Using Data Mining Methods to Build Customer Profiles

G. Adomavicius & A. Tuzhilin. IEEE Computer, vol. 34 no. 2, 2001.

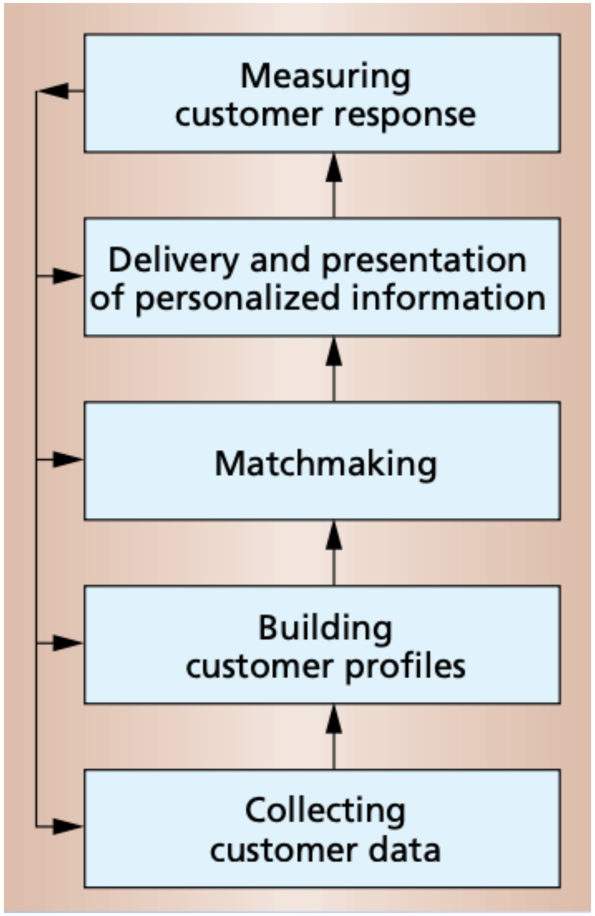
발표자 김형준

Personalization

- **Collecting Data**
  - Collecting customer data from various sources.
  - Collected Data must be prepared, cleaned and stored.

- **Customer Profiling**
  - Construct accurate customer profiles based on collected data

- **Matchmaking**
  - Match appropriate content and service.
    - Content-based : recommending similar items
    - Collaborative filtering : recommend items that similar customer preferred.

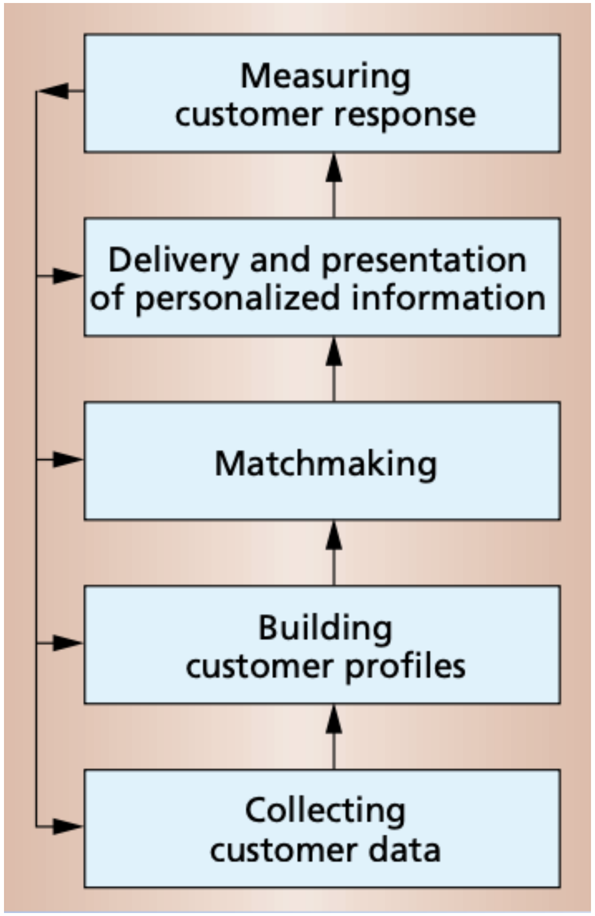


[Figure 1]

Personalization

- **Delivery and Presentation**
  - Pull, push, passive
    - Pull : display information if the customer explicitly requests
    - Push : reach to a customer who is not interacting with the system.
    - Passive : display personalized information in the context of applications.

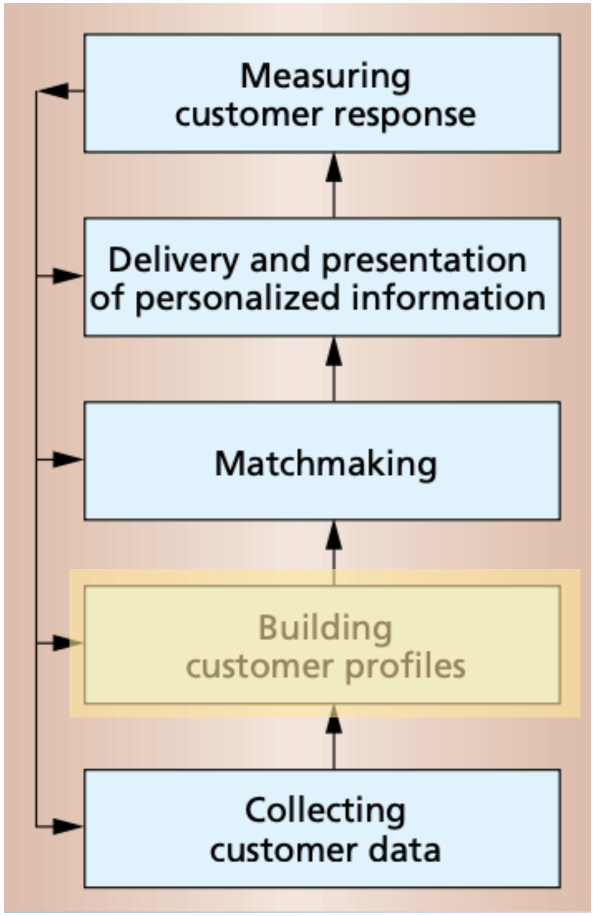
- **Measuring Customer Response**
  - Evaluate the effectiveness of personalization.
    - Time/money spent on the website.
    - Whether the service attracts new customers.
    - Whether the customer loyalty increases.
  - This process serves as feedback for possible improvements.
    - Whether to collect additional data
    - Whether to build better user profiles
    - Whether to build better matchmaking algorithms
  - Result in providing better understanding of customers, better recommendations and service



[Figure 1]

Personalization

- **Collecting Data**
  - Collecting customer data from various sources.
  - Collected Data must be prepared, cleaned and stored.
- **Customer Profiling < Our topic !**
  - Construct accurate customer profiles based on collected data
- **Matchmaking**
  - Match appropriate content and service.
    - Content-based : recommending similar items
    - Collaborative filtering : recommend items that similar customer preferred.



