Методы поиска ассоциативных правил

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Машинное обучение (курс лекций, К.В.Воронцов)»

октябрь 2012

Содержание

Определения и обозначения

X — пространство объектов;
$\mathscr{F} = \{f_1, \ldots, f_n\}$, $f_j : X \to \{0, 1\}$ — бинарные признаки (items);
$X^\ell = \{x_1, \ldots, x_\ell\} \subset X$ — обучающая выборка.

Каждому подмножеству $\varphi \subseteq \mathscr{F}$ соответствует конъюнкция

$$\varphi(x) = \bigwedge_{f \in \varphi} f(x), \quad x \in X.$$

Если $\varphi(x) = 1$, то «признаки из φ *совместно встречаются* у x».

Частота встречаемости (*поддержка*, support) φ в выборке X^ℓ

$$\nu(\varphi) = \frac{1}{\ell} \sum_{i=1}^{\ell} \varphi(x_i).$$

Если $\nu(\varphi) \geqslant \delta$, то «набор φ *частый*» (frequent itemset).
Параметр δ — *минимальная поддержка*, `MinSupp`.

Определения и обозначения

Определение

Ассоциативное правило (association rule) $\varphi \to y$ — *это пара непересекающихся наборов* $\varphi, y \subseteq \mathscr{F}$ *таких, что:*
1) наборы φ *и* y *совместно часто встречаются,*

$$\nu(\varphi \cup y) \geqslant \delta;$$

2) если встречается φ*, то часто встречается также и* y*,*

$$\nu(y|\varphi) \equiv \frac{\nu(\varphi \cup y)}{\nu(\varphi)} \geqslant \varkappa.$$

$\nu(y|\varphi)$ — *значимость* (confidence) правила.

Параметр δ — *минимальная поддержка*, `MinSupp`.
Параметр $\varkappa$ — *минимальная значимость*, `MinConf`.

Классический пример

Анализ рыночных корзин (market basket analysis) [1993]
признаки — товары (предметы, items)
объекты — чеки (транзакции)

$f_j(x_i) = 1$ — в i-м чеке зафиксирована покупка j-го товара.

Пример: «если куплен хлеб φ, то будет куплено и молоко y с вероятностью $\nu(y|\varphi) = 60\%$; причём оба товара покупаются совместно с вероятностью $\nu(\varphi \cup y) = 2\%$».

Цели анализа:
- оптимизировать размещение товаров на полках,
- формировать персональные рекомендации,
- планировать рекламные кампании (промо-акции),
- более эффективно управлять ценами и ассортиментом.

Ассоциативные правила — это логические закономерности

Определение

Предикат $\varphi(x)$ — логическая ε, δ-закономерность класса $c \in Y$

$$D_c(\varphi, X^\ell) = \frac{p_c(\varphi)}{\ell} \geqslant \delta; \qquad E_c(\varphi, X^\ell) = \frac{n_c(\varphi)}{p_c(\varphi) + n_c(\varphi)} \leqslant \varepsilon,$$

$$p_c(\varphi) = \#\{x_i : \varphi(x_i) = 1 \text{ и } y(x_i) = c\} \quad + \text{примеры класса } c;$$
$$n_c(\varphi) = \#\{x_i : \varphi(x_i) = 1 \text{ и } y(x_i) \neq c\} \quad - \text{примеры класса } c.$$

Для «$\varphi \to y$» возьмём целевой признак $y(x) = \bigwedge\limits_{f \in y} f(x)$. Тогда

$$\nu(\varphi \cup y) \equiv D_1(\varphi) \geqslant \delta; \qquad \frac{\nu(\varphi \cup y)}{\nu(\varphi)} \equiv 1 - E_1(\varphi) \geqslant 1 - \varepsilon \equiv \varkappa.$$

Вывод: различия двух определений — чисто терминологические.

Два этапа построения правил. Свойство антимонотонности

Поскольку $\varphi(x) = \bigwedge\limits_{f \in \varphi} f(x)$ — конъюнкция, имеет место

Свойство антимонотонности:
для любых $\psi, \varphi \subset \mathscr{F}$ из $\varphi \subset \psi$ следует $\nu(\varphi) \geqslant \nu(\psi)$.

