкой плотностью объектов.

Алгоритм DBSCAN [7] – один из первых алгоритмов кластеризации плотностным методом. В основе этого алгоритма лежит несколько определений:

- *ε-окрестностью* объекта называется окрестность радиуса ε некоторого объекта.
- *Корневым объектом* называется объект, ε-окрестность которого содержит не менее некоторого минимального числа *MinPts* объектов.
- *Объект p непосредственно плотно-достижим* из объекта q, если p находится в ε-окрестности q и q является корневым объектом.
- *Объект p плотно-достижим* из объекта q при заданных ε и *MinPts*, если существует последовательность объектов $p_1, \ldots, p_n$, где $p_1 = q$ и $p_n = p$, такая что p_i+1 непосредственно плотно достижим из p_i, $1 \leq i \leq n$.
- *Объект p плотно-соединён* с объектом q при заданных ε и *MinPts*, если существует объект o такой, что p и q плотно-достижимы из o.

Для поиска кластеров алгоритм DBSCAN проверяет ε-окрестность каждого объекта. Если ε-окрестность объекта p содержит больше точек чем $MinPts$, то создаётся новый кластер с корневым объектом p. Затем DBSCAN итеративно собирает объекты непосредственно плотно-достижимые из корневых объектов, которые могут привести к объединению нескольких плотно-достижимых кластеров. Процесс завершается, когда ни к одному кластеру не может быть добавлено ни одного нового объекта.

Хотя, в отличие от методов разбиения, DBSCAN не требует заранее указывать число получаемых кластеров, требуется указание значений параметров ε и $MinPts$, которые непосредственно влияют на результат кластеризации. Оптимальные значения этих параметров сложно определить, особенно для многомерных пространств данных. Кроме того, распределение данных в таких пространствах часто несимметрично, что не позволяет использовать для их кластеризации глобальные параметры плотности. Для кластеризации многомерных пространств данных на базе DBSCAN был создан алгоритм SUBCLU [8].

Сетевые методы

Общая идея методов заключается в том, что пространство объектов разбивается на конечное число ячеек, образующих сетевую структуру, в рамках которой выполняются все операции кластеризации. Главное достоинство методов этой группы в малом времени выполнения, которое обычно не зависит от количества объектов данных, а зависит только от количества ячеек в каждом измерении пространства.

Алгоритм CLIQUE [9], адаптированный под кластеризацию данных высокой размерности, является одним из классических сетевых алгоритмов. Метод основан на том предположении, что если в многомерном пространстве данных распределение объектов не равномерно – встречаются регионы плотности и разрежения, то проекция региона плотности в подпространство с меньшей размерностью будет частью региона плотности в этом подпространстве. Алгоритм CLIQUE производит кластеризацию

многомерного пространства данных следующим образом: пространство данных разбивается на не пересекающиеся ячейки фиксированного размера, среди них идентифицируются плотные ячейки – такие, плотность объектов данных в которых превышает заданное пороговое значение. Далее из найденных ячеек формируется пространство, в котором могут существовать плотные ячейки большей размерности. Процесс начинается с одномерных пространств (описанная процедура выполняется для каждого измерения) с последующим переходом к подпространствам более высокой размерности.

Этот алгоритм масштабируем для обработки большого количества данных, однако при большом количестве измерений число рассматриваемых комбинаций растёт нелинейно, следовательно, требуется использовать эвристики для сокращения количества рассматриваемых комбинаций. Кроме того, получаемый результат очень сильно зависит от выбора размера ячейки и порогового значения плотности объектов в ячейке. Это является большой проблемой, поскольку одни и те же значения этих параметров используются при рассмотрении всех комбинаций измерений. Эту проблему решает алгоритм MAFIA [10], работающий по схожему принципу, но использующий адаптивный размер ячеек при разбиении подпространств.

Модельные методы

Методы этого семейства предполагают, что имеется некоторая математическая модель кластера в пространстве данных и стремятся максимизировать сходство этой модели и имеющихся данных. Часто при этом используется аппарат математической статистики.

Алгоритм EM [11] основан на предположении, что исследуемое множество данных может быть смоделировано с помощью линейной комбинации многомерных нормальных распределений. Его целью является оценка параметров распределения, которые максимизируют функцию правдоподобия, используемую в качестве меры качества модели. Иными словами, предполагается, что данные в каждом кластере подчиняются определенному закону распределения, а именно, нормальному

распределению. С учетом этого предположения можно определить оптимальные параметры закона распределения – математическое ожидание и дисперсию, при которых функция правдоподобия максимальна. Таким образом, мы предполагаем, что любой объект принадлежит ко всем кластерам, но с разной вероятностью. Тогда задача будет заключаться в «подгонке» совокупности распределений к данным, а затем в определении вероятностей принадлежности объекта к каждому кластеру. Очевидно, что объект должен быть отнесен к тому кластеру, для которого данная вероятность выше.