[Figure 1]

Motivation

Motivation

- **Personalization has become an important marketing tool.**
  - e.g. personalized web content presentations to book, CD, stock purchase, etc.

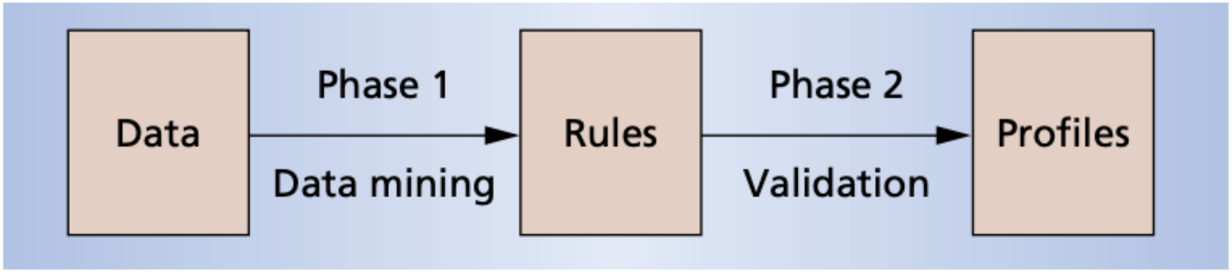
- **Some important issues of personalization :**
  - **how to extract this knowledge from the available data and store it in customer profiles.**
    - We will focus on this issue in this paper!
  - how to provide personal recommendations based on a comprehensive knowledge of who customers are, how they behave, and how similar they are to other customers

- **To deal with the first issue, this paper developed an approach using information learned from customers' transactional histories.**
  - With these information, we construct accurate, comprehensive **individual profiles**.
    - **Facts**
    - **Behavioral Rules** : describe customers' behaviors
      - Use data mining methods to derive behavioral rules.
      - Developed a method for **validating** customer profiles with help of human domain expert.

1:1 Pro System

1:1 Pro System (One-to-One Profiling system)

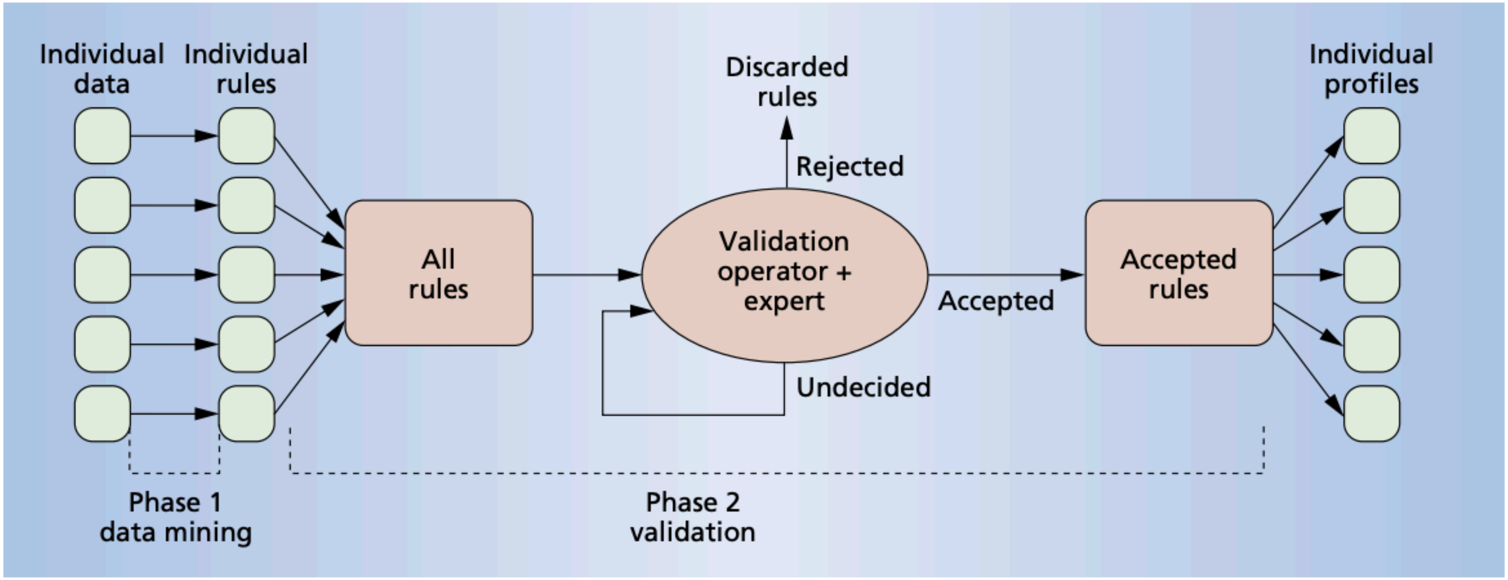
- **Building customer profiles**
  - Two main phases : **rule discovery, validation**



[Figure 2]

1:1 Pro System (One-to-One Profiling system)

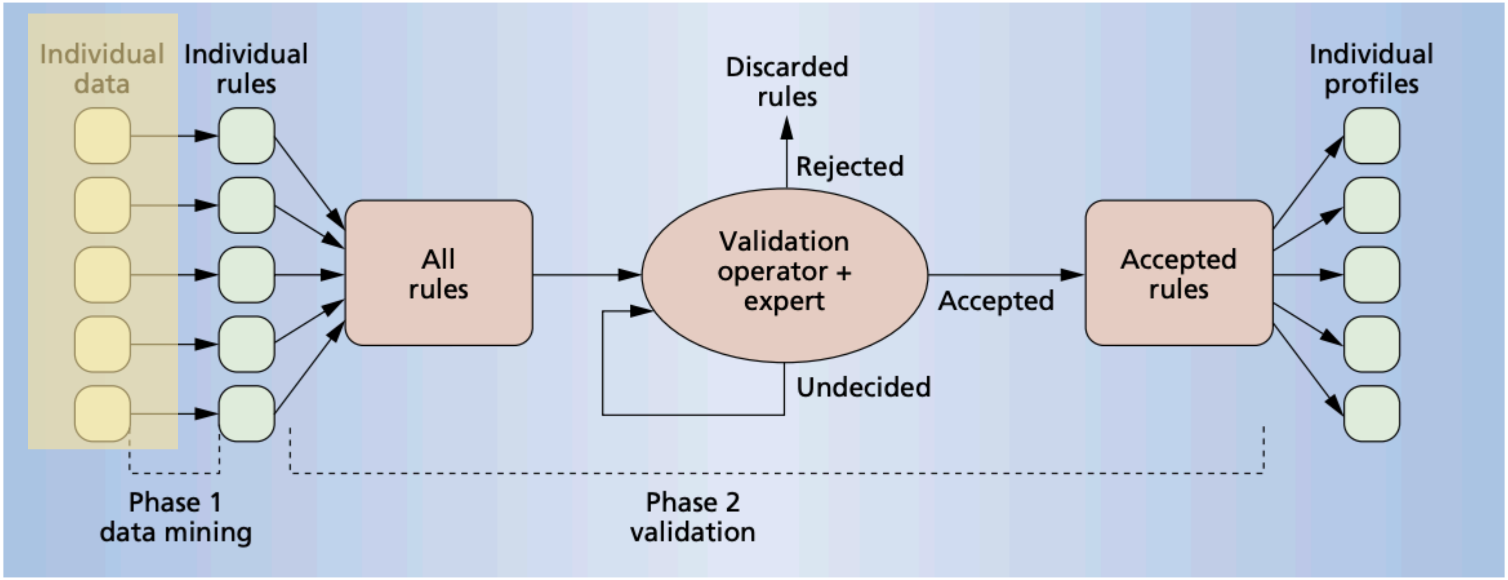
- **Building customer profiles**
  - Two main phases : **rule discovery, validation**



