УДК 004.6 doi: 10.26102/2310-6018/2019.24.1.012

А.И. Мирошников

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ ИНТЕРВАЛЬНОЗНАЧНЫХ ДАННЫХ ПРИ РЕШЕНИИ ЗАДАЧИ КЛАСТЕРИЗАЦИИ МАРОК ЧУГУНА

ФГБОУ ВО «Липецкий государственный технический университет», Липецк, Россия

Актуальность исследования обусловлена необходимостью повышения эффективности процессов обработки интервальнозначных данных в вычислительных машинах, комплексах и компьютерных сетях и в сокращении сроков их создания; поиска методов повышения надежности функционирования прикладного программного обеспечения, средств унификации его разработки. С целью решения поставленных задач разработана библиотека для работы с интервальнозначными данными, реализующая пользовательский интервальнозначный тип и базовые операции над ним, что позволяет многократно использовать надёжный и эффективный тип данных, по умолчанию осуществляющий проверки целостности в базах данных, снимая эту задачу с прикладного программиста, предоставляя ему возможность оперировать с границами интервалов, как с едиными объектами, ускоряя время разработки программного обеспечения при организации интервальных баз данных и выполнения интервальных запросов с целью аналитической обработки информации. Средства для поддержки интервальных типов данных в системах баз данных позволяют решать широкий класс задач анализа данных, оперирующих интервальными данными, в частности, задачу кластеризации в справочной информационно-аналитической системе марок чугуна для промышленного производства.

Ключевые слова: интервальнозначные данные, интервальные базы данных, пользовательские типы данных, интервальные запросы, аналитическая обработка данных, достоверные вычисления

Введение.

Современные информационные системы хранят и обрабатывают огромные объемы данных о технических и экономических процессах, статистической информации. В связи с этим встает вопрос максимально эффективного использования этих данных, извлечения скрытых знаний, потенциально полезных её владельцу (которым часто выступают крупные организации) в коммерческом плане. Однако, не всегда возможно определить точные значения ряда параметров ввиду особенностей измерительных приборов или самой природы данных. В случаях, когда можно лишь определить границы диапазона, внутри которого находится точное значение, используют теорию интервальной арифметики и интервального анализа [1].

В [2, 3] приведено описание интервальнозначного типа данных, арифметических операций, агрегирующих функций над интервальными типами данных. В [4] описаны проблемы (в частности, отсутствие

однозначного формализованного решения в случае пересечения сравниваемых интервалов) и излагается методика сравнения интервальнозначных типов данных с использованием вероятностного показателя. В [5] рассматриваются вопросы эффективного хранения интервальнозначных данных и применения индексов систем управления базами данных для ускорения операций доступа. В [6] демонстрируется преимущество разработанного интервальнозначного типа данных над существующими объектно-ориентированными технологиями доступа к данным

Данный подход к организации пользовательского типа позволяет использовать существующие технологии баз данных для интеграции методов работы с интервальнозначными типами данных в существующих системах, что позволяет достигать высокой производительности при выполнении запросов на основе использования стандартного синтаксиса языка SQL.

Одними из наиболее распространенных баз данных являются справочные системы, являющиеся, в частности, элементом информационной системы о марках чугунов. Данная система, хранящая сведения обо всех марках чугуна, может быть использована для решения задачи кластеризации или классификации марок чугунов [7]. Кластеризация и классификационный анализ могут применяться по различным характеристикам, хранящимся в базе данных.

При проведении кластеризации над интервальнозначными данными необходимо предлагается использовать интервальное сравнение в алгоритме k-средних, а при расчете метрики – квадрат евклидового расстояния, удобно иметь инструмент, позволяющий вычислять квадрат разности интервальных значений.

Материалы и методы.

Рассмотрим процесс реализации прикладного программного обеспечения, использующего библиотеку интервальнозначных данных [3], при использовании IDE Microsoft Visual Studio Express и СУБД Microsoft SQL Server на языке C#. В первую очередь в Visual Studio следует указать ссылку на библиотеку интервальнозначных данных, добавив её в разделе Reference → Add reference. Данная библиотека содержит описание базового класса iInterval, конструктор которого содержит проверки целостности данных и реализует округление в соответствии со свойствами интервальнозначного типа. Закрытые поля класса и их назначение указаны в Таблице 1. Присваивание значений закрытым полям реализуется через механизм свойств, содержащих соответствующие геттеры, и сеттеры, вызывающих методы проверки целостности интервальнозначных данных.

