УДК 004.75
doi:10.31799/1684-8853-2019-2-35-43

Научные статьи
Articles

Кластеризация данных в распределенных системах мониторинга

***А. Н. Рукавицын*[a],** *аспирант, orcid.org/0000-0001-5382-0465, rkvtsn@gmail.com*
[a]*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», Профессора Попова ул., 5, Санкт-Петербург, 197376, РФ*

Введение: *традиционные способы анализа распределенных источников данных обычно используют централизованные хранилища данных и имеют ряд недостатков, связанных с конфиденциальностью, высокой стоимостью централизованного хранения данных, ограниченной пропускной способностью и высокой нагрузкой на телекоммуникационные сети. Методики, по которым выполняется децентрализованный анализ, не учитывают вид распределения данных и особенности выбранного алгоритма. Это снижает производительность и точность анализа или может быть причиной невыполнимости его в заданных условиях.* ***Цель:*** *обзор и анализ особенностей работы распределенных систем мониторинга и алгоритмов интеллектуального анализа данных.* ***Результаты:*** *для проведения кластеризации на основе распределенных источников данных установлены требования к алгоритму в системах распределенного мониторинга: однопроходность, поддержка разных типов входных данных, работа онлайн-режима, адаптация к данным при изменении среды, масштабирование больших объемов данных, выполнение анализа без предположений о распределении входных данных, анализ данных на источниках информации без их передачи третьей стороне. Определены два основных способа распределения данных на источниках в гетерогенных системах: вертикальный и горизонтальный. Выполнена классификация методов в соответствии с их основным принципом разграничения кластеров. Классификация включает основные алгоритмы кластеризации, их принцип работы, достоинства и недостатки. Обзор и анализ существующих методов кластеризации выявил, что в распределенных системах мониторинга наиболее эффективными являются алгоритмы на основе нейронных сетей Кохонена. Декомпозирован алгоритм самоорганизующихся карт Кохонена и определены блоки работы с данными: вычисление нейрона-победителя и настройка весов нейронов. Предложены две стратегии кластеризации распределенных данных.* ***Практическая значимость:*** *предложенные стратегии позволяют выполнять кластеризацию в системах с распределенными источниками с учетом характеристик среды без передачи всех данных.*

Ключевые слова *— интеллектуальный анализ данных, кластеризация, датчики, распределенные источники данных, самоорганизующиеся карты Кохонена, распределенные системы обработки данных, виды распределения, системы мониторинга.*

Для цитирования: Рукавицын А. Н. Кластеризация данных в распределенных системах мониторинга. *Информационно-управляющие системы*, 2019, № 2, с. 35–43. doi:10.31799/1684-8853-2019-2-35-43
For citation: Rukavitsyn A. N. Data clustering in distributed monitoring systems. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 2, pp. 35–43 (In Russian). doi:10.31799/1684-8853-2019-2-35-43

Введение

Развитие информационных технологий привело к появлению новых видов устройств и методов их взаимодействия, например, облачной виртуализированной среды или интернета вещей. Данные хранятся на разных, независимо работающих устройствах, которые могут быть связаны друг с другом через локальные или глобальные сети, например, в охранных системах, мобильных или корпоративных сетях, где датчики расположены географически в разных местах. Распределенные системы мониторинга — это системы, состоящие из разного вида датчиков с возможностями обнаружения, связи и ограниченным количеством вычислительной мощности. Системы мониторинга применяются в различных сферах жизнедеятельности [1–3]: в медицине, для защиты окружающей среды, обеспечения безопасности и др.

В ряде исследований в области интеллектуального анализа распределенных данных чаще всего рассматривают алгоритмы на основе вероятностного подхода без учета различных видов распределения.

Методики, как правило, сводятся к двум этапам [4–9]:

1) кластеризации на источниках данных, где установлены модели для кластеризации и передачи результатов на следующий этап;

2) объединению результатов и группировке кластеризации.

Объединение может быть выполнено иерархически [4], и тогда каждый вычислительный узел является листом иерархии, который объединяет свои результаты с соседними листьями на узлах до корня полученного дерева. С таким подходом сложность выполнения растет с ростом количества узлов и установленных слоев. При этом выделяется дополнительное время на передачу данных между соседними узлами.

В работе Ф. Л. Горгонио (F. L. Gorgonio) [8] описан подход, использующий ансамбль моделей (нейронных сетей), построенных SOM-алго-

ритмом (Self-Organized Maps). В нем на обоих этапах используется один и тот же алгоритм кластеризации, но для разных задач. На первом этапе SOM определяет нейронную сеть с кластерами и передает глобальной модели. На втором этапе используется процесс кластеризации карт для определения кластеров из переданных карт. Такой метод может привести к увеличению степени неопределенности.