[Figure 3]

1:1 Pro System (One-to-One Profiling system)

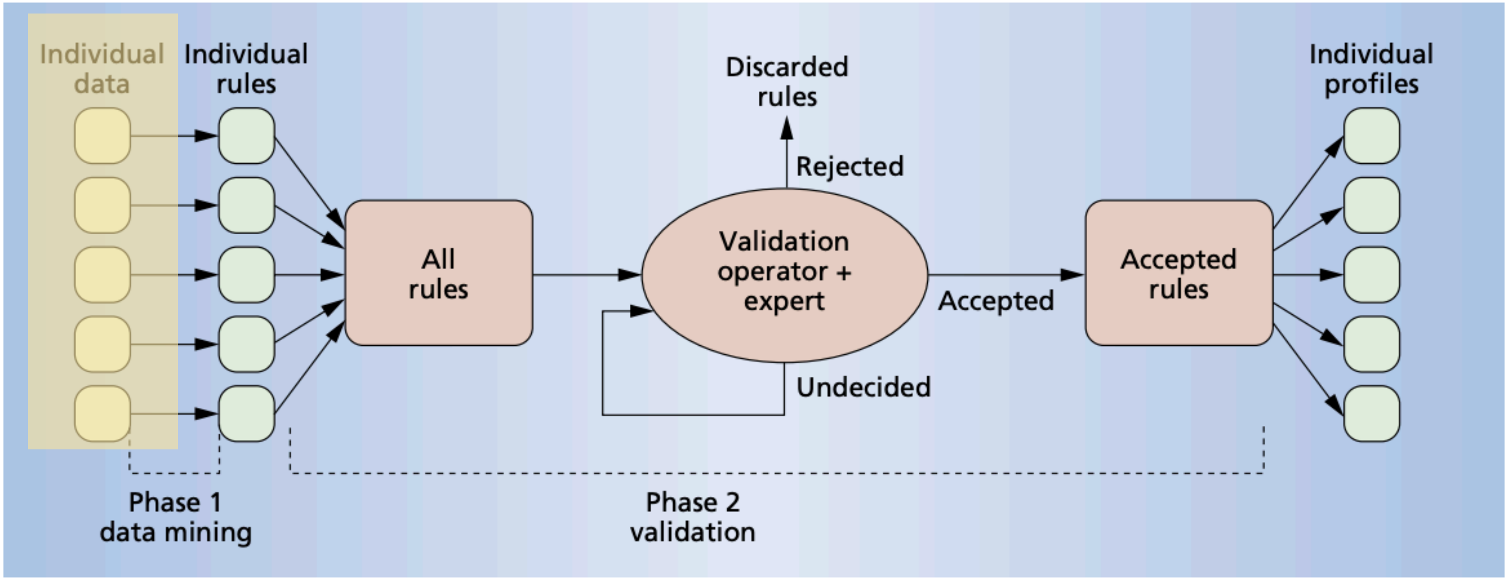
- **Building customer profiles**
  - Two main phases : **rule discovery, validation**
  - The whole process begins with **collecting the data.**



[Figure 3]

1:1 Pro System (One-to-One Profiling system)

- **Data Model**
  - **Factual** : who the customer is.
    - e.g. name, gender, birth date, address, etc.
  - **Transactional** : what the customer does.
    - e.g. purchase date, purchased product, paid amount, coupon use, etc.



[Figure 3]

1:1 Pro System (One-to-One Profiling system)

- **Data Model**
  - **Factual** : who the customer is.
    - e.g. name, gender, birth date, address, etc.
  - **Transactional** : what the customer does.
    - e.g. purchase date, purchased product, paid amount, coupon use, etc.

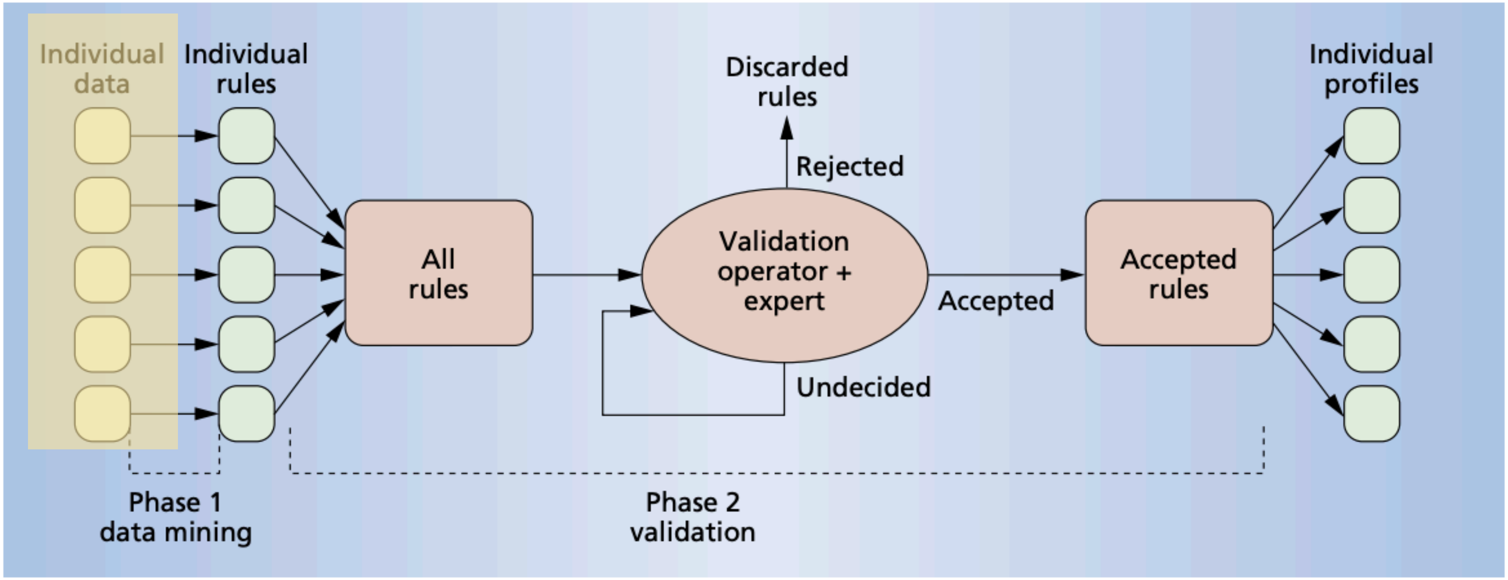
Factual	CustomerId	LastName	FirstName	BirthDate	Gender
	0721134	Doe	John	11/17/1945	Male
	0721168	Brown	Jane	05/20/1963	Female
	0730021	Adams	Robert	06/02/1959	Male

