Доклад о работах Hans-Peter Kriegel

Алгоритм DBSCAN и обнаружение «взрывов» в потоках новостей

Севастопольский Артём

ВМК МГУ

7 октября, 2015

Краткие сведения и научные достижения

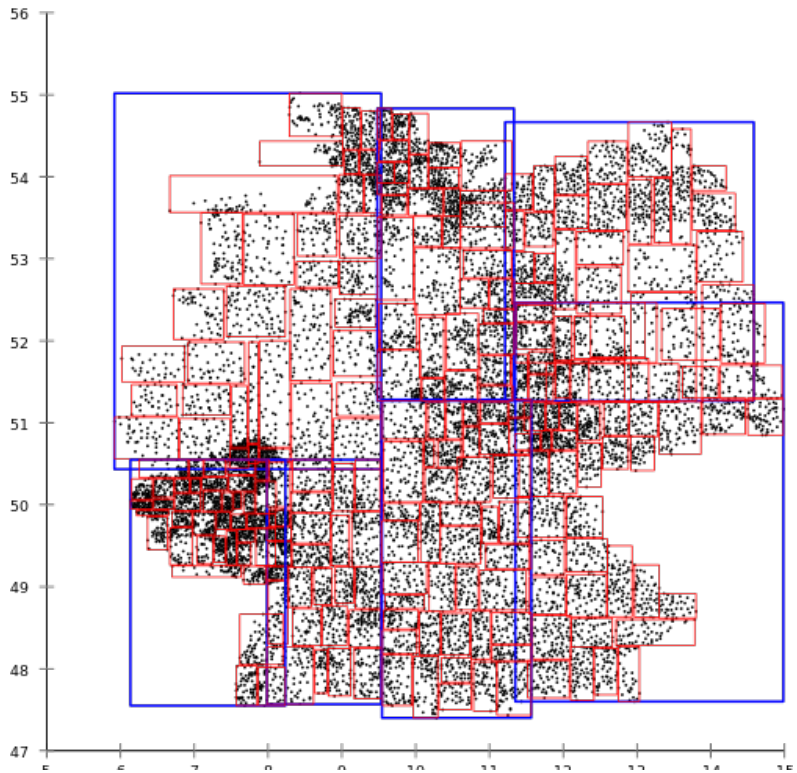
- *66 лет*
- Страна: *Германия*
- Профессор информатики в *Ludwig-Maximilians-Universität (LMU), г. Мюнхен.*

Hans-Peter Kriegel

LMU

Краткая биография и научные достижения

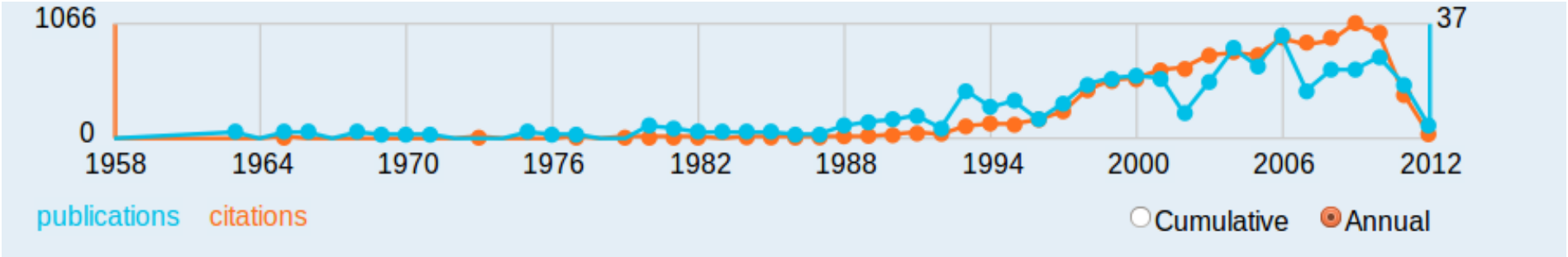
- Область интересов:
  - Кластеризация
  - Обнаружение аномалий в данных
  - Извлечение знаний из баз данных
  - Метрические методы
  - Работа с пространственными данными (базами данных местоположений объектов)
- Наиболее известен за *R*-tree*, *DBSCAN*, *OPTICS*, *LOF*.



R*-дерево

Краткая биография и научные достижения

- Премии:
  - 2009 — ACM Fellow
  - 2013 — IEEE ICDM Research Contributions Award
  - 2014 — ACM SIGKDD Test of time award
  - 2015 — ACM SIGKDD Innovation Award — считается высшей наградой в области Data Mining
- Более 450 публикаций, процитированы 35000 раз
- В начале 2015 г. – 5 место в рейтинге ученых по data mining по версии Microsoft.