Следствия:

1. если ψ частый, то все его подмножества $\varphi \subset \psi$ частые.
2. если φ не частый, то все наборы $\psi \supset \varphi$ также не частые.
3. $\nu(\varphi \cup \psi) \leqslant \nu(\varphi)$ для любых φ, ψ.

Два этапа поиска ассоциативных правил:

1. поиск частых наборов
   (многократный просмотр транзакционной базы данных).
2. выделение ассоциативных правил
   (простая эффективная процедура в оперативной памяти).

Алгоритм APriory (основная идея — поиск в ширину)

Вход: X^ℓ — обучающая выборка;
минимальная поддержка δ; минимальная значимость $\varkappa$;

Выход: $R = \{(\varphi, y)\}$ — список ассоциативных правил;

1: множество всех частых исходных признаков:
$G_1 := \{f \in \mathscr{F} \mid \nu(f) \geqslant \delta\}$;
2: **для всех** $j = 2, \ldots, n$
3: множество всех частых наборов мощности j:
$G_j := \{\varphi \cup \{f\} \mid \varphi \in G_{j-1},\ f \in G_1,\ \nu(\varphi \cup \{f\}) \geqslant \delta\}$;
4: **если** $G_j = \varnothing$ **то**
5: **выход** из цикла по j;
6: $R := \varnothing$;
7: **для всех** $\psi \in G_j,\ j = 2, \ldots, n$
8: AssocRules $(R, \psi, \varnothing)$;

Выделение ассоциативных правил

Этап 2. Простой алгоритм, выполняемый быстро, как правило, полностью в оперативной памяти.

Вход: и **Выход:**

R — список ассоциативных правил;

(φ, y) — ассоциативное правило;

ПРОЦЕДУРА AssocRules (R, φ, y);
для всех $f \in \varphi$
  $\varphi' := \varphi \setminus \{f\}$; $y' := y \cup \{f\}$;
  если $\nu(y'|\varphi') \geqslant \varkappa$ **то**
    добавить ассоциативное правило (φ', y') в список R;
    если $|\varphi'| > 1$ **то**
      AssocRules (φ', y');

Модификации алгоритмов индукции ассоциативных правил

- Более эффективные структуры данных для быстрого поиска частых наборов.
- Самплинг с последующей проверкой правил на полной выборке.
- Иерархические алгоритмы, учитывающие иерархию признаков (например, товарное дерево).
- Учёт времени: инкрементные и декрементные алгоритмы.
- Учёт времени: поиск последовательных шаблонов (sequential pattern).
- Учёт информации о клиентах.

Префиксное FP-дерево (FP — frequent pattern)

В каждой вершине v дерева T задаются:

- признак $f_v \in \mathscr{F}$;
- множество дочерних вершин $S_v \subset T$;
- поддержка $c_v = \nu(\varphi_v)$ набора признаков $\varphi_v = \{f_u : u \in [v_0, v]\}$, где $[v_0, v]$ — путь от корня дерева v_0 до вершины v.

Обозначения:

$V(T, f) = \{v \in T : f_v = f\}$ — все вершины признака f.

$C(T, f) = \sum_{v \in V(T,f)} c_v$ — суммарная поддержка признака f.

Свойства FP-дерева T, построенного по всей выборке X^ℓ:

1. T содержит полную информацию о всех $\nu(\varphi)$, $\varphi \subseteq \mathscr{F}$.
2. $C(T, f) = \nu(f)$ для всех $f \in \mathscr{F}$.

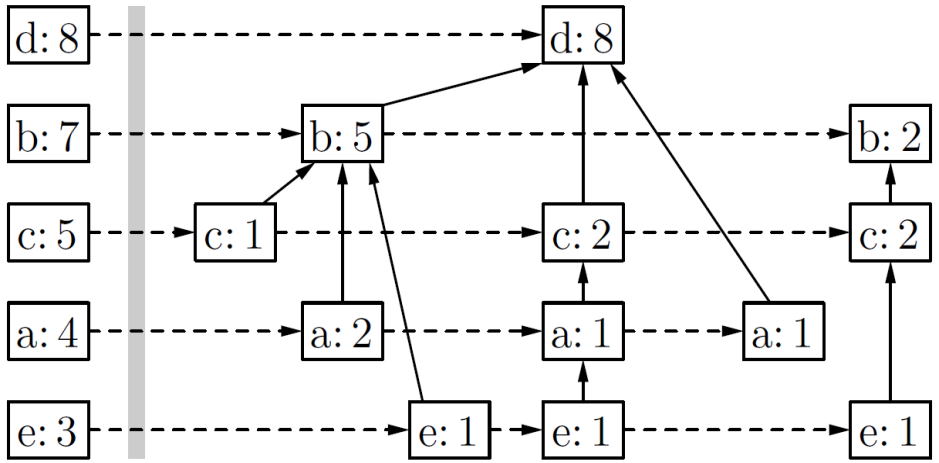
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