Одним из наиболее распространенных подходов анализа распределенных данных является централизация в хранилище данных, к которым применяются традиционные методы интеллектуального анализа [10–12]. Хранилище данных — популярная технология, которая объединяет данные из нескольких источников в один, чтобы эффективно выполнять сложные аналитические запросы [13]. Однако, несмотря на широкое распространение, этот подход может быть непрактичным или невозможным по следующим причинам [14–16]:

— огромное количество данных генерируется на разных источниках, а из-за стоимости централизации данные не могут масштабироваться для связи, хранения и вычислений;

— владельцы данных не могут или не хотят передавать информацию, например, защищая конфиденциальность или потому, что раскрытие такой информации может привести к преимуществу конкурентов или значительной коммерческой добавленной стоимости;

— ограниченная пропускная способность и высокая нагрузка при передаче данных.

Соответственно, подход для интеллектуального анализа распределенных данных должен учитывать:

— адаптацию алгоритма построения модели в распределенной сети;

— объединение данных с минимальной нагрузкой сети и достаточной точностью анализа;

— вид распределения.

Распределенная модель системы мониторинга

Рассмотрим систему мониторинга с использованием сенсорных сетей, где узлы датчиков могут быть оснащены звуковыми, вибрационными, температурными и отражающими зондами. Допустим, датчики контролируют географический регион и должны следить за данными, обрабатывать их и обмениваться данными друг с другом, чтобы отслеживать и идентифицировать объекты, интересующие пользователя. Наблюдения обычно представляют собой данные в виде временных рядов. Система мониторинга может контролировать множество объектов X:

$$X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}.$$

Каждый объект характеризуется набором атрибутов A:

$$A = \{a_1, a_2, \ldots, a_j, \ldots, a_m\}.$$

При этом система мониторинга следит за значением атрибута x каждого объекта в каждый момент времени t:

$$x_i(t) = \{a_{1i}(t), a_{2i}(t), \ldots, a_{ji}(t), \ldots, a_{mi}(t)\}.$$

Анализируемые данные для системы мониторинга можно представить в виде таблицы (табл. 1).

Стоит обратить внимание, что системы могут быть однородными (т. е. каждый узел наблюдает общее подмножество данных) или гетерогенными (т. е. каждый узел наблюдает соответствующее подмножество данных). Гетерогенная система рассматривает два вида распределения данных: горизонтальное и вертикальное. При горизонтальном распределении данные делятся на источники по строкам или для временных рядов по времени либо по объектам из X. При вертикальном распределении данные делятся на источники по столбцам или по атрибутам из A.

Одной из главных задач систем мониторинга является сегментация сцен для дальнейших

■ ***Таблица 1***. Формальное представление данных для системы мониторинга

■ ***Table 1***. Formal presentation of data for the monitoring system

t	X	A_1	A_2	...	A_j	...	A_m	M
t_1	$x_1(t_1)$	$a_{11}(t_1)$	$a_{21}(t_1)$	...	$a_{j1}(t_1)$	...	$a_{m1}(t_1)$	M_1
	$x_2(t_1)$	$a_{12}(t_1)$	$a_{22}(t_1)$	...	$a_{j2}(t_1)$	...	$a_{m2}(t_1)$	
	...	...	...	...	...	...	...	
	$x_i(t_1)$	$a_{1i}(t_1)$	$a_{2i}(t_1)$	...	$a_{ji}(t_1)$	...	$a_{mi}(t_1)$	
	...	...	...	...	...	...	...	
	$x_n(t_1)$	$a_{1n}(t_1)$	$a_{2n}(t_1)$	...	$a_{jn}(t_1)$	...	$a_{mn}(t_1)$	
...								
t_z	$x_1(t_2)$	$a_{11}(t_2)$	$a_{21}(t_2)$	...	$a_{j1}(t_2)$	...	$a_{m1}(t_2)$	M_z
	$x_2(t_2)$	$a_{12}(t_2)$	$a_{22}(t_2)$	...	$a_{j2}(t_2)$	...	$a_{m2}(t_2)$	
	...	...	...	...	...	...	...	
	$x_i(t_2)$	$a_{1i}(t_2)$	$a_{2i}(t_2)$	...	$a_{ji}(t_2)$	...	$a_{mi}(t_2)$	
	...	...	...	...	...	...	...	
	$x_n(t_2)$	$a_{1n}(t_2)$	$a_{2n}(t_2)$	...	$a_{jn}(t_2)$	...	$a_{mn}(t_2)$	

идентификации интересующих объектов (например, транспортные средства) и последующей их классификации (например, седан). Стандартным подходом сегментации является кластеризация [17]. Кластеризация данных — наиболее используемый метод интеллектуального анализа данных в распределенных системах мониторинга [18]. Цель этого метода заключается в разложении или разбиении набора данных на группы путем минимизации межгруппового несходства и максимизации внутригруппового сходства. При этом требуется организовать передачу данных от датчиков в единое хранилище для последующей кластеризации. В сенсорных системах мониторинга это сложно выполнить по таким причинам, как ограниченная пропускная способность связи или вычислительная мощность. Традиционная структура централизованных алгоритмов кластеризации плохо масштабируется в распределенных приложениях, так как предполагается передача данных на базовый узел. При этом интенсивная передача данных по ограниченному каналу полосы пропускания может привести к снижению времени отклика всей системы.

