CS 295B/CS 395B Systems for Knowledge Discovery

Lecture 3:

KDD Background

The University of Vermont

Outline

- What is Knowledge Discovery in Databases (KDD)?
    - Knowledge discovery
    - Data Mining
    - Databases
- What is KDD…today?
- Friday's readings

What is KDD?

The University of Vermont

What is knowledge discovery?

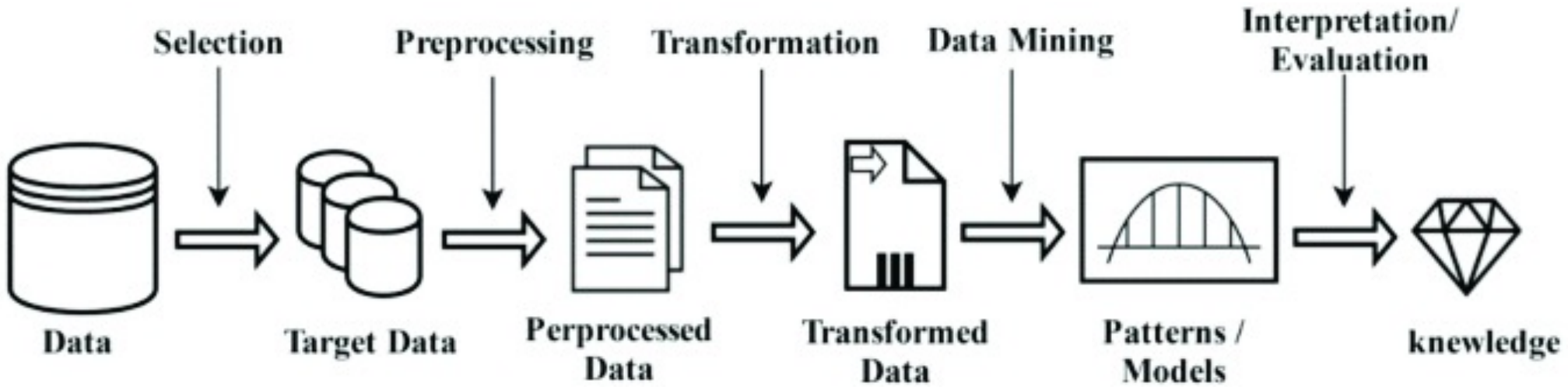


Image source: Ali et al. 2018

What is knowledge discovery?

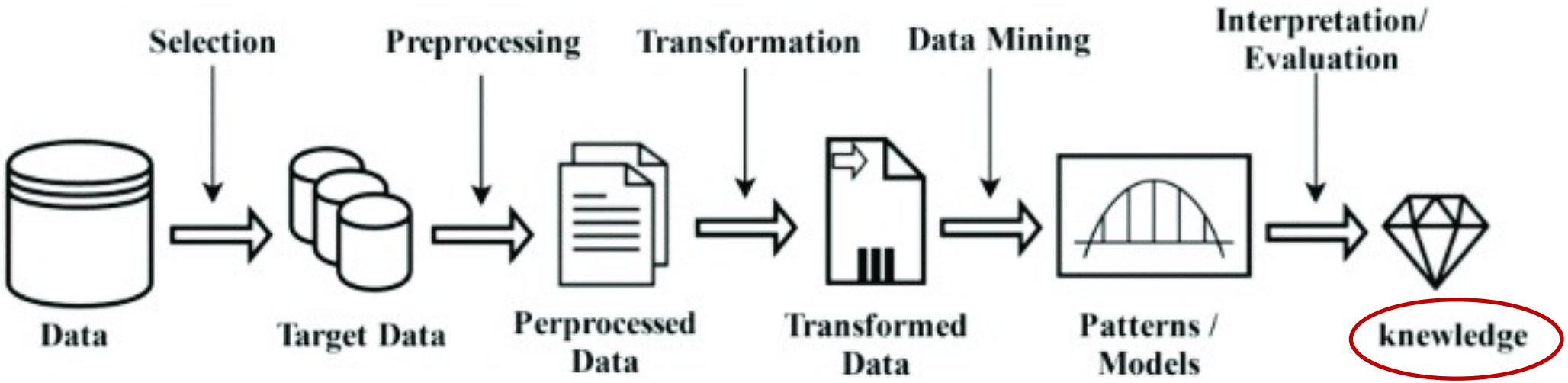


Image source: Ali et al. 2018

What is knowledge?

