Проект по Мрежово програмиране, Проект 1.133

Александър Бориславов Канджички, КН 3.2, 81427

# Съдържание:

* Тема
* Описание на решението.
* Проектиране
* Реализация
* Ръководство за инсталация.
* Резултати
* Литература
* Приложение

# Тема

клиент и сървър приложения чрез TCP протокола за търсене на зависимост на поведение на потребител.

Клиентът изпраща до сървър лог файл със структура:

Time;Event context;Component;Event name;Description;Origin;IP address

Сървърът анализира поведението на потребителите на база данните от файла с помощта на откриване на шаблон на данните чрез реализирането на  **Perform Sequence Prediction using the CPT+ Sequence Prediction Model** алгоритъм за извличането на данни.

Търсят се зависимости в поведението на навигирането на страниците на потребителите. Алгоритъма се опитва да познае коя ще бъде следващата отворена страница, с цел подобряване на сайта и евентуално кеширане/preloading на ресурсите

# Решение

Данните от колона **Time** се разделят на две компоненти - **date** и **hours**.

Данните изпратени от потребителя се групират по два критерия: **user IP address** и **Time(date)**. Всяка група се трансформира в последователност от отворени страници (взема се **Event context** колона) и тази последователност се трансформира в удобен вид за обучение на алгоритъма. След това всички последователности от групите се подават на алгоритъма за обучение.

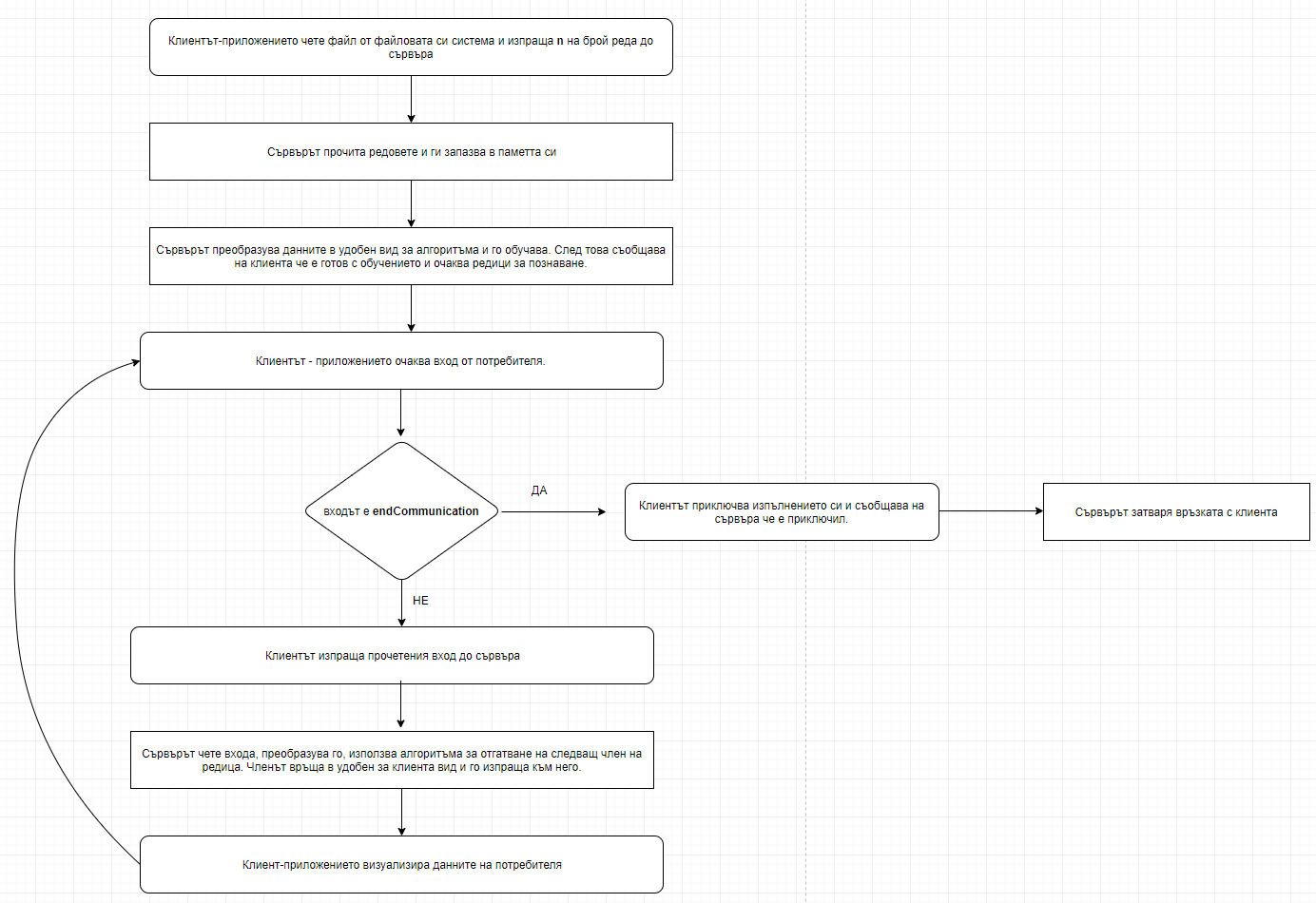
При приключване на обучението, сървърът очаква потребителя да подава редици от **Event context** данни и отговаря с предположението за бъдещи елементи за продължение на тази редица.