Еще одной типичной задачей для интеллектуального анализа данных в системах мониторинга является выявление выбросов в данных, например обнаружение химического разлива или вторжений. Кластерный анализ можно отнести к основным подходам обнаружения выбросов. Для таких задач использование централизованных методов кластеризации имеет ранее перечисленные недостатки. Поэтому важное значение приобретает разработка эффективных алгоритмов распределенной кластеризации, которые требуют небольшой полосы пропускания при построении одноранговой среды сетей датчиков.

Основные трудности при решении такой проблемы заключаются в следующем:

— количество обрабатываемых измерений чрезвычайно велико — 10^6 и более;

— пространство объектов, используемое для группировки входных данных в регионы, имеет множество измерений;

— нет предварительной информации о количестве и местонахождении искомых регионов;

— данные распределены на нескольких источниках и не могут быть объединены в центральном хранилище.

Из описанных особенностей работы распределенных систем мониторинга вытекают нижеперечисленные требования к методам кластеризации:

— однопроходность по данным — алгоритм должен выполнять кластеризацию за один проход по всем данным;

— поддержка разных типов входных данных — данные могут быть дискретными, непрерывными, категориальными и др.;

— визуализация кластеров, связей и возможность интерпретации полученной визуализации — одно из главных требований, позволяющее на основе полученных результатов производить анализ и диагностику специалистами для выявления причин выбросов в данных;

— поддержка онлайн-режима и адаптация к данным при изменении среды;

— проецирование многомерного пространства в пространство с более низкой размерностью, что увеличивает скорость обработки многомерных данных, получаемых от множества датчиков системы;

— выделение наиболее значимых атрибутов;

— масштабирование больших объемов данных;

— отсутствие ограничений входных параметров априорными знаниями, например количеством кластеров;

— выполнение анализа без предположений о распределении входных данных;

— обнаружение аномалий;

— анализ данных на источниках информации без их передачи третьей стороне;

— поддержка анализа распределенных данных.

В соответствии с сформулированными требованиями необходимо определить наиболее подходящий метод кластеризации для анализа в системах мониторинга с распределенными источниками данных.

Методы кластерного анализа

С появлением интеллектуального анализа данных было предложено множество методов кластеризации. Кластеризацию можно классифицировать по нескольким категориям [19–22] в соответствии с принципом формирования кластеров.

Иерархическая кластеризация

Иерархическая кластеризация направлена на создание иерархии вложенных разбиений исходного множества объектов. Иерархическая кластеризация представляет иерархию, т. е. структуру, которая является более информативной, чем неструктурированный набор плоских кластеров. Алгоритмы имеют в основном высокие требования по памяти и по времени, что делает их слишком медленными даже для средних наборов данных.

При *агломеративном* подходе («снизу-вверх») новые кластеры создаются путем объединения более мелких кластеров — дерево иерархии строится от листьев к стволу.

В случае *дивизимного* подхода («сверху-вниз») новые кластеры формируются путем деления бо-

лее крупных кластеров на мелкие — дерево иерархии строится от ствола к листьям.

Одним из популярных алгоритмов иерархической кластеризации является BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies). В отличие от своих аналогов, он был специально разработан для минимизации количества операций ввода-вывода. Алгоритм особенно подходит для больших объемов данных [23]. Он динамически сегментирует входящие многомерные метрические точки данных, чтобы создать кластер с наилучшим качеством и доступными ресурсами (доступной памятью и временными ограничениями) [24]. После первой кластеризации может улучшить качество с помощью нескольких дополнительных сканирований. Он также является первым алгоритмом кластеризации, предложенным в области базы данных для эффективного управления «шумом».

Преимущества:

1) гибкость в отношении уровня детализации;

2) удобство представления для иерархических данных;

3) быстрота обработки.

Недостатки:

1) невозможность внести исправления после разделения/слияния;

2) отсутствие интерпретируемости дескрипторов кластера;

3) нечеткость критерия завершения;

4) невозможность работать с асферическими кластерами разного размера;

5) неэффективность в высокоразмерных пространствах из-за проклятия размерности;

6) чувствительность порядка данных.

Центроидная кластеризация

Задачу центроидной кластеризации [25] можно формально определить следующим образом: необходимо найти центры k кластеров, назначить объекты ближайшему центру кластера и минимизировать квадраты расстояний.

Основным представителем центроидного метода является алгоритм K-Means. Он может организовывать более точные кластеры, чем при иерархической кластеризации. Модель может изменять кластеры (переходить на другой кластер), когда центроиды пересчитываются. При работе с этим методом трудно предсказать количество k кластеров. Начальная инициализация и порядок данных оказывают сильное влияние на результаты. Алгоритм имеет чувствительность к масштабируемости — нормализация и стандартизация влияют на результаты.