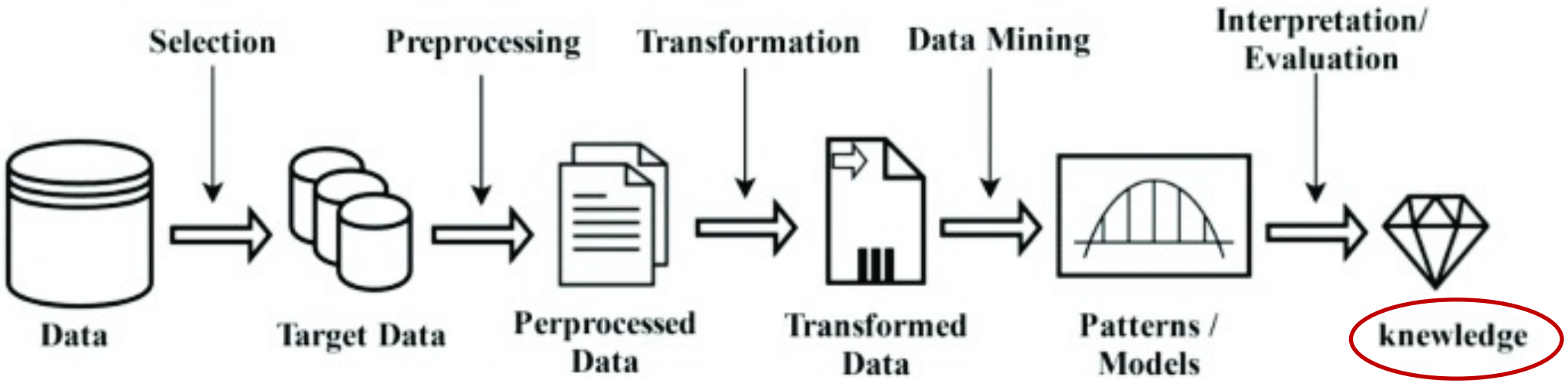


Image source: Ali et al. 2018

Mining Association Rules between Sets of Items in Large Databases

Rakesh Agrawal Tomasz Imielinski* Arun Swami

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We are given a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. We also present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

1 Introduction

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called *basket* data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This paper introduces the problem of "mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

The work reported in this paper could be viewed as a step towards enhancing databases with functionalities to process queries such as (we have omitted the confidence factor specification):

- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke.
- Find all rules that have "bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels.
- Find all rules that have "sausage" in the antecedent and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold.
- Find all the rules relating items located on shelves A and B in the store. These rules may help shelf planning by determining if the sale of items on shelf A is related to the sale of items on shelf B.
- Find the "best" k rules that have "bagels" in the consequent. Here, "best" can be formulated in terms of the confidence factors of the rules, or in terms

*Current address: Computer Science Department, Rutgers University, New Brunswick, NJ 08903

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0207...$1.50

Knowledge as Association

- Classic paper from SIGMOD 1993
  - Paper quite legible, Wikipedia article also very good!

Mining Association Rules between Sets of Items in Large Databases

Rakesh Agrawal Tomasz Imielinski* Arun Swami

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We are given a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. We also present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

1 Introduction

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called *basket* data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This paper introduces the problem of "mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

The work reported in this paper could be viewed as a step towards enhancing databases with functionalities to process queries such as (we have omitted the confidence factor specification):

- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke.
- Find all rules that have "bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels.
- Find all rules that have "sausage" in the antecedent and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold.
- Find all the rules relating items located on shelves A and B in the store. These rules may help shelf planning by determining if the sale of items on shelf A is related to the sale of items on shelf B.
- Find the "best" k rules that have "bagels" in the consequent. Here, "best" can be formulated in terms of the confidence factors of the rules, or in terms

