PatientSim: Optimizing Patient Similarity Analysis

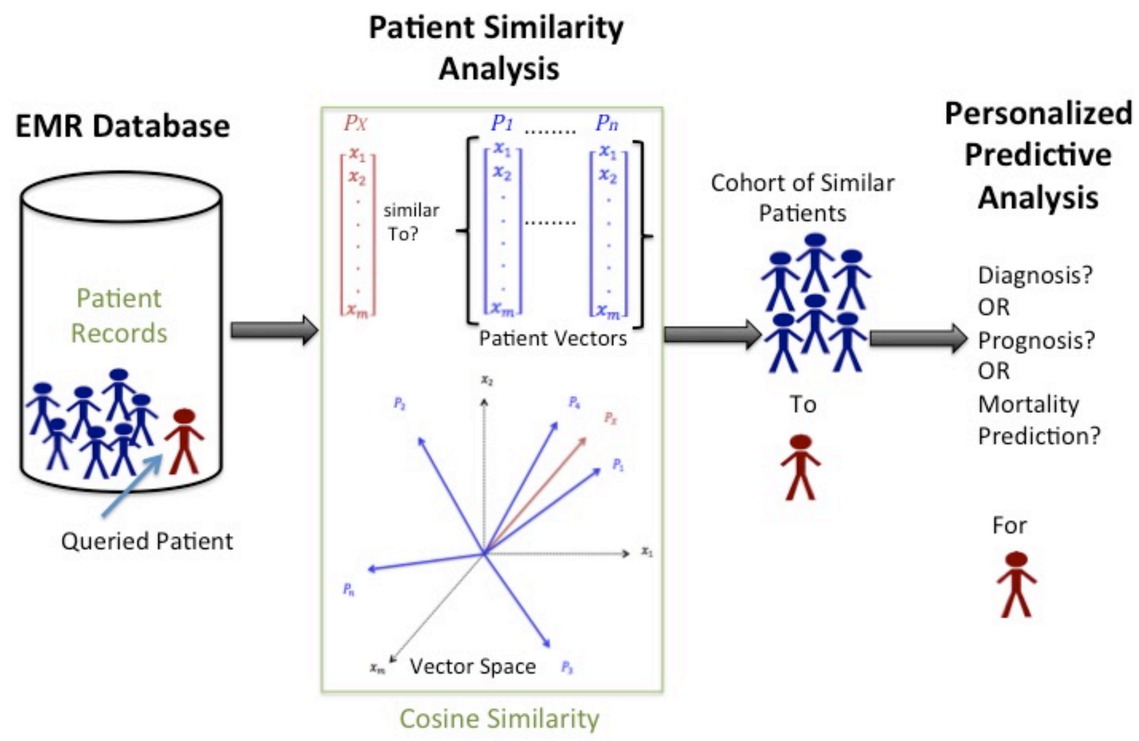
Ingmar Wiese, Nicole Sarna, Lena Wiese, Araek Tashkandi
Research Group Knowledge Engineering
Institute of Computer Science
Georg-August-University
Göttingen, Germany

Abstract

This project focuses on a comprehensive hardware-backed analysis of biomedical data sets. The efficient implementation of such an analysis is a major precondition for large-scale evaluation of biomedical data that will be necessary in future applications in the area of personalized medicine. We applied cosine similarity to a commonly used medical dataset. Performance tests were carried out inside a HANA database instance as well as in a distributed VM environment with a machine learning toolkit.

1. Introduction

The focus of the PatientSim project lies in the area of efficient computation of patient similarity: that is, for a given target patient a doctor wants to find a group of patients (a so-called "cohort") having similar features (like for example, displaying similar symptoms or having similar lab measurements). By focusing on the cohort, a doctor can decide on an appropriate treatment based on the identified prior experiences or predict future health conditions of the target patient.