Преимущества:

1) относительно масштабируемый и простой;

2) хорошо подходит для наборов данных с компактными сферическими кластерами, которые качественно разделены.

Недостатки:

1) сильное снижение эффективности в высокоразмерных пространствах, поскольку почти все пары точек находятся примерно так же далеко, как и средние;

2) плохие дескрипторы кластера;

3) опора на экспертные знания пользователя;

4) высокая чувствительность к начальным значениям, шуму и выбросам;

5) частые остановки на локальных оптимумах;

6) низкая точность анализа данных с невыпуклыми кластерами разного размера и плотности.

Плотностная кластеризация

С помощью методов, основанных на плотности, определяются кластеры относительно плотностных регионов [26]. Известны алгоритмы с одним сканированием, например DBSCAN (Density-Based Spatial Clustering of Applications with Noise) и OPTICS (Ordering Points to Identify the Clustering Structure). Ключевым недостатком таких алгоритмов является то, что они ожидают снижения плотности для обнаружения границ кластера.

Преимущества:

1) возможность определения кластеров произвольной формы с разным размером;

2) сопротивляемость шуму и выбросам.

Недостатки:

1) высокая чувствительность к настройке входных параметров;

2) плохие дескрипторы кластера;

3) низкая эффективность при анализе высокоразмерных наборов данных из-за явления проклятия размерности.

Кластеризация на основе нейронных сетей Кохонена

Эти методы используют нейронные сети, основным элементом которых является слой Кохонена. Слой Кохонена представляет собой адаптивные линейные сумматоры. Выходные сигналы слоя обычно обрабатываются по правилу [27]: наибольший сигнал превращается в единичные, остальные — в ноль. В результате нейроны, обученные на входных данных, образуют области, соответствующие кластерам в этих данных. Примерами такого типа алгоритмов являются SOM или GNG (Growing Neural Gas). GNG относится к алгоритмам [28] Topology Representing Networks [29]. Он использует процесс роста данных для постепенного создания сетевой модели прототипа входного пространства, где результирующая сетевая структура формирует индуцированную триангуляцию Делоне пространства. Это отличает алгоритм GNG от аналогичных, таких как SOM [30], который использует фиксированную топологию сети. Это позволяет GNG

более подробно отражать структуру входного пространства. GNG может создавать непересекающиеся серии, которые указывают возможные классы во входных данных. GNG [31] и другие растущие нейронные сети (такие как Growing Cell Structures [32], Growing Grid [33] и т. д.) используют механизм постепенной адаптации и самостоятельной корректировки своих собственных размеров. GNG-алгоритм имеет сложность $O(n^2)$, где n — количество нейронов (т. е. промежуточных кластеров), и не зависит от количества входных данных. GNG может работать, когда входные данные подчиняются непрерывному распределению. Этот факт является значительным преимуществом, например, по сравнению с алгоритмом K-Means, и позволяет использовать алгоритм для больших объемов входных данных.

В случае применения алгоритма SOM в качестве средства предобработки данных для дальнейшей кластеризации результаты показывают высокую эффективность для сжатия многомерных наборов данных, что делает интерпретацию и диагностику намного проще и очевиднее.

Преимущества:

1) определение кластеров произвольной формы с разным размером;

2) кластеризация без начальных знаний о данных;

3) обработка больших объемов данных;

4) сжатие размерности данных.

Недостатки:

1) интерпретация результатов кластеризации требует преобразования результатов, например U-matrix [34];

2) точность определения кластеров ниже, чем у предопределенных (например, K-Means, при условии верно подобранного количества кластеров k).

По результатам сравнительной характеристики (табл. 2) наиболее эффективными для выполнения кластеризации в распределенных системах мониторинга являются алгоритмы, построенные на основе нейронных сетей Кохонена. Для анализа данных системы мониторинга этот класс методов обладает такими преимуществами, как поддержка разных типов входных данных и работа с высокоразмерными данными. Кроме того, метод позволяет производить визуализацию кластеров и их связей. Кластеризация на основе нейронных сетей Кохонена не требует ввода начальных значений и априорных знаний об исследуемом объекте. Метод адаптируется к данным и может автоматически определять количество кластеров, что делает процесс кластеризации универсальным. Еще одним преимуществом является комбинирование с другими методами кластеризации и уменьшение размерности данных.

■ ***Таблица 2.*** Сравнительная характеристика методов кластеризации для систем мониторинга

■ ***Table 2.*** Comparative characteristics of clustering methods for monitoring systems

Требование	Методы			
	Иерархические	Центроидные	Плотностные	Нейронные сети Кохонена
Однопроходность	–	+	+	+
Работа с высокоразмерными данными	–	–	–	+
Поддержка разных типов входных данных	–	+	+	+
Поддержка онлайн-режима (адаптация к данным при их изменении)	–	+	+	+
Работа с большим количеством данных	–	+	+	+
Визуализация связей	–	+	–	+*
Визуализация высокоразмерных данных	–	–	–	+*
Графическая интерпретация данных	–	–	–	+*
Отсутствует необходимость предварительных данных	+	–	+	+
Работа с распределенными данными	+*	+*	+*	+*

* — требует предобработку специальными методами.

