Статистика в современном мире

(Proposals for improving statistics and its methodological, information and software support in the context of new technologies and analytical capabilities presented by machine learning and AI methods.)

(Предложения по совершенствованию статистики и ее методологического, информационного и программно-технологического обеспечения в условиях новых технологий и аналитических возможностей, представленных методами машинного обучения и ИИ.)

(Statistics, measurement, static and dynamic structure, adequacy criterion)
(Статистика, измерения, статическая и динамическая структуры, критерий адекватности)

В условиях цифровой трансформации экономики управление территориями субъектов Федерации и крупных городов, территориальными структурами федеральных органов исполнительной власти, а также территориально распределенными государственными и частными промышленными и финансовыми корпорациями и холдингами невозможно без применения современных географически распределенных систем, обеспечивающих принятие эффективных управленческих решений на основе результатов статистического анализа.

Вместе с тем существует, по крайней мере, два аспекта в проблеме объединения и согласования информации и работ, связанных со статистическим анализом в географически распределенных системах:

- Организационно-методологический и
- Информационно-технологический.

Эти аспекты тесно связаны между собой и не могут быть рассмотрены по отдельности и независимо один от другого.

Другими словами, необходим комплексный подход к решению соответствующих проблем на основе использования:

- **Метаданных**, как нормативной основы решения задач организационно-методологической интеграции статистических данных;
- **Контейнерных технологий**, например, Dockers, Kubernetes, Podman, как основы решения задач информационно-технологического обеспечения территориально-распределенных статистических систем;
- **Систем комплексного статистического анализа**, **разработанных с использованием машинного обучения и Искусственного Интеллекта** в качестве инструментария интерактивной поддержки пользователей.

Эти проблемы вынуждают нас вновь обратиться к анализу основных понятий статистики с целью их переосмысления в соответствии со стоящими перед статистикой новыми задачами и новыми возможностями.

Следуя традиции, сложившейся в естественных науках, круг проблем, связанных с выбором и обоснованием исходных принципов и понятий статистической методологии, мы будем называть **основаниями общей теории статистики**.

Основания науки отражают ее логику и, в силу этого, непосредственно примыкают к соответствующим разделам философии и используют их.

Понятие статистического измерения

Всякая логика использует тот или иной язык. Язык оснований общей теории статистики, на наш взгляд, должен отвечать ряду требований, которые вытекают из прикладного характера статистики как науки. Эти требования в первом приближении могут быть сформулированы следующим образом,

Во-первых, все элементы этого языка должны быть конструктивно определены. Это значит, основные понятия статистики такие как:

- Статистическое наблюдение;
- Объект статистического наблюдения;
- Статистический показатель;
- Статистическая связь;
- Система статистических показателей и др. -

должны быть не просто названы, но и должен быть указан способ их построения. Другое название для этого требования - операциональность всех определений.

Во-вторых, основные преобразования на элементах этого языка (объединение понятий, переход одного понятия в другое, вычленение более узкого понятия из общего и др.) должны быть определены однозначно и иметь четкую интерпретацию в терминах статистики.

В-третьих, этот язык должен быть полон и непротиворечив. Это требование означает, что применение допустимых преобразований к любым элементам языка не должно приводить к противоречиям, либо построению не интерпретируемого в терминах статистики логического вывода.

Формальной основой для языка оснований общей теории статистики, который отвечал бы всем выдвинутым требованиям, может служить **математическая теория измерений** [1]. Эта теория представляет собой специальный вариант реляционной алгебры и в этом смысле является естественным формализмом для представления логики оснований общей теории статистики.

Основные понятия и категории, которые используются в теории измерений не всегда совпадают по содержанию с аналогичными понятиями в статистике. Это относится и к определению термина “измерение”, которое в теории измерений понимается как отображение одной системы в другую.

В статистической интерпретации измерением является отображение объекта наблюдения (некоторой социально-экономической системы) в систему статистических показателей. Основные требования к этому отображению - однозначность и сохранение основных структурных соотношений отображаемой системы, т.е. каждое свойство или элемент наблюдаемого объекта отображается в соответствующие показатели и реквизиты, а связи между элементами объекта наблюдения - в соответствующие связи между показателями.

При этом происходит определенное огрубление описания реальной системы за счет слияния нескольких характеристик объекта наблюдения в одном показателе или реквизите. Это свойство статистического наблюдения не противоречит традиционному определению измерения, однако ряд свойств статистического наблюдения позволяют выделить его в особый класс измерений, которые мы будем называть **статистическими измерениями**.

Прежде всего, сама постановка задачи статистического измерения формулируется иначе чем это сделано в математической теории измерений, в которой известными считаются как измеряемая (эмпирическая система с отношениями), так и измеряющая (числовая система с отношениями) системы, а определению подлежит само отображение, связывающее две эти системы.

В статическом измерении задана измеряющая система (даны описания и установлены свойства), в качестве которой выступает система статистических показателей, и отображение, которое связывает объект статистического наблюдения и систему статистических показателей.

Целью же статистического измерения (как практической деятельности) является идентификация состояния и динамики наблюдаемого объекта на основе исследования состояния и динамики системы статистических показателей.

