Обзорная статья
УДК 004.042
DOI: 10.17072/1993-0550-2024-2-61-67

Сравнительная оценка методов кластеризации в работе с большими данными

Елена Викторовна Панферова[1], Роман Андреевич Матюшин[2]
[1,2]Тульский государственный педагогический университет им. Л.Н. Толстого,
Институт передовых информационных технологий, г. Тула, Россия
[1]gamma15@inbox.ru
[2]roman.matyuschin2017@yandex.ru

Аннотация. В работе рассмотрена проблематика использования методов кластерного анализа в задачах обработки, анализа и хранения структурированных и неструктурированных данных большого объема и проведена оценка целесообразности их применения при различных аспектах работы с Big Data. Целью работы является выявление наиболее предпочтительных из распространенных алгоритмов кластеризации данных. Для этого была поставлена задача проведения сравнительной оценки следующих популярных алгоритмов: иерархической кластеризации, k-means, DBSCAN, OPTICS и CURE. Рассмотрены алгоритмическая сложность методов и устойчивость алгоритмов к шумам и выбросам, также обозначены потенциальные возможности визуализации их результатов и сферы народнохозяйственного применения. Сделаны выводы о преимуществах и недостатках каждого представленного алгоритма при их использовании в сфере Big Data и о наиболее предпочтительных методах кластерного анализа при различных аспектах работы с большими данными.

Ключевые слова: *Big Data; большие данные; кластеризация; выборка; алгоритм; кластерный анализ; метрика; визуализация; алгоритмическая сложность*

Для цитирования: *Панферова Е.В., Матюшин Р.А.* Сравнительная оценка методов кластеризации в работе с большими данными // Вестник Пермского университета. Математика. Механика. Информатика. 2024. Вып. 2(65). С. 61–67. DOI: 10.17072/1993-0550-2024-2-61-67.

Статья поступила в редакцию 30.04.2024; одобрена после рецензирования 23.05.2024; принята к публикации 12.06.2024.

Review article

Comparative Evaluation of Clustering Methods in Working With Big Data

Elena V. Panferova[1], Roman A. Matushin[2]
[1,2]Tula State Lev Tolstoy Pedagogical University,
Institute of Advanced Information Technologies, Tula, Russia
[1]gamma15@inbox.ru
[2]roman.matyuschin2017@yandex.ru

Abstract. The paper considers the problems of using cluster analysis methods in the tasks of processing, analyzing and storing structured and unstructured large-volume data and evaluates the feasibility of their use in various aspects of working with Big Data. The aim of the work is to identify the most preferred of the common data clustering algorithms. To do this, the task was set to conduct a comparative evaluation of the following popular algorithms: hierarchical clustering, k-means, DBSCAN, OPTICS and CURE. The algorithmic complexity of the methods is considered, the stability of algorithms to noise and emissions is analyzed, as well as the potential possibilities of visualizing their results and the scope of

economic application are indicated. Conclusions are drawn about the advantages and disadvantages of each presented algorithm when used in the field of Big Data and about the most preferred methods of cluster analysis in various aspects of working with big data.

Keywords: *Big Data; clustering; sampling; algorithm; cluster analysis; metric; visualization; algorithmic complexity*

For citation: Panferova, E. V. and Matushin, R. A. (2024), "Comparative evaluation of clustering methods in working Big Data", *Bulletin of Perm University. Mathematics. Mechanics. Computer Science*, no. 2(65), pp. 61-67. DOI: 10.17072/1993-0550-2024-2-61-67.

The article was submitted 30.04.2024; approved after reviewing 23.05.2024; accepted for publication 12.06.2024.

Введение

В сфере информационных технологий на данный момент активно применяется термин Big Data или "Большие данные". Единого общепринятого определения данному понятию не существует. Наиболее правильным и полным, на наш взгляд, является такое: комплекс методов, средств и научно обоснованных подходов к анализу больших массивов данных с целью использования в практической деятельности, то есть, по сути, технология обработки и анализа непрерывно стремительно поступающих огромных массивов разнородной информации.

Big Data представляет собой структурированные и неструктурированные данные большого объема, а также инструменты для работы с ними, что предоставляет возможности для широкого их использования в различных народнохозяйственных сферах, таких как финансы, здравоохранение, маркетинг, средства массовой информации.