DBSCAN

A Density-Based Algorithm for Discovering Clusters
A Density-Based Algorithm for Discovering Clusters
in Large Spatial Databases with Noise

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu

Institute for Computer Science, University of Munich
Oettingenstr. 67, D-80538 München, Germany
{ester | kriegel | sander | xwxu}@informatik.uni-muenchen.de

Abstract

Clustering algorithms are attractive for the task of class identification in spatial databases. However, the application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to determine the input parameters, discovery of clusters with arbitrary shape and good efficiency on large databases. The well-known clustering algorithms offer no solution to the combination of these requirements. In this paper, we present the new clustering algorithm DBSCAN relying on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. DBSCAN requires only one input parameter and supports the user in determining an appropriate value for it. We performed an experimental evaluation of the effectiveness and efficiency of DBSCAN using synthetic data and real data of the SEQUOIA 2000 benchmark. The results of our experiments demonstrate that (1) DBSCAN is significantly more effective in discovering clusters of arbitrary shape than the well-known algorithm CLARANS, and that (2) DBSCAN outperforms CLARANS by a factor of more than 100 in terms of efficiency.

Keywords: Clustering Algorithms, Arbitrary Shape of Clusters, Efficiency on Large Spatial Databases, Handling Nlj4-275oise.

1. Introduction

Numerous applications require the management of *spatial* data, i.e. data related to space. *Spatial Database Systems (SDBS)* (Gueting 1994) are database systems for the management of spatial data. Increasingly large amounts of data are obtained from satellite images, X-ray crystallography or other automatic equipment. Therefore, automated knowledge discovery becomes more and more important in spatial databases.

Several tasks of *knowledge discovery in databases* (KDD) are often not known in advance when dealing with large databases.

(2) Discovery of clusters with arbitrary shape, because the shape of clusters in spatial databases may be spherical, drawn-out, linear, elongated etc.

(3) Good efficiency on large databases, i.e. on databases of significantly more than just a few thousand objects.

The well-known clustering algorithms offer no solution to the combination of these requirements. In this paper, we present the new clustering algorithm DBSCAN. It requires only one input parameter and supports the user in determining an appropriate value for it. It discovers clusters of arbitrary shape. Finally, DBSCAN is efficient even for large spatial databases. The rest of the paper is organized as follows. We discuss clustering algorithms in section 2 evaluating them according to the above requirements. In section 3, we present our notion of clusters which is based on the concept of density in the database. Section 4 introduces the algorithm DBSCAN which discovers such clusters in a spatial database. In section 5, we performed an experimental evaluation of the effectiveness and efficiency of DBSCAN using synthetic data and data of the SEQUOIA 2000 benchmark. Section 6 concludes with a summary and some directions for future research.

2. Clustering Algorithms

There are two basic types of clustering algorithms (Kaufman & Rousseeuw 1990): partitioning and hierarchical algorithms. *Partitioning algorithms* construct a partition of a database D of n objects into a set of k clusters. k is an input parameter for these algorithms, i.e some domain knowledge is required which unfortunately is not available for many ap-

A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise
Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu

- 1996 г., конференция KDD
- 7000 цитирований
- Премия 2014 SIGKDD Test of Time
- Алгоритм положил начало другим алгоритмам (OPTICS, LOF)

DBSCAN

- DBSCAN (Density Based Spatial Clustering of Applications with Noise)
- Алгоритм кластеризации, разделяющий объекты по их **плотности** – густоте распределения по пространству.
- Хорошо подходит для пространственных данных.
- Применим к базам данных координат объектов в 2D и 3D.

3. A Density Based Notion of Clusters

When looking at the sample sets of points depicted in figure 1, we can easily and unambiguously detect clusters of points and noise points not belonging to any of those clusters.

database 1 database 2 database 3

figure 1: Sample databases

DBSCAN. Определения

Пусть $D = \{(x_i, y_i)\}_{i=1}^{n}$ – база данных точек в 2D евклидовом пространстве, $dist(p, q), p, q \in D$ – функция расстояния.

Определение (ε-окрестность точки)

ε-окрестность точки $N_\varepsilon(p)$ – это множество точек, удаленных от p не более чем на ε:

$$N_\varepsilon(p) = \{q \in D \mid dist(p, q) \leq \varepsilon\}$$

Определение (достижимость напрямую)

Точка p **напрямую достижима** из q, если:

1. $p \in N_\varepsilon(q)$
2. $|N_\varepsilon(q)| \geq MinPts, MinPts \in \mathbb{N}$

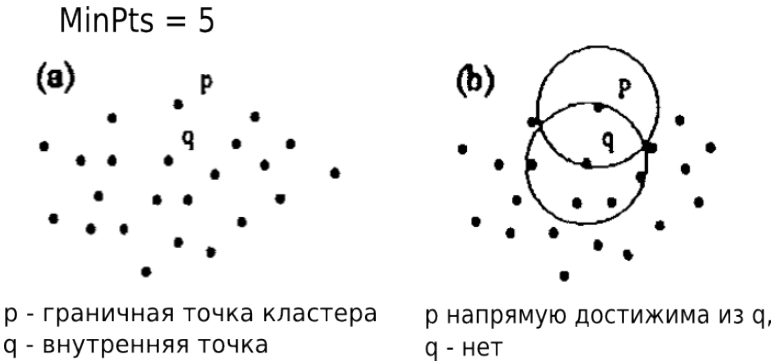


Рис. 1 : Отношение *достижимости напрямую* не симметрично.