Transactional	CustomerId	Date	Time	Store	Product	CouponUsed
	0721134	07/09/1993	10:18am	GrandUnion	WheatBread	No
	0721134	07/09/1993	10:18am	GrandUnion	AppleJuice	Yes
	0721168	07/10/1993	10:29am	Edwards	SourCream	No
	0721134	07/10/1993	07:02pm	RiteAid	LemonJuice	No
	0730021	07/10/1993	08:34pm	Edwards	SkimMilk	No
	0730021	07/10/1993	08:34pm	Edwards	AppleJuice	No
	0721168	07/12/1993	01:13pm	GrandUnion	BabyDiapers	Yes
	0730021	07/12/1993	01:13pm	GrandUnion	WheatBread	No

[Figure 4]

1:1 Pro System (One-to-One Profiling system)

- **Customer profile : What we want to build!**
  - **Factual profile** : obtained from customer's factual data + some transactional data.
    - e.g. name, gender, etc. + customer's favorite beer is Heineken, customer's biggest purchase, etc.
  - **Behavioral profile(rules)** : customer's actions mostly derived from transactional data.
    - Use rules to describe customer behavior.
    - e.g. when purchasing cereal, the customer usually buys milk, etc.
  - Other profiling methods does not include behavioral profiles(rules)

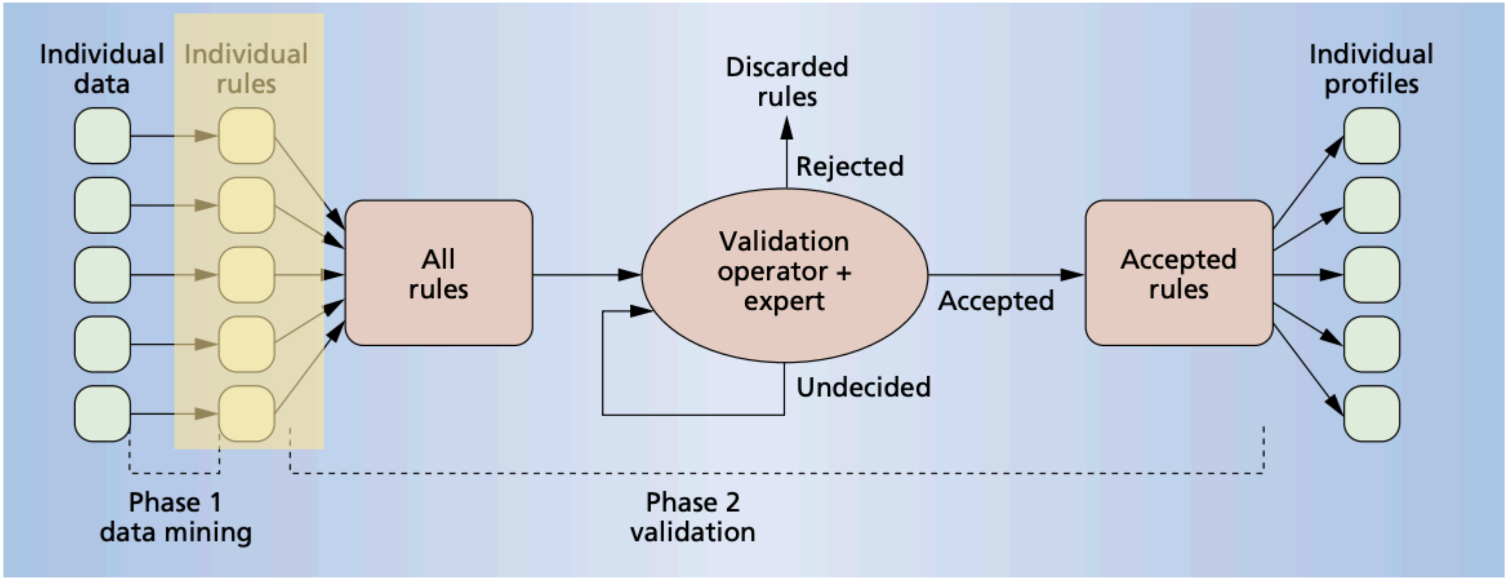


[Figure 3]

INTELLIGENT DATA SYSTEMS LABORATORY

1:1 Pro System (One-to-One Profiling system)

- **Rule discovery**
  - Apply **rule discovery methods** individually to every customer's data.
    - Rule discovery methods may vary
  - Methods work well for applications with **many transactions.**
    - Applications with less transactions, rules tend to be less reliable.
      - e.g., car purchase, vacation planning, etc.



[Figure 3]

1:1 Pro System (One-to-One Profiling system)

- **Rule discovery**
  - Example (Figure 5.)
    - 95 percent of the cases when he buys lemon juice, he buys it at RiteAid.
    - 2.4 percent of all John Doe's shopping transactions include purchasing lemon juice at RiteAid.