Исходя из определения, при таких параметрах входных данных возникает проблема их корректной фиксации, систематизации, обработки, анализа и хранения, для чего и применяются методы кластерного анализа, которые могут выступать в качестве именно начального шага в работе ввиду того, что эти алгоритмы опираются, прежде всего, на такую характеристику Big Data, как мощность рассматриваемого озера всех типов "сырых" данных, так называемый размер входа n [1]. Такие методы могут быть полезными лишь для предварительного анализа, поскольку позволяют учесть специфику больших данных в оценке лишь по одному, рассматриваемому ниже параметру – сложности алгоритмов.

Сложность алгоритма – это количественная характеристика, которая говорит о том, сколько времени либо какой объем памяти потребуется для его выполнения.

При анализе сложности для класса таких задач определяется некоторое число, характеризующее некоторый объем данных – размер входа n.

Таким образом, полагаем, что сложность алгоритма – некоторая функция размера входа. Сложность алгоритма, очевидно, может быть различной при одном и том же размере входа, но различных входных данных.

Понятие "O-сложность" алгоритмов введено для того, чтобы измерять скорость роста функции в зависимости от входных данных [2]. В математике "O" используется для обозначения "order of" (порядка) и позволяет сравнивать функции роста для оценки верхней границы (наихудшего случая), временной сложности алгоритма.

Кроме очевидной экономии различного рода ресурсов, временных и аппаратных, и ускорения процесса обработки больших объемов непрерывно поступающей информации, кластеризация может дать базовые представления о закономерностях внутри таких данных.

Кластеризация – задача разбиения заданной выборки объектов на непересекающиеся подмножества, называемые *кластерами,* так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Под схожестью обычно понимается близость друг к другу относительно выбранной метрики. Для осуществления данного процесса применяются специальные разработанные алгоритмы кластерного анализа. Однако не существует универсального алгоритма, и поэтому возникает необходимость понимания, в каких случаях какой подход предпочтительнее.

В связи с этим подробно рассмотрим наиболее известные алгоритмы и сделаем вывод относительно их применимости в сфере больших данных.

Иерархическая кластеризация

Данный метод основан на построении иерархической структуры кластеров, представленной в виде дерева или дендрограммы. (рис. 1).

Данный метод включает в себя два типа – агломеративный и дивизивный (англ. divisive).

Агломеративный вариант работает по принципу от меньшего к большему, т. е. процесс начинается с каждого объекта в собственном кластере и последовательно проводится объединение ближайших кластеров до тех пор, пока все объекты не окажутся в одном кластере.

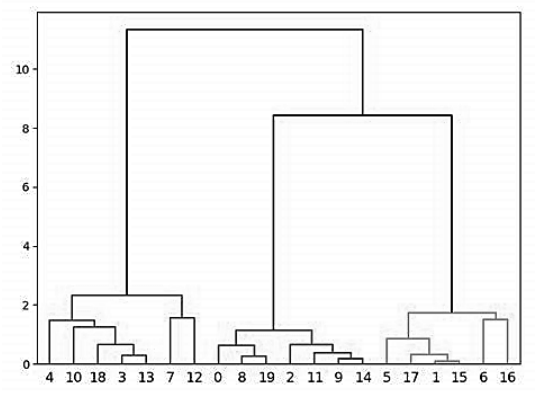

Рис. 1. *Итог работы иерархического метода кластеризации*

В случае дивизивного типа действуют наоборот: начинают с одного кластера, а затем разделяют его на более мелкие.

Для работы данного алгоритма требуется решить, какие данные между собой следует объединять в кластеры, для этого выбирается метрика, количественно характеризующая сходство или несходство данных между собой [3].

Одними из наиболее применимых метрик являются евклидово $d(p,q)$ (1) и манхэттенское d (2) расстояния:

$$d(p,q) = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2} \quad (1)$$

$$d = |x_2 - x_1| + |y_2 - y_1| \,. \quad (2)$$

Основными преимуществами данного алгоритма являются наглядная визуализация результатов, что означает высокую интерпретируемость, а также отсутствие необходимости вручную задавать необходимое количество кластеров, так как алгоритм строит все доступные уровни иерархии.