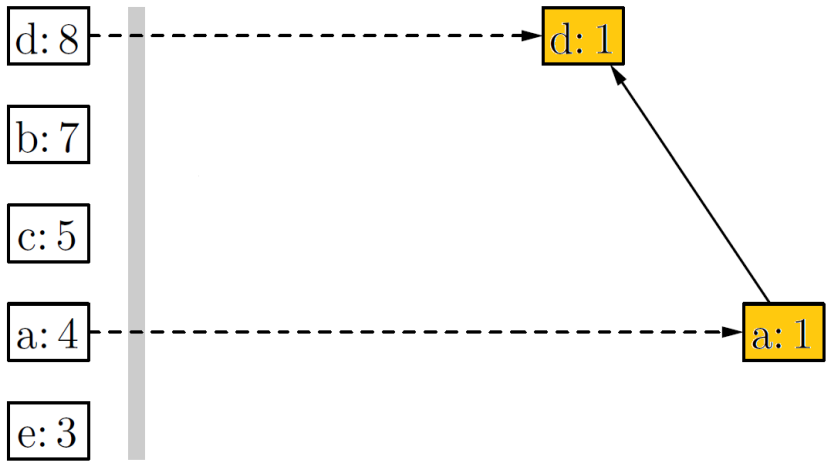
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

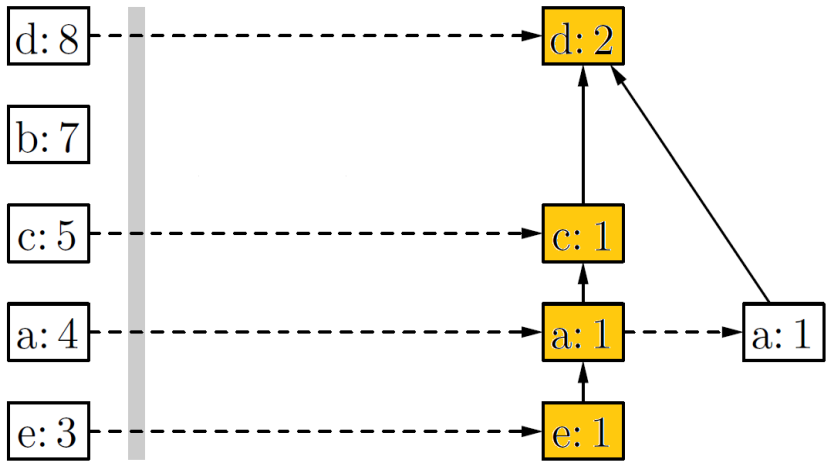
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

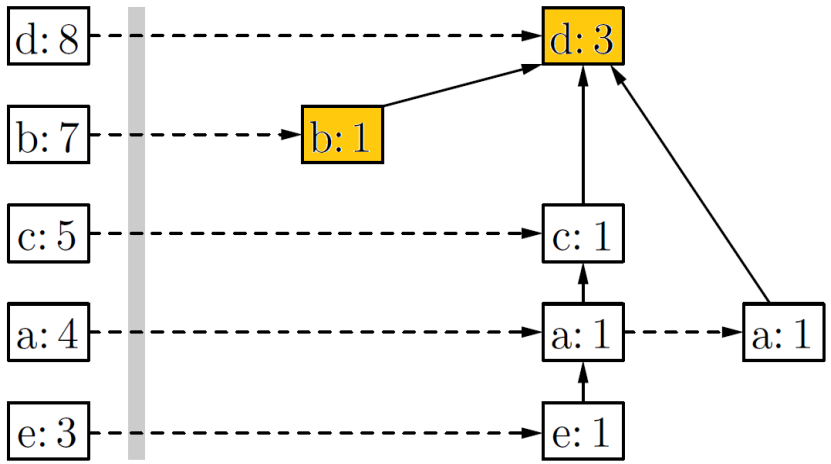
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