*Current address: Computer Science Department, Rutgers University, New Brunswick, NJ 08903

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0207...$1.50

Knowledge as Association

- Classic paper from SIGMOD 1993
  - Paper quite legible, Wikipedia article also very good!
- Highly influential

Mining Association Rules between Sets of Items in Large Databases

Rakesh Agrawal Tomasz Imielinski* Arun Swami

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We are given a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. We also present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

1 Introduction

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called *basket* data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This paper introduces the problem of "mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

The work reported in this paper could be viewed as a step towards enhancing databases with functionalities to process queries such as (we have omitted the confidence factor specification):

- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke.
- Find all rules that have "bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels.
- Find all rules that have "sausage" in the antecedent and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold.
- Find all the rules relating items located on shelves A and B in the store. These rules may help shelf planning by determining if the sale of items on shelf A is related to the sale of items on shelf B.
- Find the "best" k rules that have "bagels" in the consequent. Here, "best" can be formulated in terms of the confidence factors of the rules, or in terms

*Current address: Computer Science Department, Rutgers University, New Brunswick, NJ 08903

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0207...$1.50

Knowledge as Association

- Classic paper from SIGMOD 1993
  - Paper quite legible, Wikipedia article also very good!
- Highly influential
- Classical AI approach
  - Learning probabilistic logical implications from data
  - Discrete spaces

Do stuff on the board

The University of Vermont

Why did I go into detail here?

Discuss: why might time matter?

Why did I go into detail here?

Temporal precedence is necessary but not sufficient for establishing causal relations.

Mining Association Rules between Sets of Items in Large Databases

Rakesh Agrawal Tomasz Imielinski* Arun Swami

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We are given a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. We also present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

1 Introduction

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called *basket* data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This paper introduces the problem of "mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

The work reported in this paper could be viewed as a step towards enhancing databases with functionalities to process queries such as (we have omitted the confidence factor specification):

- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke.
- Find all rules that have "bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels.
- Find all rules that have "sausage" in the antecedent and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold.
- Find all the rules relating items located on shelves A and B in the store. These rules may help shelf planning by determining if the sale of items on shelf A is related to the sale of items on shelf B.
- Find the "best" k rules that have "bagels" in the consequent. Here, "best" can be formulated in terms of the confidence factors of the rules, or in terms

*Current address: Computer Science Department, Rutgers University, New Brunswick, NJ 08903

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0207...$1.50

How is this not just stats?

- Stats problem:
  - Given a model, learn parameters
  - Variable selection (e.g., LASSO)

Mining Association Rules between Sets of Items in Large Databases

Rakesh Agrawal Tomasz Imielinski* Arun Swami

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We are given a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. We also present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

1 Introduction

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called *basket* data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This paper introduces the problem of "mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

The work reported in this paper could be viewed as a step towards enhancing databases with functionalities to process queries such as (we have omitted the confidence factor specification):

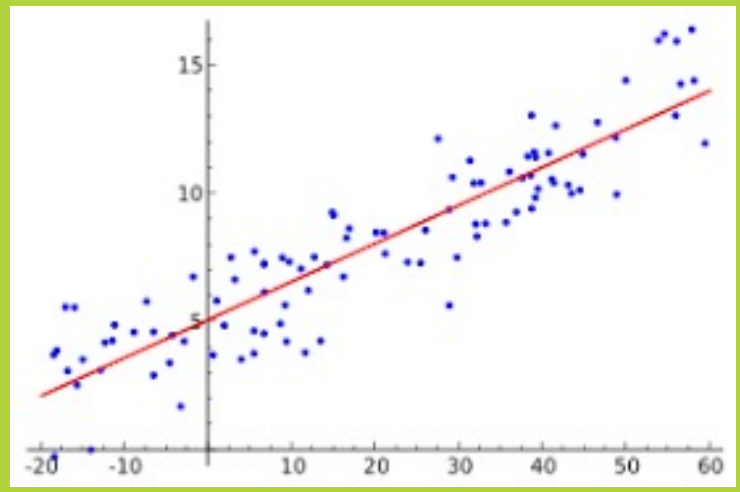
- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke.
- Find all rules that have "bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels.
- Find all rules that have "sausage" in the antecedent and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold.
- Find all the rules relating items located on shelves A and B in the store. These rules may help shelf planning by determining if the sale of items on shelf A is related to the sale of items on shelf B.
- Find the "best" k rules that have "bagels" in the consequent. Here, "best" can be formulated in terms of the confidence factors of the rules, or in terms

*Current address: Computer Science Department, Rutgers University, New Brunswick, NJ 08903

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0207...$1.50

How is this not just stats?

