داده کاوی
data Mining

دانشگاه آزاد اسلامی سیرجان

نیم سال اول ۹۵-۹۴

معرفی درس

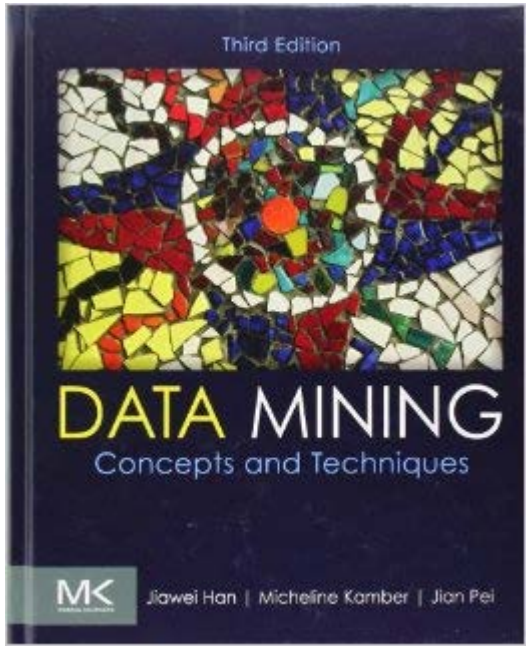
shmdi56@gmail.com

- مراحل درس
  1. مقدمه و مفاهیم اولیه
  2. استخراج الگوهای پرتکرار و قوانین انجمنی
  3. طبقه بندی داده ها
  4. خوشه بندی داده ها
  5. مباحث کاربردی

منبع درس :

Data Mining: Concepts and Techniques, Third Edition

Authors: **Jiawei Han**, **Micheline Kamber**
Publisher: **Morgan Kaufmann Publishers**



ارزشیابی درس:

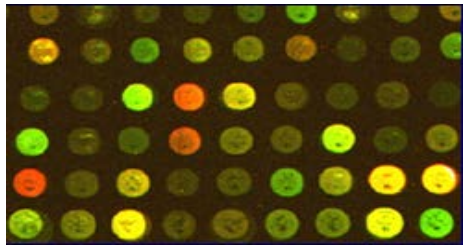
۱- امتحان پایان ترم ۸۰٪ نمره

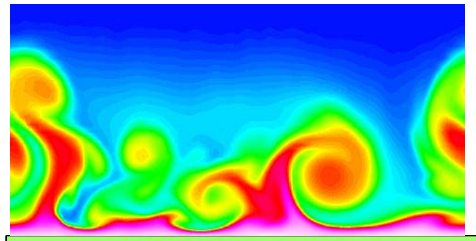
۲- ارائه شفاهی موضوعات مرتبط با

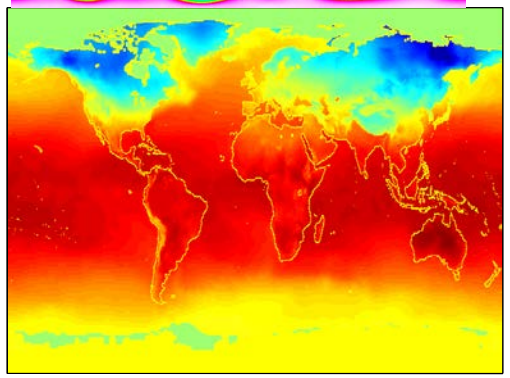
چرا داده کاوی؟ از دیدگاه تجاری

- مقدار زیادی داده در حال جمع آوری و انباشت هست
  - داده های اینترنتی
  - داده های تجارت الکترونیک
  - داده های خرید از فروشگاهها
  - داده های ترا کنش های بانکی و کارت های اعتباری
- کامپیوتر ها قوی تر و ارزان تر شده اند
- رقابت های تجاری سخت تر شده اند
  - ارائه خدمات بهتر به مشتریان و جلب نظر مشتریان (مشتری مداری)

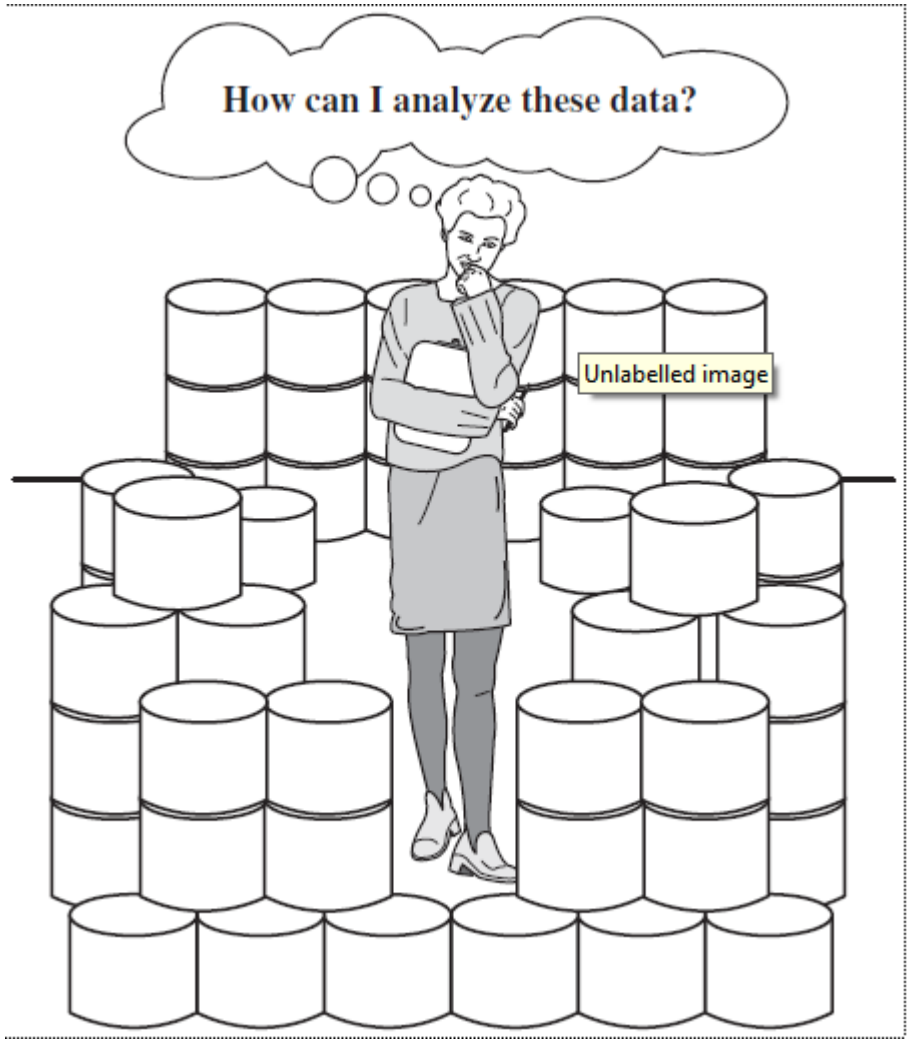
چرا داده کاوی؟ از دیدگاه علمی







- داده ها با سرعت خیلی زیاد جمع آوری و ذخیره می شوند (گیگا بایت در ثانیه)
  - حسگر های موجود در ماهواره ها
  - تلسکوپ های فضایی
  - داده های خرید ثبت شده در فروشگاهها
  - داده های ترا کنش های بانکی و کارت های اعتباری
- تکنیکهای قدیمی برای این داده ها قابل استفاده نیستند
- داده کاوی به دانشمندان امکان میدهد که:
  - داده ها را گروه بندی و دسته بندی کنند.
  - فرضیه های جدیدی را شکل دهند

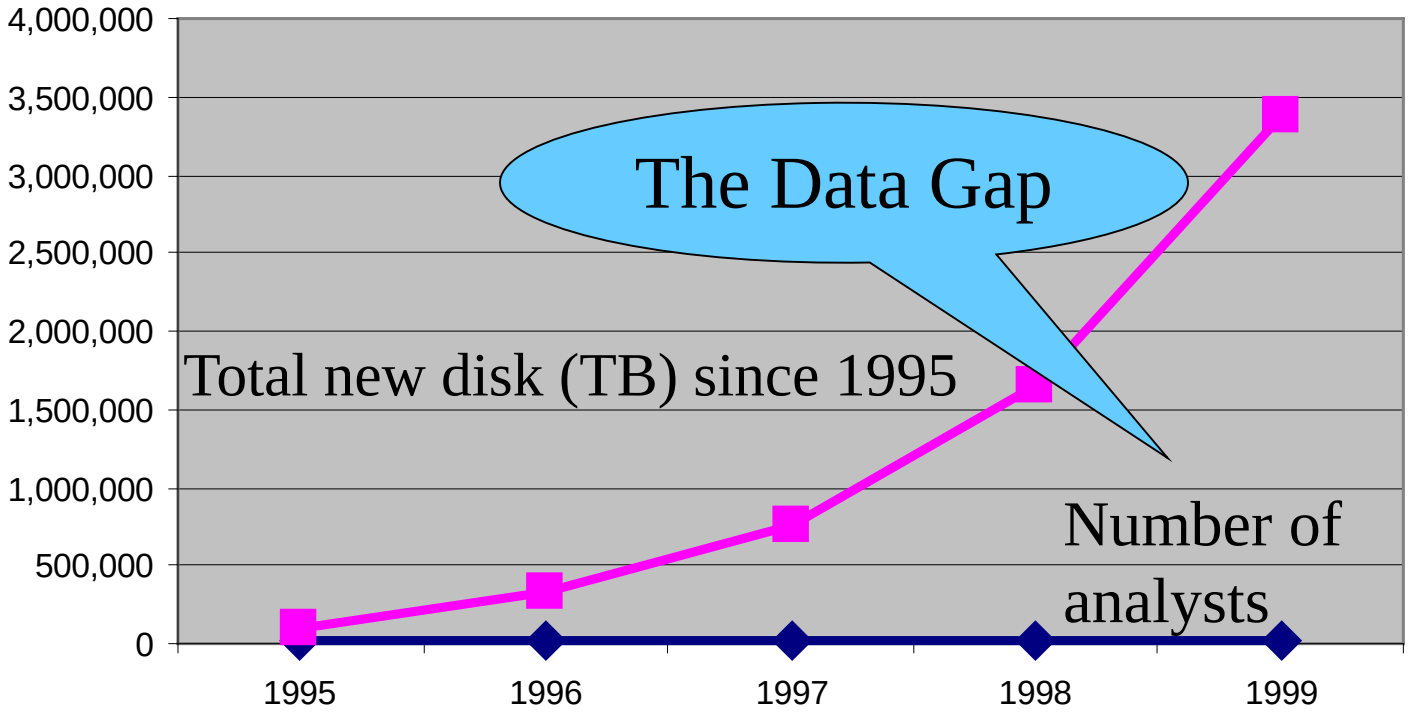


The world is data rich but information poor

کاوش مجموعه داده های بزرگ:

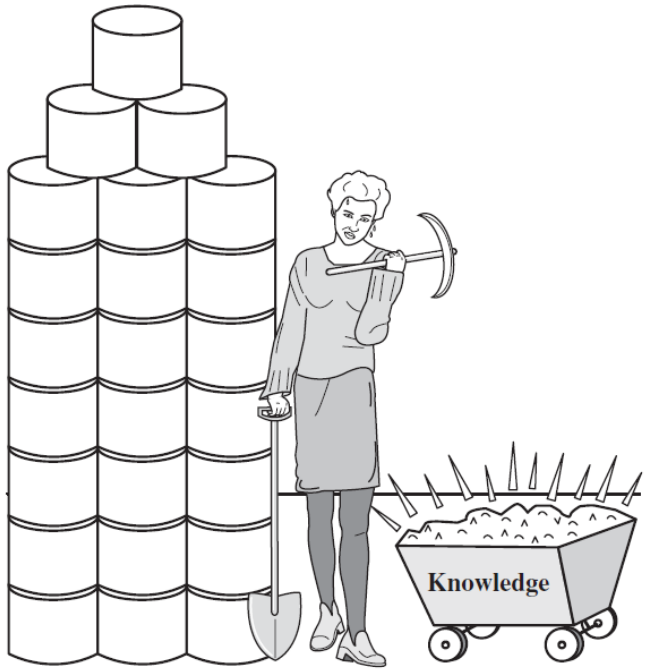
انگیزه

- معمولا اطلاعات نهفته ای در داده ها وجود دارد که تا کنون آشکار نشده است.
- برای کشف اطلاعات مفید توسط انسانها هفته ها زمان نیاز هست.
- خیلی از داده ها هنوز تحلیل نشده اند

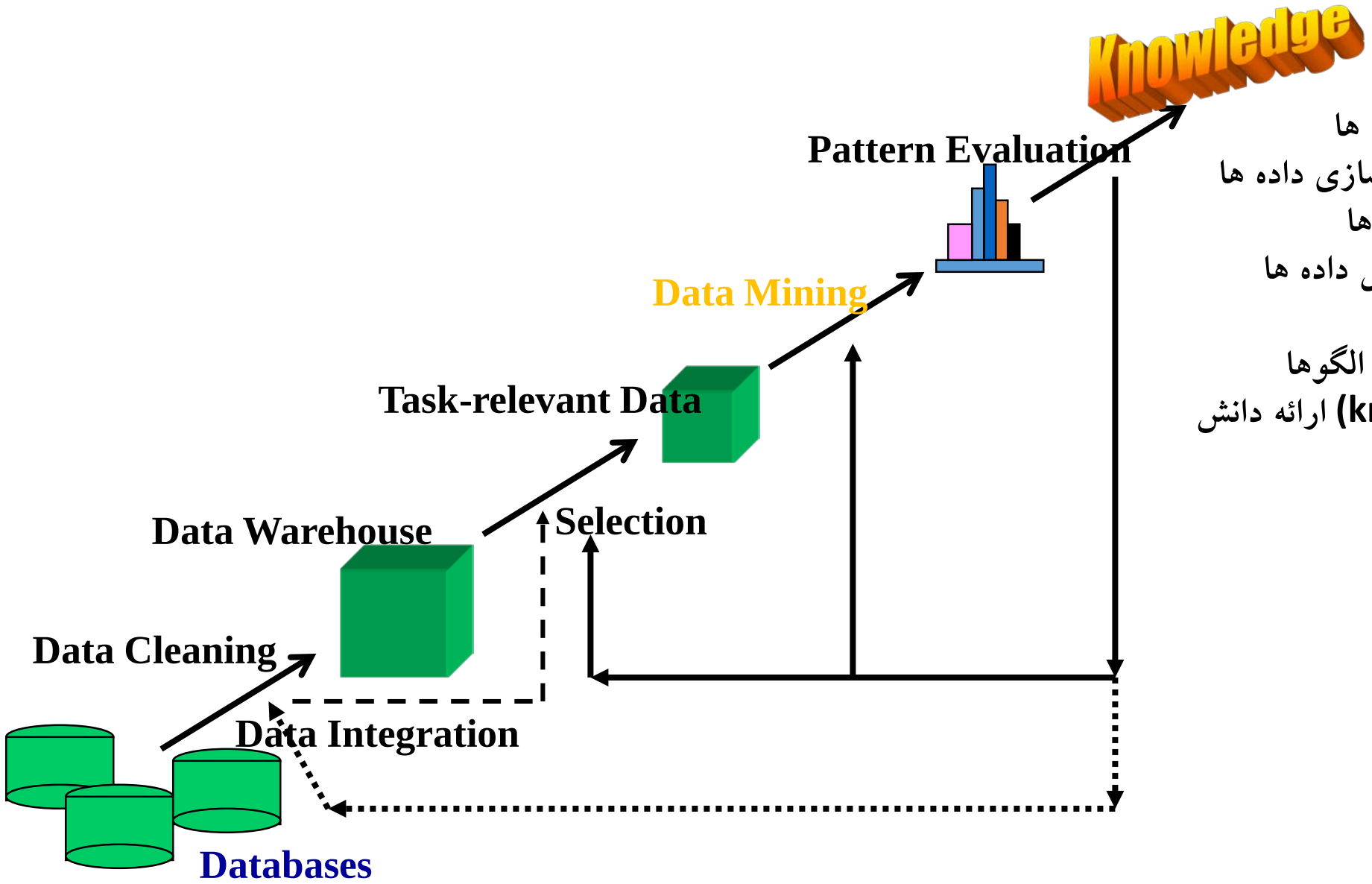


داده کاوی چیست؟

- تعریف اول: استخراج اطلاعات مفید و ناشناخته از داده ها
- تعریف دوم: آنالیز و اکتشاف از داده های با حجم زیاد، بوسیله ابزارهای خودکار و نیمه خودکار، به منظور کشف الگوهای معنادار

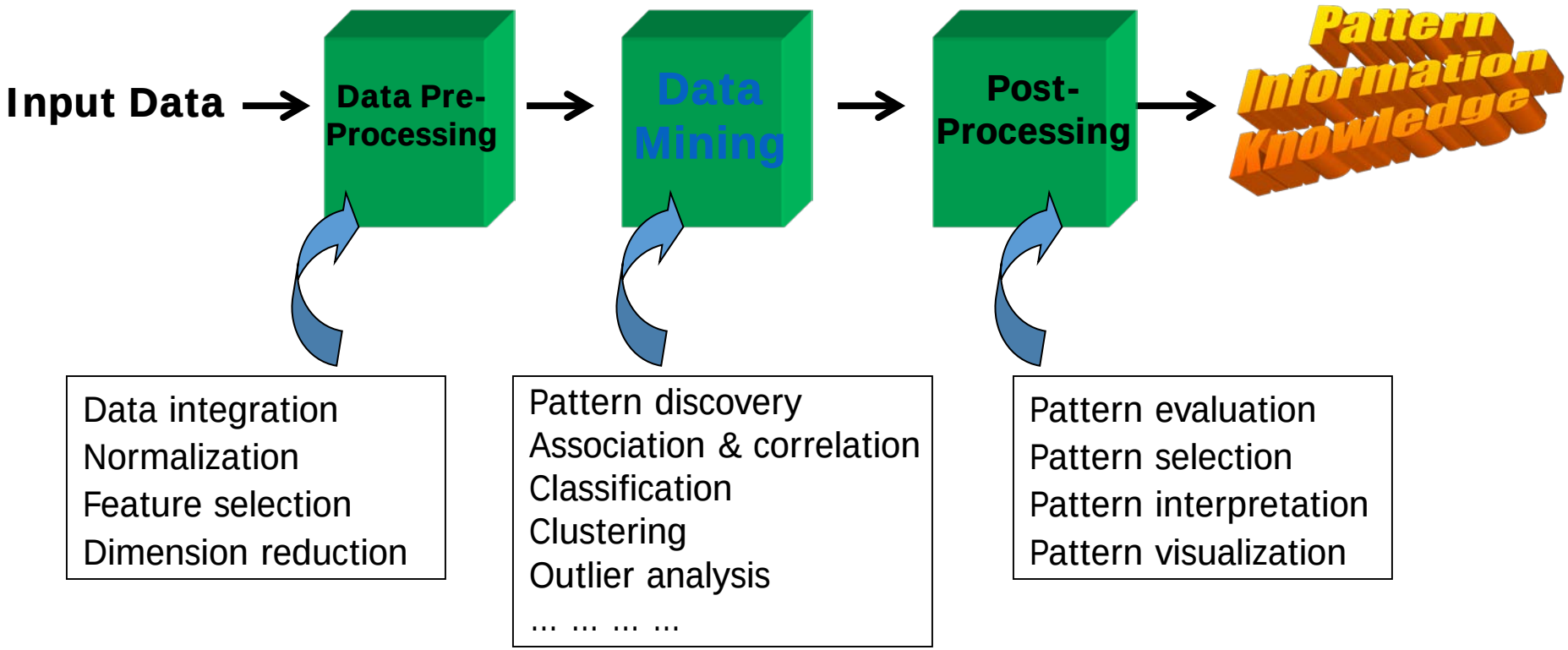


مراحل اکتشاف دانش



.X (Data Cleaning) پاکسازی داده ها
.Ц (Data Integration) یکپارچه سازی داده ها
.Ц (Data Selection) انتخاب داده ها
.Љ (Data Transformation) تبدیل داده ها
.Њ (Data Mining) داده کاوی
.a (Pattern Evaluation) ارزیابی الگوها
.Ъ (knowledge Representation) ارائه دانش

فرایند اکتشاف دانش: از دیدگاه یادگیری ماشین و علم آمار



- This is a view from typical machine learning and statistics communities

Databases

چه نوع داده هایی می تواند مورد کاوش قرار گیرد؟

Database-oriented data sets and applications

- Relational database, data warehouse, transactional database

Advanced data sets and advanced applications

- Data streams and sensor data
- Time-series data, sequence data (incl. bio-sequences)
- Structure data, graphs, social networks and multi-linked data
- Object-relational databases
- Multimedia database
- Text databases
- The World-Wide Web

چه الگوهایی می تواند مورد کاوش قرار گیرد؟ (تحلیل وابستگی و همبستگی)

Frequent patterns (or frequent itemsets)

- What items are frequently purchased together in your Walmart?

- Association, correlation vs. causality
  - A typical association rule
    - Bread→ Milk [0.5%, 75%] (support, confidence)
  - Are strongly associated items also strongly correlated?
- How to mine such patterns and rules efficiently in large datasets?
- How to use such patterns for classification, clustering, and other applications?