Таблица 1 – Описание полей базового класса

Поле базового класса	Описание поля базового класса
`private double down;`	Нижняя интервальная граница
`private double up;`	Верхняя интервальная граница
`private bool isNull;`	Значение, используемое при реализации интерфейса INullable
`public static double p;`	Параметр, используемый при реализации сравнения интервальных значений

Разработанная библиотека интервальнозначных данных содержит перегрузки стандартных арифметических операторов и операторов сравнения, представленные в Таблице 2. Заголовки методов для реализации базовых функций интервальнозначных данных указаны в Таблице 3. Методы специальных структур, позволяющие реализовывать агрегирующие операторы над пользовательским типом данных при написании прозрачных SQL-запросов, перечислены в Таблице 4.

Таблица 2 – Методы перегрузки стандартных операторов

Заголовок метода перегрузки	Описание метода перегрузки
`public static iInterval operator+(iInterval obj1, iInterval obj2)`	Сумма двух интервальнозначных данных
`public static iInterval operator-(iInterval obj1, iInterval obj2)`	Разность двух интервальнозначных данных
`public static iInterval operator*(iInterval obj1, iInterval obj2)`	Произведение двух интервальнозначных данных
`public static bool operator==(iInterval obj1, iInterval obj2)`	Сравнение «равенство» двух интервальнозначных данных
`public static bool operator!=(iInterval obj1, iInterval obj2)`	Сравнение «неравенство» двух интервальнозначных данных
`public static bool operator<(iInterval obj1, iInterval obj2)`	Сравнение «меньше» двух интервальнозначных данных
`public static bool operator<=(iInterval obj1, iInterval obj2)`	Сравнение «меньше или равно» двух интервальнозначных данных

Таблица 3 – Методы реализации базовых функций интервальнозначных данных

Заголовок метода	Описание метода
`public static SqlDouble DOWN(iInterval Value)`	Получение нижней границы интервального значения
`public static SqlDouble UP(iInterval Value)`	Получение верхней границы интервального значения
`public static SqlDouble WIDTH(iInterval Value)`	Расчет ширины интервального значения
`public static SqlDouble MID(iInterval Value)`	Расчет медианы интервального значения
`public static SqlDouble RAD(iInterval Value)`	Расчет радиуса интервального значения
`public static iInterval SQ_DIFF(iInterval obj1, iInterval obj2)`	Расчет квадрата разности интервальнозначных данных
`public static iInterval SCALAR_MULT(iInterval Value, SqlDouble a)`	Расчет произведения интервального и скалярного значений

Таблица 4 – Агрегирующие операторы над интервальнозначными данными

Заголовок структуры	Описание агрегата
`public struct ICOUNT`	Количество значений в столбце Таблицы, содержащей интервальные данные
`public struct ISUM`	Сумма значений в столбце Таблицы, содержащей интервальные данные
`public struct ISUB`	Разность значений в столбце Таблицы, содержащей интервальные данные
`public struct IMULT`	Произведение значений в столбце Таблицы, содержащей интервальные данные

Отдельно стоит отметить описание класса (Рисунок 1), позволяющего вычислить квадрат разности интервальнозначных данных, используемого при расчете Евклидового расстояния между двумя интервальными значениями. Данная реализация становится возможной после создания перегрузки стандартных операторов (Таблица 2).

После подключения ссылки на библиотеку интервальнозначных данных необходимо разрешить SQL Server выполнять код из пользовательских сборок при помощи передачи параметра «[clr enabled], 1» в хранимую процедуру «sp_configure» и дальнейшего вызова инструкции «reconfigure». Далее следует зарегистрировать сборку (CREATE ASSEMBLY), указав в разделе «Программирование» SQL Server, путь до библиотеки интервальнозначных данных.

```
public class I_SQ_DIFF
{
     [SqlFunctionAttribute()]
     public static iInterval SQ_DIFF(iInterval obj1,
                                     iInterval obj2)
     {
          iInterval s = new iInterval();
          s = (obj1 - obj2) * (obj1 - obj2);
          return s;
     }
}
```

Рисунок 1 – Класс, реализующий вычисление квадрата разности интервальнозначных данных