- Stats problem:
  - Given a model, learn parameters
  - Variable selection (e.g., LASSO)



Mining Association Rules between Sets of Items in Large Databases

Rakesh Agrawal Tomasz Imielinski* Arun Swami

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We are given a large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. We present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. We also present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

1 Introduction

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called *basket* data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information.

This paper introduces the problem of "mining" a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence, and presents an efficient algorithm for this purpose. An example of such an association rule is the statement that 90% of transactions that purchase bread and butter also purchase milk. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

The work reported in this paper could be viewed as a step towards enhancing databases with functionalities to process queries such as (we have omitted the confidence factor specification):

- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke.
- Find all rules that have "bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels.
- Find all rules that have "sausage" in the antecedent and "mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold.
- Find all the rules relating items located on shelves A and B in the store. These rules may help shelf planning by determining if the sale of items on shelf A is related to the sale of items on shelf B.
- Find the "best" k rules that have "bagels" in the consequent. Here, "best" can be formulated in terms of the confidence factors of the rules, or in terms

*Current address: Computer Science Department, Rutgers University, New Brunswick, NJ 08903

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0207...$1.50

How is this not just stats?

- Stats problem:
  - Given a model, learn parameters
  - Variable selection (e.g., LASSO)
- Knowledge Discovery problem:
  - Don't care so much about the weights
  - Care about the variables
  - Care about the relations

What is data mining?

The algorithmic process that produces knowledge

- e.g., the algorithm presented in Agrawal et al.

What is data mining?

The algorithmic process that produces knowledge

- e.g., the algorithm presented in Agrawal et al.

How does data mining differ from machine learning?

- Output: patterns, not predictors

What is data mining?

The algorithmic process that produces knowledge

- e.g., the algorithm presented in Agrawal et al.

How does data mining differ from machine learning?

- Output: patterns, not predictors

Discuss: how do you evaluate patterns vs predictors?

What is a database?

It's where the data live, duh.

What is a database?

In KDD, almost exclusively mean "relational database"

- an association of attributes (columns) with entities (tables)
- "relation" → "these things are associated"
  - "these things" → "tuples" → instance of cartesian product of attributes
- database objective → lay out data to suit retrieval/querying purposes
  - ancillary effect: useful for representing knowledge
  - structure is a form of inductive bias

What is a database?

In KDD, almost exclusively mean "relational database"

- **an association of attributes (columns) with entities (tables)**
- "relation" → "these things are associated"
  - "these things" → "tuples" → instance of cartesian product of attributes
- database objective → lay out data to suit retrieval/querying purposes
  - ancillary effect: useful for representing knowledge
  - structure is a form of inductive bias

What is a database?

In KDD, almost exclusively mean "relational database"

- an association of attributes (columns) with entities (tables)
- **"relation" → "these things are associated"**
  - "these things" → "tuples" → instance of cartesian product of attributes
- database objective → lay out data to suit retrieval/querying purposes
  - ancillary effect: useful for representing knowledge
  - structure is a form of inductive bias

What is a database?

In KDD, almost exclusively mean "relational database"

- an association of attributes (columns) with entities (tables)
- "relation" → "these things are associated"
  - **"these things" → "tuples" → instance of cartesian product of attributes**
- database objective → lay out data to suit retrieval/querying purposes
  - ancillary effect: useful for representing knowledge
  - structure is a form of inductive bias

What is a database?