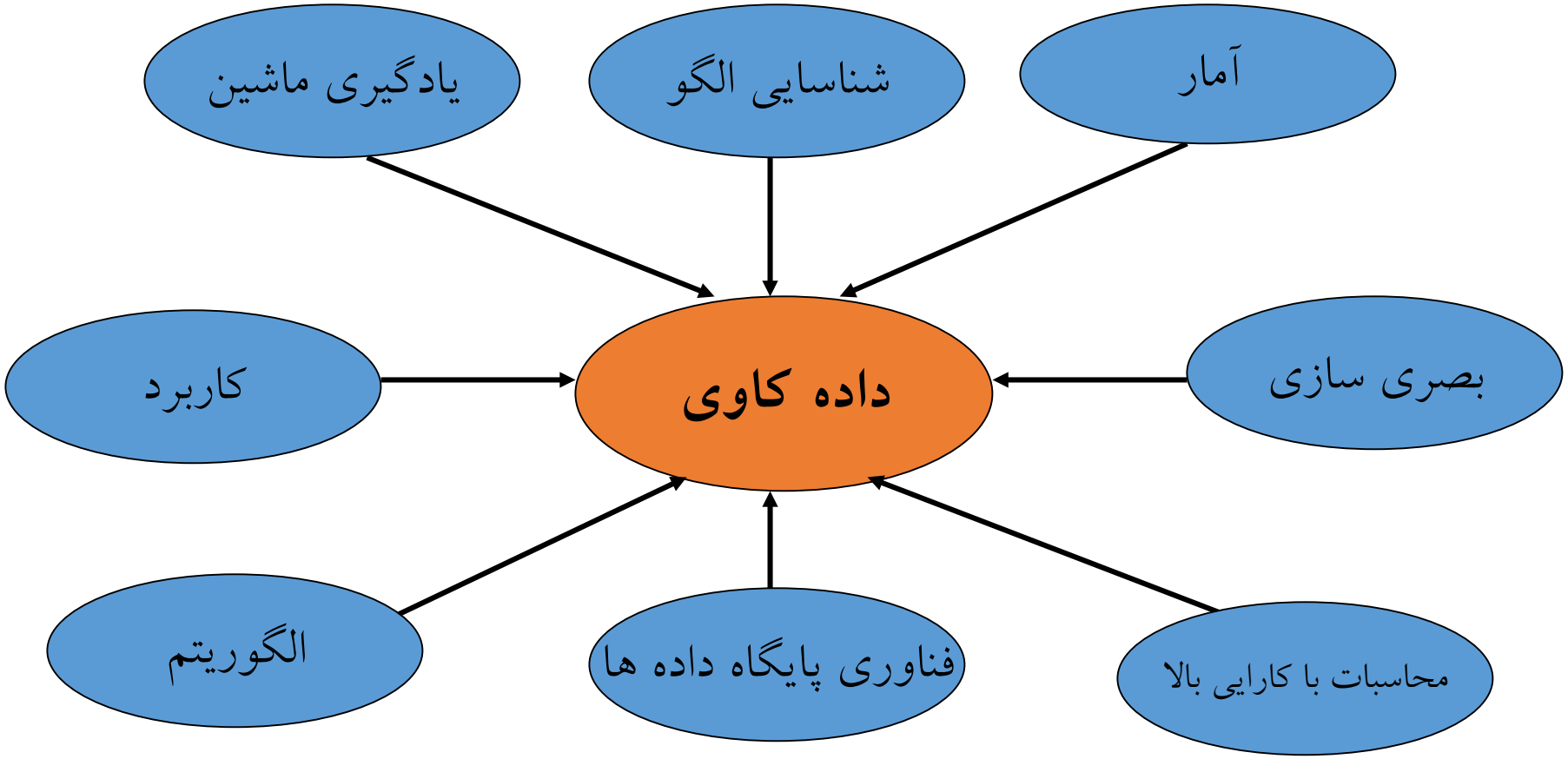
چه الگوهایی می تواند مورد کاوش قرار گیرد؟ (طبقه بندی)

- Classification and label prediction
  - Construct models (functions) based on some training examples
  - Describe and distinguish classes or concepts for future prediction
    - E.g., classify countries based on (climate), or classify cars based on (gas mileage)
  - Predict some unknown class labels
- Typical methods
  - Decision trees, naïve Bayesian classification, support vector machines, neural networks, rule-based classification, pattern-based classification, logistic regression, …
- Typical applications:
  - Credit card fraud detection, classifying stars, diseases, web-pages, …

ه الگوهایی می تواند مورد کاوش قرار گیرد؟ (خوشه بندی)

- Unsupervised learning (i.e., Class label is unknown)
- Group data to form new categories (i.e., clusters), e.g., cluster houses to find distribution patterns
- Principle: Maximizing intra-class similarity & minimizing interclass similarity
- Many methods and applications

داده کاوی: تلاقی علوم مختلف



چرا نیاز به علوم مختلف داریم

- حجم زیاد داده ها
  - الگوریتمها باید قابلیت کار با حجم زیادی از داده ها را داشته باشند
- داده ها دارای ابعاد زیادی هستند
  - مثلا میکرو آرایه ها دارای ده ها هزار ویژگی هستند
- داده ها دارای پیچیدگی زیادی هستند
  - مثل داده های جریانی و داده های حس گر ها
  - داده های دنباله زمانی و داده های ترتیبی
  - داده های ساختاری و گرافی
  - داده های چند رسانه ای، متنی و داده های وب
- کاربردهای جدید و پیچیده

کاربردهای داده کاوی

- Web page analysis: from web page classification, clustering to PageRank & HITS algorithms
- Collaborative analysis & recommender systems
- Basket data analysis to targeted marketing
- Biological and medical data analysis: classification, cluster analysis (microarray data analysis), biological sequence analysis, biological network analysis

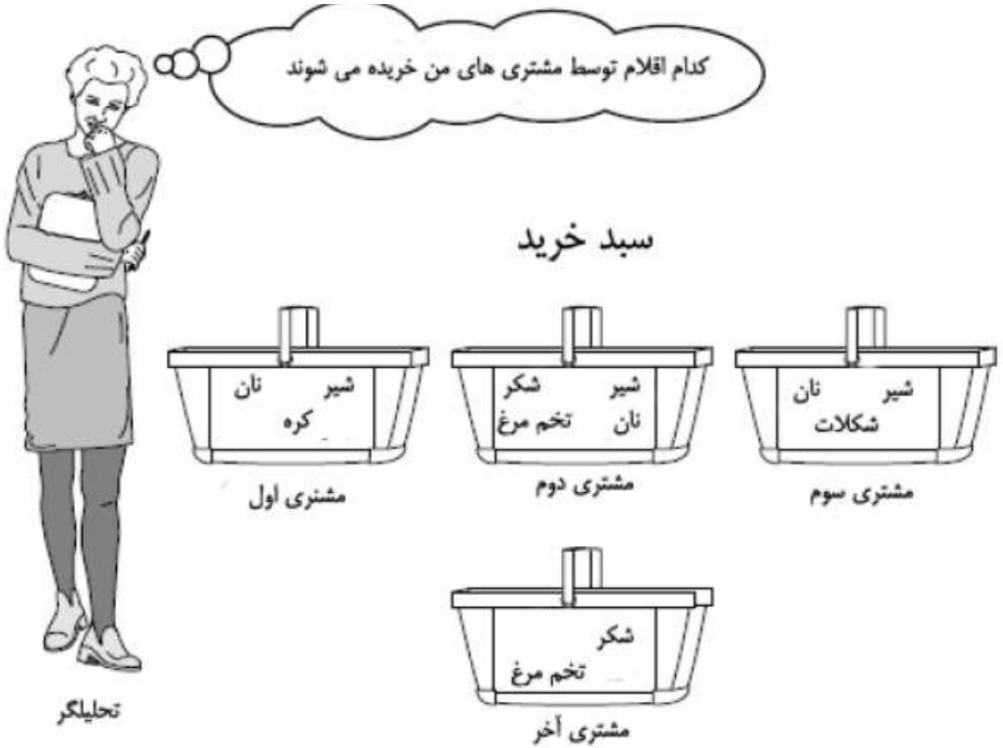
کاوش الگوهای پر تکرار، همبستگی و وابستگی:
مفاهیم اولیه و روشها

Mining frequent patterns; Association and Correlation:
Basic concepts and methods

تحلیل الگو های پر تکرار چیست؟

- **الگوی پر تکرار:** یک الگو (مجموعه ای از آیتم ها، زیر دنباله ها، ساختارها و ...) که مکرراً در مجموعه داده ها تکرار می شوند.
- **انگیزه:** یافتن نظم ذاتی در داده ها
  - چه محصولاتی معمولا با هم خریداری می شوند؟- مایع ظرف شویی و اسکاچ
  - بعد از خرید یک محصول چه محصولات دیگری خریداری میشود؟ – بعد از خرید کامپیوتر، خرید آنتی ویروس
  - چه نوع DNA به یک داروی جدید حساس است.
  - بعد از ملاقات یک صفحه وب خاص، کاربران به چه صفحه ای مراجعه می کنند.
- **کاربرد ها:** تحلیل سبد خرید مشتریان، طراحی کاتالوگ محصولات و تبلیغات، تحلیل فروش و طراحی فروشگاهها، تحلیل وبلاگ ها (click stream)، تحلیل دنباله DNA ها و ...

آنالیز سبد خرید



X. با پیدا کردن ارتباط زیاد فروش دو محصول می توان در چینش محصولات فروشگاه به شکلی عمل کرد که میزان فروش را افزایش دهد.

Ч. به فروشندگان کمک می کند تا تصمیم بگیرند چه کالاهایی را فروش ویژه اعلام کنند. در این حالت، فروش ویژه یک محصول باعث افزایش فروش دیگر محصولات مرتبط می گردد.

آنالیز سبد خرید

- فرض کنید $I = [I_1, I_2, \dots, I_m]$ مجموعه اقلام کالا باشد. هر تراکنش با نام T مجموعه ای از آیتم ها است که در آن $T \subset I$ می باشد.
- فرض کنید A مجموعه ای از آیتم ها باشد. گفته می شود تراکنش T شامل A است اگر و فقط اگر $A \subset T$ باشد.
- یک قانون انجمنی (Association rule) به صورت $A \Rightarrow B$ بیان می شود بطوریکه

$$A \cap B = \emptyset, A \subset I, B \subset I$$

مفاهیم اولیه: الگو های پر تکرار (Frequent patterns)

- مجموعه ای از یک یا چند آیتم را ***Itemset*** گوئیم
- ***Itemset*** که حاوی k آیتم ($x = \{I_1, I_2, \dots, I_k\}$) باشد را ***K-Itemset*** گویند.
- فرکانس تکرار یک ***Itemset*** را که نشان دهنده تعداد تکرار تراکنش‌هایی است که حاوی ***Itemset*** است را ***Support count*** آن ***Itemset*** گویند.
- برای یک ***Itemset*** ، نسبت تراکنش هایی که حاوی ***Itemset*** هستند به کل تراکنش‌ها را ***Support*** گویند
- ***Frequent Itemset***: یک ***Itemset*** را پر تکرار گویند اگر مقدار ***support*** آن کمتر از یک حد آستانه مشخص نباشد.

الگوهای پر تکرار و قوانین وابستگی(انجمنی)

- برای استخراج قوانین وابستگی به دو قدم زیر نیاز داریم:

X. پیدا کردن تمام الگوهای پر تکرار:

بر اساس تعریف، در این مرحله تمام الگوهای پر تکرار که مقدار ***support*** آنها از یک حد آستانه (***Min-Support***) کمتر نباشد را مشخص می کنیم.

Y. تولید قوانین وابستگی قوی از روی الگو های پر تکرار

در این مرحله قوانین را چنان استخراج می کنیم که میزان جذابیت آنها از یک حد آستانه بیشتر باشد.

برای تعیین میزان جذابیت یک قانون دو پارامتر برای هر قانون تعریف می شود.

قوانین وابستگی

- فرض کنید قانون $A \Rightarrow B$ در مجموعه تراکنش های D وجود دارد. برای این قانون دو پارامتر تعریف می شود:

۱. **Support :** نشان دهنده درصدی از تراکنش های D که حاوی A و B است. این درصد را به صورت احتمال $P(A \cup B)$ بیان می کنیم.

$$support(A \Rightarrow B) = P(A \cup B)$$

۲. **Confidence** : بیان کننده درصدی از تراکنش های D است که اگر حاوی A باشند، آنگاه B نیز در آن وجود داشته باشد.

$$Confidence(A \Rightarrow B) = P(B|A)$$

$$Confidence(A \Rightarrow B) = \frac{Support(A \cup B)}{Support(A)} = \frac{Support\_count(A \cup B)}{Support\_count(A)}$$

قوانین وابستگی

قانون $A \Rightarrow B$ برای دو *Itemset* به نامهای A و B را یک قانون وابستگی گویند اگر شرایط زیر را دارا باشد:

1. دارای حداقل مقدار *Min_Support* باشد.
2. دارای حداقل مقدار *Min_Confidence* باشد.
3. $A \cap B = \emptyset$

قوانین وابستگی (مثال)

Tid	آیتم های خریداری شده
10	Bread, Nuts, Butter
20	Bread, Coffee, Butter, Eggs
30	Bread, Butter, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Butter, Eggs, Milk

فرض کنید:

حداقل مقدار ***support*** برابر ۵۰٪ باشد

حداقل مقدار***confidence*** برابر ۵۰٪ باشد

Frequent Patterns:

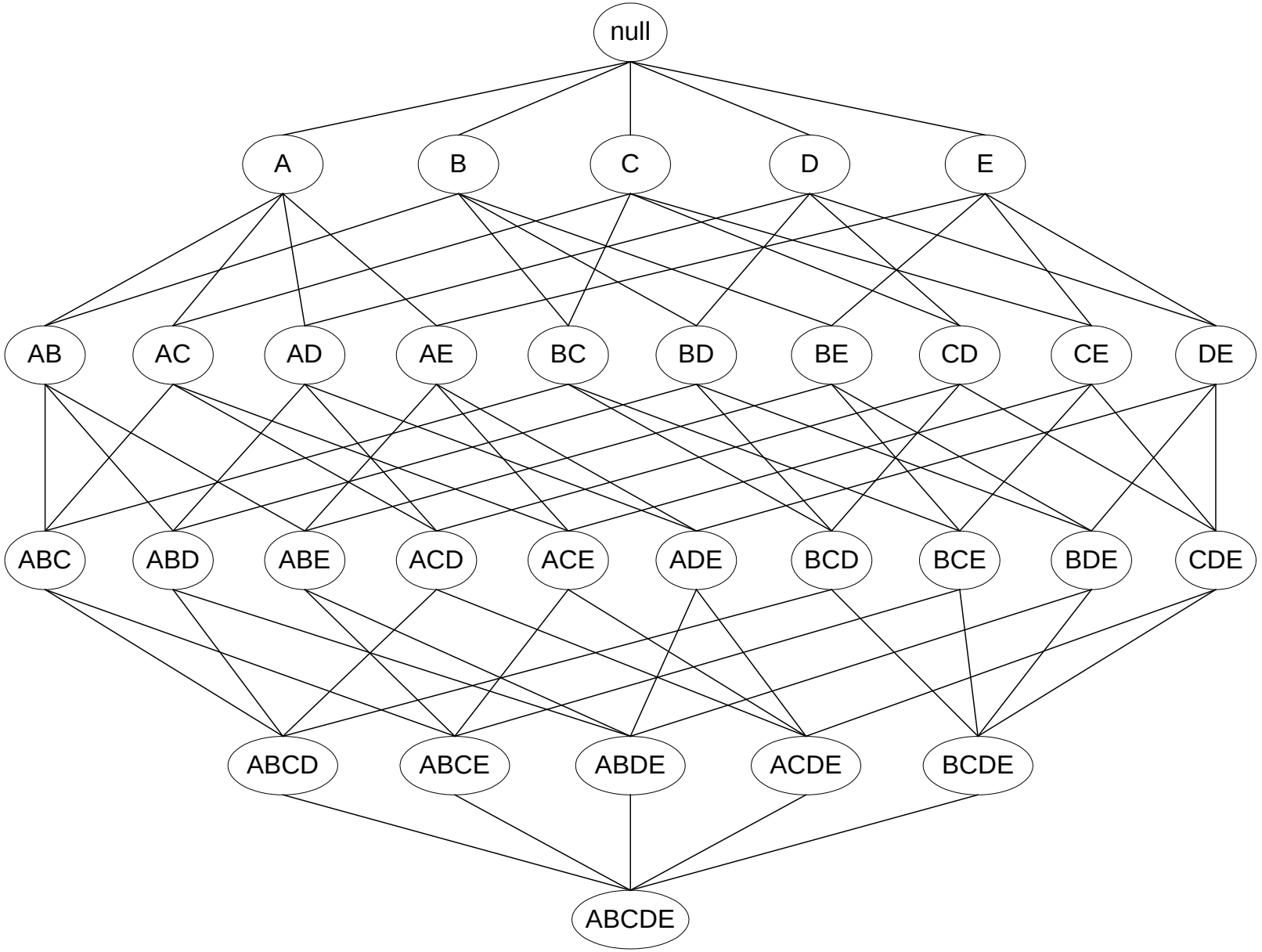
Bread:3, Nuts:3, Butter:4, Egg:4,
(Bread, Butter):3

Association Rule:

Bread⇒Butter (60%,100%)

Butter⇒Bread (60%,75%)

Frequent Itemset Gneration



یک مشکل

- فرض کنید یک الگوی پر تکرار شامل مجموعه $\{a_1,a_2,...a_{100}\}$ باشد.
- نکته مهم: تمامی زیر مجموعه های یک مجموعه پر تکرار حتما پر تکرار خواهند بود.
- تعداد زیر مجموعه های این مجموعه برابر است با :

$$\binom{100}{1}+\binom{100}{2}+\cdots+\binom{100}{100}=2^{100}-1\cong 1.27\times 10^{30}$$

- این تعداد زیاد *itemset* ها هم از نظر محاسبه و هم از نظر ذخیره سازی مشکل ساز است.
- برای غلبه بر این مشکل دو مفهوم را معرفی می کنیم:

ب. *Closed frequent itemset*

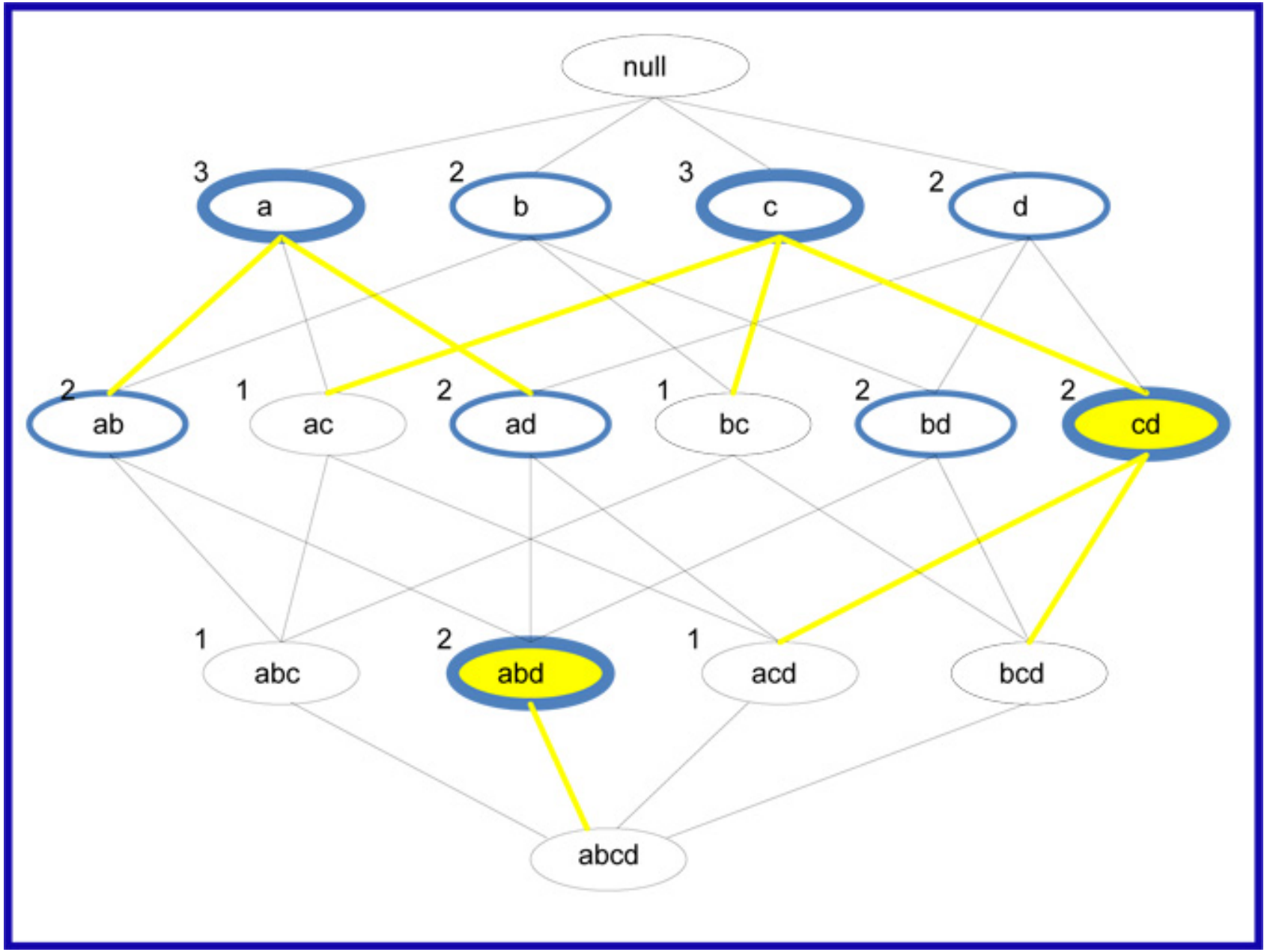
ج. *Maximal frequent itemset*

Closed Frequent Itemset

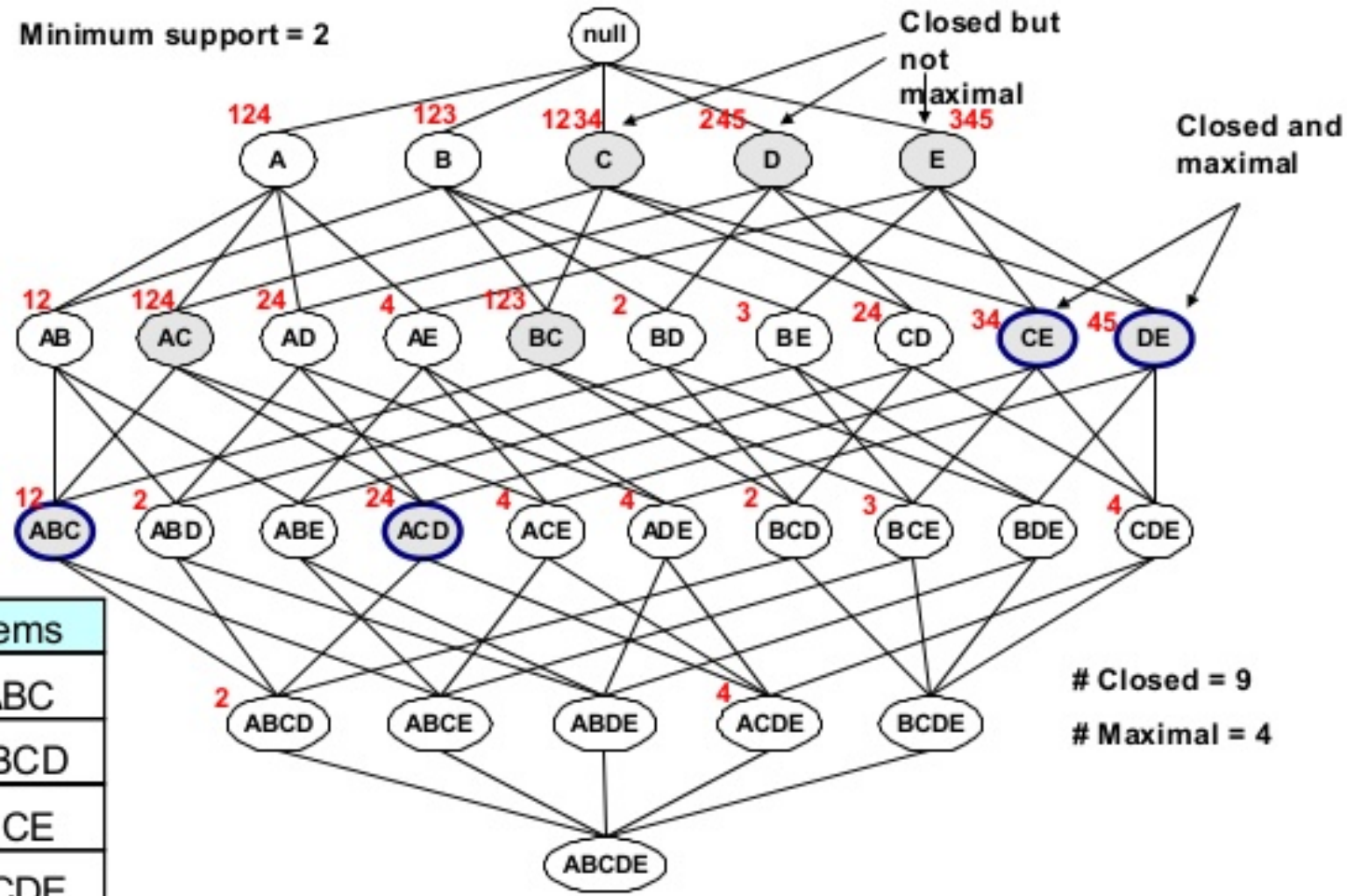
- تعریف ۱: در یک مجموعه داده S یک *Itemset* مثل X را بسته (Closed)گویند هر گاه هیچ *Supper-Itemset* مثل Y ($X \subset Y$) وجود نداشته باشد که *support_count* آن مشابه X باشد.
- تعریف ۲: یک *Itemset* در مجموعه داده S را *Closed frequent itemset* گویند اگر
  - ✓ بسته (*Closed*) باشد
  - ✓ پر تکرار باشد.

Maximal Frequent Itemset

- تعریف : در یک مجموعه داده S یک *Itemset* مثل X را *Maxima frequent Itemset* یا (*Max-itemset*) گویند هر گاه X پر تکرار باشد و هیچ *Supper-Itemset* مثل Y ($X \subset Y$) وجود نداشته باشد که Y هم پر تکرار باشد.


null
3
a
2
b
3
c
2
d
2
ab
1
ac
2
ad
1
bc
2
bd
2
cd
1
abc
2
abd
1
acd
bcd
abcd

Maximal vs Closed Frequent Itemsets



TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

مثالی دیگر

- فرض کنید در یک دیتا بیس دو تراکنش وجود داشته باشد:
$\{\langle a_1,a_2,\dots,a_{100}\rangle\langle a_1,a_2,\dots,a_{50}\rangle\}$ و فرض کنید $Min\_sup = 1$ باشد.
- ما میتوانیم دو تا $Closed\ frequent\ itemset$ پیدا کنیم که عبارتند از:
$$\mathbf{C} = \{\{a_1, a_2, \dots, a_{100}\}: 1, \{a_1, a_2, \dots, a_{50}\}: 2\}$$
- یک عدد $Maximal\ frequent\ Itemset$ نیز وجود دارد:
$$\mathbf{M} = \{\{a_1, a_2, \dots, a_{100}\}: 1\}$$
- مجموعه C و اطلاعات مربوط به تعداد تکرار آن می تواند برای استخراج اطلاعات تمامی $frequent\ itemset$ ها مورد استفاده قرار گیرد. در واقع گوییم این مجموعه تمامی اطلاعات مورد نیاز برای استخراج $frequent\ itemset$ ها را دارا می باشد.
- به عنوان مثال: از مجموعه C می توان گفت:

ب. $\{a_2,a_{45}: 2\}$ چونکه مجموعه $\{a_2,a_{45}\}$ زیر مجموعه $\{a_1,a_2,\dots,a_{50}: 2\}$ است.

ج. $\{a_8,a_{55}: 1\}$ چونکه مجموعه $\{a_8,a_{55}\}$ زیر مجموعه $\{a_1,a_2,\dots,a_{100}: 1\}$ است.

- بر اساس مجموعه M فقط می توان $frequent\ itemset$ ها را بدون اطلاعات تکرار بدست آورد.

روش های کاوش الگوهای پرتکرار

۱- الگوریتم **Apriori** :روشی است مبتنی بر تولید – و – آزمایش الگوهای کاندید

۲- روش **FPGrowth** : روشی برای ساخت الگوهای مکرر از طریق گسترش آنها

۳- روش **ECLAT** : کاوش الگوهای پر تکرار بر اساس چیدمان داده ها به صورت عمودی

ویژگی Downward Closure

این ویژگی می گوید:

- یک زیر مجموعه از الگو های پر تکرار، خود پر تکرار است.

مثلا: اگر $\{Bread, Nuts, Butter\}$ پر تکرار باشد آنگاه $\{Bread, Butter\}$ نیز پر تکرار است.

الگوریتم APRIORI

- نام این الگوریتم به دلیل اینکه از دانش پیشین استفاده می کند *A Priori* انتخاب شده است.
- در این الگوریتم از اطلاعات موجود در تولید *K-Itemset* برای تولید *(K+1)-Itemset* استفاده می شود.
- ابتدا برای یافتن *1-itemset* های پر تکرار، یکبار مجموعه تراکنش ها اسکن می شوند. نتیجه کاوش L_1 نامیده می شود.
- سپس L_1 برای یافتن L_2 که حاوی *2-itemset* های پر تکرار است، مورد استفاده قرار می گیرد.
- این مراحل تا جایی که دیگر هیچ *k-itemset* پر تکراری پیدا نشود ادامه می یابد.
- برای یافتن هر L_k نیاز به یکبار اسکن مجموعه تراکنش ها می باشد.

الگوریتم APRIORI (ادامه)

- برای تولید هوشمندانه تر مجموعه های پر تکرار می توان از خاصیت *Apriori Property* استفاده کرد.
- *Apriori Property* : همه زیر مجموعه های نا تهی از یک مجموعه پر تکرار باید پر تکرار باشند.
- این خاصیت بر اساس مشاهده زیر بنا شده است:

  اگر یک مجموعه مثل I حداقل مقدار استانه $Support$ را نداشته باشد، بنابراین پر تکرار نیست. حال اگر آیتمی مثل A به این مجموعه اضافه شود، مجموعه حاصل $(I \cup A)$ نمی‌تواند تعداد تکرار بیشتری داشته باشد. بنابراین نمی‌تواند پر تکرار باشد.
- چگونه از این ویژگی استفاده کنیم؟ (جواب در اسلایدهای بعدی)

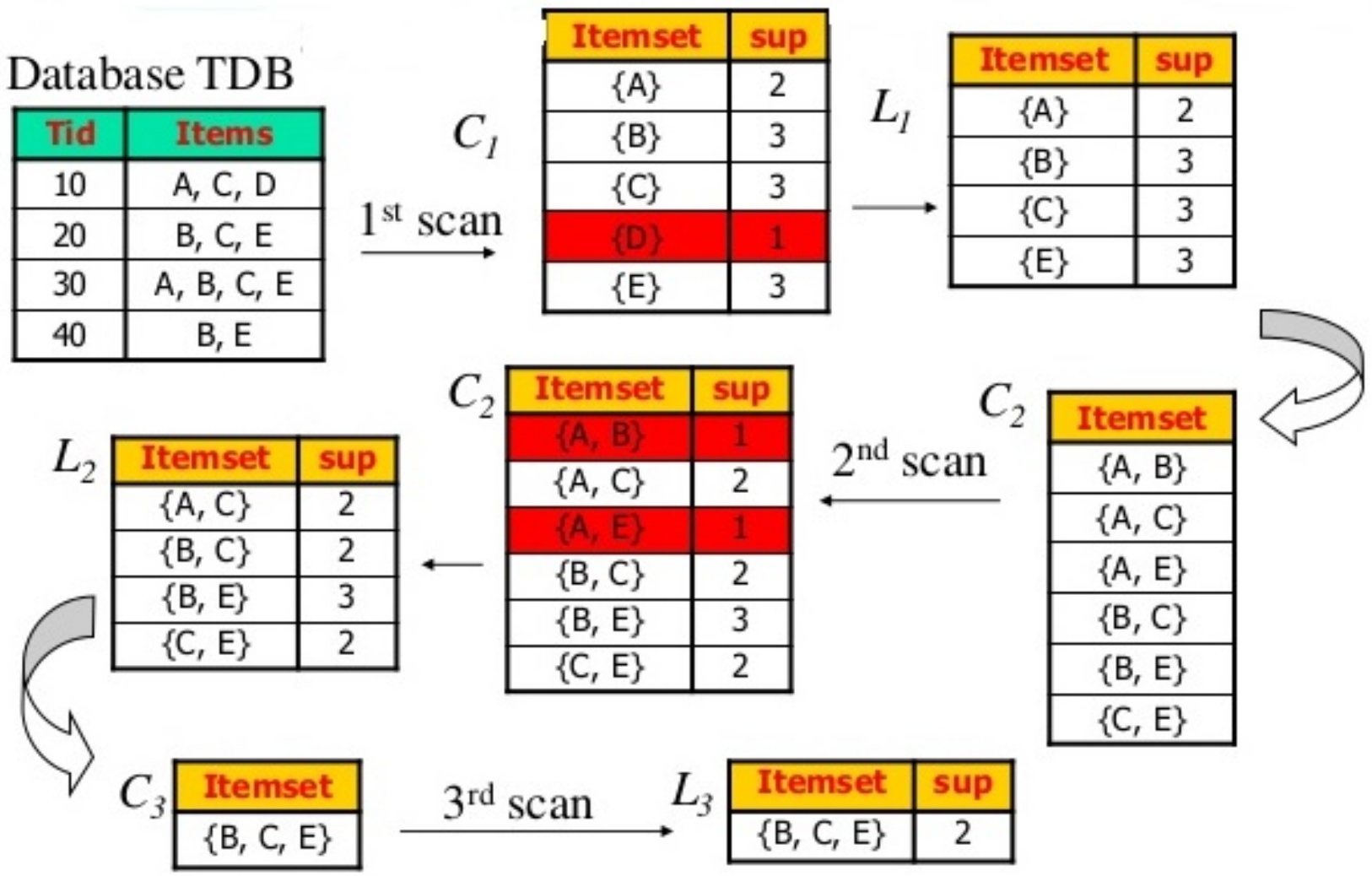
چگونگی ایجاد L_k بر اساس L_{k-1}

- فرایند دو مرحله ای برای ایجاد L_k

B. گام اتصال *(join step)*:برای ایجاد L_k که معرف مجموعه کاندیداهای k-itemset هستند، L_{k-1} را با خودش اتصال می دهیم. مجموعه بدست آمده را C_k مینامیم.

Γ. گام هرس *(prune step)*: بر اساس خاصیت Apriorit property هر عنصر k-itemset که زیر مجموعه‌هایش در L_{k-1} نباشند، نمی‌تواند پر تکرار باشد.

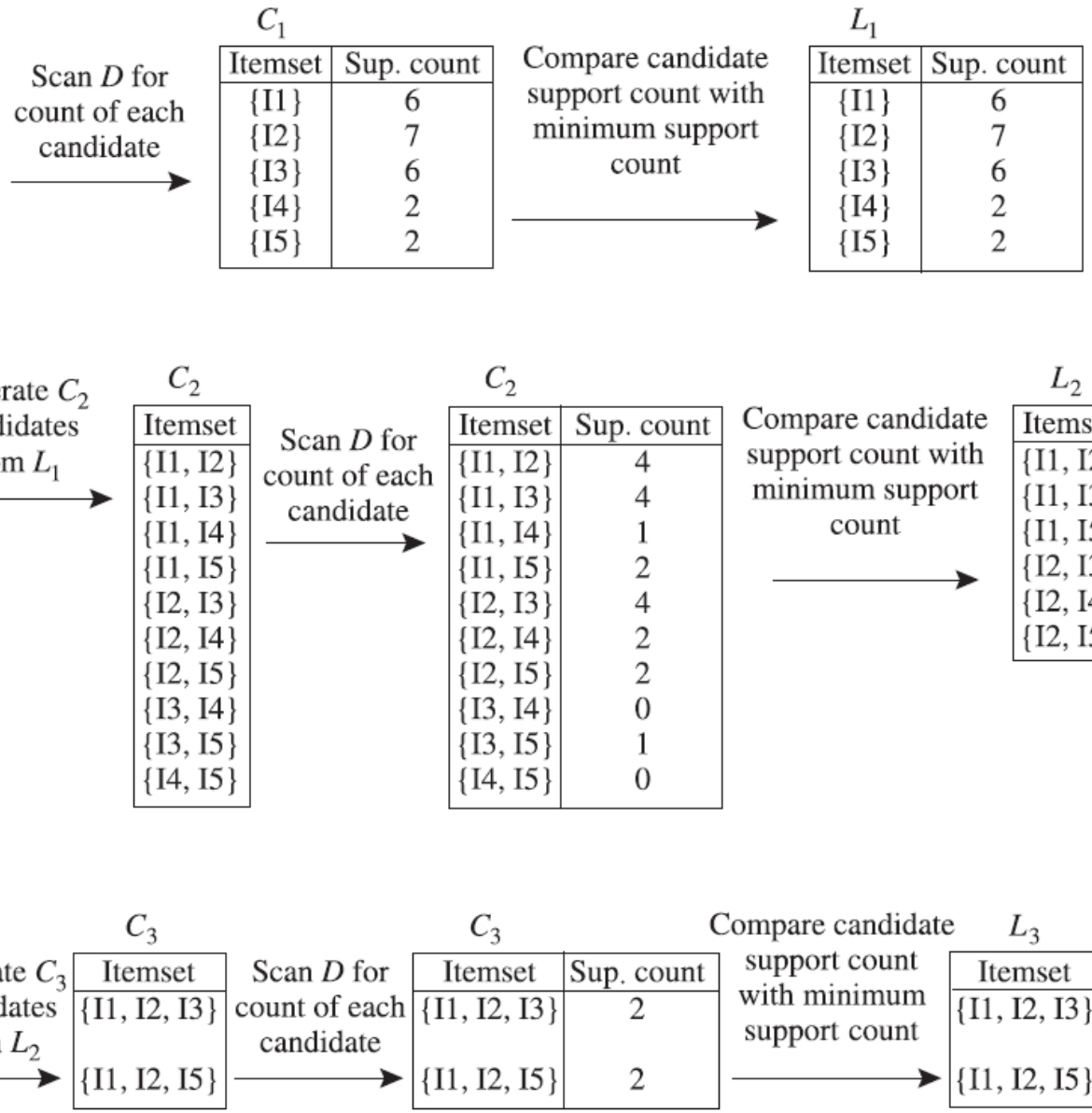
مثال۱



Min_Sup = 2

مثال ۲

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



تولید قوانین وابستگی از روی الگوهای پر تکرار

- زمانیکه الگوهای پر تکرار از روی پایگاه داده استخراج شدند، تولید قوانین وابستگی (انجمنی) قوی کار آسانی خواهد بود.
- در این حالت قانونی قوی است که: حداقل استانه های *support* و *confidence* را داشته باشد.
- نحوه تولید قوانین به شرح زیر است:

X. برای هر *itemset* پر تکرار مثل l تمام زیر مجموعه های نا تهی آن تولید می شود

Ц. برای هر زیر مجموعه غیر تهی s از l قانونی به صورت زیر تولید می شود:

$$s \Rightarrow (l - s)\ if\ \frac{support\_count(L)}{support\_count(S)} \geq min\_Conf$$

مثال

- به عنوان مثال برای الگوی پر تکرار $l = \{I1,I2,I5\}$ قوانین تولید شده عبارتند از :

$$
\begin{aligned}
&\{I1, I2\} \Rightarrow I5, && confidence = 2/4 = 50\% \\
&\{I1, I5\} \Rightarrow I2, && confidence = 2/2 = 100\% \\
&\{I2, I5\} \Rightarrow I1, && confidence = 2/2 = 100\% \\
&I1 \Rightarrow \{I2, I5\}, && confidence = 2/6 = 33\% \\
&I2 \Rightarrow \{I1, I5\}, && confidence = 2/7 = 29\% \\
&I5 \Rightarrow \{I1, I2\}, && confidence = 2/2 = 100\%
\end{aligned}
$$

- در صورتی که **min_sup = 70%** باشد ۲و۳و۶ انتخاب می شوند.

بهبود الگوریتم *Apriori*

- اصلی ترین چالشهای محاسباتی الگوریتم *Apriori*

B. مرور چند باره پایگاه داده

Г. تعداد زیاد کاندیداهای تولید شده

Д. محاسبه support برای کاندیداها

- ایده های اصلی

B. کاهش تعداد مرور پایگاه داده

Г. کاهش تعداد کاندیداها

Д. تسهیل محاسبه support کاندیداها

افزایش بهره وری الگوریتم *Apriori*

ب) استفاده از توابع در هم ساز (Hash-based techniques)

ج) کاهش تراکنش‌ها (Transaction reduction)

د) پارتیشن بندی (Partitioning)

ت) نمونه برداری (Sampling)

ی) شمارش پویای آیتم ست ها (Dynamic itemset counting)

۱) استفاده از تابع در هم ساز

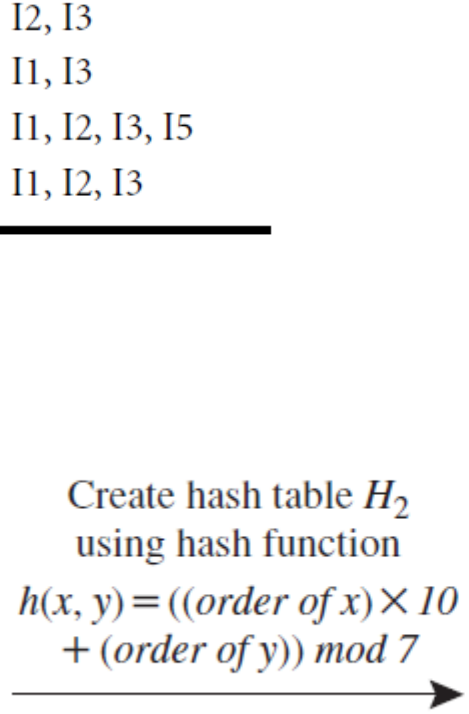
- از این روش برای کاهش مجموعه های کاندیدا و اسکن پایگاه داده ها استفاده می شود.
- در این روش اندازه C_k کاهش می یابد
- مثلا در زمان تولید 1-itemset ها می توان 2-itemset ها را نیز تولید کرد.
- در این مرحله 2-itemset ها را با استفاده از یک تابع درهم ساز بسته بندی می کنیم.
- هر 2-itemset در یک بسته (bucket) قرار می گیرد.
- هر بسته که تعداد اعضاء آن کمتر از حد آستانه support باشد پر تکرار نیست.