Таким образом, статистическое измерение, понимаемое как определенным образом организованное отображение социально-экономической системы в систему статистических показателей, можно рассматривать как предмет общей теории статистики.

Эта позиция в основе своей не противоречит установившейся точке зрения на предмет общей теории статистики, который традиционно представляется как исследование количественной стороны качественно выделенных аспектов массовых социально-экономических явлений и их развития, но предлагает более строгую дифференциацию этого понятия.

В существующем определении предмета статистики отражен только один аспект статистического отображения - его результат, представленный статистическими данными. **Мы же, кроме того, выделяем и само отображение, с помощью которого были получены эти данные**.

Новое (уточненное) определение предмета общей теории статистики будет играть критическую роль в интеграции статистистических исследований и систем машинного обучения (МО) и искусственного интеллекта (ИИ).

Статистическое отображение и его результат, представленный в виде системы статистических показателей, мы будем называть **статистическим измерением**, само же отображение мы будем называть **статистической шкалой**.

Другим существенным моментом нашего подхода, который также можно рассматривать как развитие традиционных идей общей теории статистики о взаимосвязи наблюдаемых явлений, является переход к системному рассмотрению всех основных понятий статистической науки.

Системой статистических показателей мы будем называть совокупность исходных (включенных в программу наблюдения) данных и расчетных (полученных в результате решения статистических задач) показателей, структурно объединенных целью статистического наблюдения.

Цель статистического наблюдения формируется вне статистического наблюдения потребителями статистической информации.

Социально-экономическая система (СЭС), как объект управления и статистического наблюдения, характеризуется наличием большого количества, как правило, разнородных элементов и связями между ними, которые отражают характер зависимостей элементов друг от друга. **Основным свойством такой совокупности элементов является целостность**, которая выражается в невозможности расчленения социально-экономической системы без потери некоторых ее качеств.

Целостность СЭС на содержательном уровне означает определенную “замкнутость” системы относительно тех связей, которые определены между ее элементами. Классическим примером, иллюстрирующим введенное понятие целостности и замкнутости СЭС, является система материальных потоков, отраженная в межотраслевом балансе.

Естественно допустить, что система статистических показателей, отражающая некоторую СЭС, так же должна отвечать основным требованиям системности:

- Взаимозависимости (связности) отдельных элементов (показателей) и
- Целостности.

Отношения и связи между показателями отражают **структуру статистической информации**.

Отметим, кстати, что понятие “**структура**” мы рассматриваем в более широком, чем это принято в общей теории статистике, смысле. Традиционное понятие структуры, как представление целого через доли его частей, очевидно, является более узким, поскольку ему соответствует одно из возможных отношений структуры статистической информации - отношение “**включения**”, с помощью которого может быть описана иерархия соответствующих показателей.

В структуре ССП мы будем выделять два основных типа связей (отношений) между показателями и, соответственно, два типа структур:

- **Статическую структуру ССП**, которая отражает постоянные (устойчивые) связи между единицами наблюдения. Эти связи отражают пространственное положение исследуемых единиц наблюдения в ряду других, материальные, функциональные и другие фиксированные или стабильные связи между единицами статистического наблюдения, а также между их отдельными аспектами, выраженными соответствующими статистическими показателями;
- **Динамическую структуру ССП**, которая отражает переменные, динамические отношения, представляющие корреляционные и причинно-следственные связи между параметрами единиц статистического наблюдения и объектов СЭС. Эти отношения, как правило, определены между факторными и результирующими показателями ССП.

Отношение причинной связи, положенное в основу определения динамической структуры статистической информации, непосредственно связано с понятием случайности. Следуя требованиям к языку оснований общей теории статистики, мы должны дать конструктивное определение и этому понятию. Однако, решение этой проблемы выходит за рамки собственно статистики и требует рассмотрения случайности как философской категории.

В качестве одного из возможных подходов к конструктивному определению причинности можно найти в работах P.Suppes. В ссылке [2] вы можете найти краткое изложение идеи P.Suppes и библиографию, посвященной этой теме.

Формальное определение систем статистической информации

Дискретный характер статистического наблюдения позволяет выделить, как мы уже отмечали, **статическую** (отражающую текущее состояние статистической информации) и **динамическую** структуры статистической информации.

Динамическую структуру естественно рассматривать как совокупность связей, отражающих зависимость каждого текущего состояния статистической

информации(точнее, статической структуры статистической информации в конкретный момент времени) от состояния статистической информации в предыдущие моменты времени. Дгугими словами, мы определяем **динамическую структуру** как **статистический аналог причинно-следственных связей**.

Основу системы статистической информации (ССИ) образует множество статистических показателей. В свое время М.А.Королев формализовал определение статистического показателя как **составную единицу информации** (СЕИ)

$$\mathbf{S.(P(1), P(2), P(3), \ldots, P(k - 1), Q),} \qquad (1)$$

в которой P(1), P(2), . . . , P(k-1) - соответствуют реквизитам-признакам, а Q - реквизиту-основанию.

Статическая структура отражает ассоциативные связи между показателями, часть из которых может быть установлена на основе анализа приведенного описания показателя (1).

Так например, два показателя, у которых значения реквизита-признака “министерство” совпадают, ассоциативно связаны, поскольку характеризуют объекты, относящиеся к одному министерству. Аналогичным образом могут быть определены ассоциативные связи и по другим реквизитам-признакам.