In KDD, almost exclusively mean "relational database"

- an association of attributes (columns) with entities (tables)
- "relation" → "these things are associated"
  - "these things" → "tuples" → instance of cartesian product of attributes
- **database objective → lay out data to suit retrieval/querying purposes**
  - ancillary effect: useful for representing knowledge
  - structure is a form of inductive bias

What is a database?

In KDD, almost exclusively mean "relational database"

- an association of attributes (columns) with entities (tables)
- "relation" → "these things are associated"
  - "these things" → "tuples" → instance of cartesian product of attributes
- database objective → lay out data to suit retrieval/querying purposes
  - **ancillary effect: useful for representing knowledge**
  - structure is a form of inductive bias

What is a database?

In KDD, almost exclusively mean "relational database"

- an association of attributes (columns) with entities (tables)
- "relation" → "these things are associated"
  - "these things" → "tuples" → instance of cartesian product of attributes
- database objective → lay out data to suit retrieval/querying purposes
  - ancillary effect: useful for representing knowledge
  - **structure is a form of inductive bias**

What are the database *tasks*?

Make the database *fast*

- How?
  - Data layout
  - Schema design
  - Indexing
  - Query optimization

Most work in this space focuses on performance.

Very large systems (esp. information retrieval) → correctness

What is KDD…today?

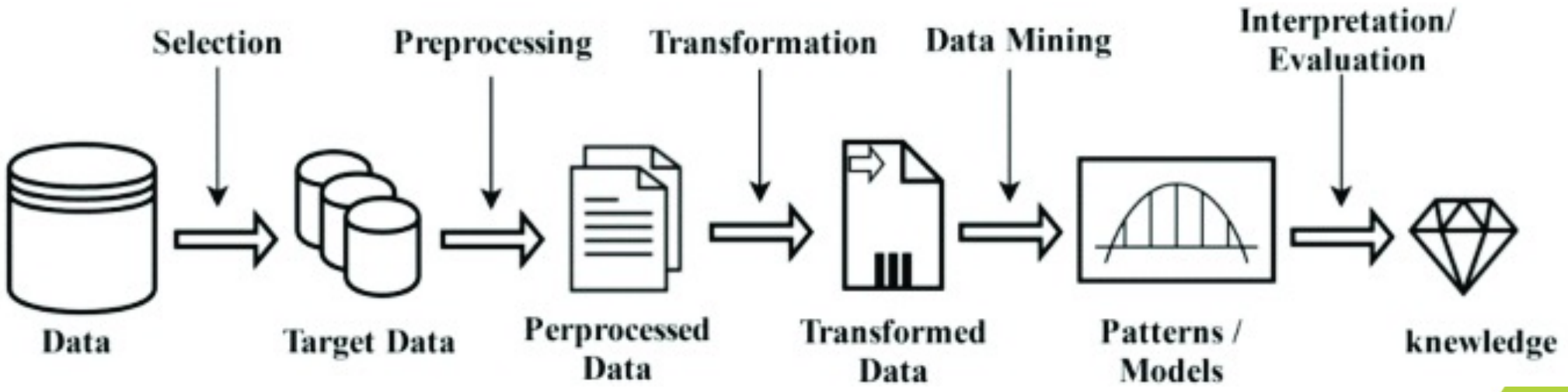
The University of Vermont

KDD today

- **30+ year evolution**
- Can think of each node in the diagram having expanded to its own (sub-)field
  - Who has the data? Much of this process is industrial.
  - Relationship to "data science"
- ACM KDD: flagship conference
  - ECML PKDD, SDM conferences

KDD today

- 30+ year evolution
- **Can think of each node in the diagram having expanded to its own (sub-)field**



KDD today

- 30+ year evolution
- **Can think of each node in the diagram having expanded to its own (sub-)field**
    - Who has the data? Much of this process is industrial.
    - Relationship to "data science"
- ACM KDD: flagship conference
    - ECML PKDD, SDM conferences

KDD today

- 30+ year evolution
- Can think of each node in the diagram having expanded to its own field
  - Who has the data? Much of this process is industrial.
  - Relationship to "data science"
- **ACM KDD: flagship conference**
  - ECML PKDD, SDM conferences