Алгоритмическая сложность данного алгоритма кубическая, то есть равна $O(n^3)$. Это означает, что время выполнения алгоритма пропорционально кубическому объему входных данных и будут требоваться значительные аппаратные мощности.

Нельзя не отметить низкую устойчивость алгоритма к шумам и выбросам, то есть к появлению точек, не принадлежащих ни одному из кластеров и экстремально отличающихся от остального массива заданных точек, например отрицательное число в массиве положительных.

Также стоит сказать о невозможности визуализации работы алгоритма в виде разбиения на кластеры как произвольной, так и какой-либо фиксированной геометрической формы, к примеру, круга или сферы, за исключением дендрограммы, а также чувствительности алгоритма к метрике расстояний. Под чувствительностью к метрике подразумевается вероятность получить разные результаты в зависимости от выбора той или иной метрики.

Как итог, можно сделать вывод о нецелесообразности использования рассматриваемого алгоритма при работе с большими данными.

K-means

Данные алгоритмы кластеризации представляют собой группу алгоритмов на основе центроидов. Центроид – средняя точка или центр массы фигуры или множества заданных точек.

Суть алгоритма k-means заключается разбивке данных на k точек, где k – заданное количество кластеров и каждый кластер представляет собой группу точек, центр которых является центроидом (рис. 2) [4].

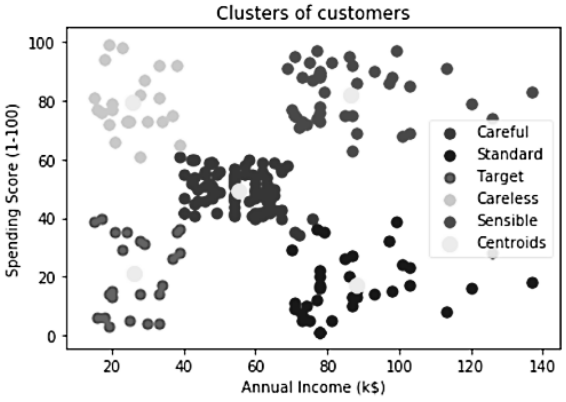

Рис. 2. *Визуальное представление алгоритма k-means*

Получение масштабируемых алгоритмов основано на идее отказа от локальной

функции оптимизации. Парное сравнение объектов между собой в алгоритме k-means есть не что иное, как локальная оптимизация, так как на каждой итерации необходимо рассчитывать расстояние от центра кластера до каждого объекта.

Данный алгоритм является довольно простым в реализации, также он легко масштабируется. Но в то же время его применение ведет к большим вычислительным затратам.

Также имеется ряд минусов, таких как слабая устойчивость к шуму, чувствительность к задаваемым параметрам и начальным центроидам, необходимость задавать число нужных кластеров, строгая форма кластеров сферической формы.

Алгоритмическая сложность алгоритма является плавающей величиной, она оценивается как O(k×n×t), где k – количество центроидов, n – общее количество точек, а t – количество итераций, и чаще всего, в том числе и в случаях значительного увеличения потока данных, она является полиномиальной.

Как итог, можно утверждать, что применять данный алгоритм при обработке больших данных нерационально.

DBSCAN

DBSCAN – алгоритм кластеризации, основанный на плотности, который используется для разделения наборов данных на группы, основанные на пространственной близости точек (рис. 3) [5]. Под плотностью понимается количество точек в заданном пространстве.

Для работы алгоритма требуется задать два параметра необходимых для создания кластеров – "eps" (радиус окрестности в которой ищутся соседи) и "min_samples" (минимальное количество точек, необходимое для определения кластер.

DBSCAN обладает следующим рядом преимуществ: автоматическое определение количества кластеров, возможность создания кластеров произвольной формы, устойчивость к шуму и простота реализации.

Но у него есть такие минусы как чувствительность к параметрам eps и min_samples.

В лучшем случае алгоритмическая сложность DBSCAN будет линейно-логарифмической. Время выполнения алгоритма, очевидно, растет быстрее, чем линейно, но медленнее, чем квадратично: O(n×log n), в худшем случае она составляет $O(n^2)$, такая сложность будет получена, если не будут использоваться пространственные индексы [6].