Далее считаем ε и $MinPts$ заданными параметрами алгоритма.

DBSCAN. Определения

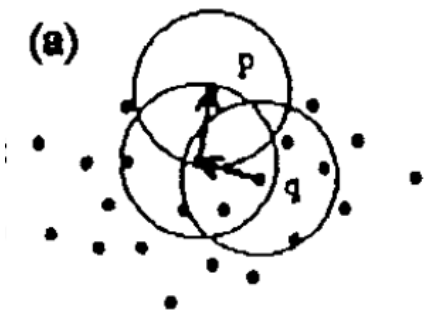
Далее считаем ε и *MinPts* заданными параметрами алгоритма.

Определение (достижимость)

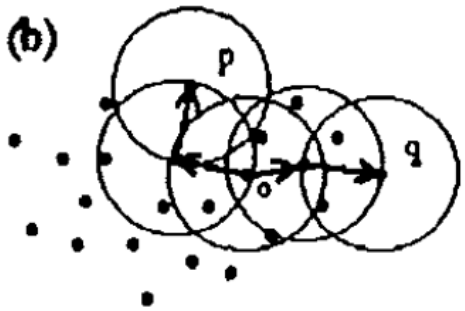
Точка p **достижима** из точки q, если $\exists$ последовательность точек $p_1, \ldots, p_n$, такая что p_{i+1} напрямую достижима из p_i.

Определение (соединённость)

Точка p **соединена** с точкой q, если $\exists$ такая точка z, что p и q достижимы из z.



p достижима из q,
но q не достижима из p



p и q соединены (за счет точки o)

DBSCAN. Определения

Определение (кластер)

Кластером будем считать непустое множество точек из базы данных D, удовлетворяющих следующим требованиям:

1. $\forall p \in C$: если q достижима из p, то $q \in C$
2. $\forall p, q \in C$, p и q соединены.

Шумом считаем всё, что не принадлежит ни одному кластеру.

DBSCAN. Основные леммы

Лемма

Пусть $p \in D$ и $N_\varepsilon(p) \geq MinPts$. ($p$ — внутренняя точка). Тогда множество всех точек, достижимых из p, — это кластер.

Лемма

Пусть C — кластер, а p — внутренняя точка C. Тогда C равно множеству всех точек, достижимых из p.

Данные леммы дают возможность составить алгоритм поиска кластеров.

DBSCAN. Алгоритм

DBSCAN. Параметры алгоритма: ε и *MinPts*.

```
DBSCAN(SetOfPoints, Eps, MinPts) {
     для каждой точки p из SetOfPoints:
         если точка p еще не классифицирована,
             ExpandCluster(p, SetOfPoints, Eps, MinPts);
}
ExpandCluster(Point, SetOfPoints, Eps, MinPts) {
     seeds := SetOfPoints.regionQuery(Point, Eps);
     если seeds.size < MinPts:
         помечаем точку Point как шум
     иначе:
         помечаем все точки из seeds частью нового кластера;
         пока seeds не пусто:
             вынимаем первую точку из seeds
             result := SetOfPoints.regionQuery(Point, Eps);
             для всех точек из result, которые не классифицированы или помечены как шум,
добавляем их в seeds.
             помечаем все точки из result частью нового кластера.
}
```

- Операция SetOfPoints.regionQuery(Point, Eps) — найти все точки в Eps-окрестности точки Point:
- если реализована полным перебором, DBSCAN работает за $O(n^2)$
- если реализована с помощью R*-tree, DBSCAN работает за $O(n \log n)$.

DBSCAN. Как подбирать ε и *MinPts*?

- Авторы предлагают способ оценить ε и *MinPts*.
- Пусть $dist_k(p), p \in D$ — это расстояние от p до её k-го ближайшего соседа. Выберем некоторое $k \in \mathbb{N}$.
  1. Сортируем все точки $p \in D$ по убыванию $dist_k(p)$.
  2. Строим график $dist_k(p)$ для всех p в отсортированном порядке.
  3. На графике можно видеть точку, в которой начинается первое плато. Пусть абсцисса этой точки – q, тогда установим $\varepsilon := dist_k(q), MinPts := k$.

- Авторы утверждают, что при $k > 4$ график мало отличается от графика при $k = 4$.

