EGCO 622 Data Mining

Association Analysis: Cardiac SPECT Diagnosis

Submitted to :

Ajarn Tanasanee Phienthrakul

Ajarn Vasin Suttichaya

By :

In-on Wiratsin 6138020

Content

Chapter 1

Introduction

Background

Perfusion is the process for delivering blood to a capillary bed, a small blood vessel that exchanges many substances with the interstitial fluid surrounding them. The perfusion information can be used in diagnosing, since diseases can be better distinguished and detected at an earlier stage. One of the method for obtaining heart perfusion information is Single Photon Emission Computed Tomography (SPECT)

SPECT is the medical imaging technique that uses gamma rays to modeling 3D information of organ. Heart SPECT data can illustrates many areas in the heart that have low blood flow. This information helps the doctor to identify many cardiovascular diseases. Early detection of cardiovascular diseases plays a huge role in improving the efficiency of their treatment.

However, it is very difficult for a human to analyze SPECT data since it contains a lot of cross section images. It is hard to understand basic trends in data to make rational decisions. Moreover, this data turns out to be less useful because it is hard to transform into intelligible format. For this reason, we require the data analyzing methods for supporting the creation of knowledge that can be used for clinical decision making.

Data mining and knowledge discovery can help the cardiovascular diseases diagnosis process by analyzing Heart SPECT data, extracting useful information, and generating the relationship among the attributes. Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. The algorithm starts by generating frequent itemsets. These itemsets will be used to determine association rules that emphasize general trends that appear in the data.

The purpose of this work was to develop an analysis approach based on Apriori association analysis to establish rules for each type of patients using heart images obtained from SPECT.

Objective

To use data mining technique with SPECT dataset in order to perform Apriori association analysis on each type of patients, and find 5 rules for each type of patients.

Scope

- Datasets was retrieved from http://archive.ics.uci.edu/ml/datasets/SPECT+Heart:
    - Given 2 datasets which are SPECT.train and SPECT.test
- Total No.of records: 267 patients
- Attributes: 23 attributes: Binary attributes 0 1
    - 1st attribute: healthy heart (diagnosis=0), unhealthy heart (diagnosis=1)
    - 2nd - 23rd attributes: high perfusion (diagnosis=0), poor perfusion (diagnosis=1)

Expected result

- The set of rules with high confidence value for patients that has normal cardiac health.
- The set of rules with high confidence value for patients that has abnormal cardiac health.

Chapter 2

Data Interpreting

Data background

The datasets were retrieved from UCI Machine Learning Repository, the original owners are Krzysztof J. Cios and Lukasz A. Kurgan. The datasets covey data regarding diagnosed patients' hearts through the use of SPECT or Single Photon Emission Computed Tomography recorded in the form of images. The process is done by using SPECT imaging to detect the emitted photon from the intravenously injected radioactive agent. The SPECT image sets for each patients contain the images of various cross-sections which is then further divided into parts to examine the perfusion of the tracer agent in each regions of the hearts. The perfusion data is then processed and used to determine the abnormality of the patient's hearts.

Perfusion is the passage of fluid through the circulatory system or lymphatic system to an organ or a tissue, usually referring to the delivery of blood to a capillary bed in tissue. Perfusion is measured as the rate at which blood is delivered to tissue, or volume of blood per unit time (blood flow) per unit tissue mass. The SI unit is m3/(s·kg), although for human organs perfusion is typically reported in ml/min/g. The word is derived from the French verb "perfuser" meaning to "pour over or through". All animal tissues require an adequate blood supply for health and life. Poor perfusion (malperfusion), that is, ischemia, causes health problems, as seen in cardiovascular disease, including coronary artery disease, cerebrovascular disease, peripheral artery disease, and many other conditions. Tests verifying that adequate perfusion exists are a part of a patient's assessment process that are performed by medical or emergency personnel. The most common methods include evaluating a body's skin color, temperature, condition (dry/soft/firm/swollen/sunken/etc), and capillary refill.

A total of 267 patients were diagnosed and categorized into two classes, patient with normal heart and patient with abnormal heart. The full dataset was divided into 2 datasets, the

first dataset “SPECT.train” contains 80 records, and the remaining 187 instances is in the “SPECT.test” dataset.

As an additional information, as conducted by the respective researchers and owner of the datasets, the “SPECT.train” dataset was used by machine learning algorithms such as CLIP3 (a combination of decision tree algorithm C4.5 and rule induction algorithm CN2). The “SPECT.test” dataset was then used to test and evaluate the efficiency and accuracy of the mentioned algorithm. Results are that with CLIP3, it achieved 84 percent accuracy as compared to cardiologists’ diagnoses.

Data Type

There are a total of 23 attributes, with none of the records having any missing values. The first is a binary class attribute which defines the class of the patients, with 0 having normal 1cardiac health and 1 having abnormal cardiac health. The other 22 binary attributes defines the partial diagnoses with 0 having high perfusion and 1 having poor perfusion. The partial diagnoses are as follows:

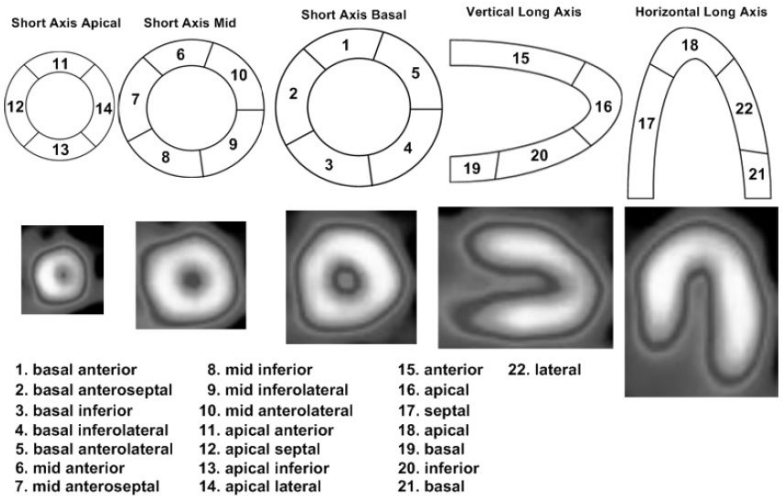


Figure 1.1: Partial SPECT scan of heart

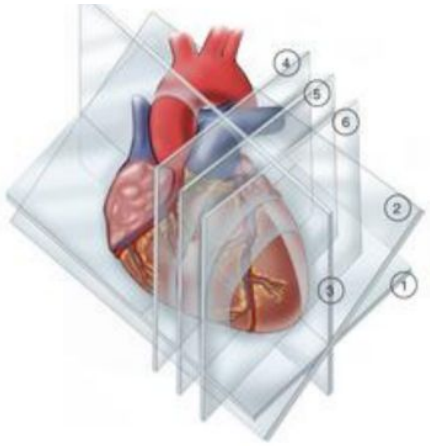


Figure 1.2: Heart cross-sections (1)

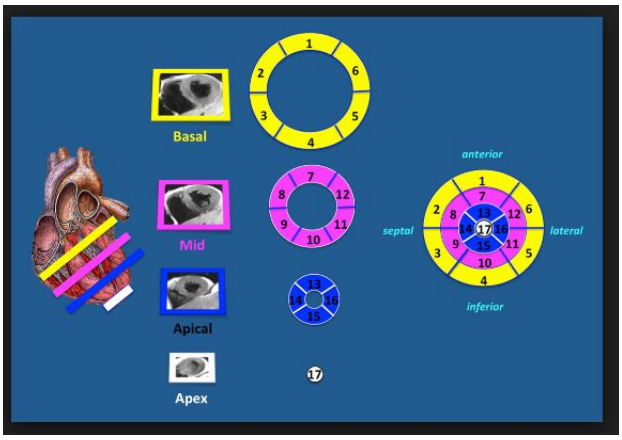


Figure 1.3: Heart cross-sections (1)

Chapter 3

Methodology

Association rule learning

In order to select interesting rules from the set of all possible rules, constraints on various measures of significance and interest are used. The best-known constraints are minimum thresholds on support and confidence.

Support: It is an indication of how frequently the itemset appears in the dataset.

$$\operatorname{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$

Figure 2.1 Support formula

Confidence: The confidence is a percentage value that shows how frequently the rule occurs. The confidence value indicates how reliable this rule is. The higher the value, the more likely the head items occur in a group if it is known that all body items are contained in that group.

$$\operatorname{conf}(X \Rightarrow Y) = \operatorname{supp}(X \cup Y) / \operatorname{supp}(X)$$

Figure 2.2 Confidence formula

Lift: The lift value is the ratio of the confidence of the rule and describes the dependency between A and C. If the rule had a lift of 1, it would imply that A and C are independent of each other. However, if the lift is > 1, it implies positive association, and negative association for lift being <1. This lets us know if both A and C are dependent on one another.

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \times \text{supp}(Y)}$$

Figure 2.3 Lift formula

Leverage: Leverage is similar to lift. Both of them are used to measure the relation between the probability of a given rule. The main difference of them is that leverage tends to prioritize items with higher frequencies or support in the dataset.

$$leve(A \rightarrow C) = sup(A \cup C) - sup(A) \times sup(C)$$

Figure 2.4 Leverage formula

Conviction: Conviction can be interpreted as the ratio of A and C such that (or a chance that) A appears without C.

$$\text{conv}(X \Rightarrow Y) = \frac{1 - \text{supp}(Y)}{1 - \text{conf}(X \Rightarrow Y)}.$$

Figure 2.5 Conviction formula

Apriori Association

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

We Perform Apriori association analysis on each type of patients:

- Run Apriori-algorithm from Python in order to find association rules.
- Remove the first attribute from the dataset which is the attribute of patient who has normal cardiac health (0) and abnormal cardiac health (1).
- Therefore, use the dataset with 22 attributes.
- The best-known constraints are minimum thresholds on support and confidence.
- Set the minimum threshold on support to be 0.5
- Set the minimum threshold on confidence to be 0.9

Run Apriori Algorithm

healthy_SPECT: with minimum support = 0.5 and minimum confidence = 0.9

Example result:

Rule: ('F11_0', 'F2_0') ==> ('F17_0', 'F6_0') , 0.900
Rule: ('F15_0', 'F9_0') ==> ('F14_0',) , 0.900
Rule: ('F2_0', 'F12_0') ==> ('F18_0',) , 0.902
Rule: ('F19_0', 'F18_0') ==> ('F17_0',) , 0.903
Rule: ('F2_0', 'F7_0') ==> ('F18_0',) , 0.903
Rule: ('F19_0', 'F6_0') ==> ('F18_0',) , 0.904
Rule: ('F7_0', 'F12_0') ==> ('F18_0',) , 0.905
Rule: ('F17_0', 'F7_0') ==> ('F12_0',) , 0.905
Rule: ('F7_0', 'F12_0') ==> ('F17_0',) , 0.905
Rule: ('F11_0', 'F2_0', 'F6_0') ==> ('F18_0',) , 0.906
Rule: ('F12_0',) ==> ('F7_0',) , 0.906
Rule: ('F19_0', 'F2_0') ==> ('F18_0',) , 0.907
Rule: ('F19_0', 'F2_0') ==> ('F17_0',) , 0.907
Rule: ('F11_0',) ==> ('F6_0',) , 0.907
Rule: ('F2_0', 'F7_0', 'F12_0') ==> ('F17_0',) , 0.908
Rule: ('F21_0',) ==> ('F18_0',) , 0.908
Rule: ('F2_0', 'F18_0') ==> ('F17_0',) , 0.908
Rule: ('F4_0', 'F15_0') ==> ('F9_0',) , 0.908
Rule: ('F11_0', 'F17_0') ==> ('F18_0',) , 0.910
Rule: ('F18_0', 'F6_0') ==> ('F11_0',) , 0.910
Rule: ('F14_0', 'F9_0') ==> ('F4_0', 'F15_0') , 0.910
Rule: ('F2_0', 'F12_0') ==> ('F17_0',) , 0.911
Rule: ('F11_0', 'F17_0', 'F6_0') ==> ('F18_0',) , 0.911
Rule: ('F18_0', 'F17_0', 'F6_0') ==> ('F11_0',) , 0.911
Rule: ('F17_0', 'F6_0') ==> ('F18_0',) , 0.911
Rule: ('F17_0', 'F15_0') ==> ('F18_0',) , 0.911
Rule: ('F17_0', 'F6_0') ==> ('F11_0',) , 0.911
Rule: ('F2_0', 'F7_0') ==> ('F17_0',) , 0.911
Rule: ('F19_0', 'F6_0') ==> ('F11_0',) , 0.912
Rule: ('F18_0', 'F7_0') ==> ('F12_0',) , 0.912
Rule: ('F12_0',) ==> ('F17_0',) , 0.914
Rule: ('F14_0',) ==> ('F15_0',) , 0.914
Rule: ('F12_0',) ==> ('F18_0',) , 0.914
Rule: ('F17_0', 'F4_0') ==> ('F18_0',) , 0.914
Rule: ('F7_0', 'F12_0') ==> ('F18_0',) , 0.905
Rule: ('F17_0', 'F7_0') ==> ('F12_0',) , 0.905
Rule: ('F7_0', 'F12_0') ==> ('F17_0',) , 0.905
Rule: ('F11_0', 'F2_0', 'F6_0') ==> ('F18_0',) , 0.906
Rule: ('F12_0',) ==> ('F7_0',) , 0.906
Rule: ('F19_0', 'F2_0') ==> ('F18_0',) , 0.907
Rule: ('F19_0', 'F2_0') ==> ('F17_0',) , 0.907
Rule: ('F11_0',) ==> ('F6_0',) , 0.907
Rule: ('F2_0', 'F7_0', 'F12_0') ==> ('F17_0',) , 0.908
Rule: ('F21_0',) ==> ('F18_0',) , 0.908
Rule: ('F2_0', 'F18_0') ==> ('F17_0',) , 0.908
Rule: ('F4_0', 'F15_0') ==> ('F9_0',) , 0.908
Rule: ('F11_0', 'F17_0') ==> ('F18_0',) , 0.910
Rule: ('F18_0', 'F6_0') ==> ('F11_0',) , 0.910
Rule: ('F14_0', 'F9_0') ==> ('F4_0', 'F15_0') , 0.910
Rule: ('F2_0', 'F12_0') ==> ('F17_0',) , 0.911
Rule: ('F11_0', 'F17_0', 'F6_0') ==> ('F18_0',) , 0.911
Rule: ('F18_0', 'F17_0', 'F6_0') ==> ('F11_0',) , 0.911
Rule: ('F17_0', 'F6_0') ==> ('F18_0',) , 0.911
Rule: ('F17_0', 'F15_0') ==> ('F18_0',) , 0.911
Rule: ('F17_0', 'F6_0') ==> ('F11_0',) , 0.911
Rule: ('F2_0', 'F7_0') ==> ('F17_0',) , 0.911
Rule: ('F19_0', 'F6_0') ==> ('F11_0',) , 0.912
Rule: ('F18_0', 'F7_0') ==> ('F12_0',) , 0.912
Rule: ('F12_0',) ==> ('F17_0',) , 0.914
Rule: ('F14_0',) ==> ('F15_0',) , 0.914
Rule: ('F12_0',) ==> ('F18_0',) , 0.914
Rule: ('F17_0', 'F4_0') ==> ('F18_0',) , 0.914
Rule: ('F3_0',) ==> ('F18_0',) , 0.915
Rule: ('F18_0', 'F14_0') ==> ('F15_0',) , 0.915
Rule: ('F2_0', 'F17_0', 'F6_0') ==> ('F18_0',) , 0.915
Rule: ('F15_0', 'F9_0') ==> ('F4_0',) , 0.915
Rule: ('F11_0', 'F2_0') ==> ('F17_0',) , 0.917
Rule: ('F18_0', 'F4_0') ==> ('F15_0',) , 0.917

Rule: ('F3_0',) ==> ('F18_0',) , 0.915
Rule: ('F18_0', 'F14_0') ==> ('F15_0',) , 0.915
Rule: ('F2_0', 'F17_0', 'F6_0') ==> ('F18_0',) , 0.915
Rule: ('F15_0', 'F9_0') ==> ('F4_0',) , 0.915
Rule: ('F11_0', 'F2_0') ==> ('F17_0',) , 0.917
Rule: ('F18_0', 'F4_0') ==> ('F15_0',) , 0.917
Rule: ('F11_0', 'F15_0') ==> ('F6_0',) , 0.918
Rule: ('F17_0', 'F7_0') ==> ('F18_0',) , 0.921
Rule: ('F2_0', 'F6_0') ==> ('F11_0',) , 0.921
Rule: ('F15_0', 'F14_0') ==> ('F9_0',) , 0.921
Rule: ('F2_0', 'F6_0') ==> ('F17_0',) , 0.921
Rule: ('F17_0', 'F4_0') ==> ('F15_0',) , 0.922
Rule: ('F11_0', 'F2_0') ==> ('F17_0', 'F6_0') , 0.900
Rule: ('F15_0', 'F9_0') ==> ('F14_0',) , 0.900
Rule: ('F2_0', 'F12_0') ==> ('F18_0',) , 0.902
Rule: ('F19_0', 'F18_0') ==> ('F17_0',) , 0.903