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

مثال

$$h(I1, I4) = (1 \times 10 + 4)\ mod\ 7 = 0$$

Create hash table H_2 using hash function $h(x, y) = ((order\ of\ x) \times 10 + (order\ of\ y))\ mod\ 7$



H_2

bucket address	0	1	2	3	4	5	6
bucket count	2	2	4	2	2	4	4
bucket contents	{I1, I4} {I3, I5}	{I1, I5} {I1, I5}	{I2, I3} {I2, I3} {I2, I3} {I2, I3}	{I2, I4} {I2, I4}	{I2, I5} {I2, I5}	{I1, I2} {I1, I2} {I1, I2} {I1, I2}	{I1, I3} {I1, I3} {I1, I3} {I1, I3}

۲) کاهش تراکنش ها

- در این روش در هر مرحله از تعداد تراکنش‌هایی که باید اسکن شوند کاسته می شود
- ایده اصلی: یک تراکنش که شامل هیچ k-itemset پر تکراری نباشد، نمیتواند. (k+1)-itemset پر تکرار داشته باشد. بنابراین این تراکنش می تواند در سنجش های بعدی حذف شود.
- بنابراین برای تولید p-itemset ها که p>k است، نیازی به در نظر گرفتن این تراکنش نیست.

الگوریتم FP-Growth

- این الگوریتم به صورت افزایشی رشد می کند.
- هدف این الگوریتم یافتن الگوهای پر تکرار بدون تولید مجموعه های کاندید است.
- الگوریتم در دو فاز عمل می کند

B.تولید درخت FP-Tree

Γ.تولید الگوهای پرتکرار از روی درخت FP-Tree

الگوریتم FP-Growth

B. تولید درخت FP-Tree

a) پایگاه داده برای محاسبه support count هر آیتم یکبار از ابتدا تا انتها پیمایش می شود. در این مرحله آیتم های کم تکرار نادیده گرفته می شوند و آیتم های پر تکرار به ترتیب نزولی بر اساس مقدار support count شان مرتب می شوند.

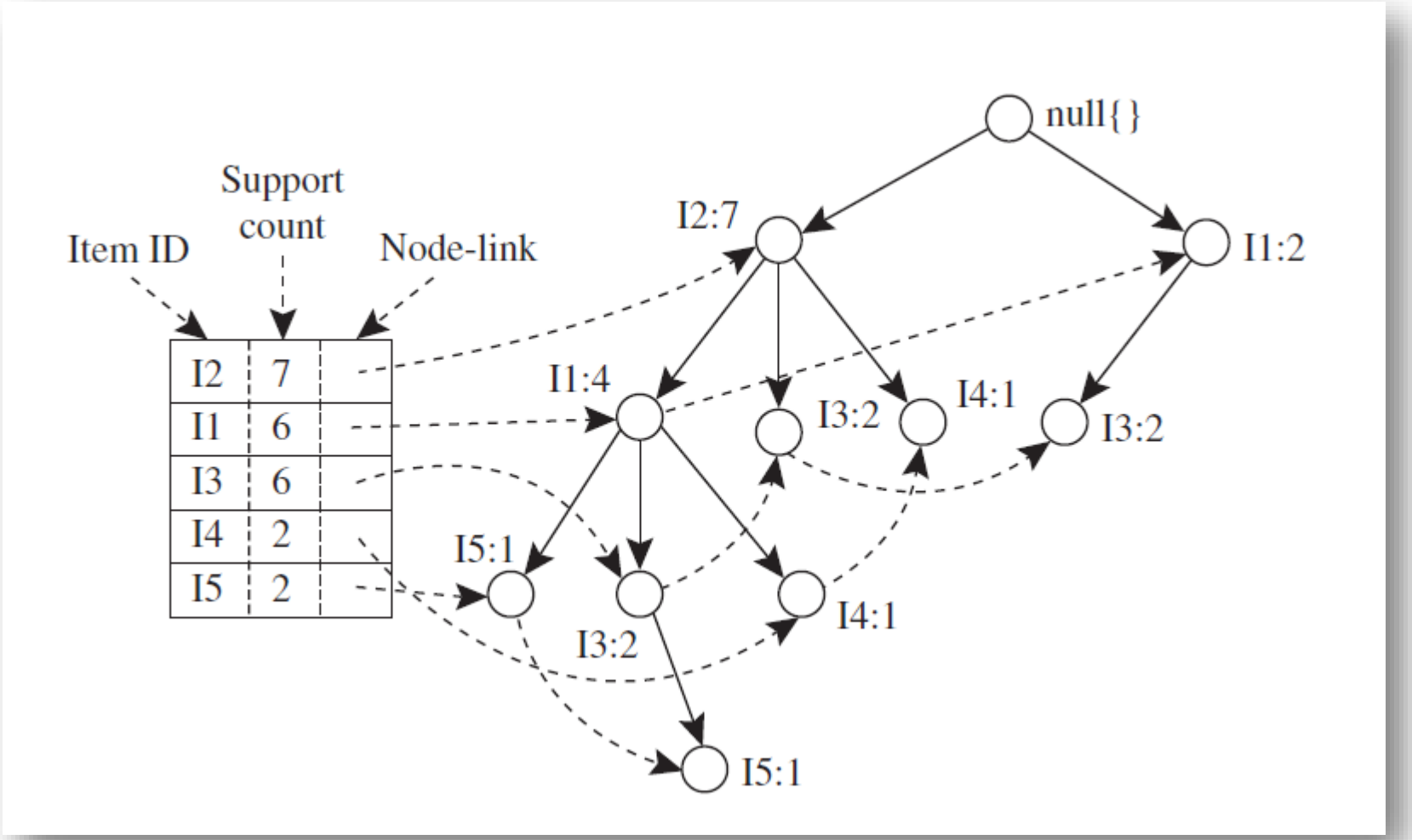
b) آیتم های تمام تراکنش های پایگاه داده بر اساس ترتیب support count ها مرتب می شود.

c) پایگاه داده برای بار دوم پیمایش می شود تا درخت FP-Tree ساخته شود.

نحوه ساخت درخت FP-Tree با استفاده از یک مثال تشریح می گردد:

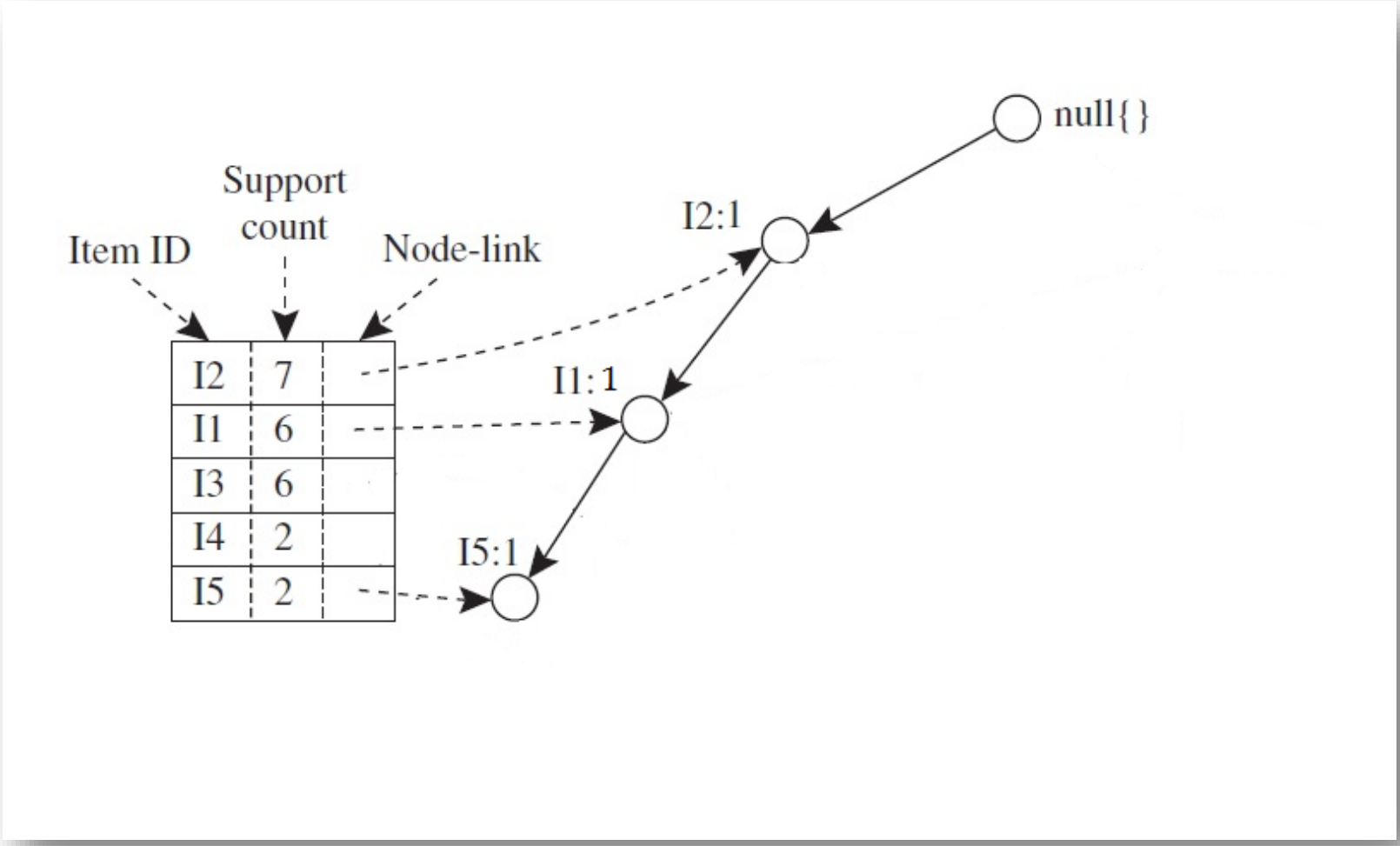
ساخت درخت FP-Tree

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



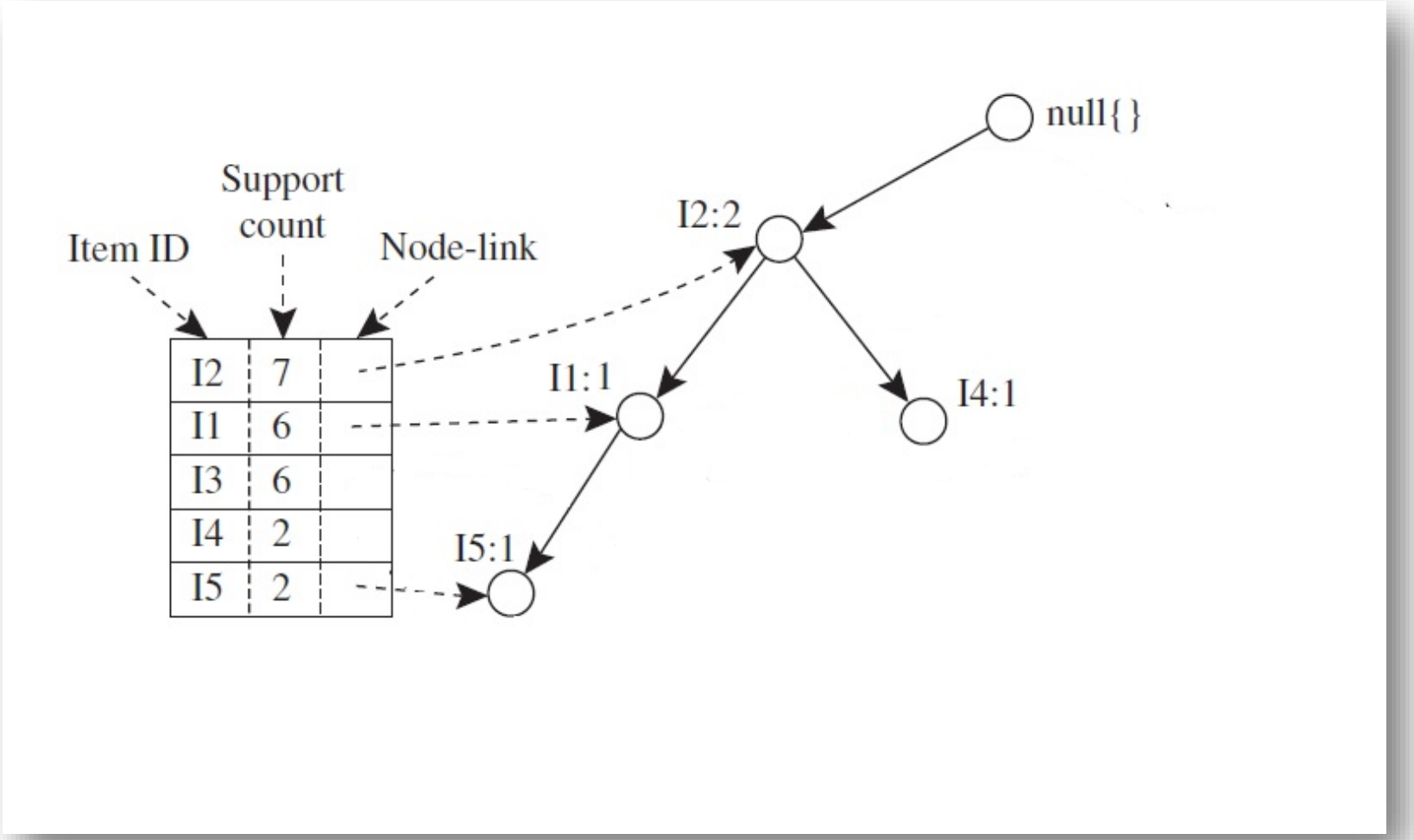
ساخت درخت FP-Tree

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



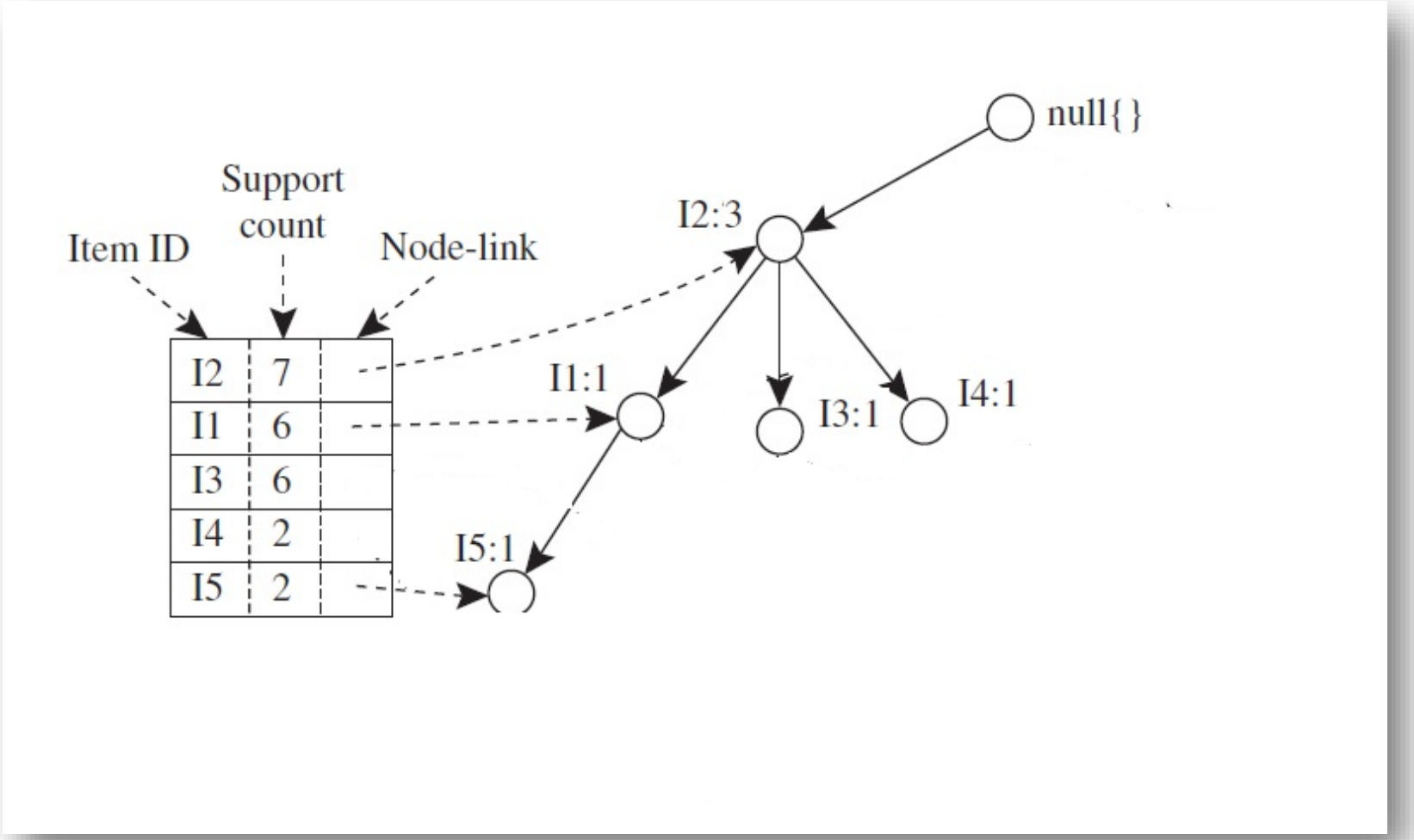
ساخت درخت FP-Tree

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



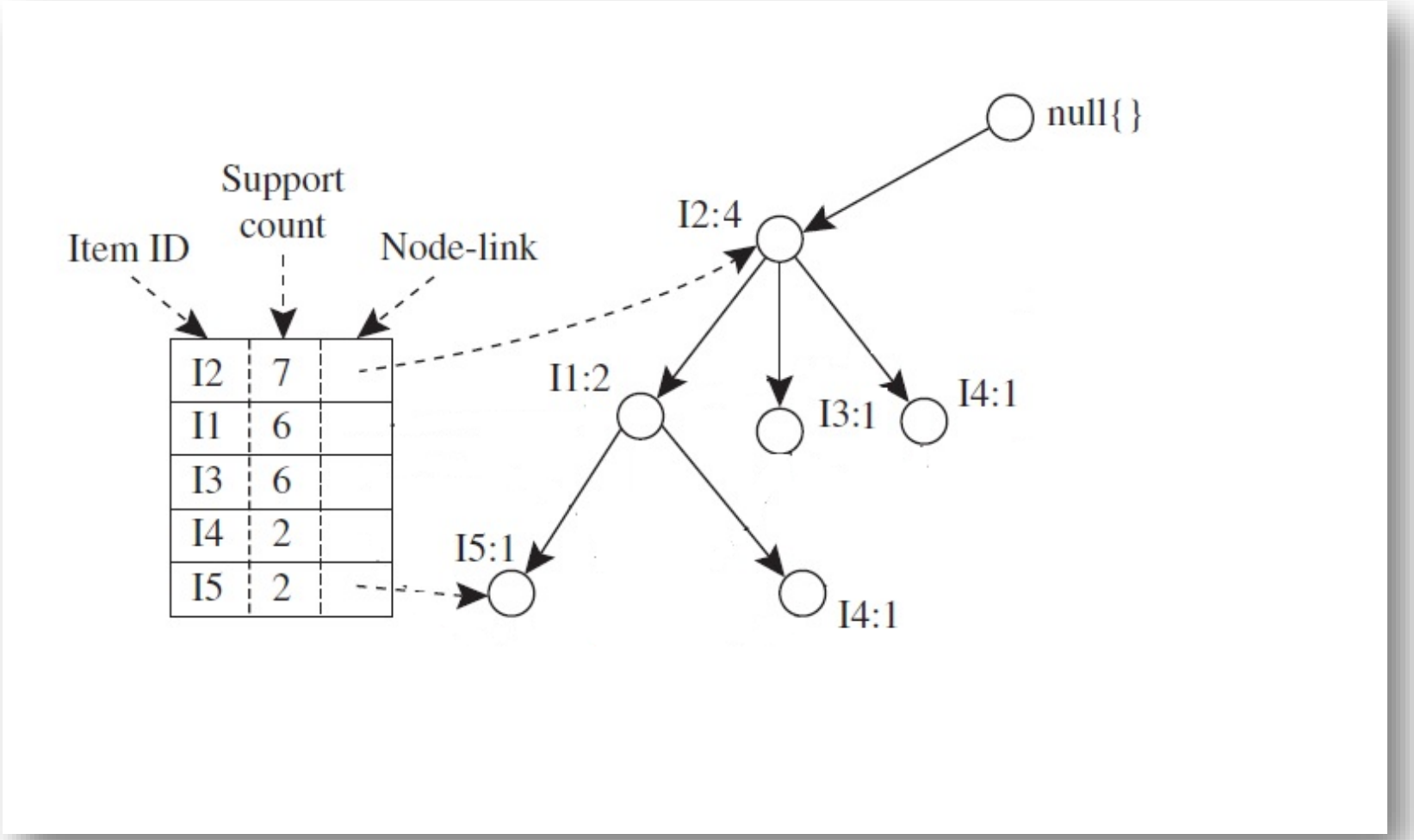
ساخت درخت FP-Tree

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



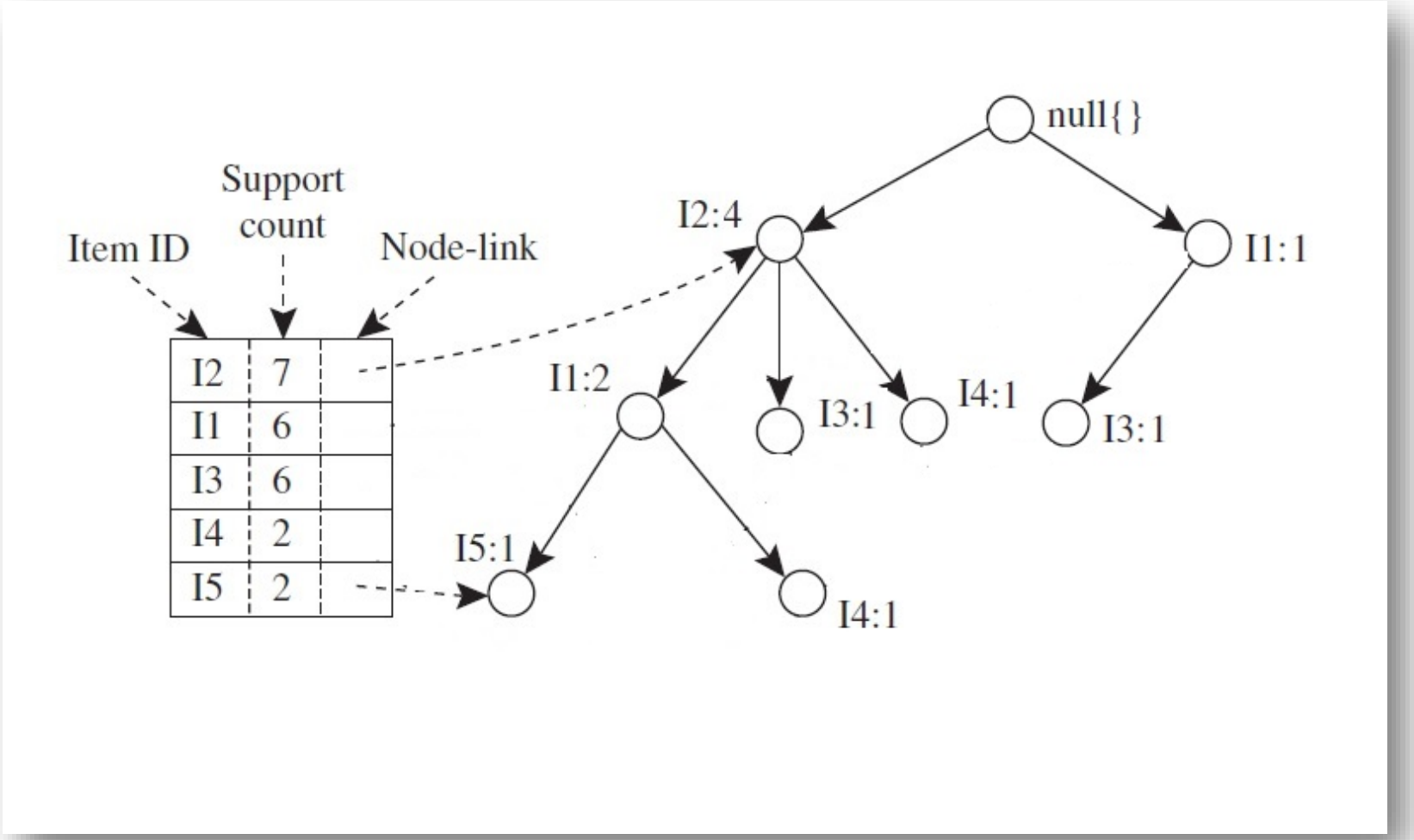
ساخت درخت FP-Tree

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



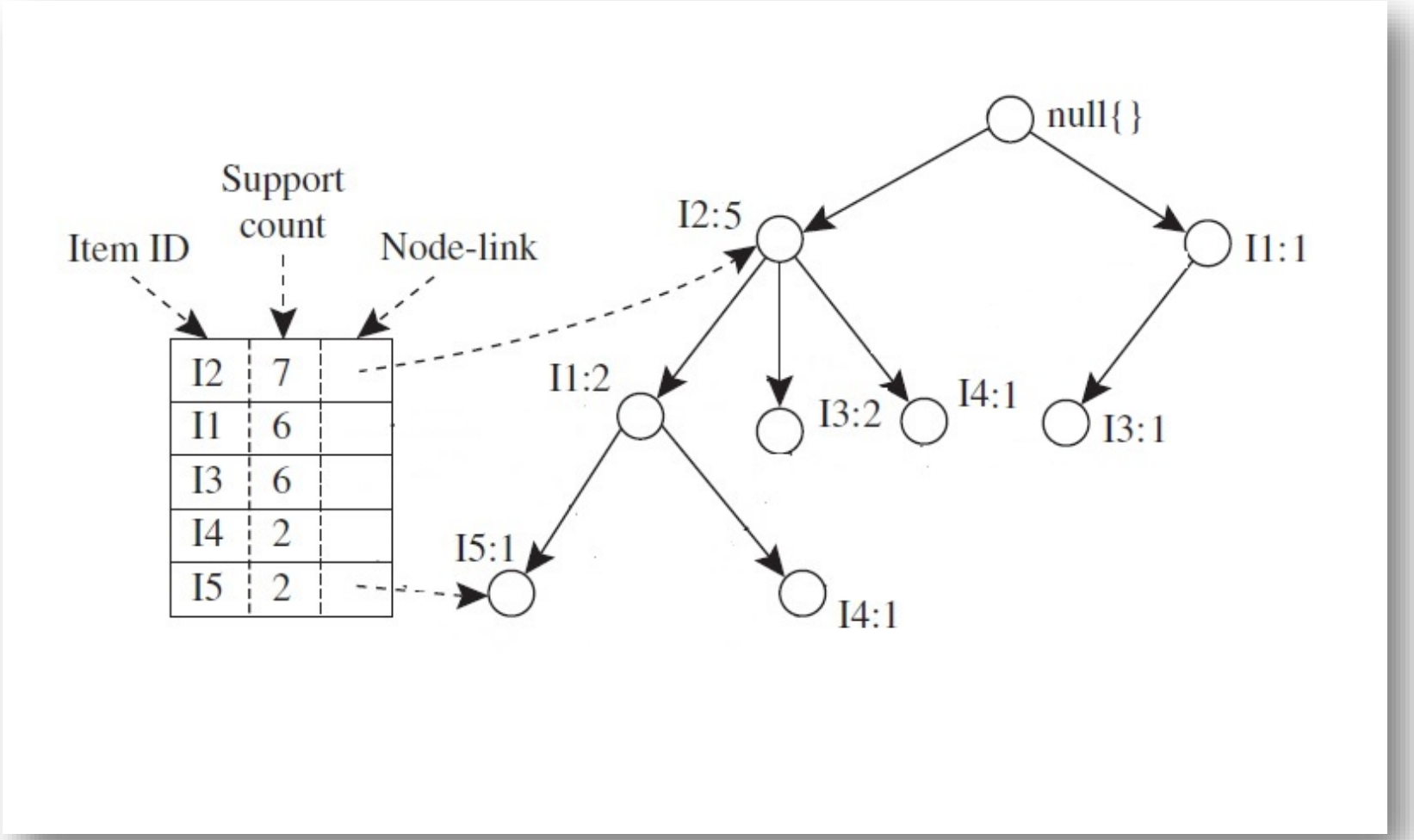
ساخت درخت FP-Tree

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



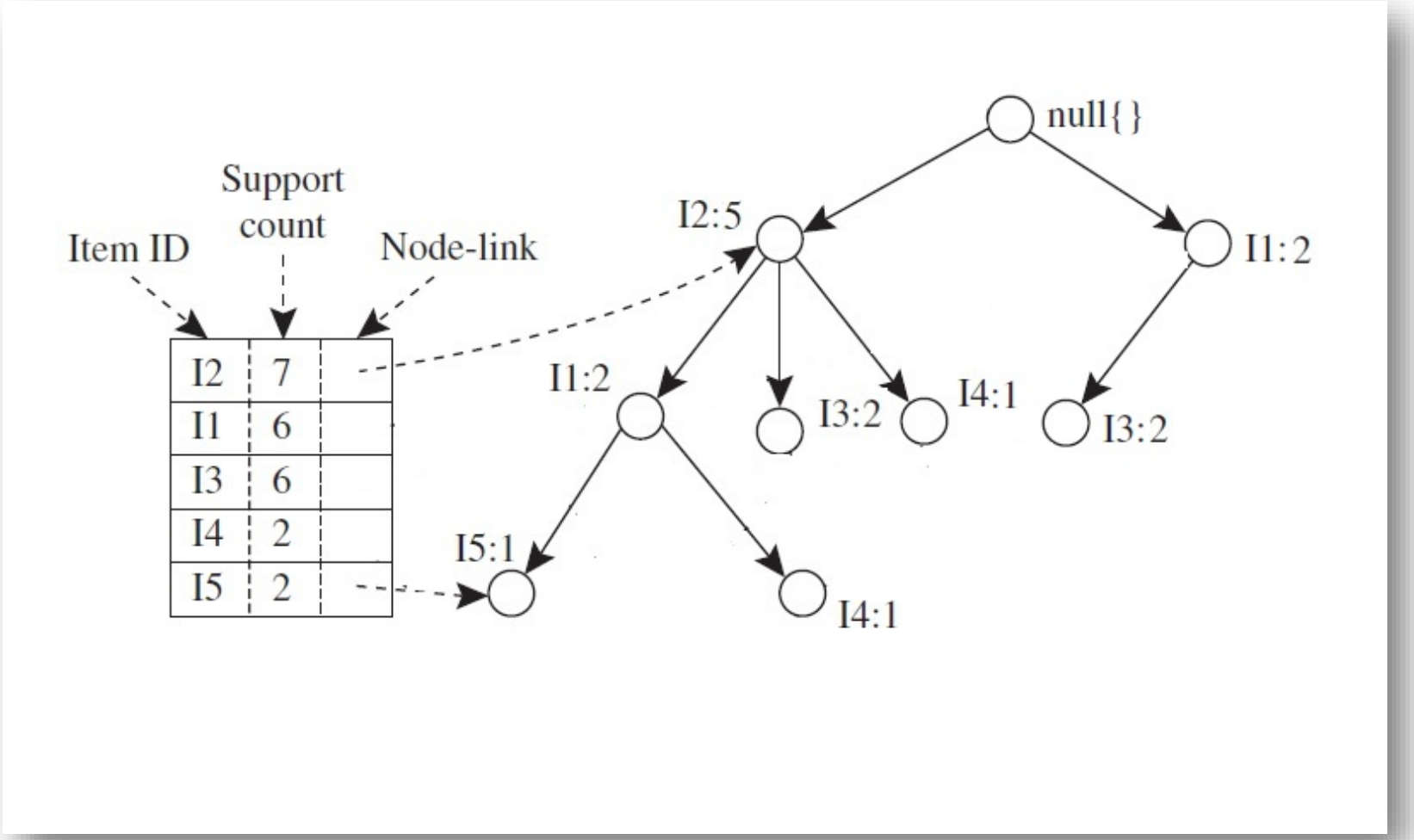
ساخت درخت FP-Tree

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



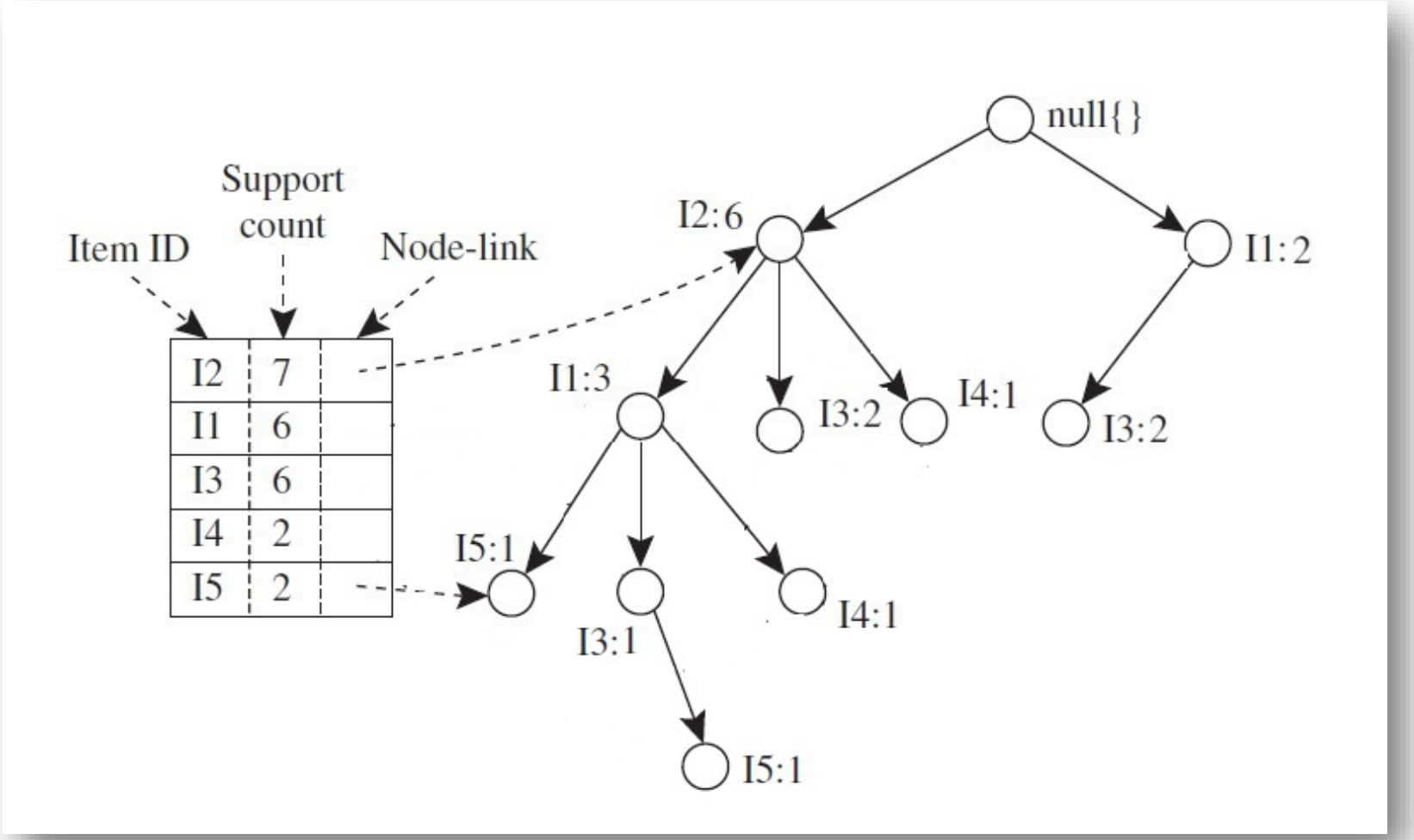
ساخت درخت FP-Tree

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



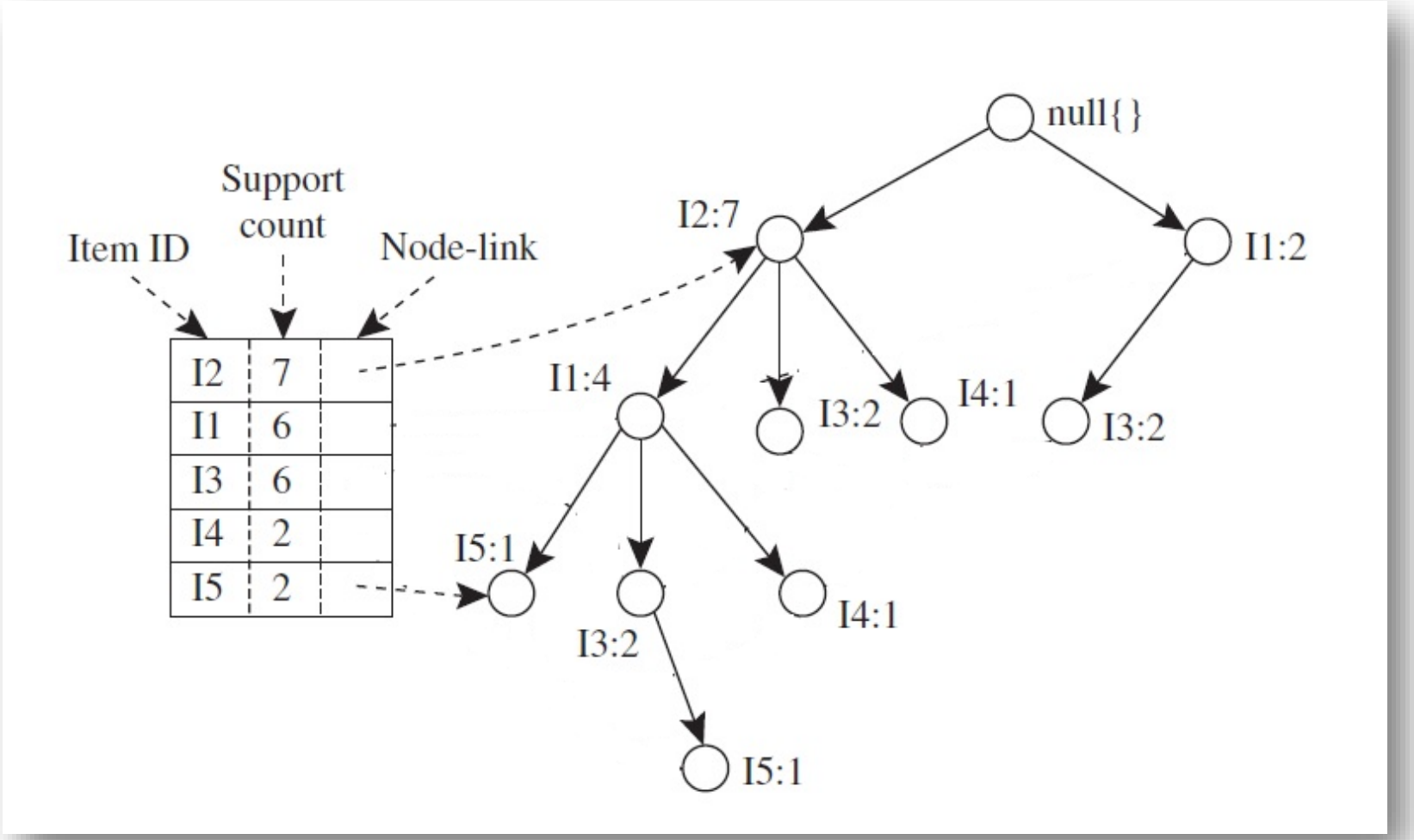
ساخت درخت FP-Tree

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



ساخت درخت FP-Tree

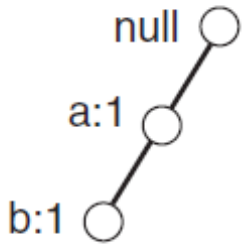
TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



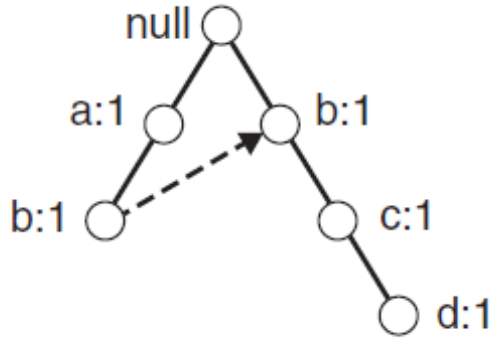
ساخت درخت FP-Tree (مثالی دیگر)

Transaction Data Set

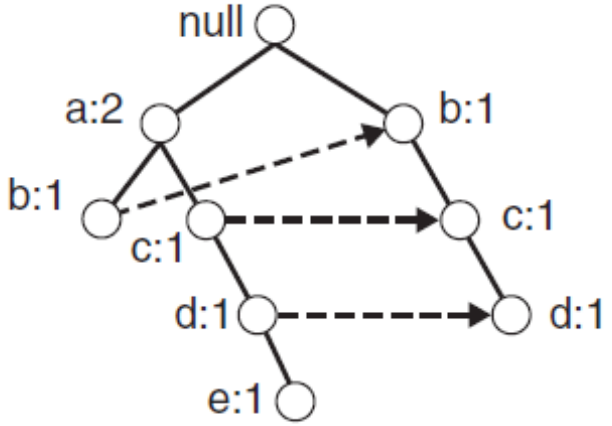
TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}



