COMPACT KNOWLEDGE DISCOVERY
A CORE BASED APPROACH

A DISSERTATION

Submitted in partial fulfilment of the requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

SAKET SANKHLA

ER & DCI
NOIDA

IIT Roorkee-ER&DCI, Noida
C-56/1, "Anusandhan Bhawan"
Sector 62, Noida-201 307

FEBRUARY, 2003

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation titled **"Compact Knowledge Discovery: A Core Based Approach"** partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Information Technology,** submitted in **IIT, Roorkee – ER&DCI Campus, Noida,** is an authentic record of my own work carried out during the period from August 2002 to February, 2003 under the guidance of **Mr. Akshay Kumar Sharma**, D.G.M. Switch, Hexacom India Limited, Oasis Cellular, Jaipur.

The matter embodied in this dissertation has not been submitted by me for award of any other degree of diploma

Date: 27/02/03

Place: ROORKEE

Sankhla

(SAKET SANKHLA)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Co-Guide

25-02-03

Mr. Munish Kumar,
Project Engineer
ER & DCI, Noida

Project Guide

A. Kumar

Mr. Akshay Kumar Sharma
D G M, (Switch)
Hexacom India Limited,
Oasis Cellular. Jaipur

CONTENTS

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation titled **"Compact Knowledge Discovery: A Core Based Approach"** partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Information Technology,** submitted in **IIT, Roorkee – ER&DCI Campus, Noida,** is an authentic record of my own work carried out during the period from August 2002 to February, 2003 under the guidance of **Mr. Akshay Kumar Sharma**, D.G.M. Switch, Hexacom India Limited, Oasis Cellular, Jaipur.

The matter embodied in this dissertation has not been submitted by me for award of any other degree of diploma

Date: 27/02/03

Place: ROORKEE

Sankhla

(SAKET SANKHLA)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Co-Guide

25-02-03

Mr. Munish Kumar,
Project Engineer
ER & DCI, Noida

Project Guide

A. Kumar

Mr. Akshay Kumar Sharma
D G M, (Switch)
Hexacom India Limited,
Oasis Cellular. Jaipur

ACKNOWLEDGEMENT

The work presented in this report would not have been completed without the guidance and support of many people.

My sincere thanks to **Prof. Prem Vratt,** Director, IIT Roorkee and **Sh. R.K.Verma,** Executive Director, ER&DCI, Noida. I would also like to thank **Prof. A.K.Awasthi,** Dean PGS&R, IIT Roorkee. I would also like to thank **Prof. R.P.Agrawal**, Course Coordinator, M.Tech.(IT), IIT' Roorkee and **Mr. V.N.Shukla**, Course Coordinator, M.Tech.(IT), ER&DCI, Noida.

I am grateful to my guide **Mr. Akshay Kumar Sharma** for his presence for motivating and inspiring me to go on when I wanted a vision to see this dissertation come true. I thank him for his constant support, incredible enthusiasm and encouragement, thank you sir for your support and confidence one me. My sincere thanks are due to **Mr. Munish Kumar.** for the encouragement and valuable suggestions he provided me with during the course of my work. and for the cooperation extended by him in the successful completion of this report. My special thanks to **Mr. Kaushal Kumar** IT Consultant of ER&DCI, Noida, having enormous knowledge in Mathematics and logical reasoning for his timely help in making me understand the Conditional Entropy.

I would like to thank God and my family for their support in all facets of my life . I would like to thank my teachers, school mates and friends who inspired me to do this thesis with full sincerity, hard work and determination. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work.

Finally, I thank members of evaluation committee for their thoughtful comments. I thank all those people who listened to me and my vague ideas when I wanted someone to listen, thank you all....... for your supportive role.

Sankhla

(SAKET SANKHLA)

CONTENTS

ABSTRACT

Every second the amount of data in and around us is increasing and there's no end to the sight. Data Mining is done to ease the understanding of large amounts of data by discovering interesting exceptions or regularities. Association rules are simple but powerful regularities in binary data. The problem in Association Rule is that there can be easily hundreds or even more association rules holding in a data set which presents a problem in their utility and understanding. Therefore, **Data Mining itself can produce such great amounts of data that there is new knowledge management problem.**

The current World realizes the need for compact knowledge discovery in databases and have attempted solutions. I present 'A Core Based Approach on Large Item Sets' that efficiently summarizes the information present in the data set. Therefore the concept of conditional entropy to measure information content in a rule is used. A Core is a set that is found in more than one set.

The Compact knowledge discovery can be identified by compactness factors like: degree of useful information conveyed, a measure of strength of a rule, suitable number of rules, measure of interdependence between items and so on.

In this thesis I end up showing a proposed approach that meets some compactness factors and promise to satisfy the rest.

Chapter 1

INTRODUCTION

1.1 Overview

Data Mining an Introduction

Every second the amount of data in and around us is increasing and there's no end to the sight. Computers make it easy to save things that previously one would have thrashed. Inexpensive multi-gigabyte disks make it too easy to postpone decisions about what to do with all this stuff – one simply buys another disk and keeps it all. These electronics record our choices, our decisions as an every day process in the supermarket, our financial habits, our comings and goings, everything. We swipe our way through the world; every swipe becomes a record in a database. For example: The World Wide Web overwhelms one with information; meanwhile, every choice we make is recorded. And all these are just personal choices: they have countless counterparts in the world of commerce and industry. One could easily testify to the growing gap between the generation of data and one's understanding of it. As the volume of the data increases, inexorably, the proportion of it that people "understand" decreases, alarmingly. Lying hidden in all this data is information, potentially useful information that is rarely made explicit or taken advantage of.

Therefore, looking for patterns becomes important task. Pattern recognition is used at every stage of our lives. Farmers seek patterns in crop growth, politicians seek patterns in voter opinion, hunters seek patters in animal migration behavior, lovers seek patterns in their partner's response. A scientist's job (like a baby's) is to make sense of data, to discover the patterns that govern how the physical world works and encapsulate them in theories that can be used for predicting what will happen in new situations. The entrepreneur's job is to identify opportunities, that is, patterns in behavior that can be turned into profitable business, and exploit them.

In data mining, the data is stored electronically and the search is automated – or at least augmented – by computer. Even this is not particularly new. Economists, statisticians, and communication engineers have long worked with the idea that patterns in data can be sought automatically, identified, validated, and used for prediction. What is new is the staggering increase in the opportunities for finding patterns in data. The unbridled growth of databases in the recent years, databases on everyday activities as customer choices, brings data mining to the forefront of new business technologies. It has been estimated that the amount of data stored in the world's databases doubles every twenty months, and although it would surely be difficult to justify this figure in any quantitative sense, one can relate to the pace of growth qualitatively. As the flood of data swells and machines that can undertake the searching become commonplace, the opportunities for data mining increases. As the world grows in complexity, overwhelming one with the data it generates, data mining is one way for elucidating the patterns that underlie it. Intelligently analyzed data is a valuable resource. It can lead to new insights and, in commercial setting, to competitive advantages.

Thus, the knowledge discovery in databases or Data Mining is a field of increasing interest combining databases, artificial intelligence, machine learning and statistics. The purpose of data mining is to facilitate understanding large amounts of data by discovering interesting regularities or exceptions.

The association rule discovery in World Wide Web gives enormous results of our finding. It gives us all the pages that holds those character in the data page, giving us all the information it scans from the web, approximately 50 to 'n' number of pages. It reduces our search but on these pages we have to search again for the relevant data. Most of the pages that are been scanned are irrelevant and not required. One has to do different permutations and combinations to scan and get the desired data and most of the times it is to be done manually.

1.2 Objective

Mining Data Mining

The project came into existence when I started discovering the important role that data is playing in our lives and the importance of mining this data and extracting relevant data or knowledge from it. It also seen that not just any arbitrary mining will yield meaningful results; we have to mine data in a compact way. There for Compact Knowledge Discovery came into picture.

Knowledge Discovery and Data Mining (KDD) is an interdisciplinary area focusing upon methodologies for extracting useful knowledge from data. The ongoing rapid growth of online data due to the Internet and the widespread use of databases have created an immense need for KDD methodologies. The challenge of extracting knowledge from data draws upon research in statistics, databases, pattern recognition, machine learning, data visualization, optimization, and high-performance computing, to deliver advanced business intelligence and web discovery solutions.

Association rules are typically used in market analysis (market basket analysis), primarily because of the utility and clarity of its results. They express how important products or services relate to each other, and immediately suggest particular actions. Association rules are used in mining categorical data - items. Besides the sole process of generating association rules, the process of application of association rules technique involves two important concerns:

(i). Choice of the right set of items

The data used for association rule analysis is typically the detailed transaction data captured at the point of sale. Gathering and using this data is a critical part of applying association rule analysis, depending crucially on the items chosen for analysis. What constitutes a particular item depends on the business (problem) need. Items in stores usually have codes that form hierarchical categories (taxonomy). These categories help in generalization, and reduction of the volume of items used for a study. Dozens or hundreds of items may be reduced to a single generalized item, often corresponding to a single department or type of a product.

(ii). **Practical limits imposed by a large number of items appearing in combinations large enough to be interesting**

Number of combinations for larger item sets rises exponentially with the number of items. Calculating the support, confidence, and improvement for a grocery store with thousands of different items, quickly rises to millions, as the number of items in the combinations grows.

The problem in Association Rule is that there can be easily hundreds or even more association rules holding in a data set which presents a problem in their utility and understanding. Therefore, Data Mining itself can produce such great amounts of data that there is new knowledge management problem.