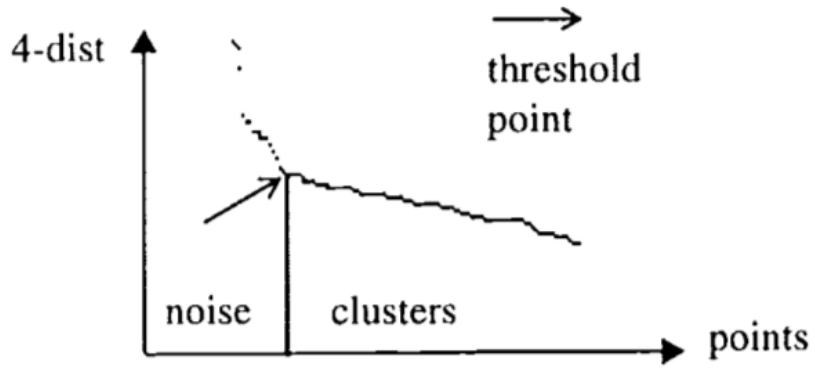


figure 4: sorted 4-dist graph for sample database 3

DBSCAN. Ссылки на литературу

1. Beckmann N., Kriegel H.-P., Schneider R, and Seeger B. 1990. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles, Proc. ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, NJ, 1990, pp. 322-331.
2. Brinkhoff T., Kriegel H.-R, Schneider R., and Seeger B. 1994 Efficient Multi-Step Processing of Spatial Joins, Proc. ACM SIGMOD Int. Conf. on Management of Data, Minneapolis, MN, 1994, pp. 197-208.
3. Ester M., Kriegel H.-P., and Xu X. 1995. A Database Interface for Clustering in Large Spatial Databases, Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining, Montreal, Canada, 1995, AAAI Press, 1995.
4. Garcia J.A., Fdez-Valdivia J., Cortijo E J., and Molina R. 1994. A Dynamic Approach for Clustering Data. Signal Processing, Vol. 44, No. 2, 1994, pp. 181-196.
5. Gueting R.H. 1994. An Introduction to Spatial Database Systems. The VLDB Journal 3(4): 357-399.
6. Jain Anil K. 1988. Algorithms for Clustering Data. Prentice Hall.
7. Kaufman L., and Rousseeuw RJ. 1990. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley and Sons.
8. Matheus C.J.; Chan P.K.; and Piatetsky-Shapiro G. 1993. Systems for Knowledge Discovery in Databases, 1EEE Transactions on Knowledge and Data Engineering 5(6): 903-913.
9. Ng R.T., and Han J. 1994. Efficient and Effective Clustering Methods for Spatial Data Mining, Proc. 20th Int. Conf. on Very Large Data Bases, 144-155. Santiago, Chile.
10. Stonebraker M., Frew J., Gardels K., and Meredith J.1993. The SEQUOIA 2000 Storage Benchmark, Proc. ACM SIGMOD Int. Conf. on Management of Data, Washington, DC, 1993, pp. 2-11.

«Взрывы» в новостях

Discovering Global and Local Bursts in a Stream of News

Max Zimmermann
School of Computer Science
Otto-von-Guericke-University
of Magdeburg,Germany
max.zimmermann@iti.cs.uni-magdeburg.de

Irene Ntoutsi
Institute for Informatics
Ludwig-Maximilians-University
of Munich,Germany
ntoutsi@dbs.ifi.lmu.de

Zaigham Faraz Siddiqui
School of Computer Science
Otto-von-Guericke-University
of Magdeburg,Germany
siddiqui@iti.cs.uni-magdeburg.de

Myra Spiliopoulou
School of Computer Science
Otto-von-Guericke-University
of Magdeburg,Germany
myra@iti.cs.uni-magdeburg.de

Hans-Peter Kriegel
Institute for Informatics
Ludwig-Maximilians-University
of Munich,Germany
kriegel@dbs.ifi.lmu.de

ABSTRACT

Reports on major events like hurricanes and earthquakes, and major topics like the financial crisis or the Egyptian revolution appear in Internet news and become (ir)regularly updated, as new insights are acquired. Tracking emerging subtopics in a major or even local event is important for the news readers but challenging for the operator: subtopics may emerge gradually or in a bursty way; they may be of some importance inside the event, but too rare to be visible inside the whole stream of news. In this study, we propose a text stream clustering method that detects, tracks and updates large and small bursts of news in a two-level topic hierarchy. We report on our first results on a stream of news from February to April 2011.

1. INTRODUCTION

People resort increasingly on Internet sources to acquire up to date information about events of global or local importance. News providers care to perform frequent updates of the arriving information, while users of social platforms experience bursts of postings in association with major and rapidly evolving events. Delivering insights on the semantics of bursty events is a major challenge. In this study, we provide a two-level approach for the detection of small and major bursts in a fast stream of news.

There are first promising results on learning bursts in streams of news [8], and there is substantial research on the discovery of emerging and evolving topics, see e.g.[10, 4]. However, these approaches fall short of capturing novelty as a *local change* in non-major events: bursts are not only major events that draw everybody's attention; there are also events of local importance, occurring inside a part of a stream, which refers to a bursty or conventional topic. This leads to the challenge of simultaneously learning both *global* and *local bursts* in a stream.