Стратегии распределенного анализа данных

Для кластеризации распределенных данных необходимо декомпозировать алгоритм на основные блоки [35, 36] — выделить блоки работы с данными [37] и определить требования относительно горизонтального и вертикального распределения данных.

В алгоритмах кластеризации на основе нейронных сетей Кохонена можно выделить общую последовательность блоков:

1) инициализация нейронов;

2) для каждого вектора:

— определение нейрона-победителя;

— вычисление окрестности;

— корректировка весов и конфигурации сети (также вычисление ошибок и соединений нейро-

нов, корректировка топологии путем добавления и удаления нейронов).

Следующие блоки предполагают работу с данными:

— определение нейрона-победителя (вычисление расстояния от каждого вектора до нейрона);

— корректировку весов (в вычислениях участвуют значения атрибутов векторов).

Соответственно, описанные блоки должны выполняться на узлах источников. Возможны следующие стратегии распределенного выполнения кластеризации:

1) с промежуточной синхронизацией нейрона-победителя, когда результаты отправляются на общий узел, объединяются и обобщенная модель рассылает на узлы источники для выполнения шага;

2) со слиянием нейронных сетей после обработки всех или части входных данных, когда карты строятся отдельно на каждом узле.

Перечисленные стратегии обладают разными характеристиками по времени выполнения и нагрузке на сеть.

Первая стратегия является медленной и создает высокую нагрузку на сеть, так как для каждого вектора выполняется два взаимодействия узлов источников с общим узлом, что замедляет работу алгоритма и увеличивает сетевой трафик.

Вторая стратегия оптимальна и требует одного или нескольких взаимодействий. При этом могут получиться разные результаты в зависимости от частоты слияния n. Так, при $n = 1$ слияние нейронных сетей выполняется для каждого вектора. Это повышает точность модели, но увеличивает число передач данных по сети и, как следствие, время анализа и сетевой трафик. При такой стратегии необходимо учитывать время завершения обработки самого медленного источника. При значении параметра n, равного количеству векторов входных данных z, снижается количество сетевого трафика, так как слияние результатов выполняется после анализа всех входных векторов данных. Возможны разные промежуточные варианты $1 < n < z$, определяющие слияние нейронных сетей:

— количество обработанных входных векторов данных;

— интервал времени между каждым слиянием (каждую минуту, каждые 10 минут и т. д.);

— промежуток времени после получения входного вектора и т. п.

Выбор стратегии и параметров слияния зависит от условий среды выполнения кластеризации и способа распределения данных.

Заключение

В работе были сформулированы основные требования методов кластеризации для распределенных систем мониторинга. Проведена классификация и обзор методов кластеризации. Выделены основные преимущества и недостатки алгоритмов и методов кластеризации.

Обзор методов кластеризации данных показал, что наиболее эффективным для анализа распределенных систем мониторинга является метод на основе нейронных сетей Кохонена. Однако для его применения к решению задачи требуется его адаптация к распределенному выполнению.

Декомпозирован алгоритм самоорганизующихся карт Кохонена и определены блоки работы с данными: вычисление нейрона-победителя и настройка весов нейронов. Предложены две стратегии кластеризации распределенных данных. Дальнейшие работы будут связаны с исследованием и разработкой кластеризации распределенных данных на основе нейронных сетей с учетом различных видов распределения.

Литература

1. **Catarinucci L.**, et al. An IoT-aware architecture for smart healthcare systems. *IEEE Internet of Things Journal*, 2015, vol. 2, no. 6, pp. 515–526.
2. **Andreu-Perez J.**, et al. From wearable sensors to smart implants — toward pervasive and personalized healthcare. *IEEE Transactions on Biomedical Engineering*, 2015, vol. 62, no. 12, pp. 2750–2762.
3. **Del Vecchio P.**, et al. Creating value from social big data: Implications for smart tourism destinations. *Information Processing & Management*, 2018, vol. 54, no. 5, pp. 847–860.
4. **Bendechache M., Kechadi M. T., Le-Khac N. A.** Efficient large scale clustering based on data partitioning. *Data Science and Advanced Analytics (DSAA), 2016 IEEE Intern. Conf.*, 2016, pp. 612–621.
5. **Aouad L. M., Le-Khac N. A., Kechadi T. M.** Lightweight clustering technique for distributed data mining applications. *Industrial Conf. on Data Mining*, Springer, Berlin, Heidelberg, 2007, pp. 120–134.
6. **Le-Khac N. A., Aouad L. M., Kechadi M. T.** A new approach for distributed density based clustering on grid platform. *British National Conf. on Databases*, Springer, Berlin, Heidelberg, 2007, pp. 247–258.
7. **Januzaj E., Kriegel H. P., Pfeifle M.** DBDC: Density based distributed clustering. *Intern. Conf. on Extending Database Technology*, Springer, Berlin, Heidelberg, 2004, pp. 88–105.
8. **Gorgonio F. L., Costa J. A. F.** Parallel self-organizing maps with application in clustering distributed data. *IEEE Intern. Joint Conf. Neural Networks (IJCNN 2008)*, 2008, pp. 3276–3283.