The current World realizes the need for compact knowledge discovery in databases and have attempted solutions. I present 'A Core Based Approach on Large Item Sets' that efficiently summarizes the information present in the data set. Therefore the concept of conditional entropy to measure information content in a rule is used. A Core is a set that is found in more than one set.

The Compact knowledge discovery can be identified by compactness factors like: degree of useful information conveyed, a measure of strength of a rule, suitable number of rules, measure of interdependence between items and so on.

Paradoxically, data mining itself can produce such great amounts of data that there is new knowledge management problem. There is need for succinct or compact summarization of the knowledge present in databases.

1.3 Organization of dissertation

The following is chapter wise summary: -

In chapter 2, titled 'Literature Survey' discuss the problems and suggestions for the project and from the field of Bio informatics, the plausible solution for it and how theory of Cores can play really a major part in solving the problem. Analysis of the data and understanding the core in conditional entropy.

In chapter 3, titled 'Core based Approach: Theory and Implementation', formulates of the theory; core based approach on the set of Large Item Sets.

In chapter 4, tilted 'Results and Discussion', discussed the results of the proposed theory.

In chapter 5, tilted 'Conclusion', discussed the conclusions arose from the work and gives many pointers for taking the theory forward.

References

Appendix A : SAS System, explanation of SAS System
Appendix B : Large Item Set File
Appendix C : The Software, the working explained
Appendix D : Integer Identifier for set
Appendix E : Top Predictor sets table

Chapter 2

LITERATURE SURVEY

2.1 Previous works/Literature survey

Many researchers realized the need for Compact Knowledge Discovery in Databases and have attempted solutions and shown results.

Knowledge Discovery and Data Mining (KDD) is an interdisciplinary area focusing upon methodologies for extracting useful knowledge from data. The ongoing rapid growth of online data due to the Internet and the widespread use of databases have created an immense need for KDD methodologies. The challenge of extracting knowledge from data draws upon research in statistics, databases, pattern recognition, machine learning, data visualization, optimization, and high-performance computing, to deliver advanced business intelligence and web discovery solutions [2].

The problem of merging multiple databases of information about common entities is frequently encountered in KDD and decision support applications in large commercial and government organizations. The problem we study is often called the Merge/Purge problem and is difficult to solve both in scale and accuracy. Large repositories of data typically have numerous duplicate information entries about the same entities that are difficult to cull together without an intelligent ''equation theory'' that identifies equivalent items by a complex, domain-dependent matching process. We have developed a system for accomplishing this Data Cleansing task and demonstrate its use for cleansing lists of names of potential customers in a direct marketing-type application. Our results for statistically generated data are shown to be accurate and effective when processing the data multiple times using different keys for sorting on each successive pass. Combing results of individual passes using transitive closure over the independent results, produces far more accurate results at lower cost. The system provides a rule programming module that is easy to program and quite good at finding duplicates especially in an environment with massive amounts of data. This paper details improvements in our system, and reports on the successful implementation for a real-world database that conclusively validates our results previously achieved for statistically generated data. [3]

It consists of the problems that arise when submitting large quantities of data to analysis by an Inductive Logic Programming (ILP) system. Complexity arguments usually make it prohibitive to analyze such data sets in their entirety. We examine two schemes that allow an ILP system to construct theories by sampling from this large pool of data. The first, "sub sampling", is a single-sample design in which the utility of a potential rule is evaluated on a randomly selected sub-sample of the data. The second, "logical windowing", is multiple-sample design that tests and sequentially includes errors made by a partially correct theory. Both schemes are derived from techniques developed to enable prepositional learning methods (like decision trees) to cope with large datasets. The ILP system CProgol, equipped with each of these methods, is used to construct theories for two datasets—one artificial (a chess endgame) and the other naturally occurring (a language tagging problem). In each case, we ask the following questions of CProgol equipped with sampling: (1) Is its theory comparable in predictive accuracy to that obtained if all the data were used (that is, no sampling was employed)?; and (2) Is its theory constructed in less time than the one obtained with all the data? For the problems considered, the answers to these questions is "yes". This suggests that an ILP program equipped with an appropriate sampling method could begin to address problems satisfactorily that have hitherto been inaccessible simply due to data extent. [4]

One of the basic problems in knowledge discovery in databases (KDD) is the following: given a data set r, a class L of sentences for defining subgroups of r, and a selection predicate, find all sentences of L deemed interesting by the selection predicate. We analyze the simple levelwise algorithm for finding all such descriptions. We give bounds for the number of database accesses that the algorithm makes. For this, we introduce the concept of the border of a theory, a notion that turns out to be surprisingly powerful in analyzing the algorithm. We also consider the verification problem of a KDD process: given r and a set of sentences S Í L determine whether S is exactly the set of interesting statements about r. We show strong connections between the verification problem and the hypergraph transversal problem. The verification problem arises in a natural way when using sampling to speed up the pattern discovery step in KDD [5]

2.2 Problem Description

I believe that a wealth of information can be extracted from the Lattice paths. The following are some of the problems and suggestions in this regard:

1. **How addition of item to the predictor set affect prediction for a given item**
   This may lead to power set of options so statically outputting this information in a tabular form may not be possible unless the number of distinct items are very small (<10). But this functionality can be achieved dynamically. At run time input the predicted item and predictor set and output the information change that takes place for the predicted item by the addition or deletion of items in the predictor set.

2. **Core strength**
   One can also think of some mathematical way of measuring strength of a core.

3. **Bio-informatics**
   Discovering the functionality of genes and the interdependence between genes is time consuming, expensive as well as difficult process. The theory developed in this dissertation can prove to be very handy tool in solving many of the problems encountered in the field of Bio-informatics

4. **Domain knowledge**
   One can use domain knowledge in form of templates (by Klemettin's [7]) or any other form on the top of core based approach to refine one's results.

5. **Time complexity**

   The focus of my dissertation was mainly to formulate a theory which can mine data in compact way. There is room for efficient implementation of the proposed theory thus improving the time complexity.

2.3 Problem in bio informatics

This problem is part of research paper 'Mining Microarray Expression Data for Classifier Gene-Cores' by Dr. Raj Bhatnagar, Dept of ECECS, University of Cincinnati [1].

Recently there has been a lot of interest in classifying various tumors using gene expression data. It has been shown that gene expression data acquired from leukemia patients can be used to build predictors that can discriminate between two acute leukemia subtypes, Acute Lymphoblastic Leukemia (ALL) and Acute Myleloid Leukemia (AML). To maximize the efficacy of cancer treatment while a the same time reducing it's toxicity, it is imperative to target specific therapies to path genetically distinct tumor types. Thus, improved cancer classification is valuable to advances in cancer treatment. Subtype of leukemia, ALL and ALL, have similar histopathological appearance. Correct prediction of the subtypes of cancer at an early stage from gene expression data can vastly improve accuracy of diagnosis and effectiveness of treatment. Currently, no single test is sufficient to make a diagnosis – leukemia classification still remains imperfect. Based on gene expression data collected from 72 patients suffering from either AML or ALL, it has been shown that a large number of genes (approximately 1100) have a higher correlation with the AML/ALL distinction than can be expected by chance. Classifiers built with this small subset of genes, selected based on their individual correlation with the cancer subtypes, have been used to predict leukemia subtypes with some accuracy. Since there is limited knowledge about the functional relevance of most genes, and a strong possibility exist of noise and biasing by inter gene dependence and gene interactions, the selection of set of genes that constitute a good and biologically meaningful classifier is very important. What genes should be chosen to form the classifier and how many genes should be included are the two important issues in designing a classifier.

2.4 Analysis of data

Almost all statistical pattern recognition methods require an abundance of data in order to generate inferences with sufficient confidence. Typically, a dataset on which statistical methods would be relevant consists of a large number (upwards of 1000s) of data points and very few variables (orders of 10s) beings measured for each data point. In the case of gene expression datasets, each data point consists of few thousand measured variables (genes) and few order of tens data points. An exploratory analysis of such a dataset requires that we generate a large number of plausible hypotheses (good classifiers, in our case) and examine them. Any inference of high confidence can be based only on its consistency with a large fraction of plausible hypotheses. This approach was adopted.

The adopted strategy works in the following two phases:

(i). During the first phase a heuristic search to identify a large number of small-sized subsets of genes that discriminate between classes better the similar sized subsets. During the second phase these classifiers have been trained by finding suitable weight vectors to optimize the discrimination potential of each subset. From relatively large population of such trained classifiers better performing individual classifiers were selected.

(ii). By using the above methodology, it was demonstrated for the ALL-AML case, existence of gene sets that have 100% accuracy in predicting the subtype of leukemia. One of the main thrust behind the strategy is the belief that it is not enough to look at the correlation between individual genes and the classes, but the effect of set of genes as a whole, given the possibility of inter-gene dependencies. It was found out that there are about 4000 good classifiers. Now, among these set of classifiers, which classifier or a small set of classifier should be used to distinguish between AML and ALL?

2.5 Understanding the Core

To scrape together insight from a large set of well-performing classifiers, an algorithm to extract or mine cores from the set of classifiers was introduced. A core is defined as a subset of genes that is an integral part of several distinct good classifiers. The number of candidate classifiers in which the core appears defines the strength of the core.