Статистические группировки представляют собой другой способ определения статической структуры. Любая статистическая группировка, так же как и показатель, может быть представлена как СЕИ. В этом случае каждый элемент СЕИ будет представлен как список значений соответствующего реквизита. **Формально СЕИ группировки представляет собой матрицу**, в которой колонки соответствуют реквизитам, а строчки - отдельным значениям реквизитов в каждом показателе, т.е. колонки - это реквизиты, а строчки - показатели.

Две статистические группировки связаны ассоциативно, если в их описании существуют одноименные составляющие, и пересечение множества значений составляющей из одной группировки с множеством значений из одноименной составляющей из другой группировки не пусто.

В качестве формальной модели представления ассоциативной части статической структуры статистической информации может быть использовано отношение сходства (толерантности) [3].

Отношение толерантности определено на множествах, но, как мы видели, в случае СЕИ, которая представляет группировку, мы имеем случай совокупности множеств. В данном случае для описания ассоциативных связей мы вынуждены использовать совокупность отношений толерантности, в которой для каждого реквизита СЕИ будет определено свое

отношение толерантности. Мы будем называть это отношение **k-толерантностью**, где k - индекс соответствующего реквизита и, связанного с ним, отношения толерантности.

Отношение k-толерантности является рефлексивным, симметричным и нетранзитивным отношением и представлено совокупностью помеченных бинарных отношений толерантности. По-существу, отношение k-толерантности представляет собой случай многомерной толерантности.

Как мы уже отмечали, мы рассматриваем динамическую структуру ССИ как совокупность причинно-следственных связей, определенных на множестве элементов ССИ. При построении конкретных алгоритмов определения причинно-следовательных связей мы будем опираться на идеи P.Suppes [4].

В своих исследованиях P.Suppes в первую очередь интересуется анализом причинно-следственных связей между событиями, используя понятие события из теории вероятностей. Соответственно, события являются подмножеством фиксированного вероятностного пространства.

Важно отметить, что он рассматривает их как мгновенные и включает время их возникновения в их формальные значения.

Таким образом,

1. **P(At)** - это вероятность того, что событие A произойдет в момент времени t,
2. **P(At|Bt')** - вероятность того, что событие A произойдет в момент времени t при условии, что событие B уже произошло в более ранний момент времени t'.

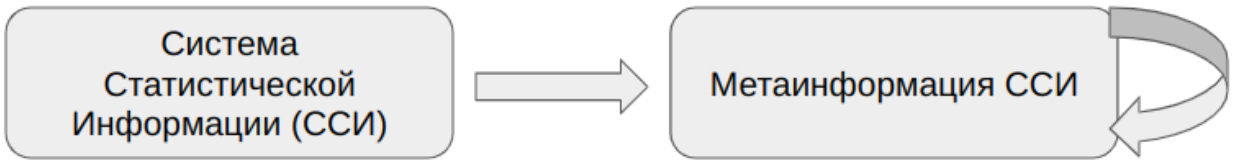
Определение **первопричины** (*prima facie*) в этом случае будет выглядеть следующим образом. Событие Bt' первопричина, если выполняются следующие условия:

1. $t' < t$;
2. $P(Bt') > 0$;
3. $P(At \mid Bt') > P(At)$.

Мы остановимся на вопросах построения динамической структуры ССИ в следующем разделе этой статьи.

От статистических данных к метаданным

Метаданные — это данные о данных. Данные — это все, что может быть представлено на компьютере в цифровом виде. С точки зрения метаданных нет никакой разницы между разными типами данных, такими как: документы, базы данных, изображения, видео, аудио, сигналы датчиков — всё это данные.

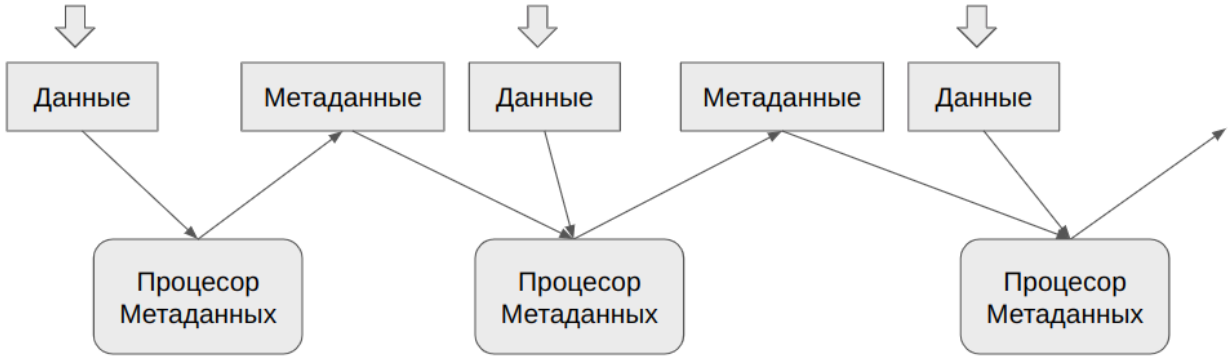


Но что мы можем сказать о метаданных для метаданных? Метаданные - это данные, т.е. **Метаданные от метаданных - это те же метаданные.**