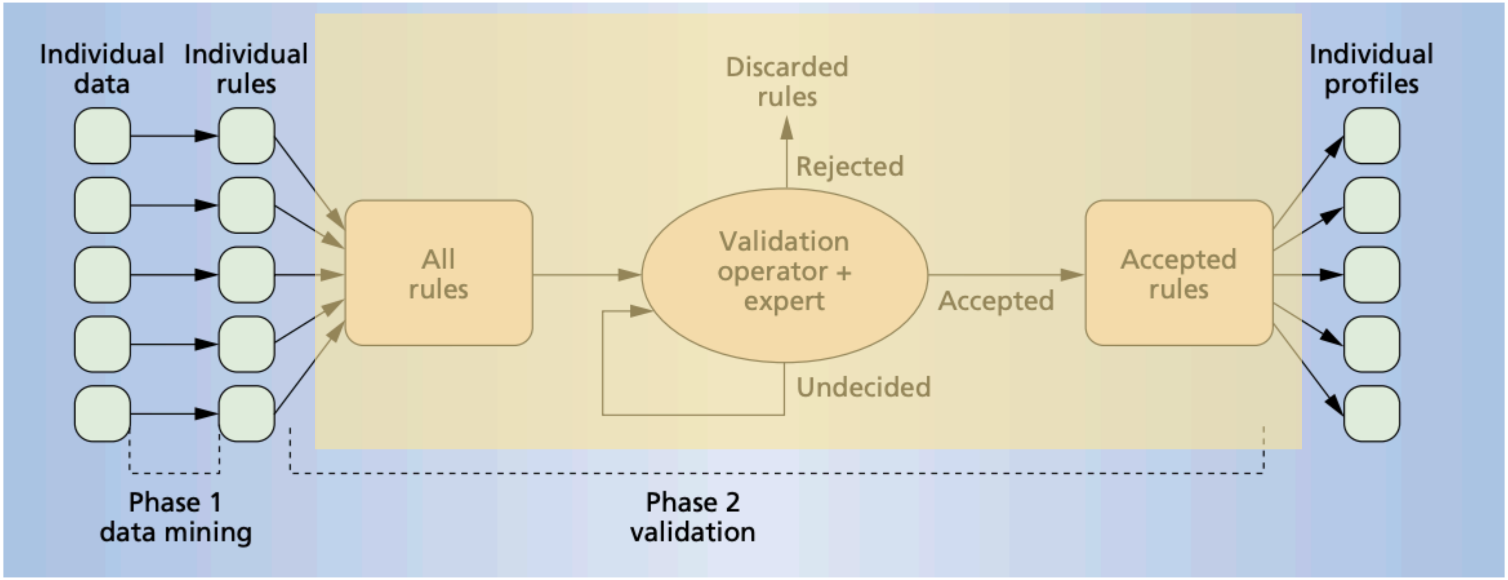
Discovered rules (for John Doe)

(1) Product = LemonJuice => Store = RiteAid (2.4%, 95%)
(2) Product = WheatBread => Store = GrandUnion (3%, 88%)
(3) Product = AppleJuice => CouponUsed = YES (2%, 60%)
(4) TimeOfDay = Morning => DayOfWeek = Saturday (4%, 77%)
(5) TimeOfWeek = Weekend & Product = OrangeJuice => Quantity = Big (2%, 75%)
(6) Product = BabyDiapers => DayOfWeek = Monday (0.8%, 61%)
(7) Product = BabyDiapers & CouponUsed = YES => Quantity = Big (2.5%, 67%)

[Figure 5]

1:1 Pro System (One-to-One Profiling system)

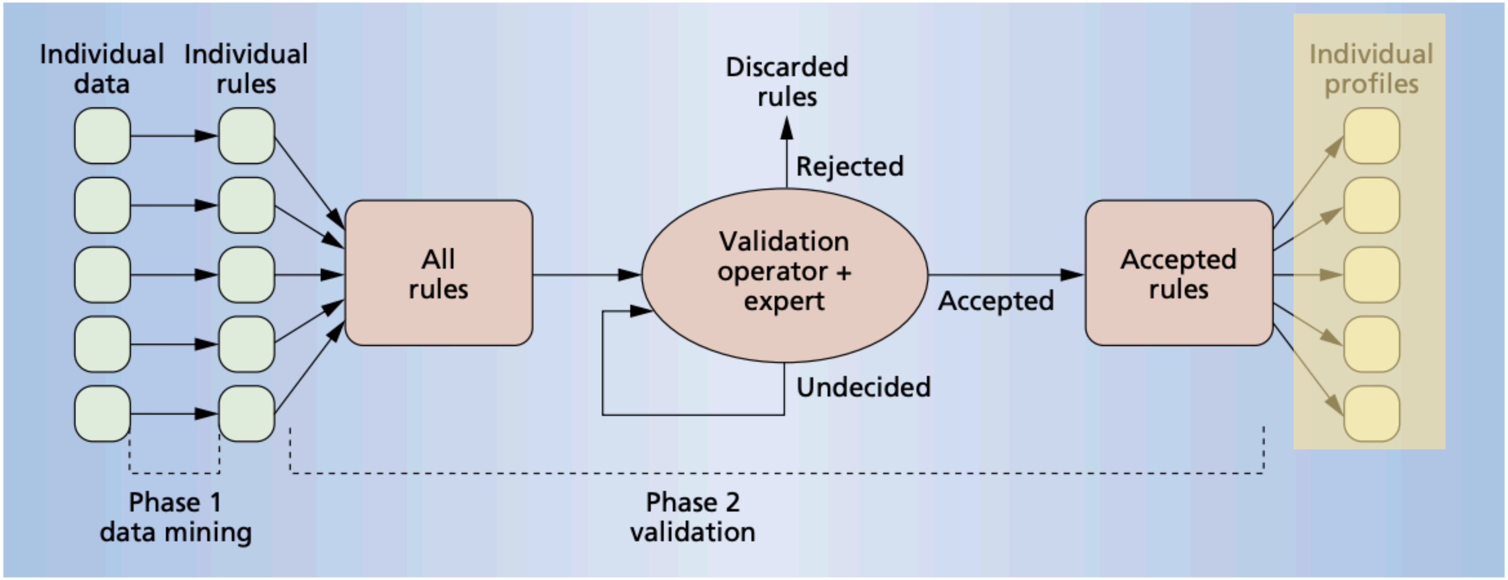
- **Rule validation**
  - Data mining methods generate large numbers of rules -> **validation** is necessary!
  - Domain expert inspect the rules -> Accept or reject rules
    - Accepted rules form the behavioral profiles.
  - **Scalability** is a big issue! It's impossible for the expert to validate all the rules one by one.
    - 1:1 Pro System uses **validation operators** that let a human expert validate large numbers of rules at a time with relatively **little input** from the expert.



[Figure 3]

1:1 Pro System (One-to-One Profiling system)

- **Profile building process**
  - After rule validation, the system places the accepted rules in the customer’s profile.



[Figure 3]

1:1 Pro System (One-to-One Profiling system)

- **Rule validation process (Notations)**
  - R_{all} : **all rules** discovered during Phase 1.
  - O_{acc} : **accepted** rules for a single validation operator.
  - O_{rej} : **rejected** rules for a single validation operator.
  - R_{unv} : remaining **unvalidated** rules.