Алгоритм EM прост и лёгок в реализации, не чувствителен к изолированным объектам и быстро сходится при удачной инициализации. Однако он требует для инициализации указания количества кластеров k, что подразумевает наличие априорных знаний о данных. Кроме того, при неудачной инициализации сходимость алгоритма может оказаться медленной или может быть получен некачественный результат. Очевидно, что подобные алгоритмы не применимы к пространствам с высокой размерностью, поскольку в этом случае крайне сложно предположить математическую модель распределения данных в этом пространстве.

Концептуальная кластеризация

В отличие от традиционной кластеризации, которая обнаруживает группы схожих объектов на основе меры сходства между ними, концептуальная кластеризация определяет кластеры как группы объектов, относящейся к одному классу или концепту – определённому набору пар атрибут-значение.

Алгоритм COBWEB [12] – классический метод инкрементальной концептуальной кластеризации. Он создаёт иерархическую кластеризацию в виде дерева классификации: каждый узел этого дерева ссылается на концепт и содержит вероятностное описание этого концепта, которое включает в себя вероятность принадлежности концепта к данному узлу и условные

вероятности вида: $P(A_i = v_{ij}|C_k)$, где $A_i = v_{ij}$ – пара атрибут-значение, C_k – класс концепта.

Узлы, находящейся на определённом уровне дерева классификации, называют срезом. Алгоритм использует для построения дерева классификации эвристическую меру оценки, называемую полезностью категории – прирост ожидаемого числа корректных предположений о значениях атрибутов при знании об их принадлежности к определённой категории относительно ожидаемого числа корректных предположений о значениях атрибутов без этого знания. Чтобы встроить новый объект в дерево классификации, алгоритм COBWEB итеративно проходит всё дерево в поисках «лучшего» узла, к которому отнести этот объект. Выбор узла осуществляется на основе помещения объекта в каждый узел и вычисления полезности категории получившегося среза. Также вычисляется полезность категории для случая, когда объект относится к вновь создаваемому узлу. В итоге объект относится к тому узлу, для которого полезность категории больше.

Однако COBWEB имеет ряд ограничений. Во-первых, он предполагает, что распределения вероятностей значений различных атрибутов статистически независимы друг от друга. Однако это предположение не всегда верно, потому как часто между значениями атрибутов существует корреляция. Во-вторых, вероятностное представление кластеров делает очень сложным их обновление, особенно в том случае, когда атрибуты имеют большое число возможных значений. Это вызвано тем, что сложность алгоритма зависит не только от количества атрибутов, но и от количества их возможных значений.

Список литературы

1. MacQueen, J. Some methods for classification and analysis of multivariate observations/ J. MacQueen // In Proc. 5th Berkeley Symp. On Math. Statistics and Probability, 1967. - С.281-297.

2. Kaufman, L. Clustering by means of Medoids, in Statistical Data Analysis Based on the l–Norm and Related Methods / L. Kaufman, P.J. Rousseeuw, Y. Dodge, 1987. - C.405-416.

3. Ng, R.T. Efficient and Effective Clustering Methods for Spatial Data Mining / R.T. Ng, J. Han // Proc. 20th Int. Conf. on Very Large Data Bases. Morgan Kaufmann Publishers, San Francisco, CA, 1994. - C.144-155.

4. Aggarwal, C.C. Fast Algorithms for Projected Clustering / C.C. Aggarwal, C. Procopiuc // In Proc. ACM SIGMOD Int. Conf. on Management of Data, Philadelphia, PA, 1999. 12 c.

5. Zhang, T. BIRCH: An Efficient Data Clustering Method for Very Large Databases / T. Zhang, R. Ramakrishnan, M. Linvy // In Proc. ACM SIGMOD Int. Conf. on Management of Data. ACM Press, New York, 1996. - C.103-114.

6. Karypis, G. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling / G. Karypis, E.-H. Han, V. Kumar // Journal Computer Volume 32 Issue 8. IEEE Computer Society Press Los Alamitos, CA, 1999. - C.68-75

7. Ester, M. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise / M. Ester, H.-P. Kriegel, J. Sander, X. Xu // In Proc. ACM SIGMOD Int. Conf. on Management of Data, Portland, OR, 1996. – C. 226-231.

8. Kailing, K. Density-Connected Subspace Clustering for High-Dimensional Data / K. Kailing, H.-P. Kriegel, P. Kröger // In Proceedings of the 4th SIAM International Conference on Data Mining (SDM), 2004. - C.246-257.

9. Agrawal, R. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications / R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan // In Proc. ACM SIGMOD Int. Conf. on Management of Data, Seattle, Washington, 1998. - C.94-105.

10. Nagesh, H. MAFIA: Efficient and Scalable Subspace Clustering for Very Large Data Sets / H. Nagesh, S. Goil, A. Choudhary // Technical Report Number

CPDC-TR-9906-019, Center for Parallel and Distributed Computing, Northwestern University, 1999. 20 c.

11. Demster, A. Maximum Likelihood from Incomplete Data via the EM Algorithm /A.P. Demster, N.M. Laird, D.B. Rubin //JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B, Vol. 39, No. 1, 1977. - C.1-38.

12. Fisher, D.H. Knowledge acquisition via incremental conceptual clustering / D.H. Fisher // Machine Learning 2, 1987. - C.139-172.