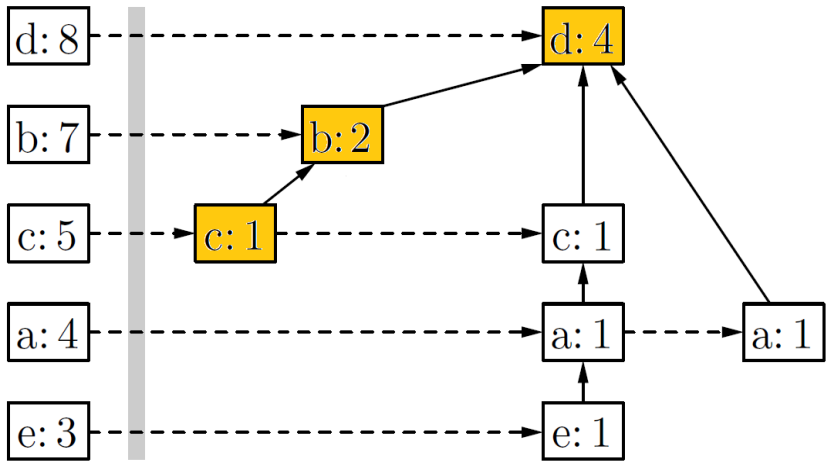
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

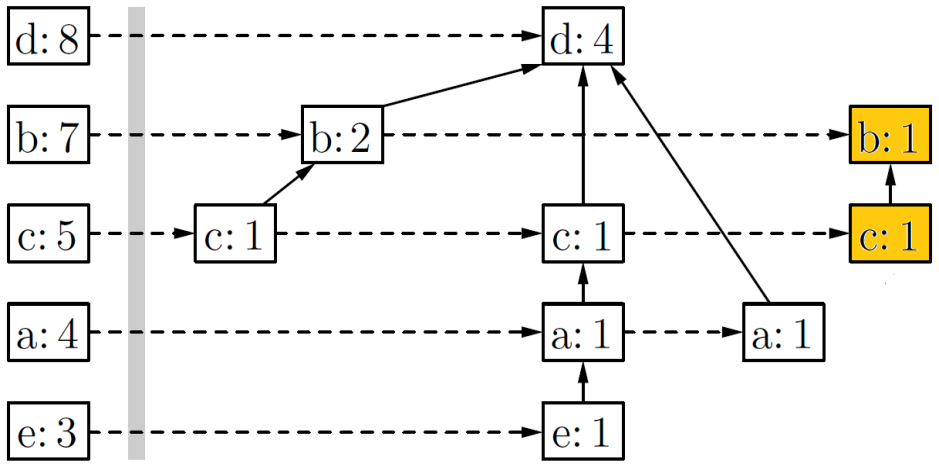
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

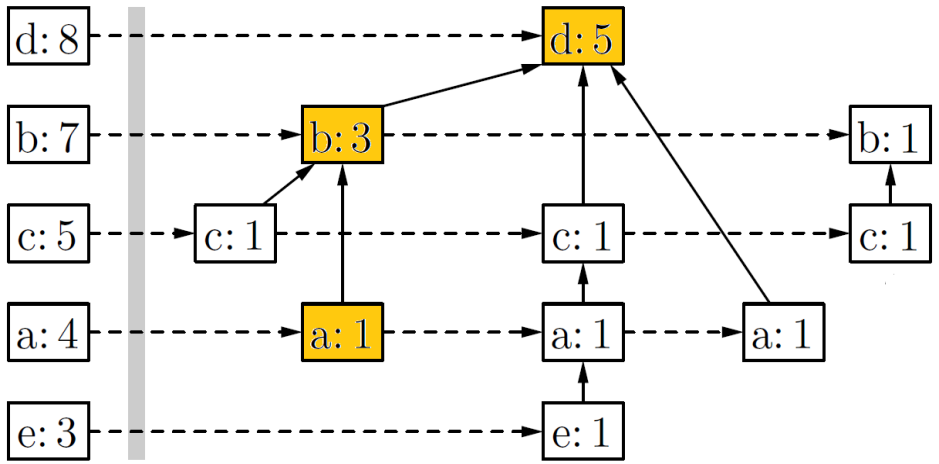
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