Causal and Interpretable Rules for Time Series Analysis

Amin Dhaou*†
TotalEnergies
Palaiseau, France
amin.dhaou@totalenergies.com

Antoine Bertoncello
TotalEnergies
Palaiseau, France
antoine.bertoncello@totalenergies.com

Sébastien Gourvénec
TotalEnergies
Palaiseau, France
sebastien.gourvenec@totalenergies.com

Josselin Garnier
CMAP, Ecole Polytechnique, Institut Polytechnique de Paris
Palaiseau, France
josselin.garnier@polytechnique.edu

Erwan Le Pennec
CMAP, Ecole Polytechnique, Institut Polytechnique de Paris
Palaiseau, France
erwan.le-pennec@polytechnique.edu

ABSTRACT

The number of complex infrastructures in an industrial setting is growing and is not immune to unexplained recurring events such as breakdowns or failure that can have an economic and environmental impact. To understand these phenomena, sensors have been placed on the different infrastructures to track, monitor, and control the dynamics of the systems. The causal study of these data allows predictive and prescriptive maintenance to be carried out. It helps to understand the appearance of a problem and find counterfactual outcomes to better operate and defuse the event.

In this paper, we introduce a novel approach combining the case-crossover design which is used to investigate acute triggers of diseases in epidemiology, and the Apriori algorithm which is a data mining technique allowing to find relevant rules in a dataset. The resulting time series causal algorithm extracts interesting rules in our application case which is a non-linear time series dataset. In addition, a predictive rule-based algorithm demonstrates the potential of the proposed method.

CCS CONCEPTS

• **Information systems** → **Association rules**; **Data mining**; • **Computing methodologies** → **Supervised learning by classification**; **Rule learning**; • **Applied computing** → **Command and control**.

KEYWORDS

Causality, Time Series, Data Mining, Case-Crossover design, Predictive maintenance

*Corresponding author
†Also with CMAP, Ecole Polytechnique, Institut Polytechnique de Paris.

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467161

ACM Reference Format:
Amin Dhaou, Antoine Bertoncello, Sébastien Gourvénec, Josselin Garnier, and Erwan Le Pennec. 2021. Causal and Interpretable Rules for Time Series Analysis. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467161

1 INTRODUCTION

Monitoring has enabled, with the help of increased storage capacity, to collect a large amount of data. The data analysis plays a crucial role in understanding the underlying mechanisms and the occurrence of incidents. In the industrial context, this consists of placing sensors and collecting temporal data like temperature, flow rates, chemical characteristics, or wind power to capture the evolution and the dynamics of the system. Exploiting these large amounts of temporal data is a real challenge facing many companies. Indeed, they contain enormous amounts of information that could help improve efficiency or optimize certain processes.

Driven by easy access to machine learning environments and the recent success of deep learning techniques, many models have been developed to predict the occurrence of these events but they do not only work on their causes but also on the correlated variables. This makes these models less robust as they could miss the incident by trusting a correlated variable. In areas where decisions and actions can have serious consequences, for example on humans in medicine or on the profitability in the industry, it is necessary to understand black-box models and therefore to carry out a causal study to act in a justified way. Hence, the objective of causality in an industrial context is to better understand the decisions taken by artificial intelligence algorithms, to find the causes of unexplained events, and to do maintenance policy by anticipating the occurrences of breakdowns. Therefore, a theoretical approach should be developed to provide a general framework that could work in an industrial environment. In particular, the approach should help the operators understand what are the mechanisms behind every decision that is taken and allow them to prevent the apparition of an incident by defusing its arrival.

The interest in causality is growing and these studies are becoming essential in industry and in many other fields of applications. For instance, it is common for distillation units to have recurrent problems occurring during petroleum refining. The causal study

KDD research today

Past: extracting patterns

Today

- Finding causes
- Producing explanations
- Design friction against automated decision making?

Friday

- Logistics: Everyone reviews, I present
  - Not representative of the papers throughout the semester
  - I will give two “sample” presentations with meta-commentary.
    - Examples of length, content, and structure
- Bids due Friday (for presentations)
- Wed. presenters assigned Fri. or Sat.