Rule: ('F11_0', 'F15_0') ==> ('F6_0',) , 0.918
Rule: ('F17_0', 'F7_0') ==> ('F18_0',) , 0.921
Rule: ('F2_0', 'F6_0') ==> ('F11_0',) , 0.921
Rule: ('F15_0', 'F14_0') ==> ('F9_0',) , 0.921
Rule: ('F2_0', 'F6_0') ==> ('F17_0',) , 0.921
Rule: ('F17_0', 'F4_0') ==> ('F15_0',) , 0.922
Rule: ('F4_0',) ==> ('F15_0',) , 0.923
Rule: ('F11_0', 'F2_0', 'F6_0') ==> ('F17_0',) , 0.923
Rule: ('F2_0', 'F17_0', 'F6_0') ==> ('F11_0',) , 0.923
Rule: ('F11_0', 'F18_0') ==> ('F17_0',) , 0.924
Rule: ('F11_0', 'F18_0') ==> ('F6_0',) , 0.924
Rule: ('F18_0', 'F6_0') ==> ('F17_0',) , 0.925
Rule: ('F11_0', 'F17_0') ==> ('F6_0',) , 0.925
Rule: ('F11_0', 'F18_0', 'F17_0') ==> ('F6_0',) , 0.926
Rule: ('F11_0', 'F18_0', 'F6_0') ==> ('F17_0',) , 0.926
Rule: ('F4_0', 'F9_0') ==> ('F14_0',) , 0.927

unhealthy_SPECT: with minimum support = 0.5 and minimum confidence = 0.9

Example result:

Rule: ('F11_0', 'F2_0') ==> ('F17_0', 'F6_0') , 0.900
Rule: ('F15_0', 'F9_0') ==> ('F14_0',) , 0.900
Rule: ('F2_0', 'F12_0') ==> ('F18_0',) , 0.902
Rule: ('F19_0', 'F18_0') ==> ('F17_0',) , 0.903
Rule: ('F2_0', 'F7_0') ==> ('F18_0',) , 0.903
Rule: ('F19_0', 'F6_0') ==> ('F18_0',) , 0.904
Rule: ('F7_0', 'F12_0') ==> ('F18_0',) , 0.905
Rule: ('F17_0', 'F7_0') ==> ('F12_0',) , 0.905
Rule: ('F7_0', 'F12_0') ==> ('F17_0',) , 0.905
Rule: ('F11_0', 'F2_0', 'F6_0') ==> ('F18_0',) , 0.906
Rule: ('F12_0',) ==> ('F7_0',) , 0.906
Rule: ('F19_0', 'F2_0') ==> ('F18_0',) , 0.907
Rule: ('F19_0', 'F2_0') ==> ('F17_0',) , 0.907
Rule: ('F11_0',) ==> ('F6_0',) , 0.907
Rule: ('F2_0', 'F7_0', 'F12_0') ==> ('F17_0',) , 0.908
Rule: ('F21_0',) ==> ('F18_0',) , 0.908
Rule: ('F2_0', 'F18_0') ==> ('F17_0',) , 0.908
Rule: ('F4_0', 'F15_0') ==> ('F9_0',) , 0.908
Rule: ('F11_0', 'F17_0') ==> ('F18_0',) , 0.910
Rule: ('F18_0', 'F6_0') ==> ('F11_0',) , 0.910
Rule: ('F14_0', 'F9_0') ==> ('F4_0', 'F15_0') , 0.910
Rule: ('F2_0', 'F12_0') ==> ('F17_0',) , 0.911
Rule: ('F11_0', 'F17_0', 'F6_0') ==> ('F18_0',) , 0.911
Rule: ('F18_0', 'F17_0', 'F6_0') ==> ('F11_0',) , 0.911
Rule: ('F17_0', 'F6_0') ==> ('F18_0',) , 0.911
Rule: ('F17_0', 'F15_0') ==> ('F18_0',) , 0.911
Rule: ('F17_0', 'F6_0') ==> ('F11_0',) , 0.911
Rule: ('F2_0', 'F7_0') ==> ('F17_0',) , 0.911
Rule: ('F19_0', 'F6_0') ==> ('F11_0',) , 0.912
Rule: ('F18_0', 'F7_0') ==> ('F12_0',) , 0.912
Rule: ('F12_0',) ==> ('F17_0',) , 0.914
Rule: ('F14_0',) ==> ('F15_0',) , 0.914
Rule: ('F12_0',) ==> ('F18_0',) , 0.914
Rule: ('F17_0', 'F4_0') ==> ('F18_0',) , 0.914