Далее требуется выполнить скрипт на языке Transact-SQL, регистрирующий пользовательские хранимые функции и агрегаты, реализованные в библиотеке интервальнозначных данных. Пример создания функции, вычисляющей Евклидовое расстояние, приведен на Рисунке 2.

```
GO
IF EXISTS (SELECT * FROM sys.objects WHERE [name] = 'I_SQ_DIFF')
DROP FUNCTION I_SQ_DIFF;
GO
CREATE FUNCTION dbo.I_SQ_DIFF(@value1 iInterval,
                              @value2 iInterval)
RETURNS iInterval
AS EXTERNAL NAME INTERVALS.[iInterval.I_SQ_DIFF].[SQ_DIFF]
```

Рисунок 2 – Регистрация пользовательской хранимой функции

Дальнейшая реализация прикладного программного обеспечения, производящая кластеризацию марок чугуна, будет использовать все возможности, предоставляемые библиотекой интервальнозначных данных. Рассмотрим последовательность действий интервального алгоритма k-средних с использованием евклидового расстояния в качестве метрики для выполнения кластеризации.

1. Создание центроидов, содержащих случайные интервально-значные данные. Для инициализации случайными значениями вызывается метод «init_centroids» (Рисунок 3). В качестве входных параметров выступает список объектов типа iIron (Рисунок 4) и задаваемое пользователем количество центроидов.

```
private void init_centroids(List<iIron> centriods,
                            int number_of_centroids)
```

Рисунок 3 – Заголовок метода, инициализирующего начальные значения центроидов

```
public class iIron
{
    public string Name { get; set; }
    public iInterval C { get; set; }
    public iInterval Si { get; set; }
    public iInterval Mn { get; set; }
    public iInterval Cr { get; set; }
    public iInterval Cu { get; set; }
    public iInterval Ni { get; set; }
    public iInterval S { get; set; }
    public iInterval P { get; set; }
    public iInterval Al { get; set; }
    public iInterval Hardness { get; set; }
    public iInterval Strength { get; set; }
    public iInterval Extension { get; set; }
}
```

Рисунок 4 – Формат хранения марки чугуна

Ввиду того, что входные параметры методов передаются по ссылке, а не по значению, тип возвращаемого значения реализуется, как void.

2. Расчет минимальных расстояний от каждого объекта, содержащего интервальнозначные данные, до центроидов (Рисунок 5). Данное расстояние определяет принадлежность каждого объекта конкретному кластеру. Сравнение интервальнозначных расстояний производится в соответствии с методикой, описанной в [4].

```
SELECT id, iron_name,
     I_SQ_DIFF(@C,  Iron.C)  +
     I_SQ_DIFF(@Si, Iron.Si) +
     I_SQ_DIFF(@Mn, Iron.Mn) +
     I_SQ_DIFF(@Cr, Iron.Cr) +
     I_SQ_DIFF(@Cu, Iron.Cu) +
     I_SQ_DIFF(@Ni, Iron.Ni) +
     I_SQ_DIFF(@S,  Iron.S)  +
     I_SQ_DIFF(@P,  Iron.P)  +
     I_SQ_DIFF(@Al, Iron.Al) +
     I_SQ_DIFF(@Hardness,  Iron.Hardness) +
     I_SQ_DIFF(@Strength,  Iron.Strength) +
     I_SQ_DIFF(@Extensio1, Iron.Extension) AS distance
FROM Iron
```

Рисунок 5 – SQL-запрос, возвращающий квадрат евклидового расстояния от центроида до каждого из объектов по 12 параметром

3. Пересчет интервальных характеристик центроидов.

4. Если выполнено определённое количество итераций предыдущих двух пунктов или значения центроидов изменяются менее заданной константы, выводится текущее разбиение на кластеры, иначе, процесс повторяется.

Результаты.

Для удобного просмотра исходных данных по химическому составу и свойствам различных марок чугунов, задания параметров кластеризации было реализовано программное обеспечение «Кластеризация марок чугуна с использованием интервального алгоритма K-средних», элемент интерфейса и вариант результата интервальной кластеризации, представлен на Рисунке 6.