9. **Kim M., Jung S., Park M.** A distributed self-organizing map for DoS attack detection. *Seventh Intern. Conf. Ubiquitous and Future Networks (ICUFN)*, 2015, pp. 19–22.
10. **Ali A. A.,** et al. Distributed data mining systems: Techniques, approaches and algorithms. *22nd Intern. Conf. on Circuits, Systems, Communications and Computers (CSCC 2018)*, Majorca, Spain, July 14–17, 2018, vol. 210, p. 04038.
11. **Song C.,** et al. Updating strategy of master-slave data consistency based on message. *Computer Engineering*, 2004, vol. 1, pp. 035.
12. **Robert H.** A majority consensus approach to concurrency control for multiple copy databases. *ACM Trans. on Database System*, 2003, vol. 4, no. 2, pp. 543–549.
13. **Moro G., Sartori C.** Incremental maintenance of multi-source views. *Proc. of the 12th Australasian Database Conf.*, IEEE Computer Society, 2001, pp. 13–20.
14. **Oussous A.,** et al. Big data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 2018, vol. 30, no. 4, pp. 431–448.
15. **Li T.,** et al. Differentially private naive bayes learning over multiple data sources. *Information Sciences*, 2018, vol. 444, pp. 89–104.
16. **Assunção M. D.,** et al. Big data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, 2015, vol. 79, pp. 3–15.
17. **Duda R. O., Hart P. E., Stork D. G.** *Pattern classification*. John Wiley & Sons, 2012. 688 p.
18. **Bendechache M., Kechadi M. T.** Distributed clustering algorithm for spatial data mining. *2nd IEEE Intern. Conf. on Spatial Data Mining and Geographical Knowledge Services (ICSDM)*, IEEE, 2015, pp. 60–65.
19. **Saraswathi S., Sheela M. I.** A comparative study of various clustering algorithms in data mining. *International Journal of Computer Science and Mobile Computing*, 2014, vol. 11, no. 11, pp. 422–428.
20. **Sisodia D.,** et al. Clustering techniques: a brief survey of different clustering algorithms. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 2012, vol. 1, no. 3, pp. 82–87.
21. **Qian W., Zhou A. Y.** Analyzing popular clustering algorithms from different viewpoints. *Journal of Software*, 2002, vol. 13, no. 8, pp. 1382–1394.
22. **Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И.** *Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAPM*. СПб., БХВ-Петербург, 2007. 384 с.
23. **Bindra K.,** et al. Effective Data Clustering Algorithms. *Soft Computing: Theories and Applications: Proc. of SoCTA 2017*, Singapore, Springer, 2019, pp. 419–432.
24. **Zhang T., Ramakrishnan R., Livny M.** BIRCH: an efficient data clustering method for very large databases. *ACM Sigmod Record*, ACM, 1996, vol. 25, no. 2, pp. 103–114.
25. **Uppada S. K.** Centroid based clustering algorithms-A clarion study. *International Journal of Computer Science and Information Technologies*, 2014, vol. 5, no. 6, pp. 7309–7313.
26. **Shah H., Napanda K., D'mello L.** Density based clustering algorithms. *International Journal of Computer Sciences and Engineering*, 2015, vol. 3, no. 11, pp. 54–57.
27. **Kerdels J., Peters G.** A sparse representation of high-dimensional input spaces based on an augmented growing neural gas. *GCAI*, 2016, pp. 303–313.
28. **Fritzke B.** A growing neural gas network learns topologies. *Proc. of the Conf. "Advances in neural information processing systems"*, 1995, pp. 625–632.
29. **Martinetz T., Schulten K.** Topology representing networks. *Neural Networks*, 1994, vol. 7, no. 3, pp. 507–522.
30. **Kohonen T.** Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 1982, vol. 43, no. 1, pp. 59–69.
31. **Fišer D., Faigl J., Kulich M.** Growing neural gas efficiently. *Neurocomputing*, 2013, vol. 104, pp. 72–82.
32. **Fritzke B.** Growing cell structures — a self-organizing network in *k* dimensions. *Proc. of the 1992 Intern. Conf. on Artificial Neural Networks (ICANN–92)*, Brighton, UK, 4–7 September, 1992, 1992, pp. 1051–1056.
33. **Fritzke B.** Growing grid — a self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 1995, vol. 2, no. 5, pp. 9–13.
34. **Ultsch A.** Self organized feature maps for monitoring and knowledge aquisition of a chemical process. *ICANN'93*, Springer, London, 1993, pp. 864–867.
35. **Kholod I., Kuprianov M., Petukhov I.** Distributed data mining based on actors for internet of things. *5th Mediterranean Conf. on Embedded Computing (MECO)*, June 12–16, 2016, Bar, Montenegro, 2016, pp. 480–484. doi:10.1109/MECO.2016.7525698
36. **Kholod I., Petuhov I., Malyshkin V.** Creation of data mining algorithms as functional expression for parallel and distributed execution. *Parallel Computing Technologies, LNCS*, Springer, 2015, vol. 9251, pp. 62–68.
37. **Kholod I., Kupriyanov M., Shorov A.** Decomposition of data mining algorithms into unified functional blocks. *Mathematical Problems in Engineering*, 2016, vol. 2016, p. 11.