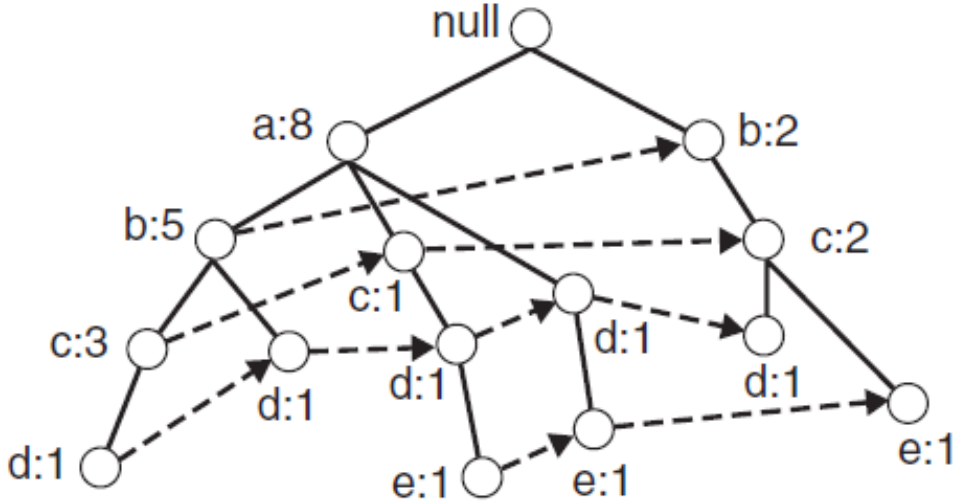
(i) After reading TID=1



(ii) After reading TID=2



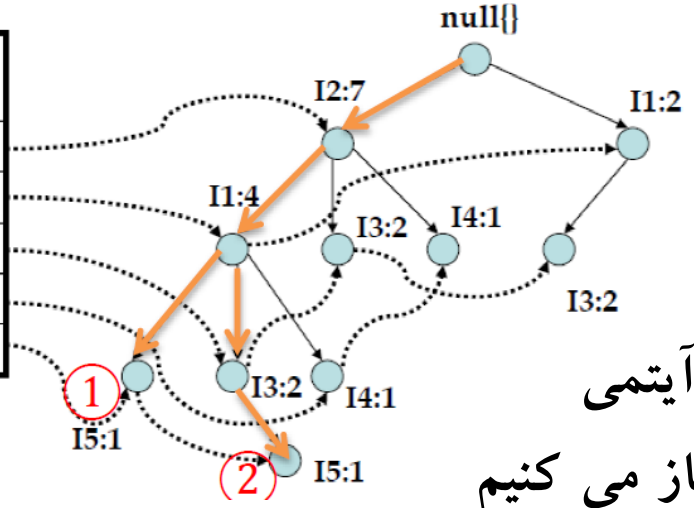
(iii) After reading TID=3



(iv) After reading TID=10

تولید الگوهای پر تکرار

Item Id	Sup Count	Node-link
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	



۲- تولید الگوهای پرتکرار از روی درخت FP-Tree

با کاوش درخت FP-Tree می توانیم الگوهای پرتکرار را بدست بیاوریم. برای این منظور چنین عمل می کنیم:

- از برگهای درخت FP-Tree شروع می کنیم و این کار را از آیتمی که در لیست آیتم های پر تکرار کمترین تکرار را داشته است آغاز می کنیم
- مثلا برای رسیدن به I_5 دو مسیر وجود دارد:

1: $(I_2, I_1, I_5{:}1)$ 2: $(I_2, I_1, I_3, I_5{:}1)$

- حالا I_5 را پسوند الگو (Suffix) می نامیم و $(I_2, I_1{:}1)$ و $(I_2, I_1, I_3{:}1)$ را Conditional Pattern Base گوییم

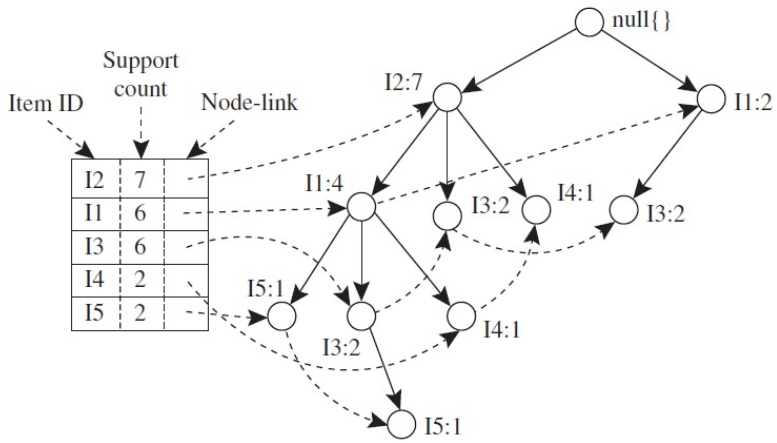
تولید الگوهای پر تکرار

- $(I_2, I_1:1)$ و $(I_2, I_1, I_3:1)$ را Conditional Pattern Base مربوط به I_5 گوییم.
- اگر Min_Sup را ۲ در نظر بگیریم، آنگاه درخت شامل مسیر $<I_2:2, I_1:2>$ می باشد.
- I_3 حذف می شود چرا که کمتر از Min_Sup است.
- حال برای تولید الگو های پرتکرار کافی است ترکیبات مختلف با I_5 را در نظر بگیریم:

$\{I_2, I_5 :2\}$

$\{I_1, I_5 :2\}$

$\{I_2, I_1, I_5 :2\}$



تولید الگوهای پر تکرار

- برای I_4 هم دو مسیر وجود دارد:

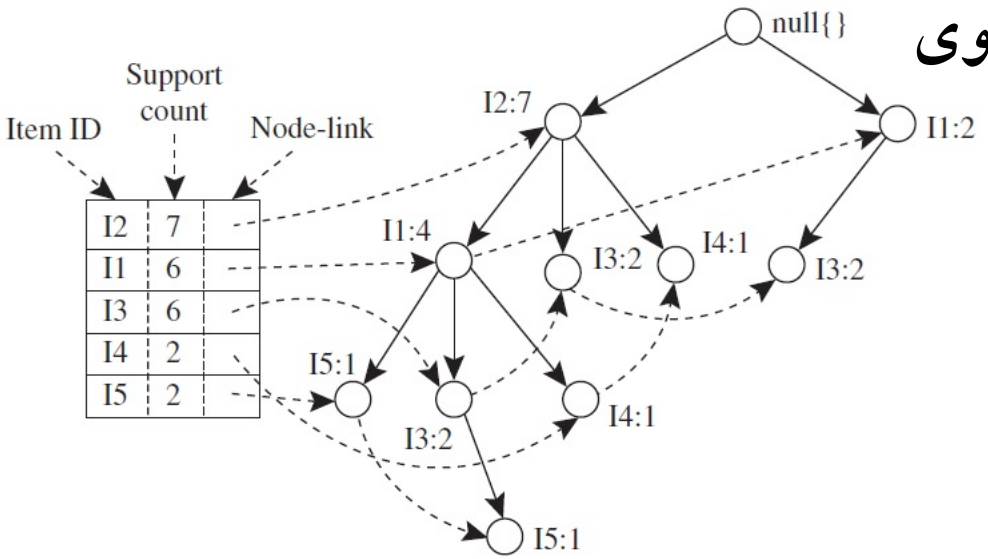
$1: (I_2, I_1, I_4 :1)$ $2: (I_2, I_4 :1)$

بنابراین Conditional Pattern Base های مربوط به I_4 عبارتند از:

$(I_2, I_1 :1)$, $(I_2 :1)$

بنابراین درخت شامل مسیر $<I_2:2>$ می باشد که الگوی زیر را تولید می کند:

$(I_2, I_4 :2)$



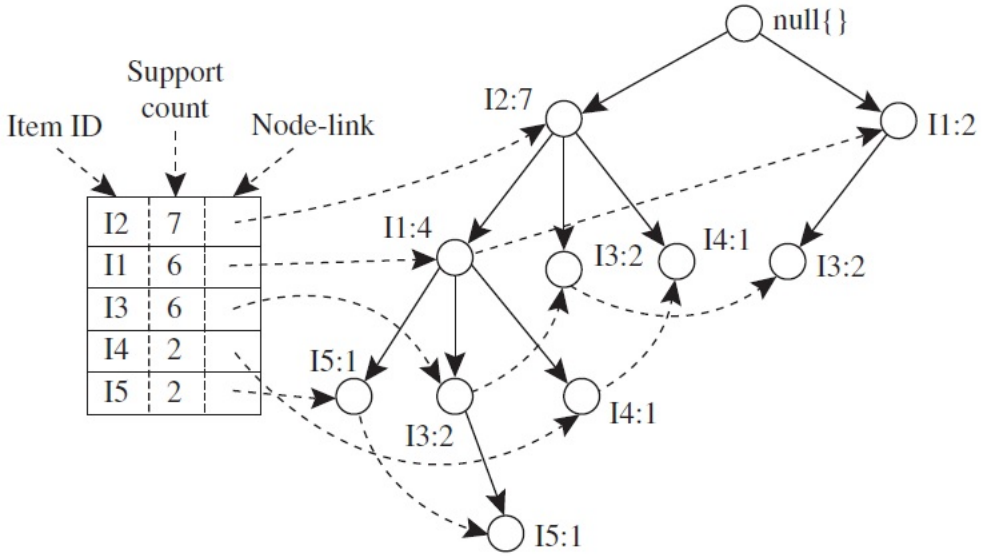
تولید الگوهای پر تکرار

- برای I_3 هم سه مسیر وجود دارد:

1: $(I_2, I_1, I_3 :2)$ 2: $(I_2, I_3 :2)$ 3: $(I_1, I_3 :2)$

بنابراین Conditional Pattern Base های مربوط به I_3 عبارتند از:

$(I_2, I_1 :2)$, $(I_2 :2)$, $(I_1 :2)$



بنابراین درخت شامل مسیر $<I_2:4, I_1:2>$ و $<I_1:2>$ می باشد که الگوهای زیر را تولید می کند:

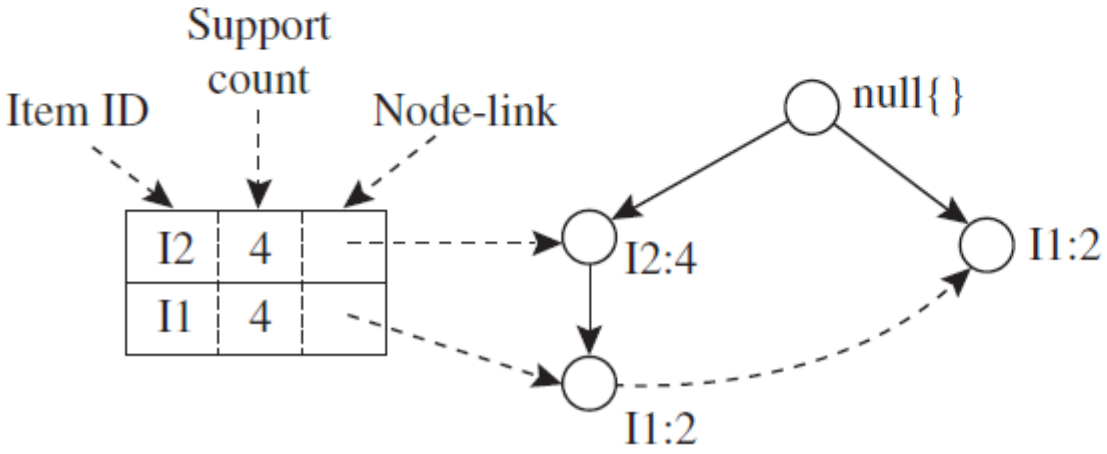
$(I_2, I_3 :4)$

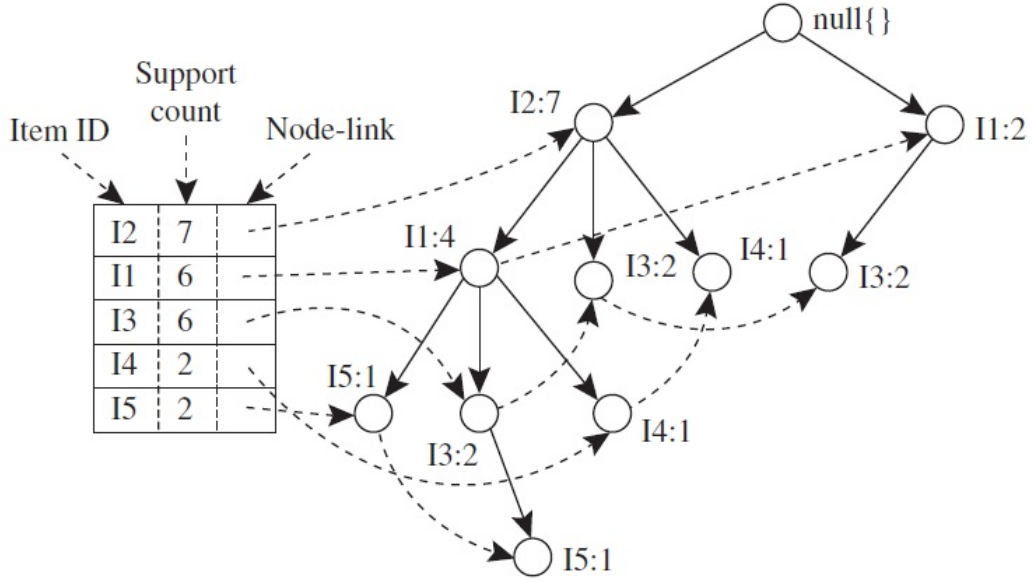
$(I_1, I_3 :4)$

$(I_2, I_1, I_3 :2)$

تولید الگوهای پر تکرار

Item	*Conditional Pattern Base*	*Conditional FP-tree*	*Frequent Patterns Generated*
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}





الگوریتم استفاده از قالب داده های عمودی

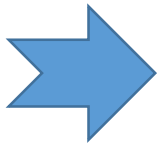
- گاهی اوقات می توان قالب داده ها را عوض کرد.
- در حالت عادی برای هر تراکنش مجموعه کالا ها را به صورت سطری در مقابل آن داریم که به آن چینش افقی (Horizontal) گوییم.
- مقابل آن قالب عمودی (Vertical) داده ها به صورت زیر است:

itemset	*TID_set*
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

الگوریتم استفاده از قالب داده های عمودی

- برای مشخص کردن آیتم های پرتکرار کافی است تعداد عناصر هر مجموعه را بدست آوریم (support_Count برابر است با تعداد عناصر هر مجموعه)
- برای ساخت 2-Itemsets کافی است دو مجموعه 1-Itemset را با هم ترکیب کنیم (اشتراک بگیریم).

itemset	*TID_set*
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

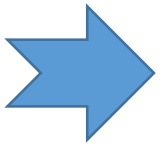


itemset	*TID_set*
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}

الگوریتم استفاده از قالب داده های عمودی

- برای ساخت 3-Itemsets کافی است دو مجموعه های 2-Itemset را با هم ترکیب کنیم (اشتراک بگیریم).

itemset	*TID_set*
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}



itemset	*TID_set*
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

تمرین

- مجموعه داده ای زیر را در نظر بگیرید:
- به روشهای گفته شده، الگوهای پر تکرار را برای این مجموعه پیدا کنید.
- چند تا از قوانین قوی را برای این الگوها با فرض Min_sup = 40% و Min_Conf = 60% بنویسید.

TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y }
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

روش های ارزیابی قوانین وابستگی (انجمنی)

- تا بدین جا فقط تعریف قوانین قوی مطرح شده است. ولی در بسیاری از مواقع قوانین قوی حتما جذاب نیستند. مخصوصاً زمانیکه Min_Sup و Min_Conf کم باشند.
- بنابراین به روشهای ارزیابی مناسبتری برای قوانین نیاز داریم.
- به مثال زیر توجه کنید:

فرض کنید مجموعه ای شامل ۱۰۰۰۰ تراکنش باشد. که ٦۰۰۰ تراکنش شامل بازی های کامپیوتری، ۷۵۰۰ تراکنش شامل فیلم ویدئویی و ٤۰۰۰ تراکنش شامل هر دو باشند.

فرض کنید Min_Sup = 30% و Min_Conf = 60% باشد.

آنگاه قانون زیر:

Computer Game ⇒ Video

دارای $support = 40\%$ و $Confidence = 66\%$ خواهد بود

بنابراین این قانون یک قانون قوی است. ولی آیا قانون کاربردی و جذاب است؟

روش های ارزیابی قوانین وابستگی (انجمنی)

- این قانون گمراه کننده است
- زیرا احتمال خرید فیلم ویدئویی ۷۵٪ است که از ٦٦٪ بیشتر است. در واقع خرید این دو محصول با هم رابطه عکس دارند.
- بنابراین خرید یک محصول، احتمال خرید محصول دیگر را کاهش میدهد.
- بدون دانستن این موضوع یک تصمیم اقتصادی نادرست ممکن است گرفته شود.

- برای بهبود معیارهای ارزیابی می توان یک معیار دیگر را نیز اضافه کردو آن آنالیز همبستگی یا Correlation Analysis است

$$A \Rightarrow B\ [support, confidence, correlation].$$

روش های ارزیابی قوانین وابستگی (انجمنی)

$$A \Rightarrow B \; [support, confidence, correlation].$$

- انواع معیار های همبستگی عبارتند از:
- Lift
- χ^2
- All confidence
- Cosine

مفهوم استقلال آماری

فرض کنید ۱۰۰۰ دانشجو داریم

- ۶۰۰ دانشجو ورزش شنا می کنند (S)
- ۷۰۰ دانشجو فوتبال بازی می کنند (B)
- ۴۲۰ دانشجو هم شنا می کنند و هم فوتبال بازی می کنند(S,B)

$P(S \cup B) = 420 / 1000 = 0.42$

$P(S) = 0.6 \quad P(B) = 0.7$

$P(S) \times P(B) = 0.42$

$P(S \cup B) = P(S) \times P(B) \Rightarrow$ Independent

$P(S \cup B) > P(S) \times P(B) \Rightarrow$ Positive Correlation

$P(S \cup B) < P(S) \times P(B) \Rightarrow$ Negative Correlation

معیار همبستگی Lift

معیار Lift با استفاده از رابطه زیر تعریف می شود:

$$\text{lift(A, B)}\ \frac{P(A \cup B)}{P(A) \times P(B)}$$

- اگر مقدار lift کمتر از ۱ باشد آنگاه دو مجموعه (آیتم) همبستگی منفی دارند.
- اگر مقدار lift بیشتر از ۱ باشد آنگاه دو مجموعه (آیتم) همبستگی مثبت دارند.
- اگر مقدار lift برابر ۱ باشد آنگاه دو مجموعه (آیتم) همبستگی ندارند (مستقل هستند).

معیار همبستگی Lift

در مثال قبل:

$$P(\{game\}) = 0.6$$

$$P(\{Video\}) = 0.75$$

$$P(\{game, video\}) = 0.4$$

$$lift(game, video) = \frac{P(\{game, video\})}{P(\{game\}) \times P(\{video\})} = \frac{0.4}{0.6 \times 0.75} = 0.89$$

بنابراین این دو آیتم همبستگی منفی دارند.

Contingency Table

	game	$\overline{game}$	Σ_{row}
video	4000	3500	7500
$\overline{video}$	2000	500	2500
Σ_{col}	6000	4000	10,000

دسته بندی (طبقه بندی)
Classification

- هدف دسته بندی استخراج مدل هایی است که بتواند کلاس های داده ها را مشخص کند.
- این مدل ها دو دسته هستند:
  - x) طبقه بندی (Classification)
  - ц) پیش بینی (Prediction)
- بعضی از کاربردهای این بحث:
  - x) دسته بندی متن (Text classification)
  - ц) تشخیص تقلب (Fraud detection)
  - ц) تشخیص پزشکی (Medical diagnosis)
  - љ) پیش بینی بازار بورس (Stock market prediction)
  - њ) طبقه بندی تصاویر (Image classification)
  - a) دسته بندی صفحات وب (Web page classification)

دسته بندی چیست؟

- مثال۱: فرض کنید مدیر یک بانک تصمیم می گیرد بر اساس داده های مربوط به مشتریان بانک و تحلیل این داده ها، مشخص کند اعطای وام به کدامیک از مشتریان امن (Safe) و به کدامیک دارای ریسک (Risky) است.
- مثال ۲: یک نرم افزار سرویس دهنده ایمیل می خواهد برای هر ایمیل ورودی مشخص کند که آیا این ایمیل یک هرز نامه (Spam) هست و یا یک ایمیل سالم (Not Spam).
- مثال ۳: یک پزشک متخصص می خواهد با تحلیل داده های برگرفته شده از آزمایشات بیمار تشخیص دهد که آیا این بیمار دچار سرطان هست(Yes) یا نه(No)؟
- در هر سه مثال نیاز به مدلی داریم که بتواند برچسب (Label) مناسبی برای داده های مورد ارزیابی انتخاب کند:

✓ در مثال ۱: Risky or Safe

✓ در مثال ۲: Spam or Not Spam

✓ در مثال ۳: Yes or No

چگونه دسته بندی انجام می شود؟

- فرایند دسته بندی می تواند در دو فاز انجام گیرد:

x. فاز آموزش یا یادگیری (Learning Step): که در این فاز یک مدل طبقه بند (Classifier) بر اساس داده های موجود ساخته می شود.

ц. فاز دسته بندی (Classification Step): که در این فاز از مدل ساخته شده برای تشخیص بر چسب (Label) مناسب برای داده های جدید استفاده می شود.

فاز اول: Learning Step

در این فاز بخشی از داده های پایگاه داده ها همراه با برچسب (Label) مربوطه به منظور آموزش مدل دسته بندی مورد استفاده قرار می گیرد. این داده ها باید برچسب مشخصی داشته باشند که اصطلاحا به آنها داده های برچسب گذاری شده (Labelled examples) گویند.