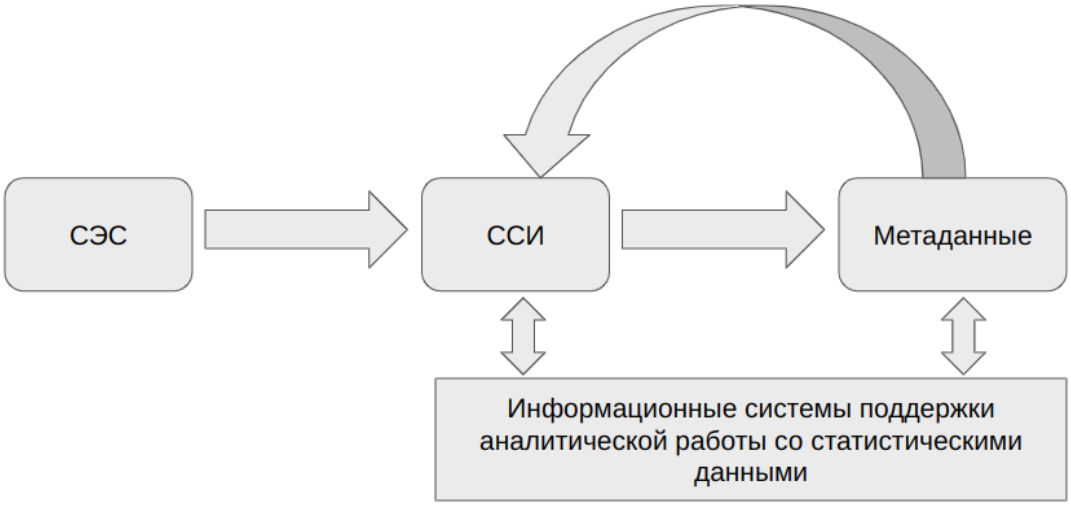
Последнее замечание очень важное. **Мы постулируем, что мы будем иметь дело только с одной системой метаданных для работы как с данными, так и с метаданными, сгенерированными на основе анализа данных (и метаданных)**.

На рисунке мы иллюстрирует этот постулат замыканием метаданные самих на себя.

Ниже мы иллюстрируем развернутый во времени процесс генерации метаданных из данных и из метаданных, сгенерированных в предыдущий момент времени.



Соединяя отображение социально-экономических систем (СЭС) в системы статистической информации (ССИ), реализованное как статистическое наблюдение, и отображение ССИ в Метаданные мы получаем следующую обобщенную схему.



На этой схеме отображения СЭС → ССИ и ССИ → (Метаданные), как мы отмечали выше, являются однозначными и сохраняют структурные связи отображаемой системы в соответствующем отображении.

Эта схема также подчеркивает важное системное качество статистики как системы - замкнутость относительно системы статистической информации (ССИ). **Любые данные входящие в систему или произведенные системой в процессе обработки включаются в ССИ как часть. Это включение, в свою очередь, является однозначным и сохраняющим структуру исходных данных.**

В настоящее время Статистическая информация включает в себя не только статистические показатели и разработочные таблицы, для презентации статистических данных и результатов статистического анализа привлекаются практические все доступные на компьютере средства, а это значит все - от таблиц и диаграмм до мультимедиа и виртуальной реальности.

С точки зрения метаданных - это всё данные, и для этих данных мы должны сгенерировать описания в форме элементов метаданных, и установить отношения (связи) между этими элементами.

Формально мы можем определить Метаинформацию как граф, в котором элементы метаинформации будут представлены узлами графа, а отношения между элементами будут определять ребра этого графа.

Графовая База Данных для представления Метаданных

Граф может быть определен следующим образом:

G = {V, E}, где

G - граф;
V - множество узлов графа, которые в случае систем статистической информации, будут представлять собой составные единицы информации (СЕИ);
E - множество ребер, соединяющих связанные узлы графа.

Узел **v** ∈ **V,** как мы отмечали, представляет собой СЕИ и может быть описан как **struct** в языках программирования C++, Rust, Julia или **Dict** в языке Python. В приведенном ниже фрагменте кода, мы использовали нотацию языка Julia.

```
struct Node <: AbstractGraphType
    sha1::String
    labels::Vector{String}
    d_sha1::String
    card::Int
    dataset::Vector{Int}
    props::Config
end
```

Приведенная структура может рассматриваться как естественное расширение СЕИ, в котором мы дополнили каждый реквизит указанием на его тип. **Эта структура является стандартной для представлением любых узлов графа**. Набор реквизитов СЕИ минимальный и служит для обеспечения однозначной идентификации каждого узла графа.

В этой структуре мы выделяем три части:

1. Внешнее описание узла, которое, в свою очередь, представлено двумя атрибутами:
    a. **sha1** - уникальный идентификатор (SHA1 хэш);
    b. **labels** - ярлыки (один или несколько), которые отражают семантику данного узла;
2. Внутреннее описание узла, которое описывает содержание СЕИ, представленного данным узлом. Атрибуты этого описания включают:
    a. **d_sha1** - SHA1 хэш сгенерированная из содержимого СЕИ;
    b. **dataset** - векторизованное представление содержимого СЕИ (мы остановимся на векторном представлении СЕИ более подробно в следующем разделе);
    c. **card** - количество неповторяющихся элементов СЕИ;
3. **props** - дополнительные атрибуты узла, представленные как JSON структура.