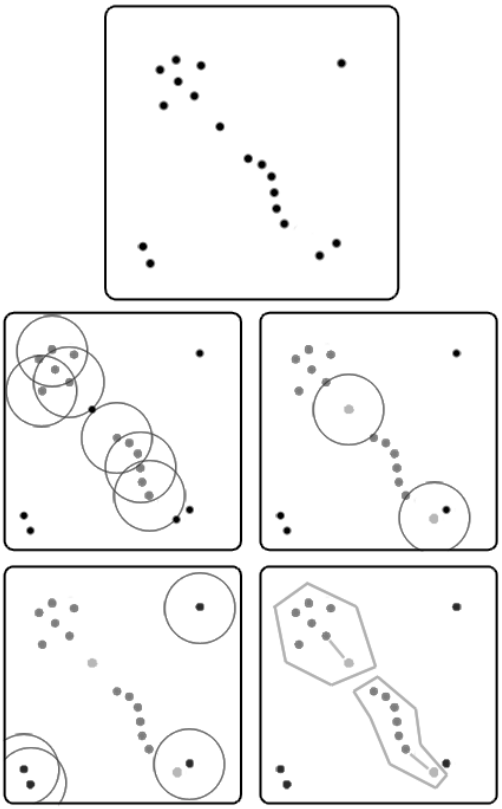

Рис. 3. *Визуальное представление алгоритма DBSCAN*

Пространственные индексы – структуры данных, используемые для оптимизации выполнения запросов, которые требуют доступа к пространственным объектам, таким как точки, линии, полигоны и т. д.

Исходя из вышеизложенного, мы можем сделать вывод, что данный алгоритм вполне применим для работы с большими данными, однако лишь в том случае, если будут созданы пространственные индексы для точек, что существенно сократит время его выполнения.

OPTICS

Данный алгоритм, так же, как и DBSCAN, основан на плотности.

OPTICS работает путем построения графа достижимости, который представляет собой граф, где вершинами являются точки данных, а ребрами – расстояния между ними (рис. 4).

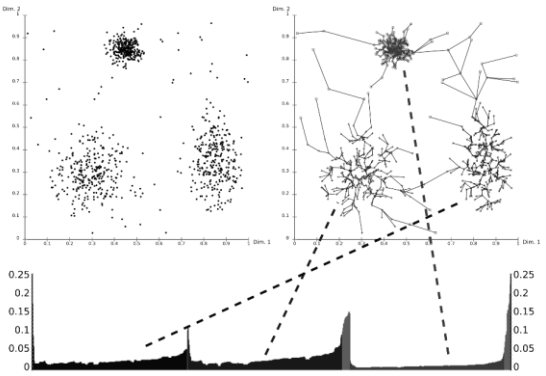

Рис. 4. *Визуальное представление алгоритма OPTICS*

Алгоритм затем упорядочивает точки в зависимости от их достижимости, которая определяется как максимальное расстояние до ближайшей точки с более низкой плотностью [7].

Кластеры определяются как связные компоненты в графе достижимости, где связность определяется порогом достижимости. Под порогом достижимости понимается максимальное расстояние, на котором две точки могут быть связаны и считаться частью одного кластера.

Говоря о преимуществах данного алгоритма, следует отметить его универсальность по отношению к различным типам данных, высокую устойчивость к шуму, но, будучи алгоритмом того же типа что и DBSCAN, он сильно зависит от параметров "eps" и "min_samples", и он более сложен в интерпретации.

Говоря об алгоритмической сложности, мы можем сказать, что по данному параметру он аналогичен DBSCAN, то есть она колеблется от O(n×log n) до $O(n^2)$ в зависимости от того, использовали мы пространственные индексы или нет, но, если сравнивать OPTICS и DBSCAN, то, в целом, первый будет быстрее на больших наборах данных [8].

В итоге мы можем сделать вывод, что данный алгоритм – наиболее предпочтительный для работы с большими данными из всех вышеперечисленных вариантов.

CURE

CURE – частный случай иерархической кластеризации, который использует набор представителей для определения принадлежности объекта к определенному кластеру (рис. 5) [9].