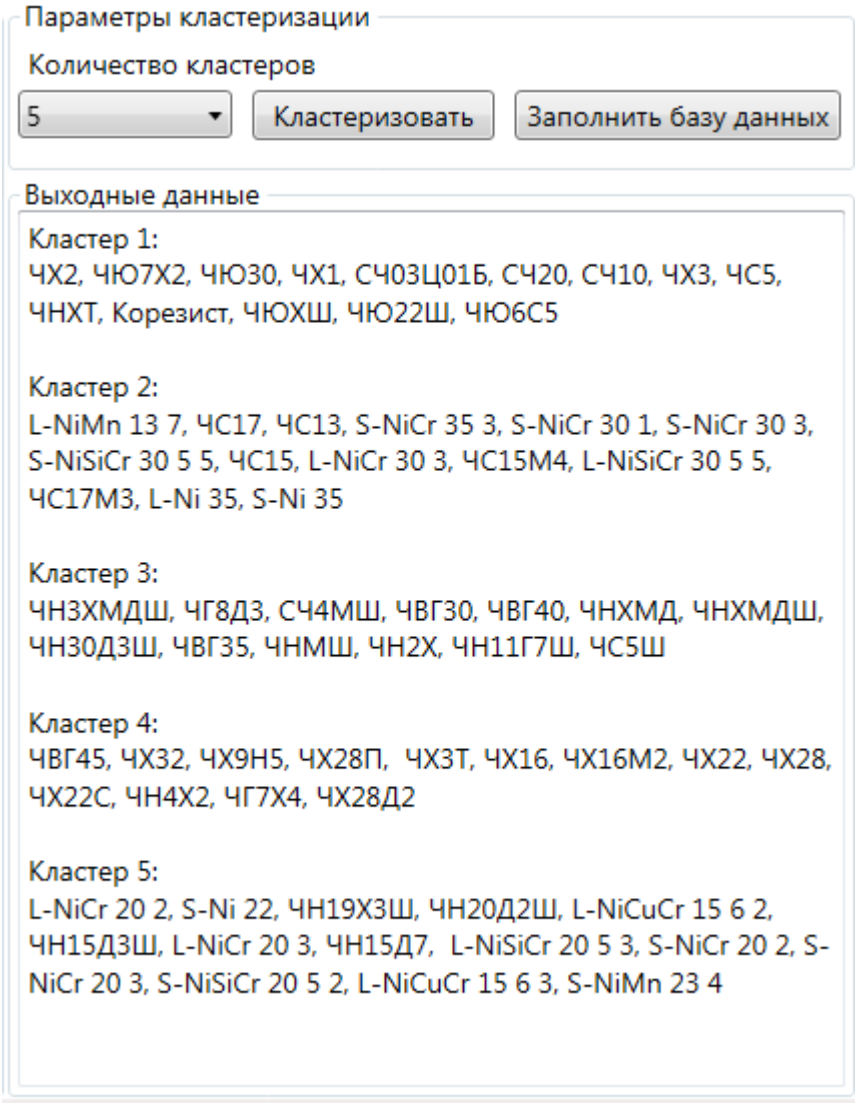

Рисунок 6 – Результаты интервальной кластеризации

Результаты, полученные в области «Выходные данные» согласуются с результатами, описанными в [8]. Так, марки хромистых

высоколегированных чугунов были отнесены к одному кластеру, а аустенитные высоколегированные чугуны с высокой жаропрочностью – к другому кластеру.

Заключение.

Разработка математического и программного обеспечения для аналитической обработки информации с использованием библиотеки интервальнозначных данных ускоряет создание приложений, гарантируя программистам надежную и эффективную обработку пользовательских типов данных, позволяя сконцентрироваться на написании основной логики программы.

Исследование выполнено при финансовой поддержке РФФИ и Администрации Липецкой области в рамках научного проекта № 16-47-480929-р_центр_а.

ЛИТЕРАТУРА

1. Шарый С.П. Конечномерный интервальный анализ. Институт вычислительных технологий СО РАН. Новосибирск: изд-во «XYZ», 2018. 628 с. URL: http://www.nsc.ru/ interval (дата обращения 20.01.2019).
2. Галкин А.В., Мирошников А.И., Погодаев А.К. Разработка интервального типа данных и операций над ним в системе MS SQL Server // Системы управления и информационные технологии, №1(67), 2017. – С. 48-51.
3. Сараев П.В., Галкин А.В., Мирошников А.И., Никольская А.А. Обработка объектов интервального типа в системе управления базами данных SQL Server // Управление большими системами: материалы XIV Всероссийской школы-конференции молодых учёных / Пермь: Изд-во Перм.нац.исслед.политехн.ун-та, 2017. – 2017. – 716 с. – С. 485 – 492.
4. Погодаев А.К., Галкин А.В., Сараев П.В., Мирошников А.И. Сравнение интервальных типов данных в системе MS SQL SERVER // Системы управления и информационные технологии. 2018. Т. 71. № 1. С. 68-72.
5. Сараев П.В., Галкин А.В., Мирошников А.И. Исследование использования индексов в базах данных с объектами интервального типа // Вести высших учебных заведений Черноземья, №4, 2017.
6. Мирошников А.И., Сараев П.В., Галкин А.В. Использование объектно-ориентированной технологии доступа к данным интервального типа // Международная конференция по мягким вычислениям и измерениям. 2018. Т. 1. С. 184-187.