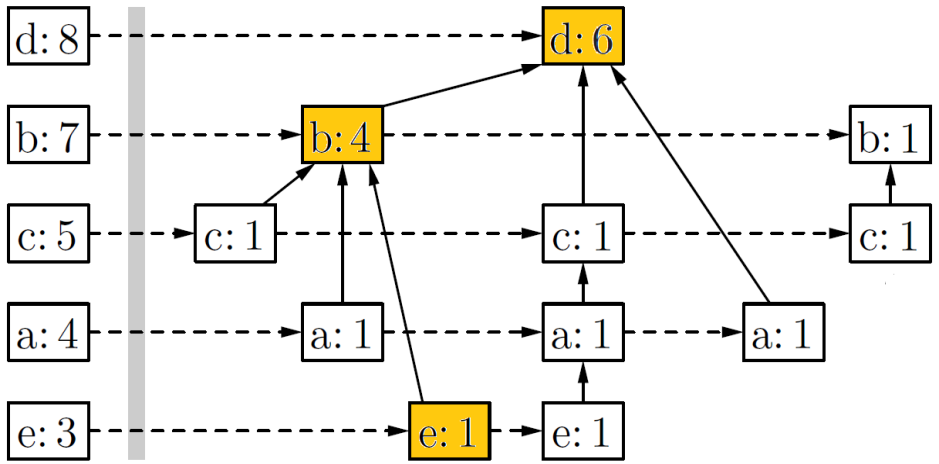
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

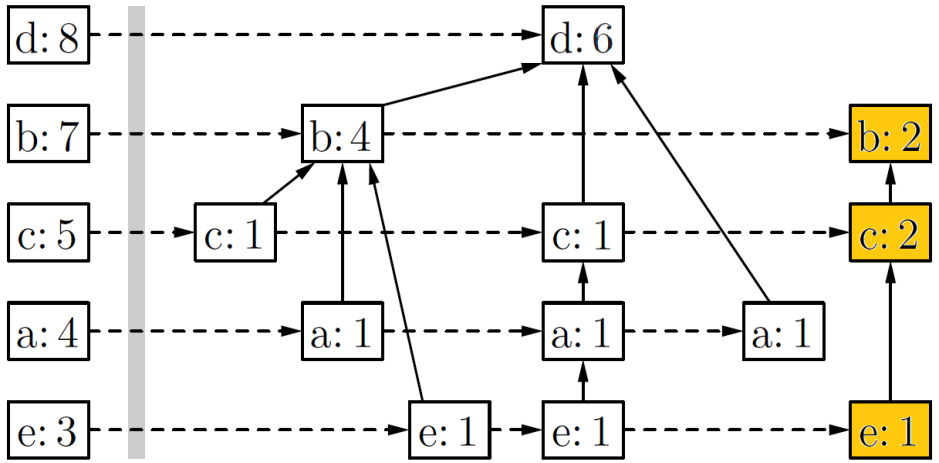
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

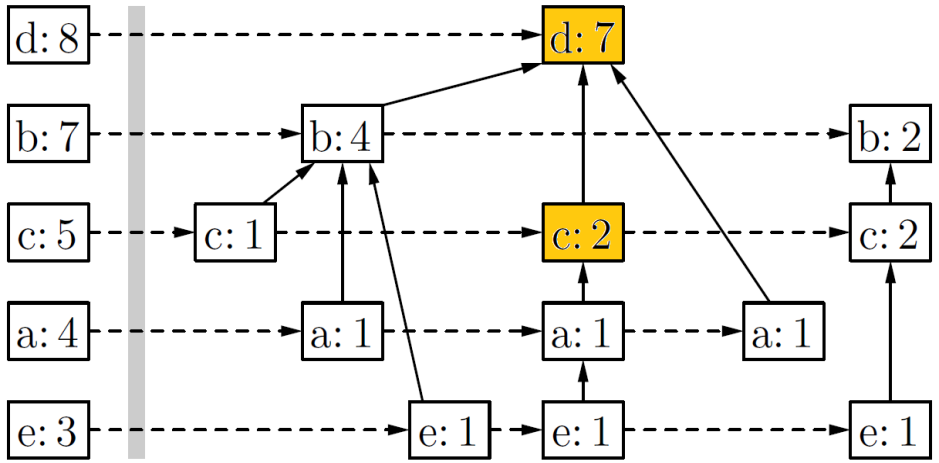
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