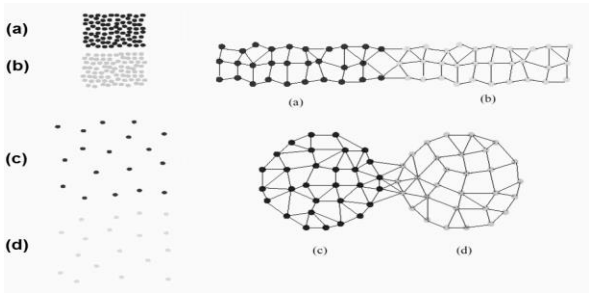

Рис. 5. *Визуальное представление алгоритма CURE*

Он хорошо подходит для кластеризации наборов числовых данных, особенно в случаях, когда:

- присутствуют выбросы: CURE менее чувствителен к выбросам, чем другие алгоритмы кластеризации;
- кластеры имеют сложную форму: CURE способен находить кластеры произвольной формы, а не только сферические или эллиптические;
- кластеры имеют разный размер: CURE может обнаруживать кластеры разных размеров, что не всегда возможно в случае других алгоритмов.

Алгоритмическая сложность фиксированная, не плавающая, составляет O(n×log n), а к преимуществам нужно отнести масштабируемость, устойчивость к выбросам и создание кластеров произвольной формы, но, к сожалению, он сложен в реализации и так же, как и DBSCAN и OPTICS, зависим от параметров [10].

Будучи алгоритмом, специально созданным для обработки больших данных, он наиболее предпочтителен при их обработке.

Суммируя ранее сделанные выводы, приведем сравнение представленных алгоритмов по следующим критериям:

a) преимущества;

b) недостатки;

c) сфера применения;

c) алгоритмическая сложность.

Иерархическая кластеризация

a) простота реализации, наглядность;

b) нежелательность использования на больших объемах данных, чувствительность к метрике расстояний, неспособность формировать произвольные формы кластеров;

c) биоинформатика, бизнес, обработка изображений, поиск информации;

d) $O(n^3)$.

k-means

a) простота реализации, предпочтительность применения на малых объемах данных;

b) низкая устойчивость к шуму, невозможность формировать произвольные формы кластеров;

c) сегментация клиентов, классификация документов, анализ записей звонков;

d) $O(k \times n \times t)$.

DBSCAN

a) устойчивость к шуму, находить кластеры произвольной формы и разного размера, простота реализации;

b) чувствительность к выбору параметров, невозможность находить кластеры иерархической структуры;

c) медицина, анализ географических данных [11];

d) от $O(n \times \log n)$ до $O(n^2)$.

OPTICS

a) устойчивость к шуму, находить кластеры произвольной формы и разного размера, простота реализации;

b) чувствительность к выбору параметров, невозможность кластеры иерархической структуры;

c) анализ геоданных, биоинформатика, телекоммуникации;

d) от $O(n \times \log n)$ до $O(n^2)$.

CURE

a) устойчивость к шуму, находит кластеры произвольной формы и разного размера;

b) чувствительность к выбору параметров, сложность реализации;

c) обработка изображений, медицина, финансы [12].

d) $O(n \times \log n)$.

Заключение

На основе проведенного анализа сделаем вывод, что среди представленных алгоритмов кластеризации наиболее предпочтительными при работе с большими данными являются DBSCAN, OPTICS и специально созданный для их обработки CURE.

Список источников

1. *Goodfellow Y., Bengio A.* Courville, Deep Learning / Adaptive Computation and Machine Learning series // The MIT Press, 2016.
2. *Даниленко А.Н.* Структуры данных и анализ сложности алгоритмов: учеб. пособие / Самара: Изд-во Самарского университета, 2018. 76 с.
3. *Data clustering:* a review / A. K. Jain, M. N. Murty, P. J. Flynn // ACM Computing Surveys. 1999. № 31(3). P. 264–323.
4. *K-means* // ScikitLearn: URL: https://scikit-learn.org/stable/modules/clustering.html#k-means (дата обращения: 03.04.2024).
5. *A density-based* algorithm for discovering clusters in large spatial databases with noise / Ester Martin, Kriegel Hans-Peter, Sander Jörg, Xu Xiaowei // Proceedings KDD'96. 1996. № 34. P. 226-231.
6. *GO-DBSCAN:* Improvements of DBSCAN Algorithm Based on Grid / Feng L., Liu K., Tang F., Meng Q. // 2017. vol. 9. no. 3, pp. 151.
7. *OPTICS:* ordering points to identify the clustering structure / Ankerst M., Breunig [и др.] // Proceedings SIGMOD '99. 1999. № 2. P. 49–60.
8. *Data mining:* Concepts and Techniques / Han J., Kamber M., Pei J. // 2012. Morgan Kaufmann Series, Waltham, USA.
9. *Basic* Understanding of CURE Algorithm // Geeksforgeeks: URL: https://www.geeksforgeeks.org/basic-understanding-of-cure-algorithm/ (дата обращения: 03.04.2024).
10. *CURE:* An Efficient Clustering Algorithm for Large Databases / Guha S., Rastogi R., Kyuseok S. // 1998. ACM SIGMOD Conference, vol. 27, no. 2, pp. 73-84.
11. *Кластеризация* пространственных данных – плотностные алгоритмы и DBCSAN // КАРТЕТИКА: URL: https://cartetika.ru/tpost/k05o2ndpf1-klasterizatsiya-prostranst-vennih-dannih (дата обращения: 11.04.2024).
12. *CURE* Algorithm // Deepgram: URL: https://deepgram.com/ai-glossary/cure-algorithm (дата обращения: 11.04.2024).