Our approach deals with this challenge as follows: We consider a two-level hierarchy of topics over the text stream. Topics of the 1st level are global ones; global bursts are captured at this level. Topics of the 2nd level (we also call them "subtopics") are local ones; they capture events of local importance inside a 1st level topic, including local bursts. Subtopics are modeled inside a topic, using a feature space that is particular to the topic. For example, the keyword "Japan" may be used to distinguish the global event of the Japan quake from other global events, but may be ignored inside this event, because it does not contribute to distinguishing among its subtopics (like the tsunami, the Fukushima disaster, the international aid etc).

To detect emerging topics of the 1st or 2nd level, we pick arriving documents that fit to no topic or fit to some topic but to no subtopic inside it: we do not force these documents into a cluster, but rather retain them in a "container". We maintain a single global container at the 1st level, and one local container for each 1st level topic. When a local container is overfilled, we adjust the feature space and perform re-clustering - but we only consider the documents in the container and in the other subtopics of this 1st level topic. This allows us to identify bursts local to a topic, and adjust the model to them without re-computing other topics. Only documents that do not fit to any 1st level topic and thus flood the global container may lead to global re-clustering.

The rest of the paper is organized as follows. Related work is found in Section 2. Our method is presented in Section 3. Section 4 discusses our experimental results. Conclusions and outlook are presented in Section 5.

2. RELATED WORK

The discovery of emerging topics in a stream of news is studied in the context of *text stream clustering* and of *dynamic topic modeling*. Also, there are methods focussing on the detection of bursts in news posted in social platforms.

The *text stream clustering* algorithm of Aggarwal and Yu maintains a fixed number of K clusters/topics over time [1]. If a new document is too far from all existing clusters, it

SAC'12 March 25-29, 2012, Riva del Garda, Italy.

Discovering Global and Local Bursts in a Stream of News
Max Zimmermann, Irene Ntoutsi, Zaigham Faraz Siddiqui, Myra Spiliopoulou, Hans-Peter Kriegel

- 2012 г.
- 27th Annual ACM Symposium on Applied Computing

«Взрывы» в новостях. Постановка задачи

- Пусть мы наблюдаем поток новостных данных. В моменты времени $t_0, t_1, \ldots, t_i, \ldots$ получаем коллекцию документов (новостей) $D = D(t_i)$.
- Основные цели:
  1. классифицировать документы по темам,
  2. обнаруживать «**взрывы**» в потоке новостей – новые темы, получившие большой резонанс.

«Взрывы» в новостях. Постановка задачи

- Пусть мы наблюдаем поток новостных данных. В моменты времени $t_0, t_1, \ldots, t_i, \ldots$ получаем коллекцию документов (новостей) $D = D(t_i)$.
- Основные цели:
  1. классифицировать документы по темам,
  2. обнаруживать «**взрывы**» в потоке новостей – новые темы, получившие большой резонанс.
- Авторы поставили задачу не просто обнаруживать взрывы, а понимать, когда произошел **глобальный** взрыв, а когда **локальный**. (когда появилась новая крупная тема или подтема крупной)

Table 1: Description of the News Threads

Topic	*keywords e.g.*	Duration	Docs/day	Total
Egypt	*protests, violence*	25/01-04/04	≈ 250	17,067
Libya	*rebels, revolution*	04/02-04/04	≈ 250	12,421
Japan	*earthquake, tsunami*	11/03-02/04	≈ 300	6,812
Cricket	*worldcup 2011, cricket*	16/02-04/04	≈ 100	4,208
				41,340

«Взрывы» в новостях. Схема решения

- Всегда храним на компьютере часть свежих документов (новостей). Поддерживаем 2-уровневую иерархию последних документов.
- Новый документ сравнивается на похожесть с темами. Если не похож ни на одну тему, попадает в *глобальный контейнер*. Иначе попадает в наиболее близкую тему.
- Внутри подтемы – тот же процесс (рекурсивно).