UDC 004.75
doi:10.31799/1684-8853-2019-2-35-43

Data clustering in distributed monitoring systems

A. N. Rukavitsyn[a], Post-Graduate Student, orcid.org/0000-0001-5382-0465, rkvtsn@gmail.com
[a]Saint-Petersburg Electrotechnical University «LETI», 5, Prof. Popov St., 197376, Saint-Petersburg, Russian Federation

Introduction: The traditional methods of analyzing distributed data sources typically use centralized data warehouses and have several disadvantages associated with confidentiality, a high cost of centralized data storage, a limited bandwidth and a high load on telecommunications networks. Techniques for a decentralized analysis do not take into account the data distribution type and the features of the selected algorithm. This reduces the performance and accuracy, or can be impracticable under certain conditions. **Purpose:** Studying and analyzing the features of distributed monitoring systems and data mining algorithms. **Results:** For clustering based on distributed data sources, requirements were set for an algorithm in distributed monitoring systems: single pass, support of different types of input data, online operation and adaptation to data when the environment changes, large data scaling, analysis without assumptions about the input data distribution, analysis of data on information sources without involving a third party. Two main ways of data distribution on sources in heterogeneous systems are defined: vertical and horizontal ones. The methods are classified according to their basic principle of cluster delimitation. The classification includes the main clustering algorithms, their operation principles, advantages and disadvantages. A review of the existing clustering methods has shown that in distributed monitoring systems, Kohonen's neural networks are most efficient. The algorithm of Kohonen self-organizing maps was decomposed, and blocks for data processing were determined: the calculation of a winner neuron and the adjustment of neuron weights. Two strategies have been proposed for clustering distributed data. **Practical relevance:** The proposed strategies allow you to perform clustering in systems with distributed sources taking into account the characteristics of the environment, without transferring all the data.

Keywords — data mining, clustering, sensors, distributed data sources, Kohonen self-organizing maps, distributed data processing systems, types of distribution, monitoring systems.

For citation: Rukavitsyn A. N. Data clustering in distributed monitoring systems. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 2, pp. 35–43 (In Russian). doi:10.31799/1684-8853-2019-2-35-43