Rule: ('F2_0', 'F17_0', 'F6_0') ==> ('F11_0',) , 0.923
Rule: ('F11_0', 'F18_0') ==> ('F17_0',) , 0.924
Rule: ('F11_0', 'F18_0') ==> ('F6_0',) , 0.924
Rule: ('F18_0', 'F6_0') ==> ('F17_0',) , 0.925
Rule: ('F11_0', 'F17_0') ==> ('F6_0',) , 0.925
Rule: ('F11_0', 'F18_0', 'F17_0') ==> ('F6_0',) , 0.926
Rule: ('F11_0', 'F18_0', 'F6_0') ==> ('F17_0',) , 0.926
Rule: ('F4_0', 'F9_0') ==> ('F14_0',) , 0.927
Rule: ('F18_0', 'F7_0') ==> ('F17_0',) , 0.928
Rule: ('F18_0', 'F12_0') ==> ('F17_0',) , 0.929
Rule: ('F17_0', 'F12_0') ==> ('F18_0',) , 0.929
Rule: ('F7_0', 'F6_0') ==> ('F11_0',) , 0.930
Rule: ('F7_0', 'F6_0') ==> ('F12_0',) , 0.930
Rule: ('F18_0', 'F6_0', 'F2_0') ==> ('F11_0',) , 0.930
Rule: ('F19_0', 'F17_0') ==> ('F18_0',) , 0.931
Rule: ('F4_0', 'F15_0', 'F9_0') ==> ('F14_0',) , 0.933
Rule: ('F19_0', 'F18_0', 'F6_0') ==> ('F17_0',) , 0.938
Rule: ('F18_0', 'F7_0', 'F12_0') ==> ('F2_0',) , 0.939
Rule: ('F18_0', 'F6_0', 'F2_0') ==> ('F17_0',) , 0.939
Rule: ('F9_0',) ==> ('F15_0',) , 0.942
Rule: ('F19_0', 'F11_0') ==> ('F6_0',) , 0.942
Rule: ('F14_0', 'F9_0') ==> ('F4_0',) , 0.943
Rule: ('F7_0', 'F12_0') ==> ('F2_0',) , 0.944
Rule: ('F11_0', 'F7_0') ==> ('F6_0',) , 0.946
Rule: ('F6_0', 'F12_0') ==> ('F7_0',) , 0.946
Rule: ('F19_0', 'F17_0', 'F6_0') ==> ('F18_0',) , 0.946
Rule: ('F7_0', 'F6_0') ==> ('F2_0',) , 0.947
Rule: ('F17_0', 'F7_0', 'F12_0') ==> ('F2_0',) , 0.947
Rule: ('F18_0', 'F9_0') ==> ('F15_0',) , 0.948
Rule: ('F4_0', 'F14_0') ==> ('F15_0', 'F9_0') , 0.949
Rule: ('F15_0', 'F14_0', 'F9_0') ==> ('F4_0',) , 0.949
Rule: ('F18_0', 'F7_0', 'F2_0') ==> ('F12_0',) , 0.955
Rule: ('F2_0', 'F17_0', 'F7_0') ==> ('F12_0',) , 0.956
Rule: ('F14_0', 'F9_0') ==> ('F15_0',) , 0.959

Rule: ('F3_0',) ==> ('F18_0',) , 0.915 Rule: ('F18_0', 'F14_0') ==> ('F15_0',) , 0.915 Rule: ('F2_0', 'F17_0', 'F6_0') ==> ('F18_0',) , 0.915 Rule: ('F15_0', 'F9_0') ==> ('F4_0',) , 0.915 Rule: ('F11_0', 'F2_0') ==> ('F17_0',) , 0.917 Rule: ('F18_0', 'F4_0') ==> ('F15_0',) , 0.917 Rule: ('F11_0', 'F15_0') ==> ('F6_0',) , 0.918 Rule: ('F17_0', 'F7_0') ==> ('F18_0',) , 0.921 Rule: ('F2_0', 'F6_0') ==> ('F11_0',) , 0.921 Rule: ('F15_0', 'F14_0') ==> ('F9_0',) , 0.921 Rule: ('F2_0', 'F6_0') ==> ('F17_0',) , 0.921 Rule: ('F17_0', 'F4_0') ==> ('F15_0',) , 0.922 Rule: ('F4_0',) ==> ('F15_0',) , 0.923	Rule: ('F2_0', 'F7_0') ==> ('F12_0',) , 0.960 Rule: ('F4_0', 'F9_0') ==> ('F15_0',) , 0.960 Rule: ('F18_0', 'F12_0', 'F2_0') ==> ('F7_0',) , 0.964 Rule: ('F6_0', 'F12_0') ==> ('F2_0',) , 0.964 Rule: ('F2_0', 'F17_0', 'F12_0') ==> ('F7_0',) , 0.964 Rule: ('F4_0', 'F14_0', 'F9_0') ==> ('F15_0',) , 0.965 Rule: ('F4_0', 'F14_0') ==> ('F15_0',) , 0.966 Rule: ('F2_0', 'F12_0') ==> ('F7_0',) , 0.967 Rule: ('F11_0', 'F2_0') ==> ('F6_0',) , 0.975 Rule: ('F11_0', 'F2_0', 'F17_0') ==> ('F6_0',) , 0.982 Rule: ('F4_0', 'F15_0', 'F14_0') ==> ('F9_0',) , 0.982 Rule: ('F4_0', 'F14_0') ==> ('F9_0',) , 0.983 Rule: ('F11_0', 'F18_0', 'F2_0') ==> ('F6_0',) , 0.991

Chapter 4

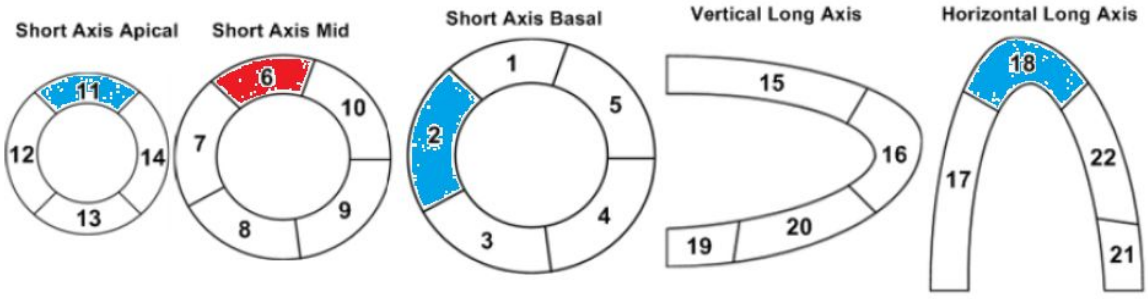
Results

Rule interpretation

For this part, from the results the interesting rules for each type of patients will be chosen to be discussed. The strong rule will be picked from the higher confidence value.