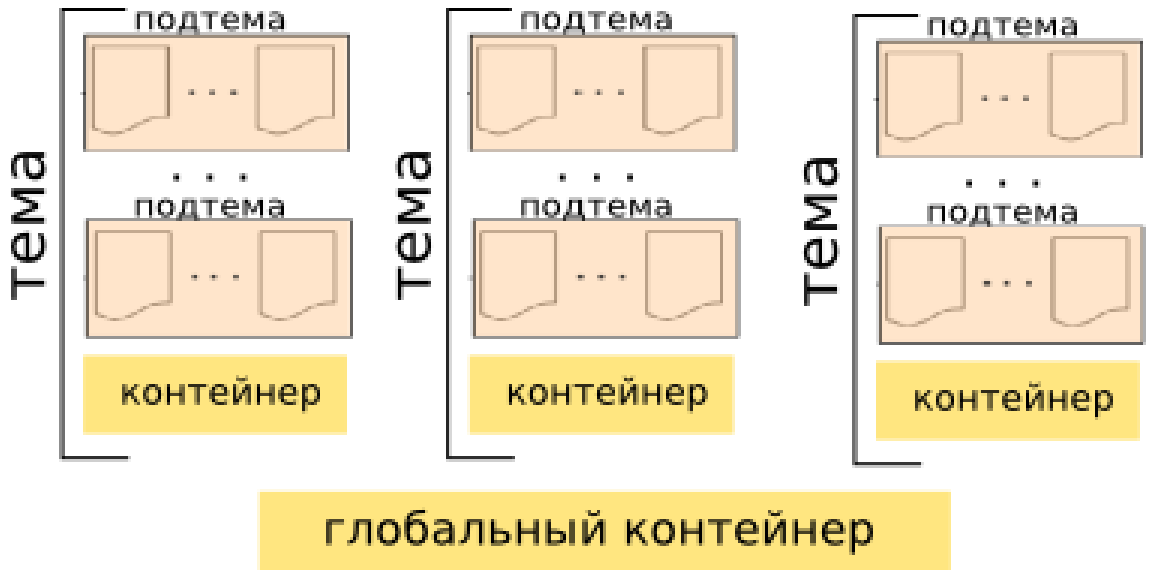


Рис. 2 : Организация сохраненных документов

«Взрывы» в новостях. Схема решения

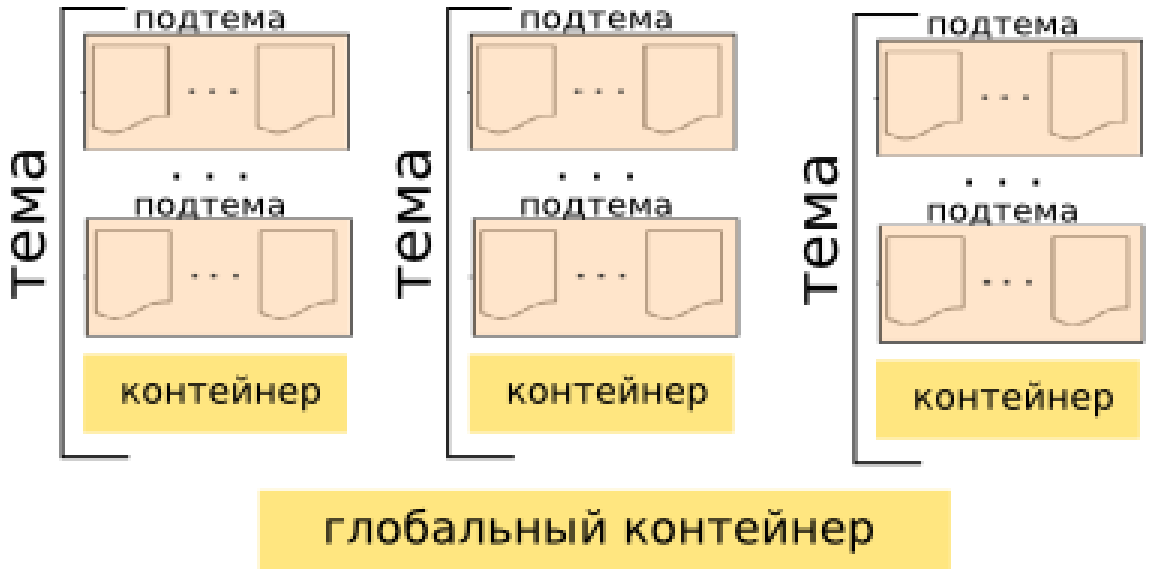


Рис. 3 : Организация сохраненных документов

- Если глобальный контейнер переполняется (его размер становится больше σ_g), производим полную перекластеризацию хранилища.
- Если контейнер темы переполняется (его размер становится больше σ_l), производим перекластеризацию внутри темы.

«Взрывы» в новостях. Более подробно

1. Каждый документ d из темы C переводится в TF-IDF представление:

$$\mathrm{tf}(t,d) = \frac{n_i}{\sum_k n_k};$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|(d_i \supset t_i)|};$$

$$\text{tf-idf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

2. TF-IDF для всей темы C – это вектор $C' = (\omega_1, \omega_2, \ldots, \omega_N)$, где ω_i – средний TF-IDF-вес слова k_i в документах из C.
3. Когда поступает очередной документ d, его TF-IDF сравнивается с TF-IDF каждой темы C_i и берется косинусная мера сходства. Документ попадает в контейнер, если

$$\max_{C_i} cosine(C_i', d) < \delta_g,$$

где δ_g – порог сходства.

«Взрывы» в новостях. Более подробно

1. Когда глобальный контейнер заполняется,
   1. оставляем в хранилище только содержимое глобального контейнера и последние W документов в потоке,
   2. для данного множества документов проводим перекластеризацию.
   3. получаем новые темы.
2. Перекластеризация производится алгоритмом *Fuzzy C-Means* (Bezdek, 1981).