7. Galkin A.V., Mihailov E., Shipelnikov A., Blyumin S.L. An Information-Analytical System of Ranging Analysis of Cast Iron Grades // Journal of Chemical Technology and Metallurgy. 2015. Vol. 50. No. 6. P. 651-658.
8. Погодаев А.К., Мирошников А.И., Никольская А.А., Сараев П.В. Применение интервального типа данных при выполнении алгоритмов в СУБД MS SQL Server // Современные сложные системы управления: материалы XII международной научно-практической конференции / Липецк, 2017. С. 60-64.

A.I. Miroshnikov

CAST IRON GRADES CLASSIFICATION USING INTERVAL DATA LIBRARY

Lipetsk State Technical University
Lipetsk, Russia

The paper is devoted to the problem of the need to improve the efficiency of interval-valued data processing in computers, complexes and computer networks and to reduce the time of its creation; search for methods to improve the reliability of the application software, the means of unifying its development. Interval-valued data library has been developed in order to solve these tasks. It implements a user-defined interval-valued type and basic operations on it which makes it possible to reuse a reliable and efficient data type which performs integrity checks in databases by default, removing this task from an application programmer, providing to operate with the boundaries of the intervals as with single objects speeding up the software development when organizing interval data bases and requesting to analytical processing. Solutions to support interval data types in database systems allow to solve a wide class of data analysis tasks operating with interval data, in particular, the clusterization of cast iron problem in the industrial reference information-analytical system.

Keywords: interval-valued data, interval databases, user-defined data types, interval queries, analytical data processing, reliable calculations

This work is partially supported by Russian Foundation for Basic Research (RFBR) and Lipetsk regional administration Grant #16-47-480929-r_a.

REFERENCES

1. Shary S.P. Finite dimensional interval analysis. Institute of Computational Technologies SB RAS. Novosibirsk: publishing house «XYZ», 2018. 628 p. URL: http://www.nsc.ru/ interval (contact date 01/20/2019).
2. Galkin A.V., Miroshnikov A.I., Pogodaev A.K. Development of interval data type and operations on it in the MS SQL Server system // Control systems and information technologies, №1 (67), 2017. - PP. 48-51.
3. Saraev P.V., Galkin A.V., Miroshnikov A.I., Nikolskaya A.A. Interval type objects processing in a SQL Server database management system // Large-

Scale Systems Control: materials of the XIV All-Russian School-Conference of Young Scientists / Perm: Perm Publishing House. Research Policy, 2017, 2017. - 2017. 716 with. - S. 485 - 492.

4. Pogodaev A.K., Galkin A.V., Saraev P.V., Miroshnikov A.I. Comparison of interval data types in the MS SQL SERVER system // Control systems and information technologies. 2018. T. 71. № 1. PP. 68-72.
5. Saraev P.V., Galkin A.V., Miroshnikov A.I. Study of the use of indices in databases with objects of the interval type // News of Higher Educational Institutions of the Chernozem Region, No. 4, 2017.
6. Miroshnikov A.I., Saraev P.V., Galkin A.V. Using Object-Oriented Interval Type Data Access Technology // International Conference on Soft Computing and Measurements. 2018. T. 1. PP. 184-187.
7. Galkin A.V., Mihailov E., Shipelnikov A., Blyumin S.L. An Information-Analytical System of Ranging Analysis of Cast Iron Grades // Journal of Chemical Technology and Metallurgy. 2015. Vol. 50. No. 6. P. 651-658.
8. Pogodaev A.K., Miroshnikov A.I., Nikolskaya A.A., Saraev P.V. Application of interval data type when executing algorithms in MS SQL Server DBMS // High technology control systems: materials of the XII international scientific-practical conference / Lipetsk, 2017. P. 60-64.