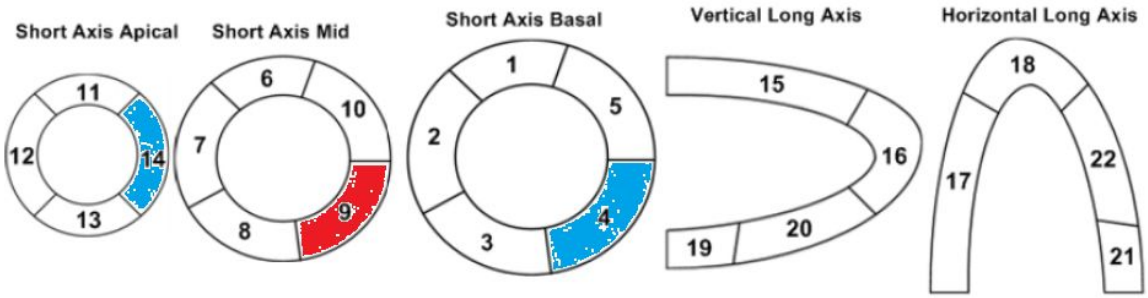
For Healthy_SPECT dataset:

1. Rule : ('F11_0', 'F18_0', 'F2_0') ==> ('F6_0',) with confidence value = 0.991



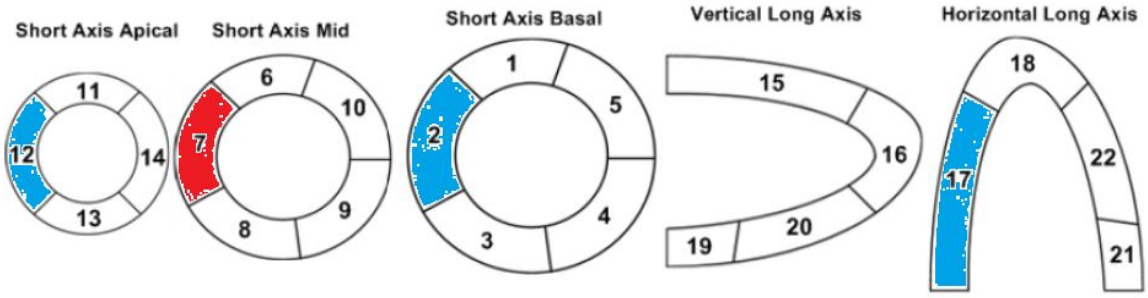
For confidence value is around 0.991, there is about 99% chance that if apical anterior (F11), apical (F18) and basal anteroseptal (F2) have poor perfusion, then mid anterior (F6) will also have poor perfusion too.

2. Rule: ('F4_0', 'F14_0') ==> ('F9_0',) with confidence value = 0.983



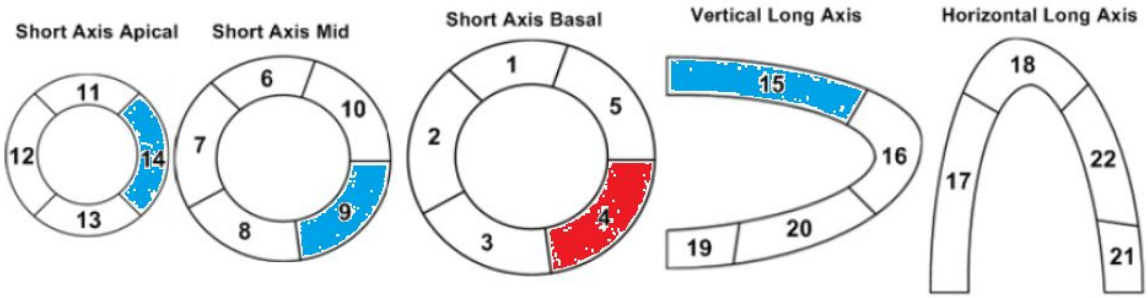
For confidence value is around 0.983, there is about 98% chance that if basal inferolateral (F4) and apical lateral (F14) have poor perfusion, then mid inferolateral (F9) will also have poor perfusion too.

3. Rule: ('F2_0', 'F17_0', 'F12_0') ==> ('F7_0',) with confidence value = 0.964



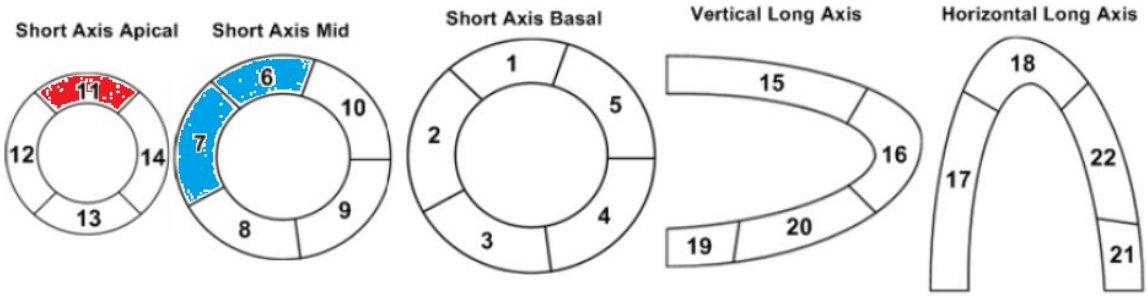
For confidence value is around 0.964, there is about 96% chance that if basal anteroseptal (F2), septal (F17) and apical septal (F12) have poor perfusion, then mid anteroseptal (F7) will also have poor perfusion too.

4. Rule: Rule: ('F15_0', 'F14_0', 'F9_0') ==> ('F4_0',) with confidence value = 0.949



For confidence value is around 0.949, there is about 95% chance that if anterior (F15), apical lateral (F14) and mid inferolateral (F9) have poor perfusion, then basal inferolateral (F4) will also have poor perfusion too.