«Взрывы» в новостях. Fuzzy c-means

- Fuzzy c-means основан на нечеткой логике. Каждой точке x_i сопоставляет не кластер, а вероятность попадания в каждый кластер C_k – w_{ik}.
  1. Выбирается число кластеров c.
  2. Случайно инициализируются все вероятности.
  3. Пока алгоритм не достиг сходимости:
     - Пересчитываются центроиды по формуле:

$$c_k = \frac{\sum_{i=1}^{n} {w_{ik}}^m x_i}{\sum_i {w_{ik}}^m} \qquad k = 1, \ldots, c$$

     - Пересчитываются вероятности принадлежности точек классам:

$$w_{ij}^m = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|\mathbf{x}_i - \mathbf{c}_j\|}{\|\mathbf{x}_i - \mathbf{c}_k\|} \right)^{\frac{2}{m-1}}}, \qquad i = 1, \ldots, N, \qquad j = 1, \ldots, c$$

- В процессе работы минимизируется целевая функция $J(X, C) = \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij}^m \left\| \mathbf{x}_i - \mathbf{c}_j \right\|^2$.
- m – «размыватель» (fuzzifier) – параметр алгоритма, отвечающий за четкость границ кластеров. Обычно выбирают $m = 2$.

«Взрывы» в новостях. Эксперименты

- BBC, поиск по ключевым словам

Table 1: Description of the News Threads

Topic	*keywords e.g.*	Duration	Docs/day	Total
Egypt	*protests, violence*	25/01-04/04	≈ 250	17,067
Libya	*rebels, revolution*	04/02-04/04	≈ 250	12,421
Japan	*earthquake, tsunami*	11/03-02/04	≈ 300	6,812
Cricket	*worldcup 2011, cricket*	16/02-04/04	≈ 100	4,208
				41,340

- Авторы утверждают, что данные содержат много шума (особенно Египет и Ливия).

«Взрывы» в новостях. Эксперименты

- Чем больше порог локального сходства δ_l, тем больше производится локальных перекластеризаций и меньше глобальных.

Table 3: Number of global and local reclusterings for different local similarity thresholds δ

global δ	local δ	# global reclusterings	# local reclusterings
0.2	0.3	9	10
0.2	0.5	5	37
0.2	0.8	3	55

Рис. 4 : Влияние δ_l на число локальных перекластеризаций

«Взрывы» в новостях. Эксперименты

- Чем больше порог локального сходства δ_l, тем больше производится локальных перекластеризаций и меньше глобальных.
- Из-за этого глобальный контейнер переполняется реже, и замечается меньше ложных взрывов.

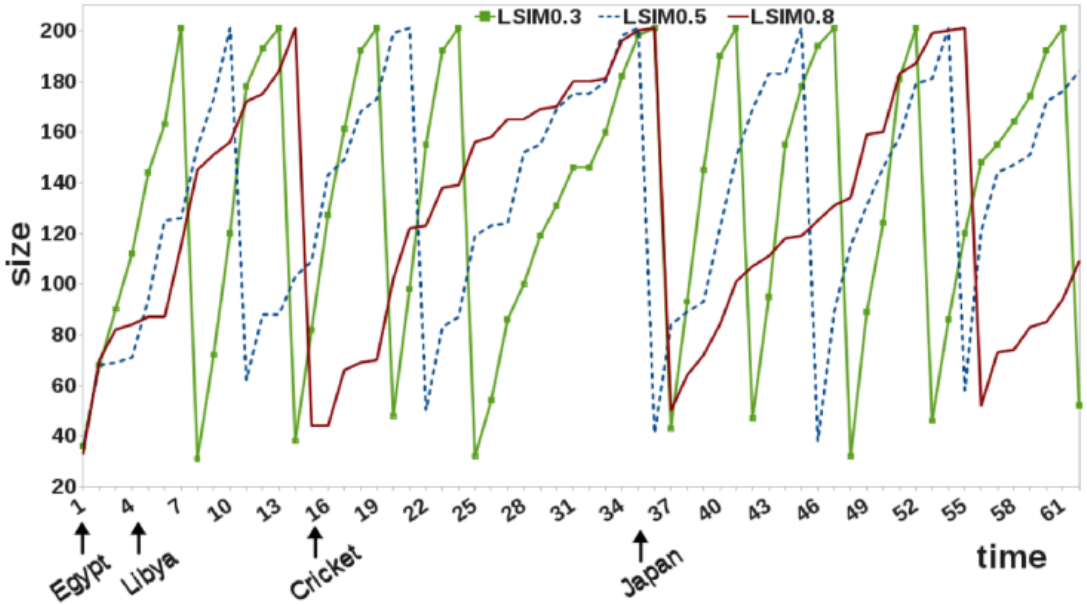


Рис. 5 : Влияние δ_l на размер глобального контейнера. Зависимость размера глобального контейнера от времени (K=5, $\delta_g = 200$, $\delta_l = 100$)

«Взрывы» в новостях. Эксперименты

- Можно заметить аналогичный эффект с качеством работы алгоритма.
- В качестве меры качества авторы статьи выбрали *purity*. Возьмем для каждой темы C_i истинный класс Y, которого среди её документов большинство. Тогда $purity_i = \frac{\#(d_i \in Y)}{\#(d_i \in C_i)}$, $purity = \frac{1}{c}\sum_{i=1}^{c} purity_i$.