The presence of many such different cores across many different classifiers can indicate that there may be several different processes leading to (Acute Lymphoblastic Leukemia) AML/ALL (Acute Myleloid Leukemia) distinction. A core itself is an indication that the genes constituting it are strongly related, at least in the context of the processes leading to the AML/ALL distinction. The examination of the cores mined can lead to the better understanding of the working of the genome with respect to tumors.

Thus these cores of genes are potentially very useful for a biologist and may reveal much about inter-gene dependencies and gene functions.

It was found out that many of the mined cores where themselves very good classifiers. Also biologist confirmed that the genes present in some of the cores do have strong inter-relationships. Thus justifying the immense knowledge that can be extracted from the set of cores as pointed out in the previous paragraph.

Chapter 3

CORE BASED APPROACH: THEORY AND IMPLEMENTATION

3.1 Focusing on Work

The project came into existence when I started discovering the important role that data is playing in our lives and the importance of mining this data and extracting relevant data or knowledge from it. It also seen that not just any arbitrary mining will yield meaningful results; we have to mine data in a compact way. With the problem of II order data mining or Compact Knowledge Discovery, I went on to see the potential of Core based approach for solving the II order data mining problem and finally, formulated a core based theory for solving it. We successfully applied the theory in case of Association rule discovery.

Compact knowledge discovery in databases has been discussed in the above chapters. The work focused on using a Core based approach on set of large item sets to do a compact knowledge discovery.

Association rules generated from a set of large item sets don't answer questions like: -

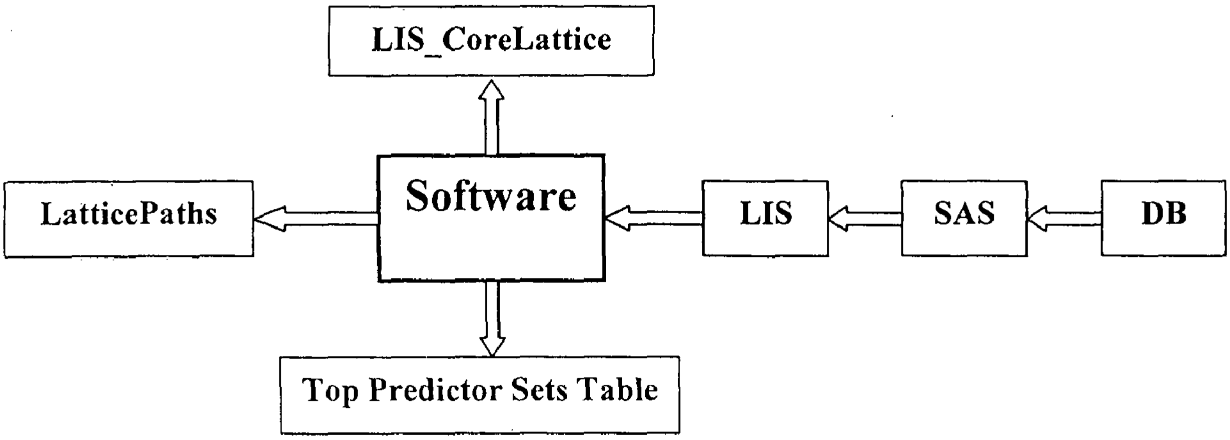
1. How is the prediction power of a set of items, for a given item change with the addition of one or more items to the set.
2. What is the interdependence between items?
3. Given an item, which is, it's most important predictors?

...These are few of the major problems faced while using association rule.

The Core based solutions mentioned in [1]. Is been continued.

3.2 System Architecture

3.2.1 Abstract design



The architecture of the above diagram is explained. It is done from right to left, that is how the system and the code works.

Database programming consists of huge data handling and so while processing it one may reach a dead end in either time or memory. While constructing table one cannot afford to generate & scan the power set of items, it's computationally too expensive. So instead we scan the core lattice to generate the necessary output.

One has to be careful about the use of space. While scanning Lattice the program was creating object for each element in the scan. By some smart ways one can avoid fatal steps like system over load.

Coding style in the software: -

- **(i).** The software is written in modular way.
- **(ii).** The code is arranged as they are compiled.
- **(iii).** Code is readable and well documented.

Source code: -

Software consists of four classes. The arrangement of code is as follows: -

(i). SummarizedMatrixElt.java

(ii). LIS_Lattice_Elt.java

(iii). LIS_Lattice.java

(iv). SummarizedMatrix.java

The whole architecture of the system as described is implemented successfully in JAVA 1.2 on Oracle J Developer 8.0 environment.

Large item set core lattice file The lattice had 5 levels. So it was a fast convergence.

Large item set core lattice paths file is generated enumerating all the possible paths that exist in core lattice. It will be sorted in the increasing order w.r.t. conditional entropy. Increasing order is chosen for the simple reason that minimum entropy means maximum information.

Top Predictor Sets table will be generated by scanning the Lattice paths. The outputted top 100 predictors for an item. Now instead of 7000 association rules we will have 100 rows and that too with much richer information content.

Also in the next chapter many ideas with some details are given for measuring the interdependence between items, core strength, how prediction measure of an item changes with addition of attribute and areas where this approach can be very effective.

Appendix C contains discussion the software and the usage of SAS system used for generating large data set.

3.2.2 Database (DB)

A transactional database was needed, which can give sufficiently large number of association rules. The required database was found in SAS system libraries. It's a super market database. The database gave 7000 association rules at 5% support and confidence.

3.2.3 SAS

SAS is a data warehousing software system, which is used to generate large item set and association rules. SAS software saved time and also the required work was done efficiently and without mistakes.

Appendix A explains SAS system.

3.2.4 Large item set file (LIS)

This file is a text file exported from large item sets generated through SAS system when the association rule discoveries were applied on the database.
The sample file is shown below (the same sample example will be used throughout as that facilitates good explanation.): -

heineken
cracker
heineken cracker
soda olives heineken cracker baguette

Appendix B lists the entire large item sets of the input database generated through SAS.

3.2.5 Software

The software that takes large items sets text file as input, parses it and outputs LIS_CoreLattice, LatticePaths and Top PredictorSets-Items table.

Appendix C gives details of the software and the coding style.

3.2.6 Large item set core Lattice (LIS_CoreLattice)

The sample file shown above represent the 0^{th} level of the lattice. The i^{th} level of lattice is construct from $(i-1)^{th}$ level in the following way.
An element in $(i-1)^{th}$ level goes to i^{th} level if there exist an element in the $(i-1)^{th}$ level which is proper super set of the element in question. Mathematically it can be written as:-
$LatticeLevel_i = \{x \mid x \in LatticeLevel_{(i-1)} \text{ and } \exists\, y \in LatticeLevel_{(i-1)} \text{ s.t. } x \subset y\}$.

The lattice will have i levels if (i +1) level has zero elements.

The LIS_CoreLattice for the above sample file is as follows: -

$LatticeLevel_0$	**$LatticeLevel_1$**	**$LatticeLevel_2$**
Heineken	*Heineken*	*Heineken*
Cracker	*Cracker*	*Cracker*
Cracker, Heineken	*cracker, heineken*	
Cracker, heineken, soda, olives, baguette		

Table 3.2.6 Large Item set Core Lattice

3.2.7 Conditional Entropy of a set

It's very important to understand the concept of conditional entropy, before I move on further in the architecture. Conditional entropy is used to measure of information. Conditional entropy captures the information changes taking place in a set by addition of items in the set.

The definition is as follows: -

LIS = $[I_1, I_2, \ldots, I_n]$ be a large item set then

CE = $[CE_1, CE_2, \ldots CE_n]$ is the conditional entropy of set LIS where

CE_1 is defined as follows: -

$$P_1 = count(I_1, I_2 \ldots I_n) \quad / \quad count(I_2, I_3, \ldots I_n).$$

$$P_2 = count(\sim I_1, I_2 \ldots I_n) \quad / \quad count(I_2, I_3, \ldots I_n).$$

$$P_3 = count(I_1, \sim(I_2 \ldots I_n)) \quad / \quad count(\sim(I_2, I_3, \ldots I_n)).$$

$$P_4 = count(\sim I_1, \sim(I_2 \ldots I_n)) \quad / \quad count(\sim(I_2, I_3, \ldots I_n)).$$

$count(I_1, I_2, \ldots, I_m)$ = # times $\{I_1, I_2, \ldots, I_m\}$ occurs in set $G(I_1, I_2, \ldots, I_n)$.

$G(I_1, I_2, \ldots, I_n)$ = $\{x \mid x \in LatticeLevel_0\}$ **if n=1**

= $\{x \mid x \in LatticeLevel_0 \text{ and } I_i \subset x \text{ and } 1 \leq i \leq n\}$

if n>1

ModifiedLog(P_i) = 0 **if $P_i = 0$**

= $\log_4 P_i$ **if $P_i > 0$**

~ is the symbol for negation.

$CE_1 = -(\Sigma (P_i * ModifiedLog(P_i)))$ where $1 \leq i \leq 4$.

Similarly we can define $CE_2, CE_3, \ldots, CE_n$.

Thus we can write: $[I_1, I_2, \ldots, I_n] \Rightarrow_{ce} [CE_1, CE_2, \ldots CE_n]$

Informally saying the value of CE_i tells us the information that other elements of set has about the item I_i.