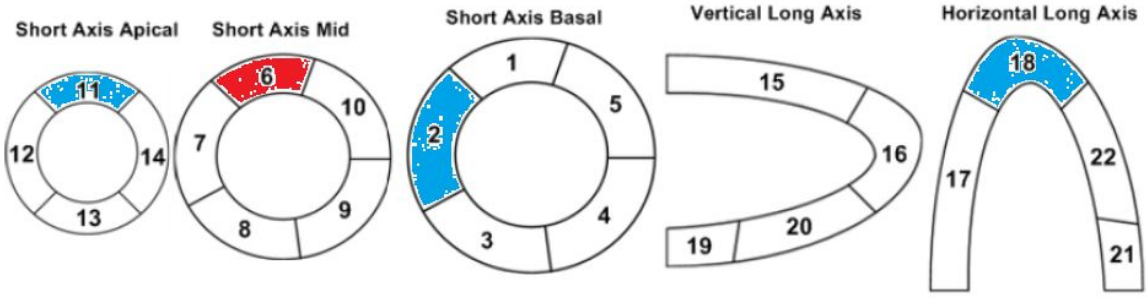
5. Rule: Rule: ('F7_0', 'F6_0') ==> ('F11_0',) with confidence value = 0.930



For confidence value is about 0.930, there is around 93% chance that if mid anteroseptal (F7) and mid anterior (F6) have poor perfusion, then apical anterior (F11) will also have poor perfusion too.

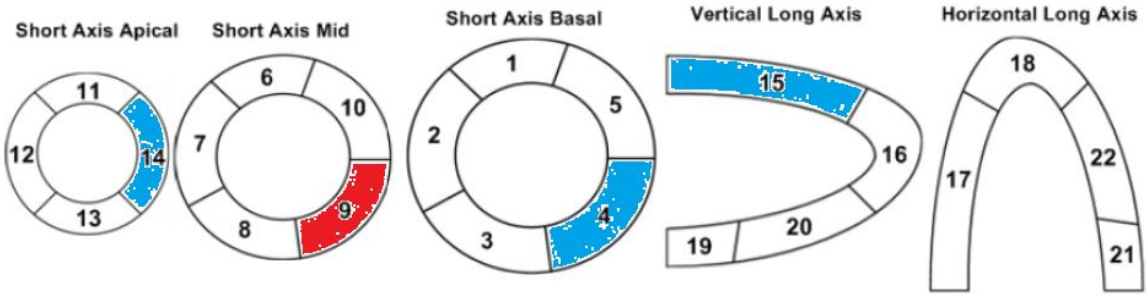
For Unhealthy_SPECT dataset:

1. Rule: ('F11_0', 'F18_0', 'F2_0') ==> ('F6_0',) with confidence value = 0.991



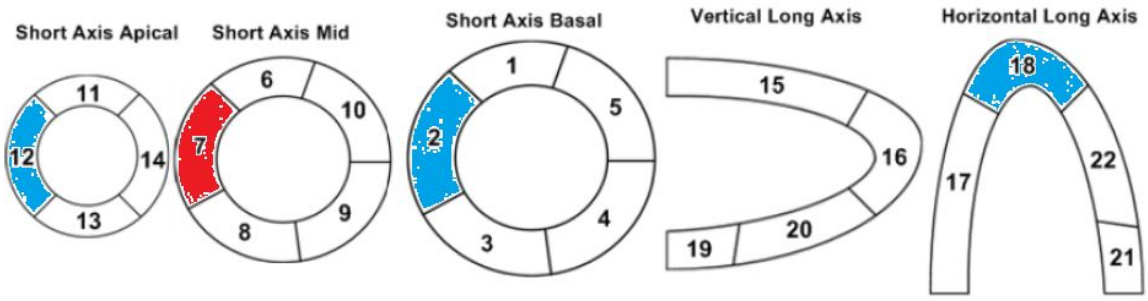
For confidence value is about 0.991, there is around 99% chance that if apical anterior (F11), apical (F18) and basal anteroseptal (F2) have poor perfusion, then mid anterior (F6) will also have poor perfusion too.

2. Rule: ('F4_0', 'F15_0', 'F14_0') ==> ('F9_0',) with confidence value = 0.982



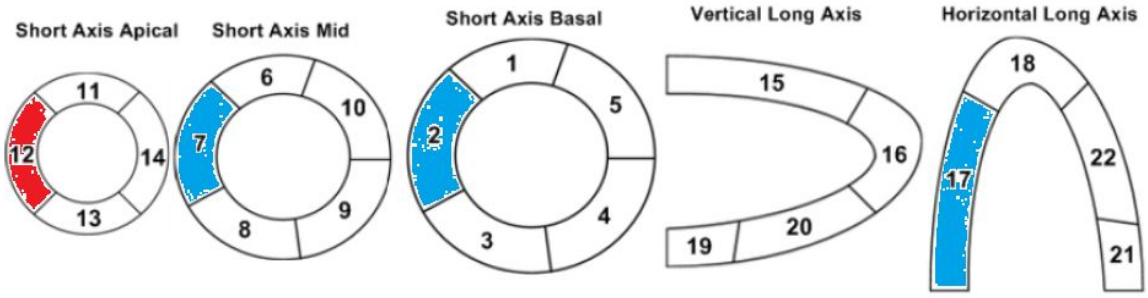
For confidence value is about 0.982, there is around 98% chance that if basal inferolateral (F4), anterior (F15) and apical lateral (F14) have poor perfusion, then mid inferolateral (F9) will also have poor perfusion too.

3. Rule: ('F18_0', 'F12_0', 'F2_0') ==> ('F7_0',) with confidence value = 0.964



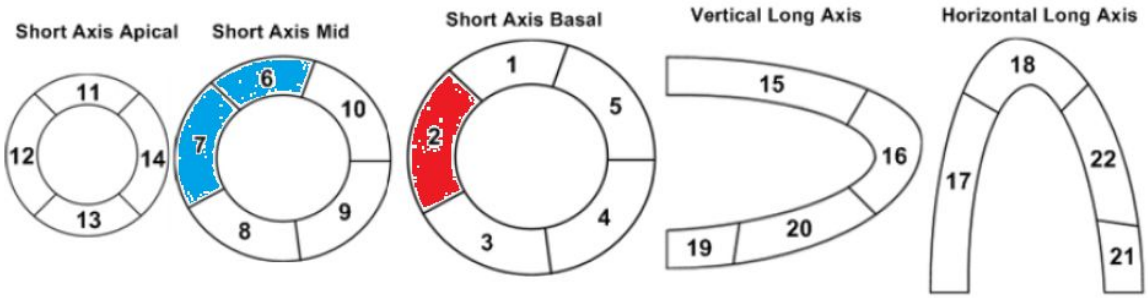
For confidence value is about 0.964, there is around 96% chance that if apical apical (F18), apical septal (F12) and basal anteroseptal (F2) have poor perfusion, then mid anteroseptal (F7) will also have poor perfusion too.