Input: Set of all discovered rules R_{all} .
Output: Mutually disjoint sets of rules R_{acc} , R_{rej} , R_{unv} ,
such that $R_{all} = R_{acc} \cup R_{rej} \cup R_{unv}$.

(1) $R_{unv} := R_{all}$, $R_{acc} := \varnothing$, $R_{rej} := \varnothing$.
(2) **while** (**not** *TerminateValidationProcess*()) **begin**
(3) Expert picks a validation operator (say, O) from the set of available validation operators.
(4) O is applied to R_{unv} . Result: disjoint sets O_{acc} and O_{rej} .
(5) $R_{unv} := R_{unv} - O_{acc} - O_{rej}$, $R_{acc} := R_{acc} \cup O_{acc}$, $R_{rej} := R_{rej} \cup O_{rej}$.
(6) **end**

[Figure 6]

1:1 Pro System (One-to-One Profiling system)

- **Rule validation process**
  - All rules are considered unvalidated. ($R_{unv} = R_{all}$)
  - Expert chooses various ***validation operators*** and applies them to the unvalidated rule set.
    - For all the validated rules, some are **accepted(O_{acc})** and some are **rejected(O_{rej})**
  - The remaining unvalidated rules(R_{unv}) go through the same process .
  - The process stops then the ***TerminateValidationProcess*** condition is met.

Input: Set of all discovered rules R_{all} .
Output: Mutually disjoint sets of rules R_{acc} , R_{rej} , R_{unv} ,
such that $R_{all} = R_{acc} \cup R_{rej} \cup R_{unv}$.

(1) $R_{unv} := R_{all}$, $R_{acc} := \varnothing$, $R_{rej} := \varnothing$.
(2) **while** (**not** *TerminateValidationProcess()*) **begin**
(3) Expert picks a validation operator (say, O) from the set of available validation operators.
(4) O is applied to R_{unv} . Result: disjoint sets O_{acc} and O_{rej} .
(5) $R_{unv} := R_{unv} - O_{acc} - O_{rej}$, $R_{acc} := R_{acc} \cup O_{acc}$, $R_{rej} := R_{rej} \cup O_{rej}$.
(6) **end**

[Figure 6]

1:1 Pro System (One-to-One Profiling system)

- **Rule validation process**
  - Validation operators
    - **Similarity-based rule grouping**
      - Put similar rules into groups.
    - **Template-based rule filtering**
      - Filters rules that match expert-specified rule templates (accepting templates / rejecting templates)
    - **Redundant-rule elimination**
      - Eliminates rules that can be derived from other facts = rules that by themselves carry no information.

Input: Set of all discovered rules R_{all} .
Output: Mutually disjoint sets of rules R_{acc} , R_{rej} , R_{unv} ,
such that $R_{all} = R_{acc} \cup R_{rej} \cup R_{unv}$.

(1) $R_{unv} := R_{all}$, $R_{acc} := \varnothing$, $R_{rej} := \varnothing$.
(2) **while** (**not** *TerminateValidationProcess()*) **begin**
(3) Expert picks a validation operator (say, O) from the set of available validation operators.
(4) O is applied to R_{unv} . Result: disjoint sets O_{acc} and O_{rej} .
(5) $R_{unv} := R_{unv} - O_{acc} - O_{rej}$, $R_{acc} := R_{acc} \cup O_{acc}$, $R_{rej} := R_{rej} \cup O_{rej}$.
(6) **end**

[Figure 6]

1:1 Pro System (One-to-One Profiling system)

- **Validation operator Examples**
  - **Similarity-based rule grouping**
    - Put similar rules into groups.

    - Put rules 1 and 2 in the same group.
      - Same structure : ***Product => Store***
      - Rule 3 would not be grouped together because the structure is : ***Product => CouponUsed***

Discovered rules (for John Doe)

(1) Product = LemonJuice => Store = RiteAid (2.4%, 95%)
(2) Product = WheatBread => Store = GrandUnion (3%, 88%) < RULE GROUPED !
(3) Product = AppleJuice => CouponUsed = YES (2%, 60%)
(4) TimeOfDay = Morning => DayOfWeek = Saturday (4%, 77%)
(5) TimeOfWeek = Weekend & Product = OrangeJuice => Quantity = Big (2%, 75%)
(6) Product = BabyDiapers => DayOfWeek = Monday (0.8%, 61%)
(7) Product = BabyDiapers & CouponUsed = YES => Quantity = Big (2.5%, 67%)

[Figure 4]

1:1 Pro System (One-to-One Profiling system)

- **Validation operator Examples**
  - **Template-based rule filtering**
    - Filters rules that match expert-specified rule templates (accepting templates / rejecting templates)
    - For a rule template : ***REJECT HEAD = {Store = RiteAid}***
      - Reject all rules that have ***Store = RiteAid*** in their heads.

[Figure 4]

1:1 Pro System (One-to-One Profiling system)

- **Validation operator Examples**
  - **Redundant-rule elimination**
    - Eliminates rules that can be derived from other facts = rules that by themselves carry no information.

    - For a rule : ***Product = AppleJuice => Store = GrandUnion(2%, 100%)***
      - Buys apple juice only at Grand Union. -> is this a meaningful rule?
      - If the customer shops exclusively at Grand Union, this rule becomes meaningless.

1:1 Pro System (One-to-One Profiling system)

- **Rule validation process**
  - Other validation operators
    - Visualization
      - view subsets of unvalidated rules in visual representations such as histograms and pie charts.
    - Statistical analysis
      - computes various statistical characteristics of unvalidated rules
    - Browsing
      - inspect individual rules or groups of rules directly by viewing them on the screen.

Input: Set of all discovered rules R_{all} .
Output: Mutually disjoint sets of rules R_{acc} , R_{rej} , R_{unv} , such that $R_{all} = R_{acc} \cup R_{rej} \cup R_{unv}$.

(1) $R_{unv} := R_{all}$, $R_{acc} := \varnothing$, $R_{rej} := \varnothing$.
(2) **while** (**not** *TerminateValidationProcess()*) **begin**
(3) Expert picks a validation operator (say, O) from the set of available validation operators.
(4) O is applied to R_{unv} . Result: disjoint sets O_{acc} and O_{rej} .
(5) $R_{unv} := R_{unv} - O_{acc} - O_{rej}$, $R_{acc} := R_{acc} \cup O_{acc}$, $R_{rej} := R_{rej} \cup O_{rej}$.
(6) **end**

[Figure 6]

1:1 Pro System (One-to-One Profiling system)

- **Rule validation process**
  - ***TerminateValidationProcess()***
    - Experts can specify the criterion in several ways.