Ребра графа описывают связи между узлами. Любая пара узлов может иметь больше чем одно ребро и каждое ребро имеет направление.

```
struct Edge <: AbstractGraphType
    source::String
```

```
    target::String
    r_type::String
    props::Config
end
```

Атрибуты описания ребра:

1. **source** - sha1 идентификатор начального узла;
2. **target** - sha1 идентификатор конечного узла;
3. **r_type** - ярлык ребра, отражающий тип связи между узлами;
4. **props** - дополнительные атрибуты ребра, представленные как JSON структура.

Множество всех значений, которые могут принимать реквизиты-признаки и реквизиты-основания, представляют собой словарный запас ССИ или просто словарь ССИ. Этот словарь не является фиксированным, а постоянно развивающимся множеством, которое включает новые понятия и термины по развития наблюдаемой совокупности и самой ССИ.

Для описания словаря ССИ мы используем следующую структуру:

```
struct Token <: AbstractGraphType
    id::Int
    bin::Int
    zeros::Int
    token::Set{String}
    tf::Int
    refs::Set{String}
end
```

В этой структуре:

1. **id** - целое число, которое вычисляется как хэш от значения слова (токена). Слово - это любой код, который представляет элементарную единицу данных вне зависимости от типа данных;
2. **bin** - k начальных битов в хэш коде, которые соответствуют позиции данного слова в векторном представлении СЕИ (подробнее в следующем разделе);
3. **zeros** - количество непрерывно повторяющихся нулей подсчитанных от конца битового представления хэши;
4. **tf** - частота появления данного слова в СЕИ (term frequency);
5. **refs** - список sha1 идентификаторов всех СЕИ, в которых появляется данное слово.

Приведенные выше три структуры представляют собой базу графового представления системы статистической информации (ССИ).

HllSet как способ аппроксимации содержимого СЕИ

Алгоритм аппроксимации СЕИ базируется на известном алгоритме вероятностной оценки количества неповторяющихся элементов в очень больших совокупностях данных, который известен как HyperLogLog алгоритм.

Определение HyperLogLog из Wiki [5]:

> В основе алгоритма HyperLogLog лежит наблюдение о том, что мощность мультимножества равномерно распределенных случайных чисел можно оценить путем вычисления максимального количества ведущих нулей в двоичном представлении каждого числа в наборе. Если максимальное количество наблюдаемых ведущих нулей равно n, оценка количества различных элементов в наборе равна 2^n .[6]

В алгоритме HyperLogLog к каждому элементу исходного мультимножества применяется хэш-функция для получения мультимножества равномерно распределенных случайных чисел с той же мощностью, что и исходное мультимножество. Затем мощность этого случайно распределенного набора можно оценить с помощью приведенного выше алгоритма.

Простая оценка мощности, полученная с помощью приведенного выше алгоритма, имеет недостаток, заключающийся в большой дисперсии. В алгоритме HyperLogLog дисперсия минимизируется путем разделения мультимножества на множество подмножеств, вычисления максимального количества ведущих нулей в числах в каждом из этих подмножеств и использования гармонического среднего для объединения этих оценок для каждого подмножества в оценку мощность всего набора.[7]

Мы опускаем технические детали реализации, которые вы можете найти в ссылках [8, 9]. Мы назвали новую структуру **HllSet** (HyperLogLog Set).

Как следует из описания оригинального алгоритма, размер HllSet постоянен и не зависит от размера исходных данных. Другими словами, размер HllSet будет одинаковым для СЕИ, представляющим один показатель, и СЕИ, представляющей группировку из нескольких тысяч показателей. Размер HllSet зависит только от количества подмножеств (регистров), которые мы хотим использовать для представления зашифрованных данных. В случае 64-битовых процессоров, в которых целое число представлено 8 байтами, размер HllSet можно определить по формуле:

$$\mathbf{M = 8 \times || S ||}, \text{ где}$$

|| S || - количество подмножеств, которые представляют регистры.

На множестве HllSet определены все стандартные операции на множествах и эти операции удовлетворяют всем требованиям теории множеств.

```
 Z - пустое множество
 U - унивесальное множество, в практических приложениях это объединение
всех HllSet определенных в конкретном приложении

 1. (A ∪ B) = (B ∪ A): true                    // коммутативность
 2. (A ∩ B) = (B ∩ A): true
 3. (A ∪ B) ∪ C) = (A ∪ (B ∪ C)): true         // ассоциативность
 4. (A ∩ B) ∩ C) = (A ∩ (B ∩ C)): true
 5. ((A ∪ B) ∩ C) = (A ∩ C) ∪ (B ∩ C): true    // дистрибутивность
 6. ((A ∩ B) ∪ C) = (A ∪ C) ∩ (B ∪ C): true
 7. (A ∪ Z) = A: true                          // идентичность
 8. (A ∩ U) = A: true
 9. (A ∪ A) = A: true
10. (A ∩ U) = A: true
```

Использование HllSet для представления табличных данных

Формально табличные наборы данных можно описать как подмножество декартова произведения двух наборов:

$$R(S_1, S_2) \subseteq S_1 \times S_2$$

Однако, в случае HllSet у нас нет доступа к элементам множеств, поэтому нам нужно использовать другой подход для создания и использования табличных структур.