From the sample example file we can give the following illustrations: -

Example 1: -

LIS	=	[heineken]
CE	=	$[CE_1]$

G([heineken]) = {

{ heineken },

{ cracker},

{cracker, heineken},

{cracker, heineken, soda, olives, baguette }

}

<u>Calculating CE_1</u>: -

P_1	=	count(heineken, ())	/	count()
	=	3	/	4
	=	0.75		
P_2	=	count(~heineken, ())	/	count()
	=	1	/	4
	=	0.25		
P_3	=	count(heineken, ~())	/	count(~()).
	=	3	/	4
	=	0.75		
P_4	=	count(~heineken, ~())	/	count(~()).
	=	1	/	4
	=	0.25		

$CE_1 \quad = \quad -(\Sigma\, P_i \,\text{ModifiedLog}(P_i)) \text{ where } 1 \le i \le 4.$

$\quad\quad = \quad 0.810$

Thus we can write the rule: **[heineken] $=>_{cc}$ [0.810]**

Example 2: -

LIS = [heineken, cracker]

CE = [CE_1, CE_2]

G([heineken]) = {

{ heineken },

{ cracker},

{cracker, heineken},

{cracker, heineken, soda, olives, baguette }

}

<u>Calculating CE_1</u>: -

P_1 = count(heineken, (cracker)) / count(cracker)

= 2 / 3

= 0.66

P_2 = count(~heineken, (cracker)) / count(cracker)

= 1 / 3

= 0.33

P_3 = count(heineken, ~(cracker)) / count(~(cracker)).

= 1 / 1

= 1

P_4 = count(~heineken, ~(cracker))/ count(~(cracker)).

= 0 / 1

$= \quad 0$

$CE_1 \quad = \quad -(\Sigma\, P_i\, \text{ModifiedLog}(P_i))$ where $1 \leq i \leq 4$.

$= \quad 0.464$

Similarly,

$CE_2 \quad = \quad 0.464$

Thus we can write the rule: **[heineken, cracker] $\Rightarrow_{cc}$ [0.464, 0.464]**

Example 3: -

LIS = [heineken, cracker, soda, olives, baguette]

CE = [CE_1, CE_2, CE_3, CE_4, CE_5]

G([heineken]) = {

{ heineken },

{ cracker},

{cracker, heineken},

{cracker, heineken, soda, olives, baguette }

}

Calculating CE_1: -

P_1 = count(heineken, (cracker, soda, olives, baguette)) / count(cracker, soda, olives, baguette)

= 1 / 1

= 1

P_2 = count(~heineken, (cracker, soda, olives, baguette)) / count(cracker, soda, olives, baguette)

= 0 / 1

= 0

P_3 = count(heineken, ~(cracker, soda, olives, baguette)) / count(~(cracker, soda, olives, baguette))

= 2 / 3

= 0.66

P_4 = count(~heineken, ~(cracker, soda, olives, baguette))

/

count(~(cracker, soda, olives, baguette))

= 1 / 3

= 0.33

CE_1 = - (Σ P_i ModifiedLog(P_i)) where $1 \leq i \leq 4$.

= 0.464

Similarly,

CE_2 = 0.464

CE_3 = 0

CE_4 = 0

CE_5 = 0

Thus we can write the rule:

[soda, olives, baguette, heineken, cracker] =>$_{ce}$ [0, 0, 0, 0.464, 0.464]

3.2.8 Large item set core lattice paths (LatticePaths)

Now, we traverse all the paths from right to left in the lattice and see how the information in the large item set changes. The measure of information is the concept of ***conditional entropy***. The LatticePaths file for the sample example would look like this: -

(Note: - Square bracket represents the large item set. Each item is given an integer identifier for the sake of readability. In the Appendix D integer identifier for each item is given. The values in the curly braces represent the values of the conditional entropy).

[1] $=>_{ce}$ [0.810]	[1, 2]$=>_{ce}$ [0.464, 0.464]	[3, 4, 5, 1, 2]$=>_{ce}$ [0, 0, 0, 0.464, 0.464]
[2]$=>_{ce}$ [0. 810]	[1, 2]$=>_{ce}$ [0.464, 0.464]	[3, 4, 5, 1, 2]$=>_{ce}$ [0, 0, 0, 0.464, 0.464]
[1]$=>_{ce}$ [0. 810]	[3, 4, 5, 1, 2]$=>_{ce}$ [0, 0, 0, 0.464, 0.464]	
[1]$=>_{ce}$ [0. 810]	[1, 2]$=>_{ce}$ [0.464, 0.464]	
[2]$=>_{ce}$ [0. 810]	[3, 4, 5, 1, 2]$=>_{ce}$ [0, 0, 0, 0.464, 0.464]	
[2]$=>_{ce}$ [0. 810]	[1, 2]$=>_{ce}$ [0.464, 0.464]	
[1]$=>_{ce}$ [0. 810]		
[2]$=>_{ce}$ [0. 810]		

Table 3.1: LatticePaths table

3.2.9 Top Predictor Sets table

Now, from the LatticePaths file we extract top predictor for a given item, for all the items. For the current running sample example the TopPredictorSet-Item table will look like this: -

(Note: - The column would represent the items)

Heineken	Cracker	Soda	Olives	baguette
[2] $=>_{ce}$ 0.464	[1] $=>_{ce}$ 0.464	[4,5,1,2] $=>_{ce}$ 0	[4, 5, 1, 2] $=>_{ce}$ 0	[4, 5, 1, 2] $=>_{ce}$ 0
[3,4,5,2] $=>_{ce}$ 0.464	[3,4,5,1]$=>_{ce}$0.464			

Table 3.2: Top Predictor set table.

Appendix E lists the top 50 predictors for some of the items.

Chapter 4

RESULTS AND DISCUSSION

Different data mining techniques are used for mining the data out of them most common techniques are Association Rule for super market data base. In which a threshold is defined in below or above which the data is extracted. This data is taken from the set of large data base in which the super market items are defined or entered. If the threshold is reduced, junk data is accumulated which does not give the desired result. The formulation of such a result depends on the threshold value and the data in the database. It can be seen that unwanted data is acquired in such a case and is a week algorithm for data mining.

The thesis uses condition entropy of a large item set to find the most probable data in the data set by scanning the core of the lattice. This further gives compact result as compared by the other conventional means. It reduces the threshold factor that is used in most of the data mining techniques.

There fore While constructing table one cannot afford to generate & scan the power set of items, it's computationally too expensive. So instead we scan the core lattice to generate the necessary output.

One has to be careful about the use of space. While scanning Lattice the program was creating object for each element in the scan. By some smart ways one can avoid fatal steps like system over load.

The Coding is done as followss: -

1. The software is written in modular way.
2. The codes is arranged as they are compiled.
3. Code is readable and well documented.

Source code: -

Software consist of four classes. The arrangement of code is as follows: -

(i) SummarizedMatrixElt.java

(ii) LIS_Lattice_Elt.java

(iii) LIS_Lattice.java

(iv) SummarizedMatrix.java

The whole architecture of the system as described in the previous chapter was implemented successfully in JAVA 1.2 on Oracle JDeveloper 8.0 environment.

Large item set core lattice file was successfully generated according to the theory mentioned in the previous chapter. The lattice had 5 levels. So it was a fast convergence.

Large item set core lattice paths file was also successfully generated enumerating all the possible paths that exist in core lattice. It was sorted in the increasing order w.r.t. conditional entropy. Increasing order was chosen for the simple reason that minimum entropy means maximum information.

By observing the file one could see that in most cases addition of item to the predictor set increases the information/prediction capacity of the predictor set (as one intuitively would expect). This results paves way to answering important questions that were raised in section 3.1.

Top Predictor Sets table was also successfully generated by scanning the Lattice paths. The outputted top 100 predictors for an item. Now instead of 7000 association rules we have 100 rows and that too with much richer information content.

Also in the next chapter many ideas with some details are given for measuring the interdependence between items, core strength, how prediction measure of an item changes with addition of attribute and areas where this approach can be very effective.

Chapter 5

CONCLUSIONS

The project came into existence when I started discovering the important role that data is playing in our lives and the importance of mining this data and extracting relevant data or knowledge from it. It also seen that not just any arbitrary mining will yield meaningful results; we have to mine data in a compact way. With the problem of II order data mining or Compact Knowledge Discovery, I went on to see the potential of Core based approach for solving the II order data mining problem and finally, formulated a core based theory for solving it. We successfully applied the theory in case of Association rule discovery.

The code gives satisfying result after the conditional entropy was set to perform under given threshold so as to get the desired results. Earlier the system use to get hanged and the output was corrupted as there was an enormous amount of data that was to be scanned and then knowledge was to be extracted from the same.

SAS System used is a data warehousing software system, which was used to generate large item set and association rules. SAS software saved time and also the required work was done efficiently and without mistakes. The Software tool is needed for analytical solutions, data mining, business visualization, rapid applications development, and much more. We used version 8 of SAS

The given theory can still be refined (we give many pointer in the next paragraph in this context.) It's not that all the problems in data mining has been solved but we surely have initiated a novel and valid way of mining data.

It's just the tip of iceberg. I believe that a wealth of information can be extracted from the Lattice paths. The following are some of the suggestions in this regard:

1. How addition of item to the predictor set affect prediction for a given item

   This may lead to power set of options so statically outputting this information in a tabular form may not be possible unless the number of distinct items are very small (<10). But this functionality can be achieved dynamically. At run time input the predicted item and predictor set and output the information change that takes place for the predicted item by the addition or deletion of items in the predictor set.

2. Core strength

   One can also think of some mathematical way of measuring strength of a core.