به مجموعه داده های برچسب گذاری شده ای که به منظور آموزش مدل دسته بند مورد استفاده قرار می گیرند مجموعه آموزشی (Training Set) گویند. این مجموعه شامل رکورد هایی از پایگاه داده هستند که هر کدام از این رکورد ها توسط یک بردار n بعدی نمایش داده می شود:

$$X = (x_1, x_2, x_3, \dots, x_n)$$

هر کدام از این مقادیر مربوط به یک ویژگی (Feature or Attribute) در مجموعه داده هاست. این ویژگی ها را با $A_1, A_2, A_3, \dots, A_n$ نشان می دهیم.

برای هر نمونه از این مجموعه داده با یک ویژگی خاص که به آن بر چسب گفته می شود به یک کلاس خاص تعلق می گیرد.

نکته: در بحث دسته بندی اصطلاحات زیر برای هر رکورد اطلاعات میتواند مورد استفاده قرار گیرد:

Sample, Example, instance, Data point, object

فاز اول: Learning Step

- بدلیل اینکه برچسب داده های آموزشی در مرحله یادگیری مشخص است، به این شیوه آموزشی روش آموزش با نظارت (Supervised learning) گویند.

name	*age*	*income*	*loan_decision*
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

یادگیری با نظارت در مقابل یادگیری بدون نظارت

- یادگیری با نظارت
  - ✓داده های آموزشی همراه با برچسب هستند که مشخص کننده کلاس آنهاست.
  - ✓داده های جدید بر اساس یادگیری از روی داده های برچسب دار(labelled) دسته بندی می شوند.
- یادگیری بدون نظارت (Unsupervised learning or Clustering)
  - ✓برای نمونه های داده ها برچسب کلاس مشخص نیست.
  - ✓معمولا تعداد و نوع کلاس ها هم مشخص نیست.

فاز اول: Learning Step

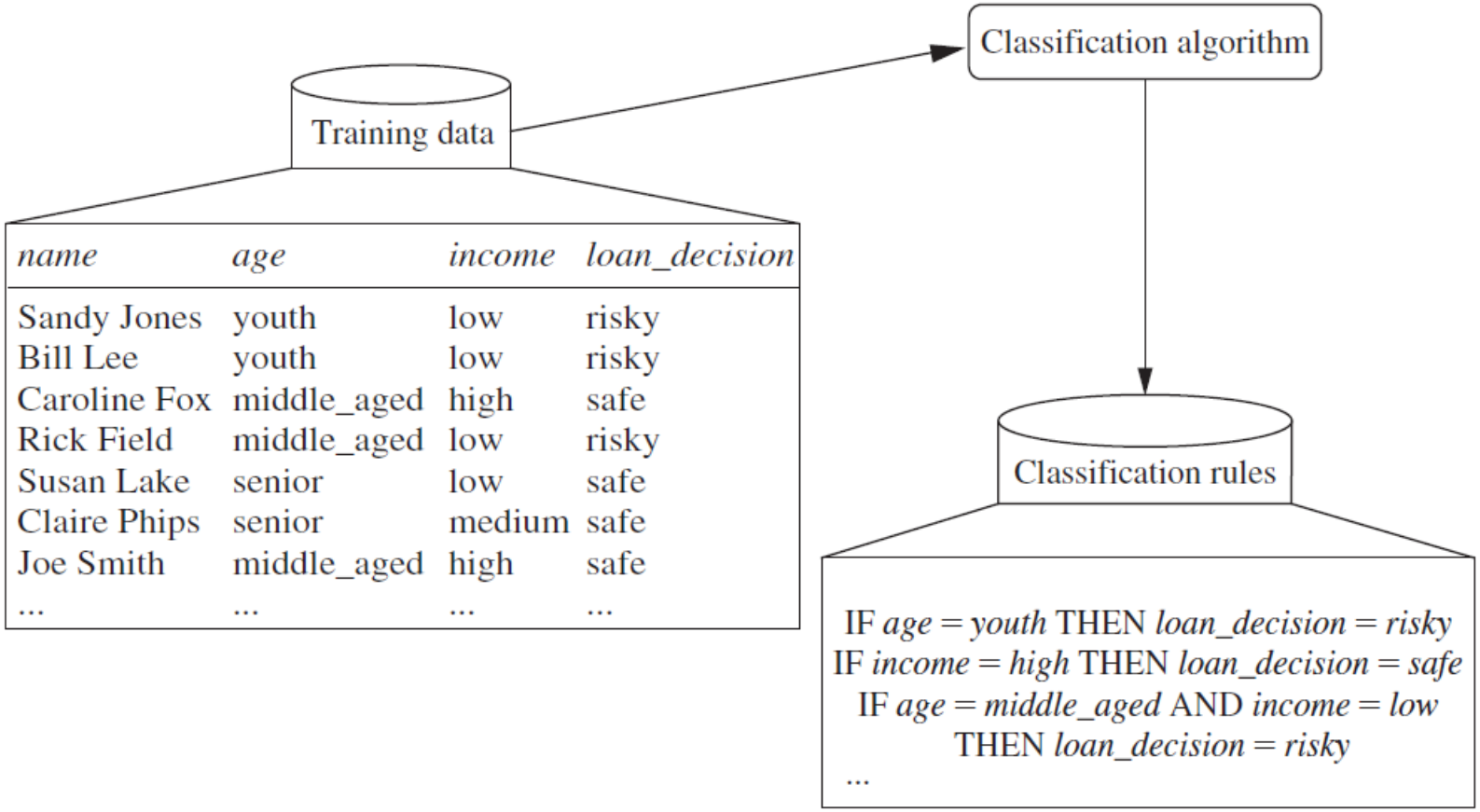
- اولین مرحله از فرایند دسته بندی می تواند به عنوان یادگیری یک نگاشت تابع $y = f(X)$ تلقی شود که می خواهد برچسب y را برای رکورد X پیش بینی کند.
- این نگاشت یا تابع معمولا به شکل های
  - قوانین طبقه بندی(Classification Rules)
  - درخت های تصمیم (Decision Tree)
  - مدل های ریاضی (Mathematical models)

نشان داده می شوند.

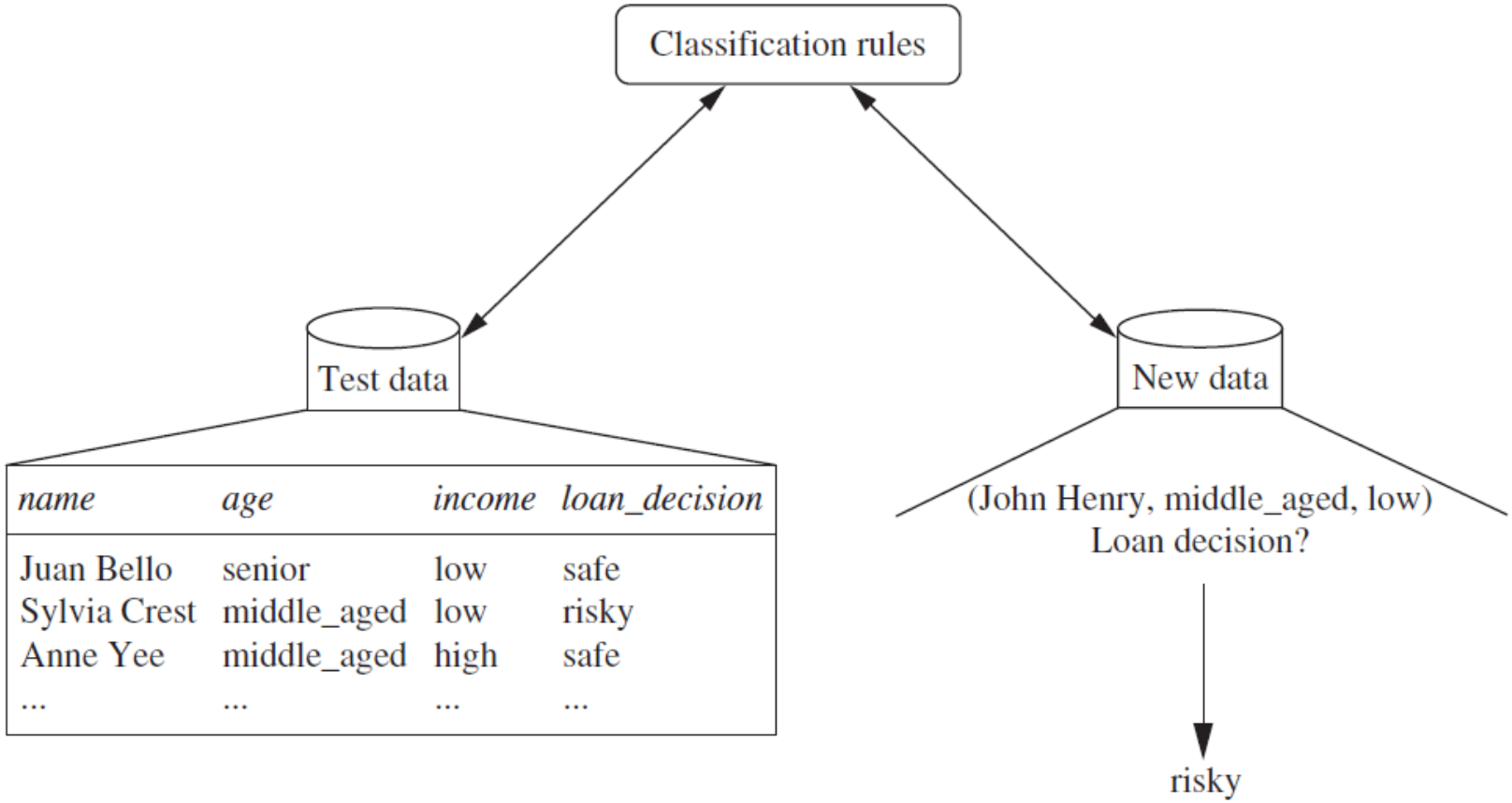
فاز دوم: Classification Step

- تا این مرحله مدل (Classifier) ساخته شد.
- در این مرحله باید مدل ساخته شده مورد ارزیابی قرار گیرد.
- برای ارزیابی مدل از یک مجموعه تست (Test set) استفاده می شود که از مجموعه آموزشی کاملا مستقل است.
- تعریف دقت (Accuracy): درصدی از مجموعه تست که توسط Classifier به درستی دسته بندی شده است.
- زمانی که صحت Classifier تائید شد می توان از آن برای دسته بندی داده های جدید بدون برچسب از آن استفاده کرد

Training Step :مثال

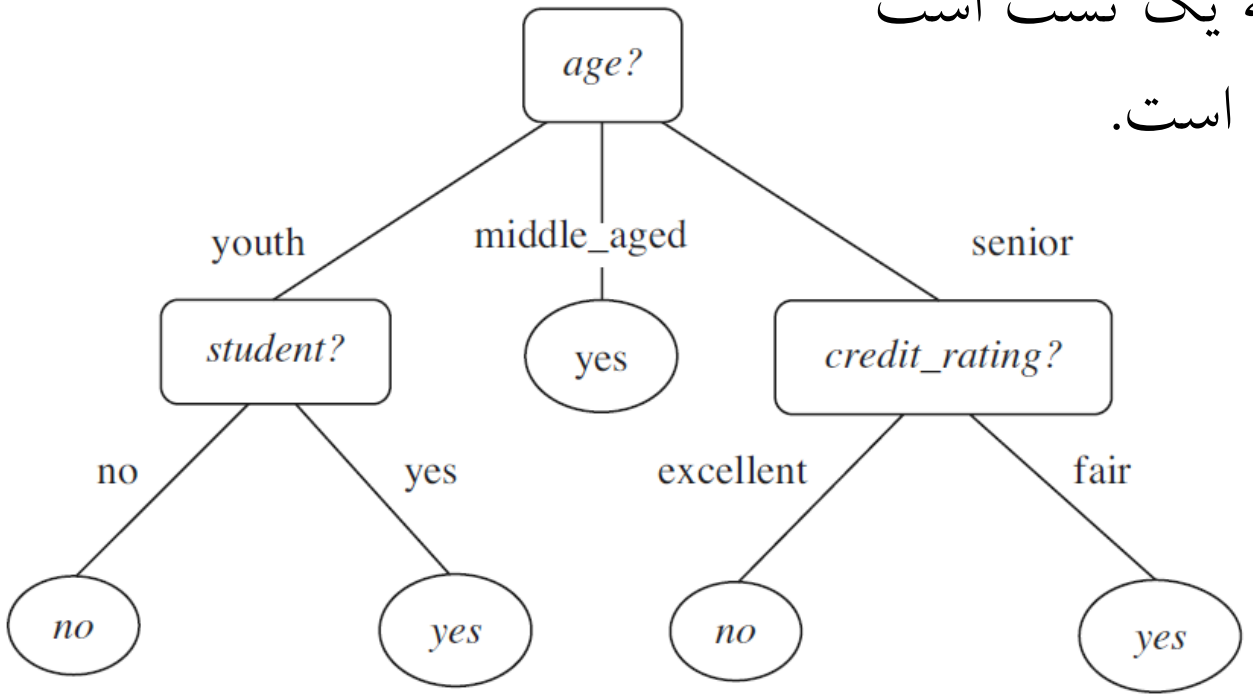


مثال: Test Step



درخت تصمیم Decision Tree

- درخت تصمیم یک ساختار درختی فلو چارت مانند است که در آن:
  - هر گره غیر پایانی (غیر برگ) یک تست بر روی یک ویژگی انجام می دهد.
  - هر شاخه (یال) نشان دهنده نتیجه یک تست است
  - هر گره برگ یک برچسب کلاس است.

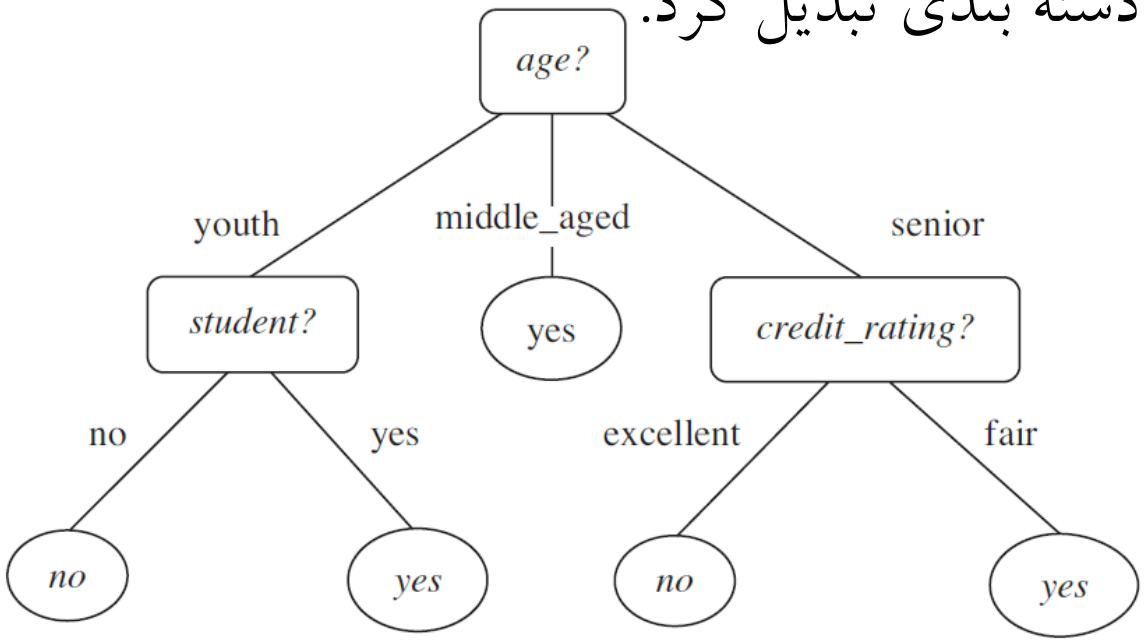


درخت تصمیم چگونه استفاده می شود؟

برای یک نمونه داده جدید که بر چسب کلاس آن نا مشخص است، بر اساس مقادیر ویژگی های آن یک مسیر از ریشه به برگ پیدا می کنیم.

برچسب برگ نشان دهنده کلاس پیش بینی شده برای آن نمونه داده است.

درخت تصمیم را به راحتی می توان به قوانین دسته بندی تبدیل کرد.



چرا درخت تصمیم محبوب است؟

X. ایجاد درخت تصمیم به هیچ دانش تخصصی و یا تنظیم پارامتری نیاز ندارد.

Ц. این درخت می تواند داده های با ابعاد زیاد (تعداد زیاد ویژگیها) را پوشش دهد.

Ц. نمایش دانش بدست آمده از درخت تصمیم قابل فهم برای انسان است.

Лъ. آموزش و دسته بندی با استفاده از این درخت ساده و سریع است.

Нь. دسته بندی با این مدل معمولا با دقت مناسبی انجام می گیرد.

a. الگوریتم درخت تصمیم در بسیاری از زمینه های کاربردی برای دسته بندی قابل استفاده است.

مشکلات الگوریتم ایجاد درخت تصمیم

- مشکل۱: کدام ویژگی ها در هر سطح درخت تصمیم استفاده شوند؟
- مشکل۲: بعد از ساخت درخت برخی از شاخه ها حاوی داده های نویز و یا پرت هستند.
- مشکل۳: چگونه این شاخه ها را پیدا کرده و هرس کنیم (Tree pruning)

الگوریتم های ساخت درخت تصمیم

- الگوریتم ID3
- الگوریتم C4.5
- الگوریتم CART (Classification And Regression Tree)

این الگوریتم ها از روش حریصانه برای ساخت درخت استفاده می کنند

روش کار آنها بالا به پایین و بر اساس استراتژی تقسیم و حل است.

الگوریتم درخت تصمیم

- الگوریتم پایه:
- درخت به صورت بازگشتی با روش تقسیم و حل، از بالا به پایین ساخته می‌شود.
- در آغاز همه نمونه های آموزشی در گره هستند
- ویژگیهای داده ها چند مقداری هستند(اگر مقادیر پیوسته باشند، گسسته می شوند)
- داده ها به صورت بازگشتی و بر اساس ویژگیهای انتخاب شده قسمت بندی می شوند.
- ویژگی های آزمون بر اساس معیارهای اکتشافی و یا آماری انتخاب می شوند

الگوریتم درخت تصمیم

- شرایط برای توقف قسمت بندی:

X. همه نمونه ها برای یک گره متعلق به یک کلاس باشند.

Ч. هیچ ویژگی برای قسمت بندی بیشتر وجود نداشته باشد (در این شرایط برچسب اکثریت نمونه های برگ به عنوان برچسب برگ انتخاب می شود).

Ч. هیچ نمونه بیشتری وجود نداشته باشد.