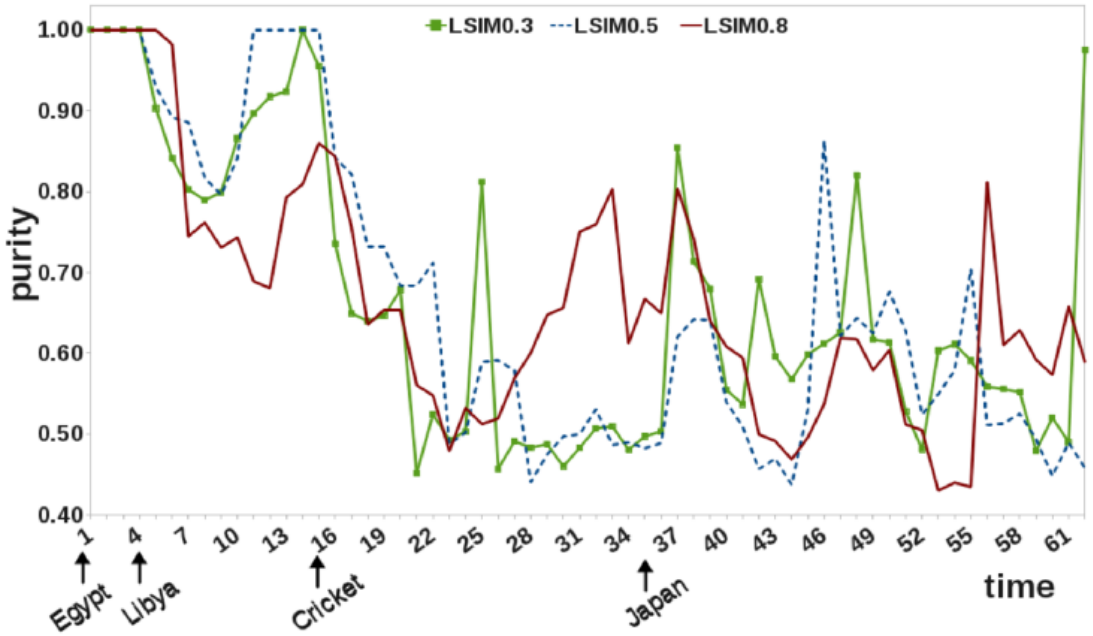


Рис. 6 : Влияние δ_l на среднюю чистоту глобальных тем. Зависимость средней частоты кластеров (purity) от времени (K=5, $\delta_g = 200$, $\delta_l = 100$)

«Взрывы» в новостях. Ссылки на литературу

1. C. C. Aggarwal and P. S. Yu. A framework for clustering massive text and categorical data streams. In SDM, 2006.
2. L. AlSumait, D. Barbara, and C. Domeniconi. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In ICDM, 2008.
3. A. Bifet and E. Frank. Sentiment knowledge discovery in Twitter streaming data. In DS, 2010.
4. D. M. Blei and J. D. Lafferty. Dynamic topic models. In ICML, 2006.
5. C. Borgelt and A. Nurnberger. Document clustering using cluster specific term weights. In TIR, 2004.
6. Y. Chi, X. Song, D. Zhou, K. Hino, and B. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In KDD, 2007.
7. A. Gohr, A. Hinneburg, R. Schult, and M. Spiliopoulou. Topic evolution in a stream of documents. In SDM, 2009.

«Взрывы» в новостях. Ссылки на литературу

8. H. Gu, X. Xie, Q. Lv, Y. Ruan, and L. Shang. ETree: Effective and efficient event modeling for real-time social networks. In WIC, 2011.
9. D. He and S. D. Parker. Topic Dynamics: An alternative model of 'bursts' in streams of topics. In KDD, 2010.
10. Y.-B. Liu, J.-R. Cai, J. Yin, and A. Fu. Clustering text data streams. Journal of Computer Science and Technology, 23(1):112–128, 2008.
11. M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In SIGMOD, 2010.
12. Q. Mei and C. Zhai. Discovering Evolutionary Theme Patterns from Text - An Exploration of Temporal Text Mining. In KDD, 2005.
13. R. Schult and M. Spiliopoulou. Discovering emerging topics in unlabelled text collections. In ADBIS, 2006.
14. M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. MONIC – modeling and monitoring cluster transitions. In KDD, 2006.
15. X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In KDD, 2006.