3. Bio-informatics

   Discovering the functionality of genes and the interdependence between genes is time consuming, expensive as well as difficult process. The theory developed in this dissertation can prove to be very handy tool in solving many of the problems encountered in the field of Bio-informatics

4. Domain knowledge

   One can use domain knowledge in form of templates (by klemettin's [7]) or any other form on the top of core based approach to refine one's results.

5. Time complexity

   The focus of my dissertation was mainly to formulate a theory which can mine data in compact way. There is room for efficient implementation of the proposed theory thus improving the time complexity.

REFERENCES

[1] Dr. Raj Bhatnagar et al. 'Mining Microarray Expression Data for Classifier Gene-Cores', BIOKDD01: Workshop on Data Mining in Bio-informatics (with SIGKDD01 conference).

[2] Gregory Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, Knowledge Discovery in Databases, pages 229-248. AAAI Press / The MIT Press, Menlo Park, CA, 1991.

[3] Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem Mauricio A. Hernández, Salvatore J. Stolfo pp. 9-37

[4] A Study of Two Sampling Methods for Analyzing Large Datasets with ILP Ashwin Srinivasan pp. 95-123

[5] Levelwise Search and Borders of Theories in Knowledge Discovery Heikki Mannila, Hannu Toivonen pp. 241-258

[6] Peter Hoschka and Willi Klosgen. A support system for interpreting statistical data. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, Knowledge Discovery in Databases, pages 325-345. AAAI Press / The MIT Press, Menlo Park, CA, 1991.

[7] Mika Klemettinen et al. Finding Interesting Rules from Large Sets of Discovered Association Rules. Appeared in the Third International Conference on Information and Knowledge management, Ed. Nabil R. Adam, Bharat K. Bhargava and Yelena Yesha, 401 – 407. Nov 29 – Dec 2, 1994, Gaithursburg, Maryland. ACM Press.

[8] R.Agrawal, T.Imielinski and A.Swami. Mining association rules between sets of items in large databases. In *Proc. 1993 ACM SIGMOD Int. Conf. Management of Data (SIGMOD'93)*, pages 207-216, Washington, DC, May 1993.

[10] D.W.Cheung, J.Han,V.Ng and C.W.Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In *Proc. 1996 Int. Conf. Data Engineering (ICDE'96)*, pages 106-114, New Orleans, La, Feb. 1996.

Appendix A: SAS System

SAS System: -SAS System, is the leading decision support and data warehousing software suite that brings together all the tools you need for analytical solutions, data mining, business visualization, rapid applications development, and much more. We used version 8 of SAS. When you open the SAS system there is plenty of help that user can access from the help menu.

Enterprise Miner: -For this dissertation I needed a Data mining tool. The Data mining tool that comes with SAS is Enterprise miner. One can activate it by going to solutions menu and then selecting analysis option, which will give one option of Enterprise Miner. Once you select the Enterprise Miner (EM) option from the help menu one can get access to EM reference. All the information that one needs one can get from the EM reference (if not a little bit struggle here and there should give you the required information. In Computer Science to learn one has to struggle).

Association Node: - We used SAS system to generate Large item sets and Association rules for a given database. All this functionality can be achieved using Association Node. Reading about the Association Node from the EM reference is sufficient to make proficient in the use of this node.

The menu bar of SAS gives you options depending on which area you are working. If you have selected the Explorer section then the File menu bar gives you option of Import/Export. The Import/Export option is also active if the result window is active. It has been found that results can correctly exported in '.txt' format, any other format like Access,.. is not guaranteed to get exported correctly.

For my dissertation we used databases SAMPSIO.ASSOCS from SAS libraries.

Note: - Don't put your data source in the work library because it's a temporary library and with every new invocation of SAS system a new work library is created deleting old work library and it's contents.

SAS home page url: www.sas.com

Appendix B: Large item set file

heineken
cracker
hering
olives
bourbon
baguette
corned_b
avocado
soda
chicken
apples
ice_crea
ham
artichok
sardines
peppers
coke
turkey
steak
bordeaux

heineken	cracker
hering	heineken
heineken	baguette
soda	heineken
olives	hering
heineken	artichok
soda	cracker
hering	baguette

heineken	avocado
olives	bourbon
hering	corned_b
cracker	bourbon
olives	corned_b
turkey	olives
ice_crea	coke
baguette	avocado
avocado	artichok
heineken	bourbon
olives	heineken
hering	cracker
heineken	chicken
olives	cracker
ice_crea	heineken
sardines	heineken
cracker	baguette
soda	hering
heineken	ham
heineken	coke
hering	avocado
cracker	corned_b
soda	olives
cracker	chicken
cracker	avocado
peppers	cracker
hering	ham
ham	corned_b
olives	apples
olives	ham
hering	artichok

hering	apples
cracker	artichok
soda	bourbon
olives	ice_crea
peppers	bourbon
soda	baguette
corned_b	bourbon
sardines	ice_crea
corned_b	apples
turkey	hering
ham	cracker
steak	hering
olives	coke
sardines	coke
chicken	bourbon
baguette	apples
steak	olives
ice_crea	bourbon
corned_b	chicken
turkey	bourbon
ice_crea	chicken
coke	bourbon
peppers	baguette
coke	chicken
avocado	apples
peppers	corned_b
ham	avocado
sardines	chicken
baguette	artichok
turkey	ham
turkey	corned_b

steak	corned_b
sardines	baguette
peppers	apples
ham	artichok
sardines	avocado
turkey	ice_crea
sardines	apples
peppers	avocado
heineken	corned_b
peppers	chicken
sardines	peppers
turkey	coke
steak	apples
hering	bourbon
heineken	apples
peppers	heineken
olives	baguette
bourbon	baguette
hering	chicken
turkey	heineken
corned_b	baguette
steak	heineken
cracker	apples
ice_crea	cracker
bourbon	apples
peppers	olives
olives	avocado
corned_b	avocado
steak	cracker
sardines	cracker
peppers	hering

olives	chicken
ham	bourbon
chicken	baguette
olives	artichok
bourbon	artichok
soda	avocado
sardines	olives
cracker	coke
corned_b	artichok
turkey	cracker
sardines	hering
sardines	bourbon
ice_crea	hering
ice_crea	baguette
ice_crea	ham
coke	apples
soda	corned_b
ice_crea	apples
turkey	baguette
chicken	artichok
hering	coke
soda	peppers
chicken	apples
bourbon	avocado
steak	baguette
soda	chicken
sardines	corned_b
ice_crea	corned_b
coke	baguette
steak	avocado
artichok	apples

ham	coke
soda	artichok
peppers	ice_crea
corned_b	coke
steak	bourbon
ham	chicken
ice_crea	artichok
soda	apples
turkey	apples
turkey	avocado
soda	ice_crea
ham	baguette
chicken	avocado
turkey	artichok
soda	ham
sardines	artichok
peppers	coke
soda	sardines
ice_crea	avocado
peppers	ham
ham	apples
coke	artichok
steak	ham
soda	coke
sardines	ham
turkey	soda
steak	soda
turkey	sardines
steak	artichok
peppers	artichok
turkey	steak

turkey	peppers	
steak	peppers	
steak	ice_crea	
steak	chicken	
turkey	chicken	
coke	avocado	
steak	sardines	
heineken	bordeaux	
steak	coke	
olives	bordeaux	
hering	bordeaux	
cracker	bordeaux	
soda	heineken	cracker
hering	heineken	baguette
olives	hering	corned_b
heineken	avocado	artichok
hering	heineken	cracker
heineken	cracker	bourbon
soda	hering	heineken
heineken	cracker	baguette
heineken	cracker	artichok
heineken	cracker	avocado
soda	hering	cracker
olives	heineken	cracker
ice_crea	heineken	coke
soda	heineken	baguette
soda	cracker	bourbon
heineken	baguette	avocado
hering	heineken	avocado
olives	heineken	bourbon
sardines	ice_crea	coke

olives	cracker	bourbon
hering	heineken	artichok
soda	hering	baguette
hering	cracker	baguette
hering	baguette	avocado
turkey	olives	bourbon
soda	olives	heineken
soda	cracker	baguette
soda	olives	cracker
heineken	ham	cracker
soda	olives	bourbon
soda	heineken	bourbon
sardines	ice_crea	heineken
olives	ham	corned_b
ice_crea	coke	chicken
hering	ham	corned_b
heineken	baguette	artichok
turkey	olives	hering
turkey	olives	corned_b
turkey	olives	ham
olives	hering	ham
sardines	heineken	coke
sardines	heineken	chicken
ice_crea	heineken	chicken
heineken	coke	chicken
sardines	coke	chicken
peppers	cracker	bourbon
ice_crea	coke	bourbon
hering	avocado	artichok
heineken	ham	avocado
steak	olives	hering

sardines	ice_crea	chicken
olives	corned_b	apples
hering	baguette	artichok
olives	ice_crea	coke
olives	hering	apples
turkey	hering	corned_b
steak	hering	corned_b
hering	corned_b	apples
heineken	ham	artichok
ham	avocado	artichok
cracker	chicken	bourbon
cracker	avocado	sartichok
cracker	corned_b	bourbon
baguette	avocado	artichok
baguette	avocado	apples
turkey	olives	ice_crea
steak	olives	corned_b
olives	coke	bourbon
olives	ice_crea	bourbon
turkey	ice_crea	coke
turkey	hering	ham
turkey	olives	coke
turkey	ham	corned_b
ham	cracker	avocado
steak	corned_b	apples
peppers	cracker	corned_b
peppers	baguette	avocado
sardines	baguette	avocado
peppers	chicken	bourbon
ham	cracker	artichok
cracker	corned_b	chicken