4. Rule: ('F2_0', 'F17_0', 'F7_0') ==> ('F12_0',) with confidence value = 0.956



For confidence value is about 0.956, there is around 99% chance that if basal anteroseptal (F2), septal (F17) and mid anteroseptal (F7) have poor perfusion, then apical septal (F12) will also have poor perfusion too.

5. Rule: ('F7_0', 'F6_0') ==> ('F2_0',) with confidence value = 0.94



For confidence value is about 0.947, there is around 94% chance that if mid anteroseptal (F7) and mid anterior (F6) have poor perfusion, then basal anteroseptal (F2) will also have poor perfusion too.

Chapter 5

Python Implementation

This section provides the description of each function for generating the association rule.

`def subsets(arr):`

This function takes list or tuple, `arr`, as input. The list represents set structure. Then, it returns non empty subset of `arr`.

`def joinSet(itemSet, length):`

This function takes a list of set, `itemSet`, and the length of output, `length`, as inputs. Then it joins a set with itself and returns the *n*-element itemsets. For example,

```
itemSet = [{1,2,3},{1,2,4},{5}]
joinSet(itemSet, 4)
```

will get the output

```
[{1, 2, 3, 4}, {1, 2, 3, 5}, {1, 2, 3, 4}, {1, 2, 4, 5}, {1, 2,
3, 5}, {1, 2, 4, 5}]
```

`def runApriori(data_iter, minSupport, minConfidence):`

This function takes the record of item (**`data_iter`**), minimum support (**`minSupport`**), and minimum confidence (**`minConfidence`**) as inputs. The algorithm starts by searching the frequent patterns that occurs more than minimum support value. Then, function joinSet is used for generating the list of joint item. The association rule is generated form frequent itemset. The association rule is of the form

LHS => RHS.

The confidence is also calculated by the following formula:

confidence(LHS, RHS) = support(LHS, RHS) / support(LHS)

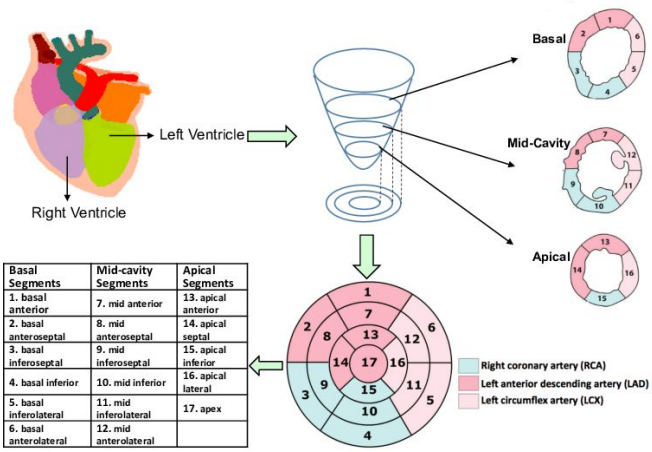
Chapter 6

Conclusion

Discussion

In conclusion, at the beginning provide a full dataset, which was divided into 2 datasets the first dataset "SPECT.train" contains 80 records, and the remaining 187 instances is in the "SPECT.test" dataset. Then, run Apriori-algorithm on each type of patients in order to find association rules, remove the first attribute which is the attribute of patient who has normal cardiac health (0) and abnormal cardiac health (1). Basically, run Apriori-algorithm on 22 attributes with constraint: minimum thresholds on support is 0.5 and minimum thresholds on confidence is 0.9.

For both healthy_SPECT dataset and unhealthy_SPECT dataset, we choose 5 rules from both datasets that seems interesting and interpret the data, consider on the rule that has high confidence. Also, look at the heart cross section diagram to see which attribute will effect on other attributes. According to the best rules found that generated from the Apriori-algorithm which we found that all of them are positive association, and we try to see the pattern of the result. So we focus on 22 attributes which used to define the patients with poor perfusion (1) and high perfusion (0). To clarify, the left ventricle can be divided into three sections which are basal, mid-cavity, and apical. After running the Apriori algorithm, it could be seen from generated rules that if there is a poor perfusion occured in any segments, other nearby neighbors are likely to have the same condition as well. For example, if apical (F16) have a poor perfusion, apical anterior (F11), mid anterolateral (F12), anterior (F15), mid anterior (F6) have a higher chance of having poor perfusion too.



Basal Segments	Mid-cavity Segments	Apical Segments
1. basal anterior	7. mid anterior	13. apical anterior
2. basal anteroseptal	8. mid anteroseptal	14. apical septal
3. basal inferoseptal	9. mid inferoseptal	15. apical inferior
4. basal inferior	10. mid inferior	16. apical lateral
5. basal inferolateral	11. mid inferolateral	17. apex
6. basal anterolateral	12. mid anterolateral	

Limitation

Apriori algorithm requires a lot of memory resources. The size of the dataset greatly affects the performance of the proposed method because large dataset will not fit with RAM.

Future development

Using Partition algorithm to reduces the number of reading operation in database.

References

Datasets: http://archive.ics.uci.edu/ml/datasets/SPECT+Heart:

Paper: https://link.springer.com/chapter/10.1007/978-3-642-21073-0_38

Slide:
https://docs.google.com/presentation/d/1jHn1Q5ecHPYmm_S3jAacYaHxC6Zx7q9i7p_K8bBCJ9k/edit#slide=id.g49db7f4782_0_75