References

1. Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*, Adaptive Computation and Machine Learning series, the MIT Press.
2. Danilenko, A.N. (2018), Struktury dannykh i analiz slozhnosti algoritmov [Data structures and algorithm complexity analysis], № 1272, Samara University Press, Samara, Russia.

3. Jain, A. K., Murty, M. N. and Flynn, P. J. (1999), "Data clustering: a review", ACM Computing Surveys, vol. 31, no. 3, pp. 264-323.
4. ScikitLearn (2024), "K-means", available at: https://scikit-learn.org/stable/modules/clustering.html#k-means (Accessed 03 April 2024).
5. Ester M., Kriegel Hans H.-P., Sander J. and Xu X. (1996), "A density-based algorithm for discovering clusters in large spatial databases with noise", Proceedings KDD'96, vol. 34, pp. 226-231.
6. Feng L., Liu K., Tang F. and Meng Q. (2017), "GO-DBSCAN: Improvements of DBSCAN Algorithm Based on Grid", vol. 9, no. 3, pp. 151.
7. Ankerst M., Breunig M. M., Kröger P. and Sander J. (1999), "OPTICS: ordering points to identify the clustering structure", Proceedings SIGMOD '99, vol. 2, pp. 49-60.
8. Han J., Kamber M. and Pei J., (2012), "Data mining: Concepts and Techniques", Morgan Kaufmann Series, Waltham, USA.
9. Geeksforgeeks (2021), "Basic Understanding of CURE Algorithm", available at: https://www.geeksforgeeks.org/basicunder-standing-of-cure-algorithm/ (Accessed 03 April 2024).
10. Guha, S., Rastogi, R. and Kyuseok, S., (1998), "CURE: An Efficient Clustering Algorithm for Large Databases", ACM SIGMOD Conference, vol. 27, no. 2, pp. 73-84.
11. Cartetika (2023), "Clustering of spatial data – density algorithms and DBCSAN", available at: https://cartetika.ru/tpost/k05o2ndpf1-klasteri-zatsiya-prostranstvennih-dannih (Accessed 11 April 2024).
12. Deepgram (2024), "CURE Algorithm", available at: https://deepgram.com/ai-glossary/cure-algorithm (Accessed 11 April 2024).

Информация об авторах:

Е. В. Панферова – кандидат технических наук, доцент, доцент института передовых информационных технологий Тульского государственного педагогического университета (300026, Россия, г. Тула, пр. Ленина, 125, корпус 3), SPIN-код: 3937-4236, AuthorID: 814520;

Р. А. Матюшин – студент-магистрант института передовых информационных технологий Тульского государственного педагогического университета (300026, Россия, г. Тула, пр. Ленина, 125, корпус 3).

Information about the authors:

Elena V. Panferova – Candidate of Technical Sciences, Associate Professor, Associate Professor, Institute of Advanced Information Technologies, Tula State Pedagogical University, (125, Lenin Ave., Tula, Russia, 300026), SPIN-code: 3937-4236, AuthorID: 814520;

Roman A. Matushin – Master's Student of the Institute of Advanced Information Technologies, Tula State Pedagogical University (125, Lenin Ave., Tula, Russia, 300026).