    - Validation continues until some **predetermined percentage** of rules is validated.
      - e.g., 95% of the rules.
    - Validation terminates when the validation operators validate only a **few rules** at a time.

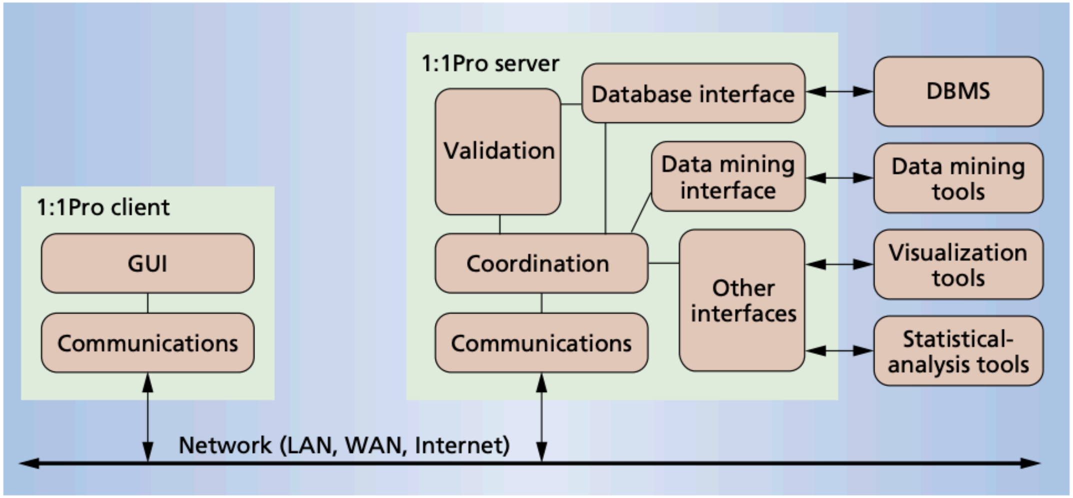
Input: Set of all discovered rules R_{all} .
Output: Mutually disjoint sets of rules R_{acc} , R_{rej} , R_{unv} ,
such that $R_{all} = R_{acc} \cup R_{rej} \cup R_{unv}$.

$R_{unv} := R_{all}$, $R_{acc} := \varnothing$, $R_{rej} := \varnothing$.
while (**not** *TerminateValidationProcess()*) **begin**
    Expert picks a validation operator (say, O) from the set of available validation operators.
    O is applied to R_{unv} . Result: disjoint sets O_{acc} and O_{rej} .
    $R_{unv} := R_{unv} - O_{acc} - O_{rej}$, $R_{acc} := R_{acc} \cup O_{acc}$, $R_{rej} := R_{rej} \cup O_{rej}$.
end

[Figure 6]

1:1 Pro System (One-to-One Profiling system)

- **1:1 Pro System architecture**
  - Takes **factual** data and **transactional** data as input.
  - Follows the **client-server model**.

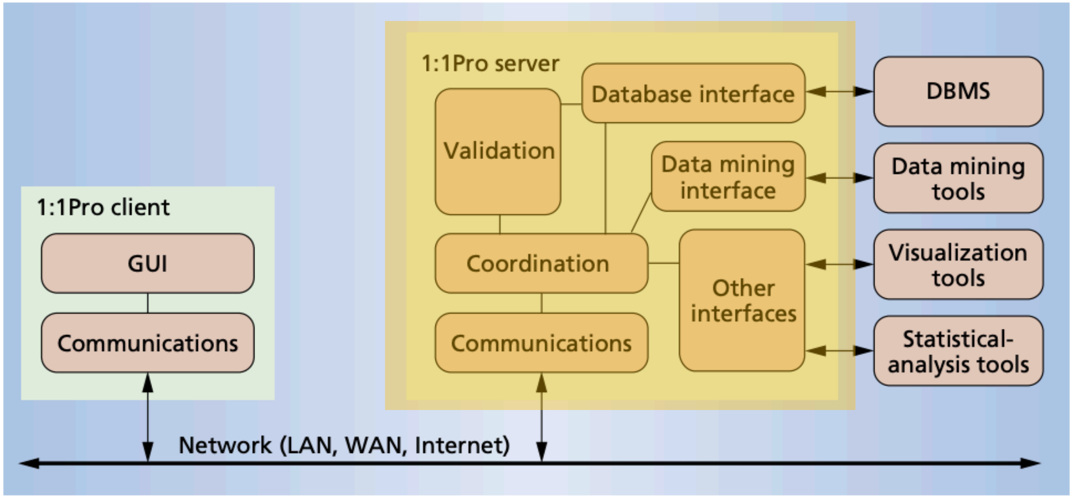


[Figure 7]

INTELLIGENT DATA SYSTEMS LABORATORY

1:1 Pro System (One-to-One Profiling system)

- **1:1 Pro System architecture**
  - **Server** component
    - Coordination module : **coordinates** profile construction (rule generation, validation process)
    - Validation module : **validates** the rules.
      - Supports similarity-based grouping, template-based filtering, redundant-rule elimination, and browsing operators.
    - Communications module : handles all communications with the client component.
    - Separate interface to external modules
      - e.g., DBMS, data mining tools, and visualization tools

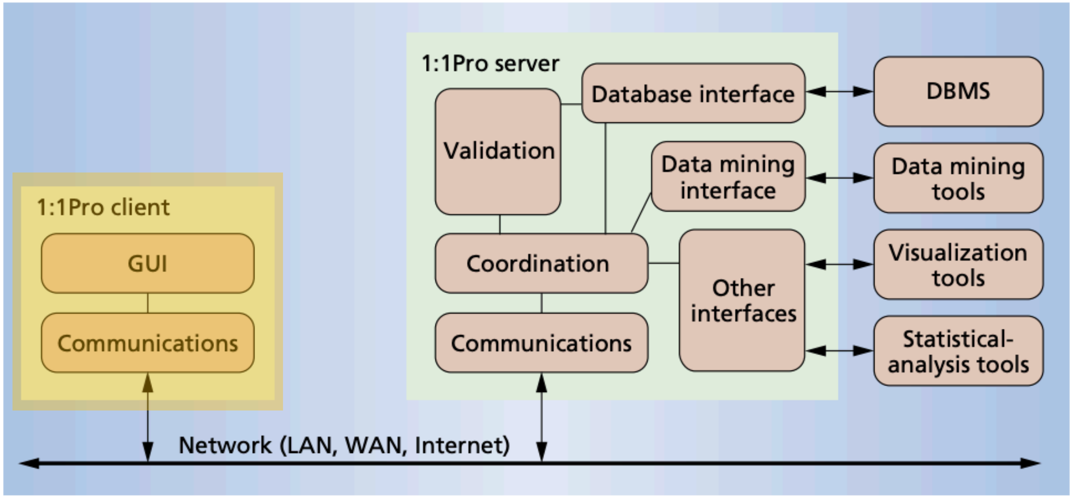


[Figure 6]