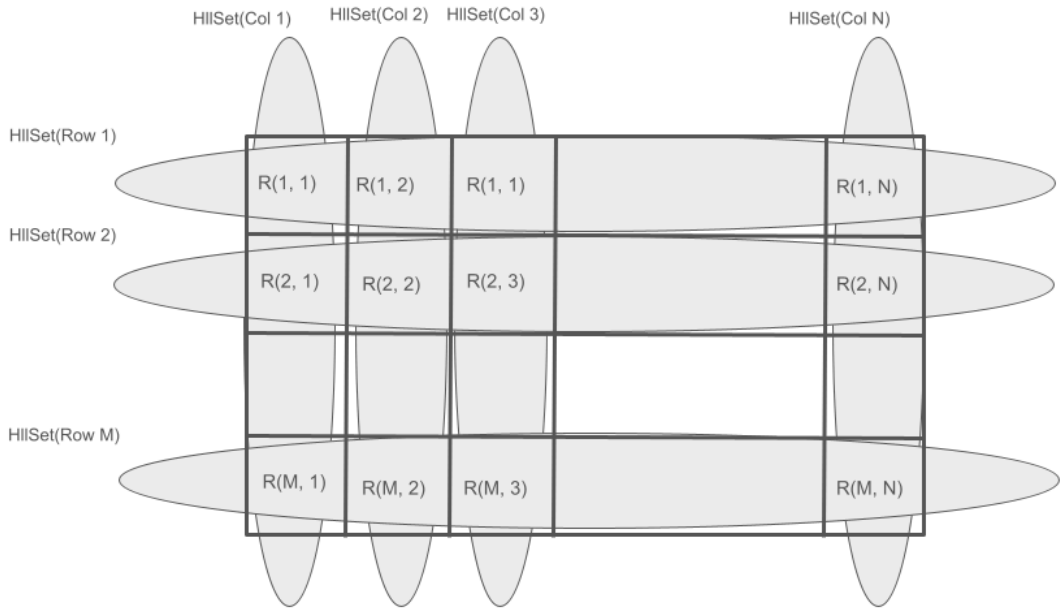


Рисунок выше — иллюстрация того, как мы можем создавать бинарные отношения между HllSets. Например, в случае файлов CSV это будет работать следующим образом:

- $\mathbf{S_1}$ - коллекция HllSets, представляющих строки из CSV файла;
- $\mathbf{S_2}$ - коллекция HllSets, представляющих столбцы CSV файла;
- Каждая ячейка табличного представления представляет собой пересечение строки HllSet и столбца HllSet. Итак, это пересечение — HllSet.

Все элементы, представленные HllSet, присутствуют в общем словаре ССИ (tokens), и они внесены в словарь с теми же идентификаторами, которые мы использовали при построении HllSet. Это означает, что мы всегда можем отследить содержимое любого HllSet до соответствующих ему слов в словаре ССИ, и восстановить содержимое ячейки.

При работе с HllSets следует учитывать следующие вещи:

- Содержимое ячейки представляет собой неупорядоченную коллекцию токенов из исходной ячейки CSV-файла. Итак, если вы хотите знать порядок этих токенов, вам следует перейти к исходному CSV файлу.
- Во многих случаях мы используем только выборку строк т.е. мы должны помнить об этом, когда работаем с этими данными.

Коррекция обычно проста:

- Чтобы получить фактический HllSet для столбца, мы должны использовать объединение HllSets ячеек данного столбца;
- То же самое следует сделать и с наборами строк HllSet.

Пример использования метаданных для поиска в ССИ

Мы рассмотрим два примера использования метаданных:

1. Восстановление табличных данных из CSV файлов;
2. Поиск CSV файлов и колонок в них, которые содержат слова из запроса.

Восстановление табличных данных

```
'''
Вначале мы должны сгенерировать метаданные для колонок и строк файлов, с которыми мы
собираемся работать.
В процессе обработки мы добавим метаданные в две основные таблицы Графовой Базы
Данных:
- nodes;
- tokens.
В таблице nodes мы разместим данные о строчках и столбцах для каждого файла. Каждая
строчка и столбец получат уникальный SHA1 идентификатор.
Все новые слова из файлов будут занесены в таблицу tokens с указанием SHA1
идентификаторов строк или столбцов файлов, из которых эти слова были выбраны.
Это установит связь между токенами (словами) и элементами файлов, в которых они были.

db - Графовая База Данных;
row.name() - функция, которая извлекает полное имя файла из row объекта, который
представляет собой файл.
'''
for row in eachrow(file_list)
    Store.ingest_csv_by_row(db, row.name())
end

for row in eachrow(file_list)
    Store.ingest_csv_by_column(db, row.name())
end
```