استنتاج بالا به پائین درخت تصمیم
ID3 (Iterative Dichotomiser 3)

1. فرض کنید بهترین ویژگی تصمیم برای گره بعد A باشد.
2. A را به عنوان ویژگی تصمیم برای گره قرار بده.
3. برای هر مقدار از A یک فرزند جدید ایجاد کن
4. مرتب سازی نمونه‌های آموزشی برای گره برگ با توجه به مقدار ویژگی شاخه
5. اگر همه نمونه‌های آموزشی کاملا طبقه بندی شده باشند(همان مقدار ویژگی هدف)، متوقف شو. در غیر اینصورت برای گره های جدید برگ تکرار کن

معیار انتخاب ویژگی Information Gain (IG)

- این معیار در ID3 و C4.5 به کار گرفته شده است.
- در هر گام از الگوریتم ساخت درخت تصمیم ویژگی با بالاترین IG برای بخش بندی انتخاب می شود.
- در صورتی که P_i احتمال اینکه یک رکورد در مجموعه داده D متعلق به کلاس C_i توسط $\frac{|C_{i,D}|}{|D_i|}$ تخمین زده شود، میزان اطلاعات مورد انتظار (آنتروپی) برای دسته بندی یک نمونه داده در مجموعه داده های D برابر است با:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i),$$

معیار انتخاب ویژگی Information Gain (IG)

- حال فرض کنید که می خواهیم مجموعه داده های D را با استفاده از یکی از ویژگی ها (مثلا A_i) که دارای v مقادیر مشخص $\{a_1,a_2,a_3, \dots ,a_v\}$ است بخش بندی کنیم. ویژگی A_i می تواند برای بخش بندی مجموعه داده های D به v زیر مجموعه مورد استفاده قرار گیرد.
- میزان اطلاعات مورد نیاز بعد از بخش بندی داده ها (با استفاده از ویژگی A_i به v بخش مجزا) برای رسیدن به دسته بندی نهایی از رابطه زیر بدست می آید:

$$Info_{A_i}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j).$$

- بنابراین IG بدست آمده بوسیله انشعاب بر روی ویژگی A_i از رابطه زیر بدست می آید:

$$IG\ (A_i) = info(D) - info_{A_i}(D)$$

مثال: پیش بینی خرید کامپیوتر توسط مشتریان

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Class 1: Yes

Class 2: No

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right)$$
$$+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right)$$
$$+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$
$$= 0.694 \text{ bits.}$$

$$IG(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246$$

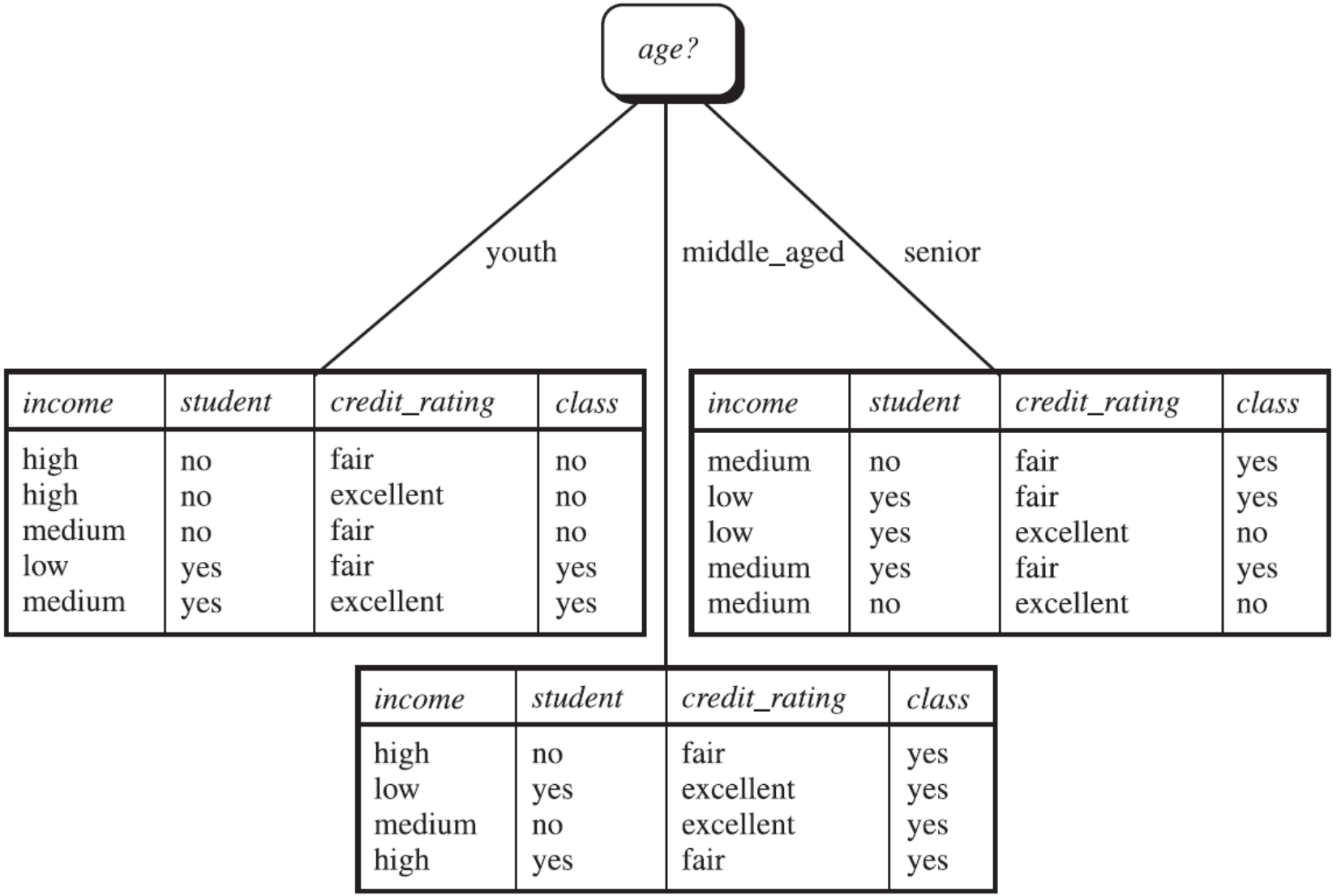
ادامه مثال

این عمل را برای ویژگی های دیگر
نیز انجام میدهیم.

$$IG(income) = 0.029$$
$$IG(student) = 0.151$$
$$IG(credi - rating) = 0.048$$

با توجه به مقادیر بدست آمده، بالاترین Information Gain را ویژگی age بدست می آورد و بنابراین به عنوان ویژگی بخش بندی کننده داده ها انتخاب می شود.

خروجی اولین مرحله

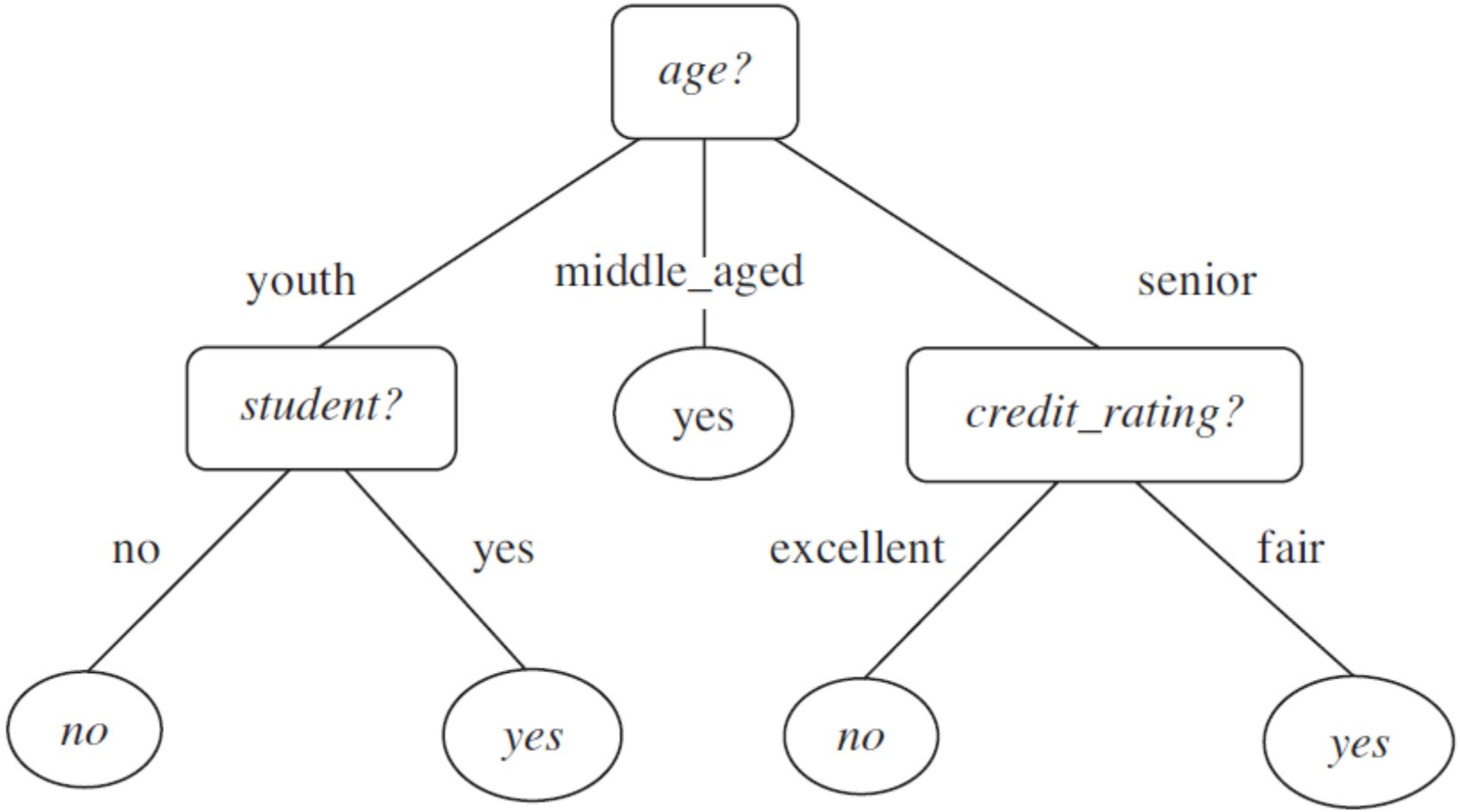


income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

ادامه الگوریتم



محاسبه Information Gain برای مقادیر پیوسته

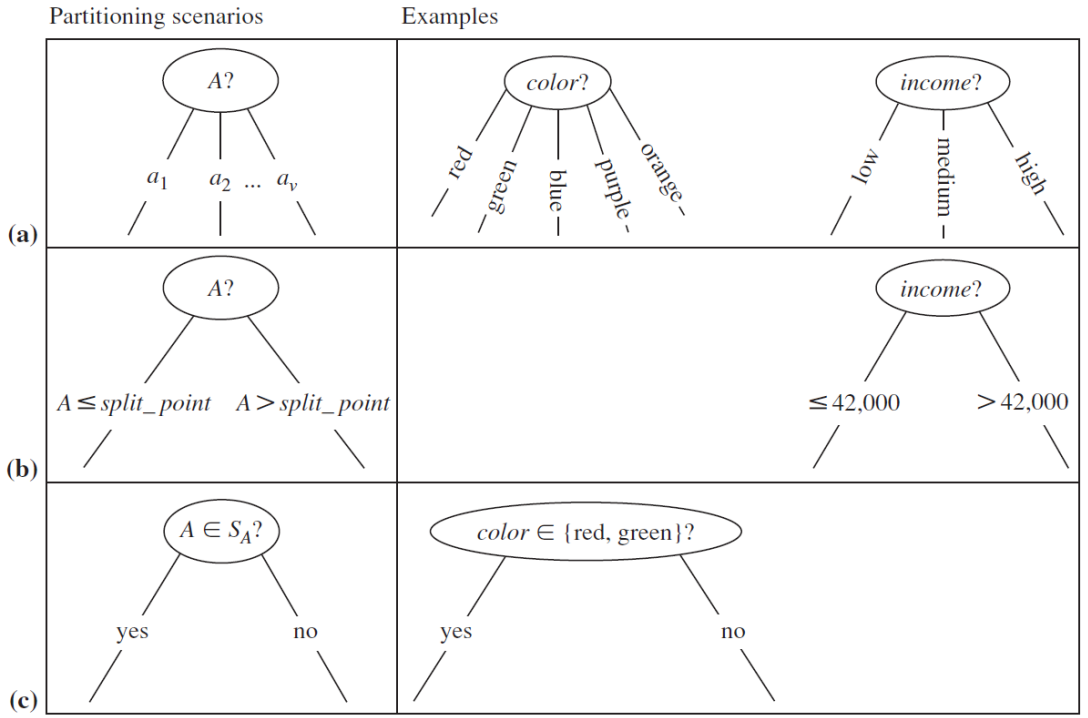
- فرض کنید ویژگی A یک ویژگی با مقدار پیوسته باشد. مثلا بجای مقادیر گسسته برای ویژگی age مقادیر واقعی سن افراد ذخیره شده باشد.
- برای چنین ویژگی هایی مثل A باید بهترین نقطه تقسیم (Split Point) را تعیین کنیم. برای این منظور:
  - مقادیر A را به صورت صعودی مرتب می کنیم
  - معمولا نقطه وسط بین هر جفت مقادیر مجاور می تواند به عنوان نقطه تقسیم در نظر گرفته شود. با فرض v مقدار برای A آنگاه v-1 نقطه تقسیم می تواند ارزیابی شود.
  - مقدار نقطه تقسیم برای دو مقدار a_i و a_{i+1} از ویژگی A از فرمول زیر بدست می آید.

$$\frac{a_i + a_{i+1}}{2}$$

  - نقطه ای که کمترین میزان اطلاعات مورد نیاز (Expected information requirement) را داشته باشد به عنوان نقطه تقسیم در نظر گرفته می شود.

محاسبه Information Gain برای مقادیر پیوسته

- تقسیم :
  - D_1 مجموعه ای از رکورد ها در D است که مقدار ویژگی A در آنها کمتر یا مساوی نقطه تقسیم است.
  - D_2 مجموعه ای از رکورد ها در D است که مقدار ویژگی A در آنها بیشتر از نقطه تقسیم است.



دسته بندی بیزین Bayesian Classifiers

- این روش یک روش دسته بندی آماری Statistical classifier است که احتمال تعلق یک رکورد داده ای را به یک کلاس محاسبه می کند.
- اساس این روش بر مبنای قضیه بیز بنا شده است.
- این روش عملکرد قابل قبولی در مقایسه با بقیه روش ها دارد. از لحاظ سرعت هم عملکرد قابل قبولی دارد.
- ساده ترین مدل این روش بنام Naïve Bayes Classifier شناخته می شود.
- در این روش فرض می شود میزان تاثیر مقادیر یک ویژگی از مقادیر سایر ویژگی ها مستقل است که این فرض با نام Class Conditional Independence شناخته میشود.
- این فرضیه باعث می شود که محاسبات کمتر شود.

تئوری بیز (Bayesian theory)

- فرض کنید xیک مشاهده (evidence) باشد.
- در مجموعه داده‌ی مورد بررسی، x با تعدادی ویژگی نشان داده می شود.
- در نظر بگیرید که H یک فرضیه است که تعلق x را به یک کلاس خاص مثل C را نشان می دهد.
- برای اعمال دسته بندی نیاز داریم P(H|x) را بسنجیم. یعنی اینکه اگر x را داشته باشیم H درست است.
- در واقع می خواهیم احتمال تعلق نمونه داده x را به کلاس C در صورت دانستن ویژگی های x بسنجیم.

تئوری بیز (Bayesian theory)

- فرمول بیز به این صورت است:

$$P(H|x) = \frac{P(x|H) \times P(H)}{P(x)}$$

- به $P(H|x)$ احتمال پسین یا Posterior Probability گویند.
- در مقابل مقادیر $P(H)$ و $P(x)$ را از روی داده های آموزشی می دانیم و به آنها احتمالات پیشین یا Prior Probabilities گویند.
- به $P(x|H)$ که احتمال مشاهده نمونه x به شرط در نظر گرفتن فرضیه H است اصطلاحا Likelihood گویند.

تئوری بیز (Bayesian theory)

- فرمول بیز به این صورت است:

$$P(H|x) = \frac{P(x|H) \times P(H)}{P(x)}$$

- به $P(H|x)$ احتمال پسین یا Posterior Probability گویند.
- در مقابل مقادیر $P(H)$ و $P(x)$ را از روی داده های آموزشی می دانیم و به آنها احتمالات پیشین یا Prior Probabilities گویند.
- به $P(x|H)$ که احتمال مشاهده نمونه x به شرط در نظر گرفتن فرضیه H است اصطلاحا Likelihood گویند.
- به عنوان مثال: اگر بدانیم که مشتری x کامپیوتر خواهد خرید، احتمال اینکه x دارای درآمد متوسط باشد چقدر است؟

مثالی از تئوری بیز

- دکتر می داند که منژیت در ۵۰٪ مواقع باعث خشکی گردن می شود. اگر منژیت را H و خشکی گردن را x در نظر بگیریم داریم $P(x|H) = 0.5$ که همان likelihood است.
- فرض کنید احتمال اینکه یک بیمار منژیت داشته باشد برابر $\frac{1}{50000}$ باشد.
  بنابر این $P(H) = 0.00002$ که همان Prior probability است.
- فرض کنید احتمال اینکه یک بیمار خشکی گردن داشته باشد ۵٪ باشد بنابراین داریم
  $P(x) = 0.05$.
- سوال: اگر بیماری دارای خشکی گردن باشد، احتمال اینکه منژیت داشته باشد چقدر است؟

$$P(H|x) = \frac{P(x|H) \times P(H)}{P(x)} = \frac{0.5 \times 0.00002}{0.05} = 0.0002 = 0.02\%$$

Naïve Bayes Classification

- فرض کنید D مجموعه داده های آموزشی باشد که دارای مجموعه ویژگی های با تعداد n ویژگی $A_1, A_2, \ldots, A_n$ می باشد.
- فرض کنید m کلاس با مقادیر $C_1, C_2, \ldots, C_m$ وجود دارد.
- با فرض وجود یک نمونه داده $X = (x_1, x_2, \ldots, x_n)$ می خواهیم کلاسی را برای X پیش بینی کنیم که بیشترین مقدار احتمال پسین (posterior probability) را داشته باشد.
- در واقع Naïve Bayes Classifier پیش بینی می کند که نمونه داده X متعلق به کلاس C_i است اگر و فقط اگر $P(C_i|\boldsymbol{X}) > P(C_j|\boldsymbol{X}) \quad \text{for } 1 \leq j \leq m, j \neq i.$

Naïve Bayes Classification

- بنابر این ما به دنبال کلاس C_i ی هستیم که بیشترین مقدار $P(C_i|X)$ را داشته باشد. به کلاسی که بیشترین مقدار احتمال پسین را داشته باشد *Maximum posteriori hypothesis* گویند.
- بر اساس تئوری بیز داریم:

$$P(C_i|X) = \frac{P(X|C_i) \times P(C_i)}{P(X)}$$

- با توجه به اینکه $P(X)$ برای تمام کلاس ها یکسان و ثابت است، بنا بر این فقط مقدار عبارت $P(X|C_i) \times P(C_i)$ باید ماکزیمم شود.
- اگر تعداد ویژگی های مجموعه داده ها زیاد باشد محاسبه $P(X|C_i)$ خیلی هزینه بر است.

Naïve Bayes Classification

- برای کاهش محاسبات، Naïve فرض می کند که همه ویژگی ها مستقل از هم هستند بنابر این داریم:

$$P(\boldsymbol{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$
$$= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i).$$

که مقادیر $P(x_1|C_i)$ تا $P(x_n|C_i)$ به راحتی از روی مجموعه داده های آموزشی (Training Set) قابل محاسبه هستند.

توجه: x_k معرف مقدار مربوط به ویژگی A_k برای نمونه داده X است.

Naïve Bayes Classification

- برای محاسبه $P(X|C_i)$ بر اساس نوع ویژگی ها به صورت زیر عمل می کنیم

a) اگر A_k از نوع مقادیر گسسته و مجزا (Categorical) باشد، آنگاه $P(x_k|C_i)$ برابر است با تعداد نمونه های مجموعه داده های آموزشی D که دارای کلاس C_i هستند و مقدار ویژگی A_k برای آنها برابر x_k است تقسیم بر تعداد کل نمونه هایی که دارای کلاس C_i هستند.

b) اگر A_k از نوع مقادیر پیوسته باشد] نیاز به محاسبات بیشتری است. یک مقدار پیوسته معمولا از یک توزیع گوسی با میانگین μ و انحراف معیار σ پیروی می کند که با فرمول زیر تعریف می شود:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

در نتیجه داریم:

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

مثال برای Naïve Bayes Classifier

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

در این مثال داریم:

C_1 = buys_computer = yes

C_2 = buys_computer = no

می خواهیم برای نمونه داده زیر یک کلاس پیش بینی کنیم

$$X = (age = youth, income = medium, student = yes, credit\_rating = fair)$$

مثال برای Naïve Bayes Classifier

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$X = (age = youth,\ income = medium,\ student = yes,\ credit\_rating = fair)$$

ابتدا مقادیر $P(C_i)$ ها را محاسبه می کنیم:

$$P(buys\_computer = yes) = 9/14 = 0.643$$
$$P(buys\_computer = no) = 5/14 = 0.357$$

برای محاسبه $P(X|C_i)$ برای کلاس های

۱و۲ (i=1,2) احتمالات شرطی زیر باید محاسبه شود

$$P(age = youth \mid buys\_computer = yes) = 2/9 = 0.222$$
$$P(age = youth \mid buys\_computer = no) = 3/5 = 0.600$$
$$P(income = medium \mid buys\_computer = yes) = 4/9 = 0.444$$
$$P(income = medium \mid buys\_computer = no) = 2/5 = 0.400$$
$$P(student = yes \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(student = yes \mid buys\_computer = no) = 1/5 = 0.200$$
$$P(credit\_rating = fair \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(credit\_rating = fair \mid buys\_computer = no) = 2/5 = 0.400$$

مثال برای Naïve Bayes Classifier

$\boldsymbol{X} = (age = youth, income = medium, student = yes, credit\_rating = fair)$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$
\begin{aligned}
P(age = youth \mid buys\_computer = yes) &= 2/9 = 0.222\\
P(age = youth \mid buys\_computer = no) &= 3/5 = 0.600\\
P(income = medium \mid buys\_computer = yes) &= 4/9 = 0.444\\
P(income = medium \mid buys\_computer = no) &= 2/5 = 0.400\\
P(student = yes \mid buys\_computer = yes) &= 6/9 = 0.667\\
P(student = yes \mid buys\_computer = no) &= 1/5 = 0.200\\
P(credit\_rating = fair \mid buys\_computer = yes) &= 6/9 = 0.667\\
P(credit\_rating = fair \mid buys\_computer = no) &= 2/5 = 0.400
\end{aligned}
$$

با توجه به مقادیر محاسبه شده داریم:

$$
\begin{aligned}
P(\boldsymbol{X}|buys\_computer = yes) = {} & P(age = youth \mid buys\_computer = yes)\\
& \times P(income = medium \mid buys\_computer = yes)\\
& \times P(student = yes \mid buys\_computer = yes)\\
& \times P(credit\_rating = fair \mid buys\_computer = yes)\\
= {} & 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044.
\end{aligned}
$$

و به صورت مشابه:

$$P(\boldsymbol{X}|buys\_computer = no) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$$

برای یافتن کلاس مناسب برای X که مقدار $P(X|C_i) \times P(C_i)$ را ماکزیمم می کند مقادیر زیر محاسبه می شوند:

$$
\begin{aligned}
P(\boldsymbol{X}|buys\_computer = yes)P(buys\_computer = yes) &= 0.044 \times 0.643 = 0.028\\
P(\boldsymbol{X}|buys\_computer = no)P(buys\_computer = no) &= 0.019 \times 0.357 = 0.007
\end{aligned}
$$

بنابراین Naïve Bayes Classifier برای نمونه داده X پیش بینی buys_computer = yes را دارد.