turkey	ice_crea	bourbon
steak	olives	apples
steak	hering	apples
peppers	cracker	chicken
turkey	coke	bourbon
corned_b	chicken	bourbon
peppers	corned_b	bourbon
peppers	baguette	apples
peppers	corned_b	chicken
peppers	avocado	apples
sardines	baguette	apples
sardines	peppers	avocado
sardines	peppers	baguette
sardines	avocado	apples
sardines	peppers	apples
heineken	cracker	chicken
olives	hering	heineken
peppers	heineken	cracker
heineken	cracker	corned_b
olives	hering	cracker
soda	olives	hering
olives	hering	bourbon
hering	heineken	bourbon
olives	corned_b	bourbon
heineken	chicken	bourbon
hering	corned_b	baguette
ice_crea	heineken	cracker
soda	heineken	artichok
sardines	heineken	cracker
hering	heineken	corned_b
soda	heineken	avocado

olives	hering	baguette
hering	baguette	apples
hering	heineken	ham
hering	cracker	corned_b
heineken	chicken	baguette
heineken	chicken	artichok
sardines	heineken	baguette
soda	peppers	cracker
soda	cracker	chicken
ice_crea	heineken	bourbon
hering	heineken	chicken
hering	cracker	bourbon
steak	heineken	cracker
olives	cracker	corned_b
hering	corned_b	bourbon
heineken	bourbon	artichok
ham	cracker	corned_b
peppers	cracker	baguette
cracker	baguette	avocado
hering	cracker	avocado
heineken	cracker	apples
heineken	coke	bourbon
steak	hering	heineken
soda	heineken	chicken
olives	heineken	coke
olives	ham	bourbon
hering	heineken	apples
heineken	cracker	coke
heineken	bourbon	baguette
heineken	baguette	apples
turkey	hering	heineken

soda	cracker	artichok
olives	ice_crea	heineken
olives	heineken	baguette
heineken	ham	corned_b
soda	cracker	corned_b
soda	cracker	avocado
hering	corned_b	avocado
hering	avocado	apples
cracker	bourbon	baguette
sardines	hering	heineken
sardines	heineken	avocado
sardines	heineken	artichok
peppers	olives	bourbon
peppers	corned_b	apples
olives	heineken	corned_b
olives	bourbon	apples
hering	ham	avocado
hering	cracker	artichok
heineken	corned_b	bourbon
turkey	olives	heineken
soda	baguette	avocado
peppers	heineken	bourbon
peppers	heineken	baguette
olives	hering	avocado
ice_crea	heineken	artichok
steak	olives	bourbon
hering	corned_b	artichok
heineken	chicken	avocado
heineken	avocado	apples
turkey	olives	cracker
turkey	olives	apples

steak	hering	baguette
soda	hering	corned_b
ice_crea	heineken	baguette
hering	corned_b	chicken
heineken	corned_b	chicken
avocado	artichok	apples
soda	olives	corned_b
peppers	bourbon	apples
ice_crea	chicken	bourbon
hering	ham	baguette
heineken	corned_b	artichok
turkey	steak	olives
turkey	hering	baguette
soda	hering	bourbon
peppers	hering	corned_b
olives	heineken	ham
hering	ham	cracker
heineken	bourbon	avocado
cracker	bourbon	artichok
turkey	heineken	cracker
turkey	heineken	bourbon
peppers	hering	baguette
peppers	heineken	avocado
ice_crea	heineken	ham
ice_crea	ham	coke
ice_crea	cracker	bourbon
hering	cracker	chicken
heineken	corned_b	baguette
heineken	corned_b	avocado
coke	chicken	bourbon
turkey	soda	olives

steak	soda	cracker
soda	peppers	bourbon
soda	heineken	ham
soda	ham	cracker
sardines	ice_crea	bourbon
peppers	heineken	chicken
peppers	cracker	apples
peppers	bourbon	baguette
olives	ice_crea	corned_b
olives	hering	artichok
olives	heineken	chicken
olives	heineken	artichok
olives	cracker	baguette
olives	chicken	bourbon
ice_crea	hering	heineken
hering	chicken	baguette
heineken	coke	baguette
heineken	coke	artichok
heineken	artichok	apples
ham	cracker	bourbon
cracker	chicken	baguette
cracker	bourbon	apples
corned_b	avocado	artichok
turkey	ham	bourbon
turkey	corned_b	bourbon
steak	soda	heineken
soda	peppers	heineken
soda	ice_crea	heineken
soda	hering	avocado
sardines	heineken	ham
sardines	coke	bourbon

peppers	olives	corned_b
peppers	hering	cracker
olives	heineken	avocado
olives	corned_b	chicken
ice_crea	coke	baguette
hering	heineken	coke
hering	artichok	apples
heineken	ham	chicken
ham	corned_b	bourbon
ham	corned_b	avocado
cracker	corned_b	baguette
corned_b	chicken	apples
turkey	cracker	bourbon
olives	ice_crea	hering
olives	ham	cracker
ice_crea	heineken	avocado
ice_crea	coke	apples
hering	ham	artichok
hering	ham	apples
steak	hering	cracker
steak	hering	avocado
soda	sardines	heineken
soda	olives	baguette
sardines	olives	ice_crea
sardines	heineken	apples
olives	hering	coke
olives	corned_b	coke
cracker	corned_b	artichok
corned_b	bourbon	apples
turkey	olives	baguette
steak	olives	heineken

steak	ham	corned_b
soda	heineken	coke
soda	corned_b	bourbon
soda	chicken	bourbon
sardines	ice_crea	apples
sardines	hering	baguette
sardines	heineken	bourbon
sardines	cracker	baguette
peppers	cracker	avocado
olives	ice_crea	apples
olives	cracker	chicken
olives	corned_b	baguette
olives	corned_b	avocado
olives	coke	apples
olives	bourbon	baguette
olives	baguette	apples
ice_crea	cracker	coke
ice_crea	coke	artichok
hering	bourbon	baguette
cracker	coke	bourbon
turkey	ice_crea	ham
steak	olives	ham
steak	hering	ham
steak	heineken	artichok
steak	cracker	bourbon
soda	peppers	baguette
soda	heineken	apples
sardines	peppers	heineken
sardines	cracker	avocado
olives	ice_crea	ham
heineken	ham	coke

heineken	ham	bourbon
cracker	corned_b	apples
cracker	baguette	artichok
corned_b	baguette	avocado
bourbon	baguette	avocado
turkey	hering	cracker
turkey	hering	bourbon
turkey	hering	avocado
turkey	heineken	artichok
steak	soda	hering
steak	heineken	baguette
steak	heineken	avocado
sardines	olives	coke
sardines	coke	apples
peppers	ice_crea	bourbon
ice_crea	cracker	chicken
cracker	baguette	apples
corned_b	baguette	apples
corned_b	avocado	apples
baguette	artichok	apples
turkey	hering	artichok
steak	olives	cracker
steak	baguette	avocado
soda	olives	apples
sardines	ice_crea	baguette
sardines	avocado	artichok
peppers	olives	cracker
peppers	hering	heineken
peppers	corned_b	baguette
peppers	avocado	artichok
olives	ham	coke

olives	ham	apples
olives	cracker	artichok
olives	bourbon	artichok
ham	corned_b	chicken
chicken	avocado	artichok
turkey	peppers	olives
turkey	ice_crea	heineken
turkey	heineken	coke
steak	olives	ice_crea
steak	hering	artichok
steak	corned_b	chicken
steak	avocado	apples
soda	hering	artichok
soda	heineken	corned_b
soda	cracker	apples
sardines	peppers	ice_crea
sardines	olives	heineken
sardines	heineken	corned_b
sardines	chicken	bourbon
peppers	ice_crea	coke
peppers	heineken	artichok
peppers	ham	cracker
peppers	ham	corned_b
olives	ice_crea	cracker
olives	hering	chicken
olives	cracker	coke
olives	avocado	apples
ice_crea	corned_b	chicken
ham	cracker	chicken
ham	corned_b	artichok
ham	corned_b	apples

bourbon	avocado	artichok	
turkey	heineken	baguette	
turkey	corned_b	apples	
steak	soda	olives	
steak	peppers	apples	
steak	baguette	apples	
steak	avocado	artichok	
soda	sardines	baguette	
soda	baguette	artichok	
soda	avocado	artichok	
sardines	peppers	coke	
sardines	peppers	chicken	
sardines	peppers	bourbon	
sardines	coke	baguette	
peppers	olives	apples	
peppers	cracker	artichok	
olives	heineken	apples	
olives	cracker	savocado	
olives	cracker	apples	
olives	corned_b	artichok	
ice_crea	hering	corned_b	
heineken	ham	baguette	
cracker	corned_b	avocado	
cracker	chicken	artichok	
corned_b	coke	chicken	
soda	hering	heineken	cracker
soda	hering	heineken	baguette
soda	heineken	cracker	baguette
hering	heineken	cracker	baguette
soda	olives	heineken	cracker
soda	heineken	cracker	bourbon