INTELLIGENT DATA SYSTEMS LABORATORY

1:1 Pro System (One-to-One Profiling system)

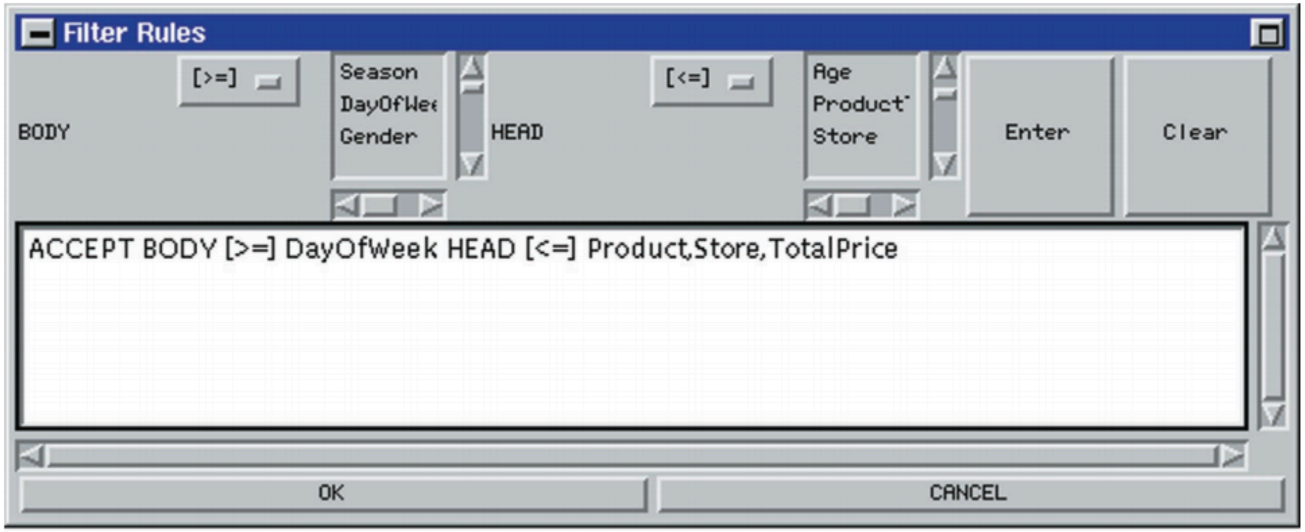
- **1:1 Pro System architecture**
  - **Client** component
    - Graphical user interface (GUI)
      - Specify validation operations and view the results of the iterative validation process.
    - Communications modules
      - Sends the expert-specified validation request to the server.
      - Server receives validation operators and passes to the **coordination** module
      - The coordination module passes to the **validation** component for processing



[Figure 7]

1:1 Pro System (One-to-One Profiling system)

- **1:1 Pro System architecture**
  - **Client** component
    - Graphical user interface (GUI)
      - Specify validation operations and view the results of the iterative validation process.
    - Communications modules
      - Sends the expert-specified validation request to the server.
      - Server receives validation operators and passes to the **coordination** module
      - The coordination module passes to the **validation** component for processing



[Figure 8]

INTELLIGENT DATA SYSTEMS LABORATORY

1:1 Pro System (One-to-One Profiling system)

- **1:1 Pro System architecture**
  - Log files
    - Log file records the **entire validation process**.
    - ResultId : the instance of the validation operator used.
    - Operator : operator's type (grouping, browsing, filtering)
    - SourceId : the instance of the previously applied validation operator
    - Date/Time : time stamp
    - Notes : expert's comments.

ResultId	Operator	SourceId	Date/Time	Notes
...	...	...	...	...
6	Filter	5	11/23/1998 5:26pm	Rejecting: demogr. in the body
7	Group	3	11/23/1998 5:37pm	Used attribute-level setting here
8	Browse	7	11/23/1998 5:51pm	Accepted: 7 groups, rejected: 11
9	Filter	3	11/23/1998 6:28pm	Rejecting: 'age' in the head
...	...	...	...	...

[Figure 9]

Experiments

Experiments

- **Experiment 1**
  - Seasonality analysis : Finding rules describing **season-related** customer behaviors.
  - Settings :
    - 1,903 households purchasing various nonalcoholic **beverages** over **a one-year period**.
    - **21 fields** characterizing the purchase transactions and **353,421 records**
      - 186 records per household
    - Validated rules themselves.
  - Results :
    - Data mining module generated 1,022,812 association rules
      - About 537 rules per household.
    - Many rules capture **specific behavior** of individual households
      - most rules represent a **very small number** of households.
      - 40 percent of the 407,716 discovered rules are about 5 or fewer of the 1,903 households.
      - Of that 40 percent, nearly half apply to only **1 household.**
    - **One hour** to perform the whole process
    - Validated 97.2 percent of the rules : 4.0 percent accepted / 93.2 percent rejected
      - reduced the average customer profile size from **537** unvalidated rules to **21 accepted rules.**

Experiments

- **Experiment 1**

Table 1. A validation process for the seasonality analysis of market research data.

Validation operator	Accepted rules	Rejected rules	Unvalidated rules
Redundancy elimination	0	186,727	836,085
Filtering	0	285,528	550,557
Filtering	0	424,214	126,343
Filtering	0	48,682	77,661
Filtering	10,052	0	67,609
Grouping (652 groups)	23,417	6,822	37,370
Grouping (4,765 groups)	7,181	1,533	28,656
Total	40,650	953,506	1,724,281

[Figure 10]

Experiments

- **Experiment 2**
  - Seasonality analysis : Finding rules describing **season-related** customer behaviors.
  - Settings :
    - Help of a marketing **expert**
    - Apply **redundant-rule** elimination
    - Apply several **template-based** filtering rejection operators
      - e.g., reject all rules not referring to the ***Season*** or ***DayOfWeek*** attributes
    - grouped the remaining unvalidated rules
  - Results :
    - Accepted 42,496 rules.
      - 4.2 percent of all discovered rules
    - About 40 minutes on the entire process

Problems

- **Problems of 1:1 Pro**
  - The rule evaluation process is **subjective**
    - Different experts can arrive at **different evaluation results** using the same validation process.
  - Problem of generating many **irrelevant rules**.
    - Mostly are rejected during the validation process.
    - Solution : specify **constraints** on the types of rules.
  - Ideal solution is to combine the **constraint specification**, **data mining**, and rule **validation** stages in one system.
    - Currently working on integrating the stages.

Thank you