# Проектиране

* Клиентът ще изпрати всички свои данни, ред по ред към сървъра. Той ще ги запази, ще обучи с тях алгоритъма и след това ще очаква клиента да подава редици. На всяка подадена редица, сървърът ще се опитва да познае следващ елемент и ще го изпраща на клиента.
* Нужна ни е библиотеката **SPMF** - алгоритъм за **sequence prediction**
* Нужни са ни отделни сървър, клиент, клас който ще съхранява данните от клиента в удобен формат, както и клас който ще използваме за обучаване на алгоритъма

# Реализация

* Създава се отделен клиент и сървър, клас **DataEntry** който ще съхранява и трансформира всеки ред от подадените данни, както и клас **MiningHelper** който ще се грижи за извличането на данните в удобен за алгоритъма вид, обучение на алгоритъма и предвиждане на данни.
* Главната функция създава комуникация между клиента и сървъра. Клиента изпраща данните ред по ред, сървъра ги чете и запазва в отделни **DataEntry** инстанции. След това използва **MiningHelper** за обработка на данните и съобщава на клиента при приключване на обучението, че е готов и очаква последователности които да предвижда. Получава такива докато не получи **endCommunicationString**, при което комуникацията приключва и клиента и сървъра приключват работа.
* Ще използваме **алгоритъм CPT+**  и класовете от папка **database** за подготовка на данните в удобен вид за **CPT+**
* **MiningHelper** трябва да може да трансформира данните от потребителя в числа и обратно във всеки един момент. При вземането им от потребителя, той трябва да преведе всичко до числови редици които да използва за обучение на алгоритъма.
* При получаване на редица от потребителя, **MiningHelper** може да я трансформира до числова, да вземе предположение от алгоритъма и да я върне обратно в удобен за потребителя вид.



# Ръководство за инсталация.

В главната папка се създава **csv** файл и данните се поставят в указания формат. Приложението се стартира с начална точка **main.java** и комуникацията се извършва чрез стандартния вход/изход - конзолата.

# Резултати

Потребителя може да използва алгоритъма за познаване на следващи членове на поредици от отворени страници/файлове.

# Литература

* Документация и алгоритми от **SPMF:**  <http://www.philippe-fournier-viger.com/spmf/>
* Работа с потоци, функционални **map, filter, foreach, collect** за Java: <https://stackoverflow.com/questions/43877296/java-map-an-array-of-strings-to-an-array-of-integers>
* Четене и писане при I/O потоци между сървър и клиент: <https://stackoverflow.com/questions/8560395/how-to-use-readline-method-in-java>
* Създаване на сървър и клиент и свързването им: <https://medium.com/@martinomburajr/java-create-your-own-hello-world-server-2ca33b6957e>

# Приложение

<https://github.com/aleksandar-kandzhichki/java_homework_projects>