hering	heineken	baguette	avocado
olives	hering	ham	corned_b
sardines	ice_crea	heineken	coke
sardines	ice_crea	heineken	chicken
sardines	ice_crea	coke	chicken
sardines	heineken	coke	chicken
ice_crea	heineken	coke	chicken
soda	hering	cracker	baguette
hering	heineken	baguette	artichok
hering	heineken	avocado	artichok
turkey	olives	hering	corned_b
heineken	cracker	avocado	artichok
heineken	ham	avocado	artichok
steak	olives	hering	corned_b
olives	hering	corned_b	apples
olives	heineken	cracker	bourbon
soda	olives	cracker	bourbon
soda	olives	heineken	bourbon
turkey	olives	ham	corned_b
heineken	baguette	avocado	artichok
turkey	olives	hering	ham
turkey	hering	ham	corned_b
heineken	ham	cracker	avocado
heineken	ham	cracker	artichok
hering	baguette	avocado	artichok
ham	cracker	avocado	artichok
steak	olives	hering	apples
steak	olives	corned_b	apples
steak	hering	corned_b	apples
olives	ice_crea	coke	bourbon
turkey	olives	ice_crea	bourbon

turkey	ice_crea	coke	bourbon
turkey	olives	ice_crea	coke
turkey	olives	coke	bourbon
peppers	cracker	chicken	bourbon
peppers	cracker	corned_b	bourbon
cracker	corned_b	chicken	bourbon
sardines	baguette	avocado	apples
peppers	cracker	corned_b	chicken
peppers	corned_b	chicken	bourbon
peppers	baguette	avocado	apples
sardines	peppers	baguette	avocado
sardines	peppers	baguette	apples
sardines	peppers	avocado	apples
soda	heineken	cracker	artichok
soda	heineken	cracker	avocado
hering	heineken	cracker	avocado
olives	hering	heineken	cracker
soda	olives	hering	heineken
soda	olives	hering	cracker
hering	heineken	cracker	artichok
peppers	heineken	cracker	bourbon
ice_crea	heineken	coke	bourbon
turkey	olives	ham	bourbon
steak	soda	heineken	cracker
soda	peppers	heineken	cracker
soda	hering	cracker	bourbon
soda	heineken	cracker	chicken
soda	peppers	cracker	bourbon
hering	heineken	cracker	bourbon
heineken	cracker	chicken	bourbon
heineken	cracker	baguette	avocado

soda	hering	heineken	bourbon
sardines	ice_crea	coke	bourbon
hering	heineken	chicken	baguette
hering	heineken	baguette	apples
soda	heineken	ham	cracker
olives	ice_crea	heineken	coke
heineken	cracker	corned_b	bourbon
heineken	corned_b	avocado	artichok
soda	hering	heineken	avocado
soda	hering	baguette	avocado
heineken	baguette	avocado	apples
turkey	olives	heineken	bourbon
soda	heineken	baguette	avocado
soda	cracker	chicken	bourbon
olives	hering	heineken	bourbon
olives	hering	corned_b	bourbon
hering	heineken	corned_b	baguette
hering	cracker	baguette	avocado
heineken	avocado	artichok	apples
soda	olives	heineken	baguette
olives	hering	corned_b	baguette
olives	hering	corned_b	avocado
olives	heineken	cracker	baguette
heineken	cracker	baguette	artichok
steak	olives	hering	ham
steak	olives	ham	corned_b
steak	hering	ham	corned_b
soda	cracker	corned_b	bourbon
sardines	olives	ice_crea	coke
olives	hering	heineken	baguette
olives	hering	ham	apples

olives	ham	corned_b	apples	
hering	heineken	bourbon	baguette	
hering	ham	corned_b	apples	
heineken	cracker	bourbon	artichok	
heineken	chicken	avocado	artichok	
baguette	avocado	artichok	apples	
turkey	olives	ice_crea	ham	
turkey	olives	hering	bourbon	
turkey	olives	corned_b	bourbon	
steak	hering	heineken	baguette	
soda	olives	cracker	baguette	
soda	hering	heineken	artichok	
soda	heineken	cracker	corned_b	
soda	heineken	cracker	apples	
olives	hering	heineken	corned_b	
olives	heineken	coke	bourbon	
sardines	ice_crea	heineken	coke	chicken
soda	hering	heineken	cracker	baguette
soda	olives	heineken	cracker	bourbon
turkey	olives	hering	ham	corned_b
hering	heineken	baguette	avocado	artichok
heineken	ham	cracker	avocado	artichok
steak	olives	hering	corned_b	apples
turkey	olives	ice_crea	coke	bourbon
peppers	cracker	corned_b	chicken	bourbon
sardines	peppers	baguette	avocado	apples
soda	olives	hering	heineken	cracker
soda	hering	heineken	cracker	bourbon
soda	hering	heineken	baguette	avocado
hering	heineken	cracker	baguette	avocado
steak	olives	hering	ham	corned_b

olives	hering	ham	corned_b	apples
soda	olives	heineken	cracker	baguette
soda	olives	hering	heineken	cracker
hering	heineken	baguette	avocado	artichok
heineken	ham	cracker	avocado	artichok
steak	olives	hering	corned_b	apples

Appendix C: The Software

Database programming is different from normal programming. This is mainly because in database programming one is dealing with huge amount of information. So while processing it one may reach a dead end in either time or memory.

While constructing table one cannot afford to generate & scan the power set of items, it's computationally too expensive. So instead we scan the core lattice to generate the necessary output.

One has to be careful about the use of space. While scanning Lattice the program was creating object for each element in the scan. After 2-3 hours of run the program threw an memory exception. By some smart ways one can avoid fatal steps.

Coding style in the software: -

- **(i)** The software is written in modular way.
- **(ii)** The codes is arranged as they are compiled.
- **(iii)** Code is readable and well documented.

Source code: -

Software consist of four classes. The arrangement of code is as follows: -

- **(i)** SummarizedMatrixElt.java
- **(ii)** LIS_Lattice_Elt.java
- **(iii)** LIS_Lattice.java
- **(iv)** SummarizedMatrix.java

Appendix D: Integer identifier for items

Item	**Integer identifier**
Heineken	1
Cracker	2
Soda	3
Olives	4
Baguette	5

Appendix E: Top Predictor Sets table

This appendix enlists top 50 predictor sets for some randomly chosen items i.e. lists the table column by column.

APPLES

[ham, corned_b, hering, olives]	==>	e 0.506
[steak, corned_b, olives, hering]	==>	e 0.506
[sardines, peppers, avocado, baguette]	==>	e 0.526
[avocado, heineken, baguette]	==>	e 0.554
[steak, olives, hering]	==>	e 0.554
[ham, olives, hering]	==>	e 0.555
[peppers, steak]	==>	e 0.559
[peppers, avocado, baguette]	==>	e 0.562
[heineken, baguette, hering]	==>	e 0.564
[sardines, avocado, baguette]	==>	e 0.565
[steak, hering, corned_b]	==>	e 0.566
[steak, olives, corned_b]	==>	e 0.566
[heineken, artichok, avocado]	==>	e 0.567
[sardines, peppers, baguette]	==>	e 0.569
[avocado, baguette, artichok]	==>	e 0.570
[ham, hering, corned_b]	==>	e 0.570
[ham, olives, corned_b]	==>	e 0.572
[sardines, peppers, avocado]	==>	e 0.573

[corned_b, olives, hering]	==>	e 0.577
[soda, heineken, cracker]	==>	e 0.579
[olives, avocado]	==>	e 0.585
[olives, peppers]	==>	e 0.587
[baguette, steak]	==>	e 0.604
[avocado, steak]	==>	e 0.605
[corned_b, baguette]	==>	e 0.615
[olives, baguette]	==>	e 0.627
[corned_b, avocado]	==>	e 0.628
[corned_b, turkey]	==>	e 0.637
[heineken, sardines]	==>	e 0.642
[corned_b, cracker]	==>	e 0.644
[heineken, olives]	==>	e 0.645
[baguette, artichok]	==>	e 0.649
[corned_b, peppers]	==>	e 0.650
[avocado, peppers]	==>	e 0.652
[olives, cracker]	==>	e 0.653
[hering, artichok]	==>	e 0.654
[avocado, sardines]	==>	e 0.655
[olives, soda]	==>	e 0.656
[corned_b, bourbon]	==>	e 0.658
[olives, coke]	==>	e 0.662
[baguette, cracker]	==>	e 0.663

[olives, ice_crea]	==>	e 0.664
[corned_b, chicken]	==>	e 0.664
[cracker, peppers]	==>	e 0.666
[peppers, bourbon]	==>	e 0.666
[corned_b, steak]	==>	e 0.667
[baguette, peppers]	==>	e 0.668
[sardines, coke]	==>	e 0.670
[sardines, ice_crea]	==>	e 0.670
[baguette, sardines]	==>	e 0.671

ARTICHOK

[ham, avocado, heineken, cracker]	==>	e 0.477
[avocado, baguette, heineken, hering]	==>	e 0.495
[heineken, avocado, corned_b]	==>	e 0.496
[heineken, avocado, chicken]	==>	e 0.514
[ham, heineken, cracker]	==>	e 0.517
[heineken, avocado, apples]	==>	e 0.518
[heineken, avocado, ham]	==>	e 0.527
[heineken, cracker, hering]	==>	e 0.535
[heineken, cracker, baguette]	==>	e 0.538
[ham, avocado, cracker]	==>	e 0.539
[heineken, cracker, bourbon]	==>	e 0.539
[heineken, avocado, hering]	==>	e 0.540