2. Method

For two patients P_1 and P_2 we compute the cosine similarity

$$CosineSimilarity = cos(\theta) = \frac{P_1 \cdot P_2}{||P_1||||P_2||}$$

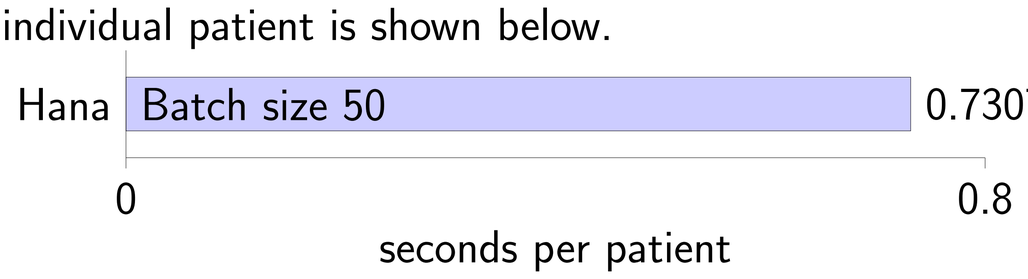
As a test dataset we used the diabetes dataset (see https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008) from the UCI Machine Learning Repository.

3. Results

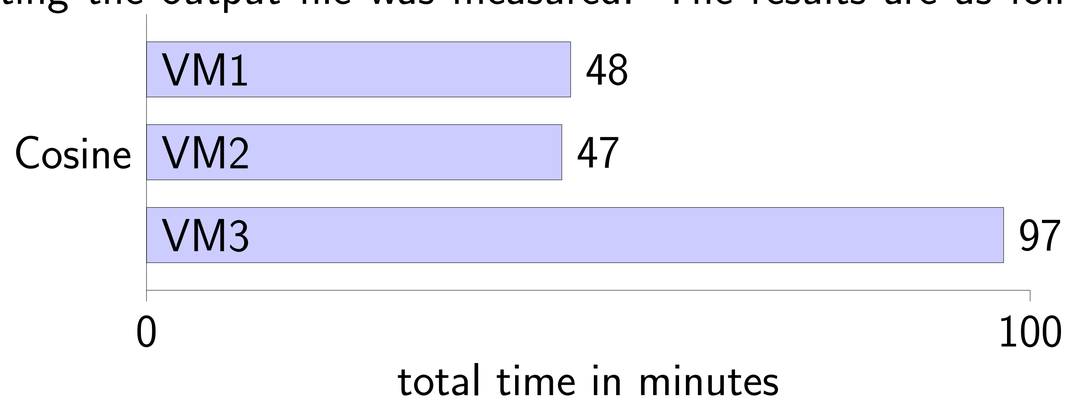
In HANA the data set was imported into a column table. The norm $\|P\|$ for each patient P was calculated beforehand and stored in an additional column. The cosine similarity was run as a SQL code shown below

```
SELECT p1.id, p2.id
       (p1.pid_1 * p2.pid_1 +
        ...
        p1.pid_m * p2.pi_m)
       / (p1.norm *p2.norm)
FROM   patients p1
       JOIN patients p2
       ON p1.id < p2.id;
```

We applied batch processing in the sense that the similarity for 50 patients at the same time was calculated. The runtime per individual patient is shown below.



The overall computation of all pairwise similarities took roughly 20 hours which leaves room for optimization.
In the three provided FSOC VMs, we utilized the Environment for Developing KDD-Applications Supported by Index-Structures (ELKI, https://elki-project.github.io/).
To calculate the distances between each patient vector with another efficiently, we let the three VMs calculate the vector distances in parallel. The time for calculating and writing the output file was measured. The results are as follows:



Contact

Research Group Knowledge Engineering
Institut für Informatik
Georg-August-Universität Göttingen
Goldschmidtstraße 7, 37077 Göttingen
Homepage: http://wiese.free.fr
Book page: http://wiese.free.fr/adm.html