Графовая База Данных (объект Store) имеет специализированные методы для извлечения матриц данных из пересечений строк и столбцов CSV файлов.

```
"""
    Здесь мы собираемся извлечь узлы строк и столбцов из файла csv.
    Полученная матрица покажет количество элементов в пересечение узлов строки   и
столбца.
"""
matrix = Store.get_card_matrix(db, source_id)
for row in eachrow(matrix)
    println(row)
```

```
end
```

Вывод после исполнения этой программы:

```
[2.0, 2.0, 2.0, 1.0, 2.0, 4.0, 3.0, 3.0]
[2.0, 2.0, 1.0, 1.0, 2.0, 4.0, 5.0, 4.0]
[2.0, 3.0, 2.0, 1.0, 1.0, 4.0, 4.0, 4.0]
[2.0, 2.0, 2.0, 1.0, 1.0, 4.0, 4.0, 3.0]
[2.0, 2.0, 2.0, 1.0, 2.0, 4.0, 3.0, 3.0]
[2.0, 3.0, 2.0, 1.0, 1.0, 4.0, 3.0, 3.0]
[2.0, 2.0, 2.0, 1.0, 2.0, 4.0, 3.0, 3.0]
[2.0, 3.0, 2.0, 1.0, 2.0, 4.0, 3.0, 3.0]
[2.0, 2.0, 1.0, 1.0, 2.0, 4.0, 3.0, 3.0]
. . .
```

Как мы видим многие ячейки матрицы имеют несколько элементов.

```
matrix = Store.get_value_matrix(db, source_id)
for row in eachrow(matrix)
  println(row)
end
```

А это вывод из этой программы (показаны только несколько первых строк вывода):

```
["[\"Serious\"]", "[\"Pedestrian\"]", "[\"Male\"]", "[]", "[\"Friday\"]",
"[\"Kensington\",\"Chelsea\",\"and\"]", "[\"Motorcycle\",\"over\",\"and\"]",
"[\"Not\",\"Pedestrian\",\"pedestrian\"]"]
["[\"Serious\"]", "[\"Pedestrian\"]", "[\"Female\"]", "[]", "[\"Wednesday\"]",
"[\"Kensington\",\"Chelsea\",\"and\"]", "[\"car\",\"Taxi/Private\",\"and\",\"hire\"]",
"[\"crossing\",\"Pedestrian\",\"Crossing\",\"ped.\",\"facility\"]"]
["[\"Serious\"]", "[\"rider\",\"Driver\"]", "[\"Male\"]", "[]", "[\"Monday\"]",
"[\"Kensington\",\"Chelsea\",\"and\"]", "[\"cycle\",\"Motor\",\"and\",\"under\"]",
"[\"crossing\",\"carriageway\",\"elsewhere\"]"]
["[\"Serious\"]", "[\"Pedestrian\"]", "[\"Male\"]", "[]", "[\"Monday\"]",
"[\"Kensington\",\"Chelsea\",\"and\"]", "[\"cycle\",\"Motor\",\"and\",\"under\"]",
"[\"Not\",\"Pedestrian\",\"pedestrian\"]"]
["[\"Serious\"]", "[\"Pedestrian\"]", "[\"Male\"]", "[]", "[\"Sunday\"]",
"[\"Kensington\",\"Chelsea\",\"and\"]", "[\"Car\",\"and\"]",
"[\"Not\",\"Pedestrian\",\"pedestrian\"]"]
["[\"Serious\"]", "[\"rider\",\"Driver\"]", "[\"Male\"]", "[]", "[\"Tuesday\"]",
"[\"Kensington\",\"Chelsea\",\"and\"]", "[\"Motorcycle\",\"over\",\"and\"]",
"[\"Not\",\"pedestrian\"]"]
. . .
```

Поиск узлов графа, содержащий слова из запроса

В этом примере мы использовали Neo4J Graph, в качестве сервера для нашей Графовой Базы Данных. Для работы с Neo4J мы используем специально разработанный интерфейс LisaNeo4J. Этот интерфейс содержит базовую поддержку для работы с графами, включая поддержку для поиска данных.

Ниже приводится образец программы, в котором мы используем упрощенный Julia код для описания основных шагов обработки запроса.

```
# Поиск узлов и ребер (SHA1 идентификаторов), которые содержат "sex", "taxi" and "day"
rows = LisaNeo4j.search_by_tokens(db.sqlitedb, "sex", "taxi", "day")
# Выборка всех ссылок на ребра
edges = Vector()
edges_refs = LisaNeo4j.select_edges(db.sqlitedb, rows, edges)

# Выборка всех ссылок на узлы
nodes = Vector()
LisaNeo4j.select_nodes(db.sqlitedb, refs, nodes)

# Построение презентации в Neo4J
# 1. Начнем с узлов
for node in nodes
  labels = replace(string(node.labels), ";" => "")
  query = LisaNeo4j.add_neo4j_node(labels, node)
  data = LisaNeo4j.request(url, headers, query)
end
# 2. После этого соединим узлы ребрами
for edge in edges
  query = LisaNeo4j.add_neo4j_edge(edge)
  data = LisaNeo4j.request(url, headers, query)
end
```

Результаты поиска позволяют нам провести базовую аналитику. В этом примере мы делаем попытку оценки связей между CSV файлами, которые удовлетворяют требованиям нашего запроса. Мы рассчитываем две метрики:

1. Jaccard расстояние и
2. Cosine толерантность (похожесть).

```
# Определение Cypher запроса
query = LisaNeo4j.cypher("MATCH (n:csv_file) RETURN n.labels, n.sha1, n.d_sha1,
n.dataset, n.props LIMIT 20")

hlls = LisaNeo4j.collect_hll_sets(query, hlls)
```

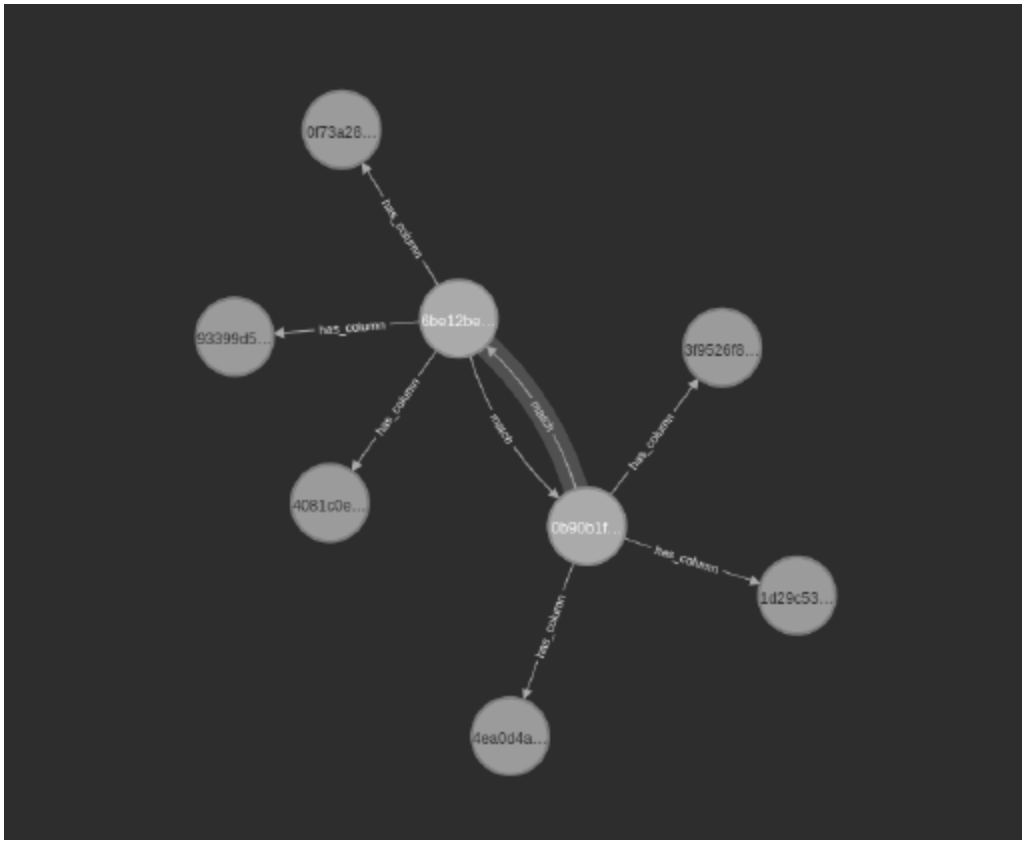
```
# Определение отношений между связанными csv файлами
for (k, v) in hlls
  for (k1, v1) in hlls
    if k != k1
      jaccard = SetCore.jaccard(v.hll_set, v1.hll_set) # jaccard сравнение
       println("jaccard: ", jaccard)
      cosine = SetCore.cosine(v.hll_set, v1.hll_set)    # cosine сравнение
       println("cosine: ", cosine)
    end
  end
end
```

А это вывод из этой программы.

```
jaccard: 78
cosine: 87.0
jaccard: 78
cosine: 87.0
```

В нашем примере мы использовали только 2 файла, поэтому мы получили только одну связь, измеренную с использованием двух методов измерения. Необходимо также отметить, что мы получили две связи для каждой метрики. Это связано с тем, что наш граф ориентированный. Симметричные связи в ориентированных графах представлены двумя ребрами: из первого узла во второй и из второго - в первый. Другое важное наблюдение - это разница значений для этих двух метрик. Нужно отметить, что в моделях машинного обучения и нейронных сетях ИИ, **cosine** является одной из наиболее часто используемых метрик и считается более адекватным измерителем близости сравниваемых объектов.

Иллюстрация графа в Neo4J браузере:



Полные тексты программ и дополнительная документация могут быть найдены по ссылке [9].

Ссылки

1. Johann Pfanzagl. Theory of Measurement. Springer-Verlag Berlin Heidelberg 1971
2. https://durham-repository.worktribe.com/preview/1421323/17400.pdf
3. https://ru.wikipedia.org/wiki/Отношение_толерантности
4. Suppes: Scientific Philosopher, Volume 1. Probability and Probabilistic Causality, ed. P. Humphreys. 339-366. Dordrecht: Springer.
5. https://en.wikipedia.org/wiki/HyperLogLog
6. https://algo.inria.fr/flajolet/Publications/FlFuGaMe07.pdf
7. https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/40671.pdf
8. https://github.com/alexmy21/lisa/blob/main/README.md
9. https://github.com/alexmy21/lisa_meta