References

1. Catarinucci L., et al. An IoT-aware architecture for smart healthcare systems. *IEEE Internet of Things Journal*, 2015, vol. 2, no. 6, pp. 515–526.
2. Andreu-Perez J., et al. From wearable sensors to smart implants — toward pervasive and personalized healthcare. *IEEE Transactions on Biomedical Engineering*, 2015, vol. 62, no. 12, pp. 2750–2762.
3. Del Vecchio P., et al. Creating value from social big data: Implications for smart tourism destinations. *Information Processing & Management*, 2018, vol. 54, no. 5, pp. 847–860.
4. Bendechache M., Kechadi M. T., Le-Khac N. A. Efficient large scale clustering based on data partitioning. *Data Science and Advanced Analytics (DSAA), 2016 IEEE Intern. Conf.*, 2016, pp. 612–621.
5. Aouad L. M., Le-Khac N. A., Kechadi T. M. Lightweight clustering technique for distributed data mining applications. *Industrial Conf. on Data Mining*, Springer, Berlin, Heidelberg, 2007, pp. 120–134.
6. Le-Khac N. A., Aouad L. M., Kechadi M. T. A new approach for distributed density based clustering on grid platform. *British National Conf. on Databases*, Springer, Berlin, Heidelberg, 2007, pp. 247–258.
7. Januzaj E., Kriegel H. P., Pfeifle M. DBDC: Density based distributed clustering. *Intern. Conf. on Extending Database Technology*, Springer, Berlin, Heidelberg, 2004, pp. 88–105.
8. Gorgonio F. L., Costa J. A. F. Parallel self-organizing maps with application in clustering distributed data. *IEEE Intern. Joint Conf. Neural Networks (IJCNN 2008)*, 2008, pp. 3276–3283.
9. Kim M., Jung S., Park M. A distributed self-organizing map for DoS attack detection. *Seventh Intern. Conf. Ubiquitous and Future Networks (ICUFN)*, 2015, pp. 19–22.
10. Ali A. A., et al. Distributed data mining systems: Techniques, approaches and algorithms. *22nd Intern. Conf. on Circuits, Systems, Communications and Computers (CSCC 2018)*, Majorca, Spain, July 14–17, 2018, vol. 210, p. 04038.
11. Song C., et al. Updating strategy of master-slave data consistency based on message. *Computer Engineering*, 2004, vol. 1, pp. 035.
12. Robert H. A majority consensus approach to concurrency control for multiple copy databases. *ACM Trans. on Database System*, 2003, vol. 4, no. 2, pp. 543–549.
13. Moro G., Sartori C. Incremental maintenance of multi-source views. *Proc. of the 12th Australasian Database Conf.*, IEEE Computer Society, 2001, pp. 13–20.
14. Oussous A., et al. Big data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 2018, vol. 30, no. 4, pp. 431–448.
15. Li T., et al. Differentially private naive bayes learning over multiple data sources. *Information Sciences*, 2018, vol. 444, pp. 89–104.
16. Assunção M. D., et al. Big data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, 2015, vol. 79, pp. 3–15.
17. Duda R. O., Hart P. E., Stork D. G. *Pattern classification*. John Wiley & Sons, 2012. 688 p.
18. Bendechache M., Kechadi M. T. Distributed clustering algorithm for spatial data mining. *2nd IEEE Intern. Conf. on Spatial Data Mining and Geographical Knowledge Services (ICSDM)*, IEEE, 2015, pp. 60–65.
19. Saraswathi S., Sheela M. I. A comparative study of various clustering algorithms in data mining. *International Journal of Computer Science and Mobile Computing*, 2014, vol. 11, no. 11, pp. 422–428.
20. Sisodia D., et al. Clustering techniques: a brief survey of different clustering algorithms. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 2012, vol. 1, no. 3, pp. 82–87.
21. Qian W., Zhou A. Y. Analyzing popular clustering algorithms from different viewpoints. *Journal of Software*, 2002, vol. 13, no. 8, pp. 1382–1394.
22. Barsegyan A. A., Kupriyanov M. S., Stepanenko V. V., Kholod I. I. *Tekhnologii analiza dannykh. Data Mining, Visual Mining, Text Mining, OLAP* [Technologies of data analysis. Data Mining, Visual Mining, Text Mining, OLAP]. Saint-Petersburg, BKhV-Peterburg Publ., 2007. 384 p. (In Russian).
23. Bindra K., et al. Effective Data Clustering Algorithms. *Soft Computing: Theories and Applications (SoCTA 2017)*, Springer, Singapore, 2019, pp. 419–432.
24. Zhang T., Ramakrishnan R., Livny M. BIRCH: an efficient data clustering method for very large databases. *ACM Sigmod Record*, ACM, 1996, vol. 25, no. 2, pp. 103–114.
25. Uppada S. K. Centroid based clustering algorithms-A clarion study. *International Journal of Computer Science and Information Technologies*, 2014, vol. 5, no. 6, pp. 7309–7313.
26. Shah H., Napanda K., D'mello L. Density based clustering algorithms. *International Journal of Computer Sciences and Engineering*, 2015, vol. 3, no. 11, pp. 54–57.

27. Kerdels J., Peters G. A sparse representation of high-dimensional input spaces based on an augmented growing neural gas. *GCAI*, 2016, pp. 303–313.
28. Fritzke B. A growing neural gas network learns topologies. *Proc. of the Conf. "Advances in neural information processing systems"*, 1995, pp. 625–632.
29. Martinetz T., Schulten K. Topology representing networks. *Neural Networks*, 1994, vol. 7, no. 3, pp. 507–522.
30. Kohonen T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 1982, vol. 43, no. 1, pp. 59–69.
31. Fišer D., Faigl J., Kulich M. Growing neural gas efficiently. *Neurocomputing*, 2013, vol. 104, pp. 72–82.
32. Fritzke B. Growing cell structures — a self-organizing network in *k* dimensions. *Proc. of the 1992 Intern. Conf. on Artificial Neural Networks (ICANN–92)*, Brighton, UK, 4–7 September, 1992, 1992, pp. 1051–1056.
33. Fritzke B. Growing grid — a self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 1995, vol. 2, no. 5, pp. 9–13.
34. Ultsch A. Self organized feature maps for monitoring and knowledge aquisition of a chemical process. *ICANN'93*, Springer, London, 1993, pp. 864–867.
35. Kholod I., Kuprianov M., Petukhov I. Distributed data mining based on actors for internet of things. *5th Mediterranean Conf. on Embedded Computing (MECO)*, June 12–16, 2016, Bar, Montenegro, 2016, pp. 480–484. doi:10.1109/MECO.2016.7525698
36. Kholod I., Petuhov I., Malyshkin V. Creation of data mining algorithms as functional expression for parallel and distributed execution. *Parallel Computing Technologies, LNCS*, Springer, 2015, vol. 9251, pp. 62–68.
37. Kholod I., Kupriyanov M., Shorov A. Decomposition of data mining algorithms into unified functional blocks. *Mathematical Problems in Engineering*, 2016, vol. 2016, p. 11.