[heineken, avocado, cracker]	==>	e 0.540
[heineken, soda, hering]	==>	e 0.554
[heineken, avocado, baguette]	==>	e 0.561
[heineken, baguette, hering]	==>	e 0.567
[avocado, hering, baguette]	==>	e 0.571
[avocado, bourbon]	==>	e 0.572
[avocado, baguette, apples]	==>	e 0.574
[avocado, chicken]	==>	e 0.587
[heineken, cracker, soda]	==>	e 0.591
[heineken, turkey]	==>	e 0.596
[heineken, steak]	==>	e 0.600
[heineken, peppers]	==>	e 0.602
[heineken, apples]	==>	e 0.605
[avocado, steak]	==>	e 0.607
[heineken, corned_b]	==>	e 0.608
[avocado, corned_b]	==>	e 0.631
[avocado, soda]	==>	e 0.636
[cracker, corned_b]	==>	e 0.643
[avocado, peppers]	==>	e 0.643
[heineken, ice_crea]	==>	e 0.644
[avocado, ham]	==>	e 0.644
[heineken, ham]	==>	e 0.644
[avocado, sardines]	==>	e 0.645

[heineken, olives]	==>	e 0.645
[hering, turkey]	==>	e 0.647
[hering, apples]	==>	e 0.647
[cracker, olives]	==>	e 0.647
[heineken, coke]	==>	e 0.648
[heineken, sardines]	==>	e 0.648
[cracker, ham]	==>	e 0.653
[cracker, chicken]	==>	e 0.663
[heineken, chicken]	==>	e 0.663
[cracker, hering]	==>	e 0.666
[cracker, peppers]	==>	e 0.666
[baguette, soda]	==>	e 0.669
[heineken, bourbon]	==>	e 0.669
[cracker, baguette]	==>	e 0.669
[hering, ham]	==>	e 0.671

AVOCADO

[baguette, heineken, hering, cracker]	==>	e 0.501
[ham, artichok, heineken, cracker]	==>	e 0.507
[artichok, baguette, heineken, hering]	==>	e 0.510
[soda, baguette, heineken, hering]	==>	e 0.514
[artichok, heineken, corned_b]	==>	e 0.523
[heineken, artichok, apples]	==>	e 0.534

[heineken, baguette, apples]	==>	e 0.537
[heineken, artichok, chicken]	==>	e 0.537
[ham, heineken, cracker]	==>	e 0.538
[sardines, peppers, apples, baguette]	==>	e 0.542
[heineken, artichok, ham]	==>	e 0.551
[baguette, apples, artichok]	==>	e 0.554
[heineken, artichok, hering]	==>	e 0.555
[baguette, hering, cracker]	==>	e 0.555
[heineken, hering, cracker]	==>	e 0.559
[heineken, artichok, baguette]	==>	e 0.560
[ham, artichok, cracker]	==>	e 0.560
[artichok, sardines]	==>	e 0.562
[artichok, hering, baguette]	==>	e 0.562
[heineken, baguette, cracker]	==>	e 0.564
[heineken, artichok, cracker]	==>	e 0.567
[artichok, steak]	==>	e 0.570
[hering, baguette, soda]	==>	e 0.571
[apples, peppers, baguette]	==>	e 0.573
[sardines, apples, baguette]	==>	e 0.575
[heineken, baguette, soda]	==>	e 0.575
[cracker, sardines]	==>	e 0.576
[artichok, peppers]	==>	e 0.576
[sardines, peppers, baguette]	==>	e 0.576

[heineken, soda, hering]	==>	e 0.576
[sardines, peppers, apples]	==>	e 0.579
[corned_b, olives, hering]	==>	e 0.584
[artichok, chicken]	==>	e 0.590
[heineken, baguette, hering]	==>	e 0.595
[artichok, bourbon]	==>	e 0.595
[baguette, steak]	==>	e 0.597
[baguette, bourbon]	==>	e 0.602
[heineken, steak]	==>	e 0.609
[heineken, soda, cracker]	==>	e 0.609
[artichok, apples]	==>	e 0.611
[artichok, soda]	==>	e 0.613
[heineken, peppers]	==>	e 0.615
[artichok, corned_b]	==>	e 0.615
[baguette, corned_b]	==>	e 0.618
[heineken, apples]	==>	e 0.623
[heineken, corned_b]	==>	e 0.624
[apples, steak]	==>	e 0.630
[artichok, ham]	==>	e 0.639
[apples, sardines]	==>	e 0.647

BAGUETTE

[avocado, heineken, hering, cracker]	==>	e 0.515
[soda, cracker, heineken, olives]	==>	e 0.523
[avocado, soda, heineken, hering]	==>	e 0.525
[artichok, avocado, heineken, hering]	==>	e 0.528
[soda, heineken, hering, cracker]	==>	e 0.531
[heineken, hering, apples]	==>	e 0.533
[heineken, chicken, hering]	==>	e 0.534
[heineken, hering, corned_b]	==>	e 0.535
[steak, heineken, hering]	==>	e 0.538
[avocado, heineken, apples]	==>	e 0.551
[heineken, hering, bourbon]	==>	e 0.551
[sardines, peppers, apples, avocado]	==>	e 0.552
[heineken, hering, olives]	==>	e 0.553
[heineken, cracker, olives]	==>	e 0.554
[avocado, hering, cracker]	==>	e 0.556
[soda, sardines]	==>	e 0.557
[hering, sardines]	==>	e 0.566
[heineken, artichok, hering]	==>	e 0.566
[avocado, apples, artichok]	==>	e 0.566
[soda, heineken, olives]	==>	e 0.567
[hering, avocado, soda]	==>	e 0.567
[heineken, avocado, soda]	==>	e 0.569

[avocado, artichok, hering]	==>	e 0.571
[avocado, bourbon]	==>	e 0.575
[cracker, sardines]	==>	e 0.575
[heineken, avocado, cracker]	==>	e 0.575
[apples, peppers, avocado]	==>	e 0.576
[heineken, artichok, cracker]	==>	e 0.576
[heineken, avocado, hering]	==>	e 0.576
[sardines, apples, avocado]	==>	e 0.576
[sardines, peppers, apples]	==>	e 0.576
[sardines, peppers, avocado]	==>	e 0.577
[soda, cracker, olives]	==>	e 0.577
[heineken, hering, cracker]	==>	e 0.578
[avocado, steak]	==>	e 0.581
[hering, peppers]	==>	e 0.583
[soda, cracker, hering]	==>	e 0.586
[soda, peppers]	==>	e 0.594
[heineken, soda, hering]	==>	e 0.594
[apples, artichok]	==>	e 0.599
[hering, chicken]	==>	e 0.601
[corned_b, hering, olives]	==>	e 0.601
[soda, artichok]	==>	e 0.603
[heineken, avocado, artichok]	==>	e 0.603
[cracker, apples]	==>	e 0.609

[heineken, turkey]	==>	e 0.609
[apples, steak]	==>	e 0.613
[heineken, steak]	==>	e 0.615
[heineken, peppers]	==>	e 0.620

HEINEKEN

[bordeaux]	==>	e 0.104
[soda, coke]	==>	e 0.431
[artichok, sardines]	==>	e 0.431
[artichok, coke]	==>	e 0.433
[soda, sardines]	==>	e 0.437
[artichok, steak]	==>	e 0.440
[soda, ice_crea]	==>	e 0.440
[artichok, turkey]	==>	e 0.442
[artichok, ice_crea]	==>	e 0.442
[sardines, ham]	==>	e 0.444
[avocado, ice_crea]	==>	e 0.456
[artichok, chicken]	==>	e 0.457
[coke, ham]	==>	e 0.461
[artichok, peppers]	==>	e 0.464
[soda, steak]	==>	e 0.465
[chicken, ham]	==>	e 0.465
[soda, chicken]	==>	e 0.469

[avocado, chicken]	==>	e 0.472
[coke, turkey]	==>	e 0.473
[soda, ham]	==>	e 0.473
[baguette, coke]	==>	e 0.475
[avocado, steak]	==>	e 0.476
[ice_crea, ham]	==>	e 0.481
[baguette, turkey]	==>	e 0.482
[baguette, ice_crea]	==>	e 0.482
[soda, artichok]	==>	e 0.485
[baguette, ham]	==>	e 0.486
[sardines, corned_b]	==>	e 0.486
[coke, ice_crea, chicken]	==>	e 0.488
[baguette, chicken]	==>	e 0.490
[sardines, coke, chicken]	==>	e 0.491
[ice_crea, turkey]	==>	e 0.491
[artichok, avocado, chicken]	==>	e 0.492
[coke, ice_crea, sardines]	==>	e 0.492
[sardines, chicken]	==>	e 0.492
[soda, apples]	==>	e 0.492
[soda, peppers]	==>	e 0.493
[baguette, steak]	==>	e 0.493
[chicken, coke]	==>	e 0.493
[sardines, ice_crea, chicken]	==>	e 0.496

[chicken, peppers]	==>	e 0.497
[hering, sardines]	==>	e 0.498
[sardines, coke, ice_crea, chicken]	==>	e 0.498
[avocado, sardines]	==>	e 0.498
[hering, coke]	==>	e 0.499
[artichok, apples]	==>	e 0.500
[cracker, sardines]	==>	e 0.501
[sardines, peppers]	==>	e 0.503
[sardines, bourbon]	==>	e 0.505
[avocado, bourbon]	==>	e 0.506
[cracker, soda, artichok]	==>	e 0.507
[chicken, ice_crea]	==>	e 0.507
[artichok, bourbon]	==>	e 0.508
[artichok, avocado, apples]	==>	e 0.508
[artichok, avocado, corned_b]	==>	e 0.518
[artichok, avocado, baguette, hering]	==>	e 0.533