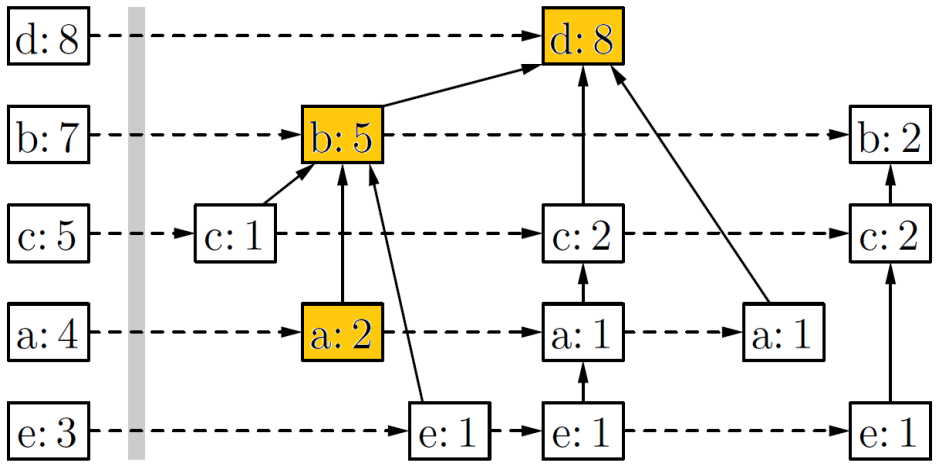
Пример: построение префиксного FP-дерева

Упорядочим все признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$.

Тогда каждый объект описывается словом в алфавите $\mathscr{F}$;
FP-дерево — это эффективный способ хранения словаря;
уровни дерева соответствуют признакам, по убыванию $\nu(f)$.

матрица	слова
a - - d - f -	d a
a - c d e - -	d c a e
- b - d - - -	d b
- b c d - - -	d b c
- b c - - - -	b c
a b - d - - -	d b a
- b - d e - -	d b e
- b c - e - g	b c e
- - c d - f -	d c
a b - d - - -	d b a

(корень v_0 не показан)



при $\delta = 3$ признаки f, g не частые

Алгоритм FP-growth

Вход: X^ℓ — обучающая выборка;
Выход: FP-дерево T, $\langle f_v, c_v, S_v \rangle_{v \in T}$;

упорядочить признаки $f \in \mathscr{F}$: $\nu(f) \geqslant \delta$ по убыванию $\nu(f)$;
ЭТАП 1: построение FP-дерева T по выборке X^ℓ
для всех $x_i \in X^\ell$
  $v := v_0$;
  для всех $f \in \mathscr{F}$ таких, что $f(x_i) \neq 0$
    если нет дочерней вершины $u \in S_v$: $f_u = f$ **то**
      создать новую вершину u; $S_v := S_v \cup \{u\}$;
      $f_u := f$; $c_u := 0$; $S_u := \varnothing$;
    $c_u := c_u + 1/\ell$; $v := u$;
ЭТАП 2: рекурсивный поиск частых наборов по FP-дереву T
FP-find $(T, \varnothing, \varnothing)$;

Этап 2: рекурсивный поиск частых наборов по FP-дереву

Вход: FP-дерево T, набор $\varphi \subset \mathscr{F}$, список правил R;
Выход: добавить в R все частые наборы, содержащие φ;

1: **ПРОЦЕДУРА** FP-find (T, φ, R);
2: **для всех** $f \in \mathscr{F}: V(T, f) \neq \varnothing$ по уровням снизу вверх
3: **если** $C(T, f) \geqslant \delta$ **то**
4: добавить частый набор $\varphi \cup \{f\}$ в список R:
$R := R \cup \{\varphi \wedge f\}$;
5: построить условное FP-дерево $T' := T|f$, а именно:
$T' :=$ FP-дерево по подвыборке $\left\{x_i \in X^\ell : f(x_i) = 1\right\}$;
6: найти по T' все частые наборы, включающие φ и f:
FP-find $\left(T', \varphi \cup \{f\}, R\right)$;

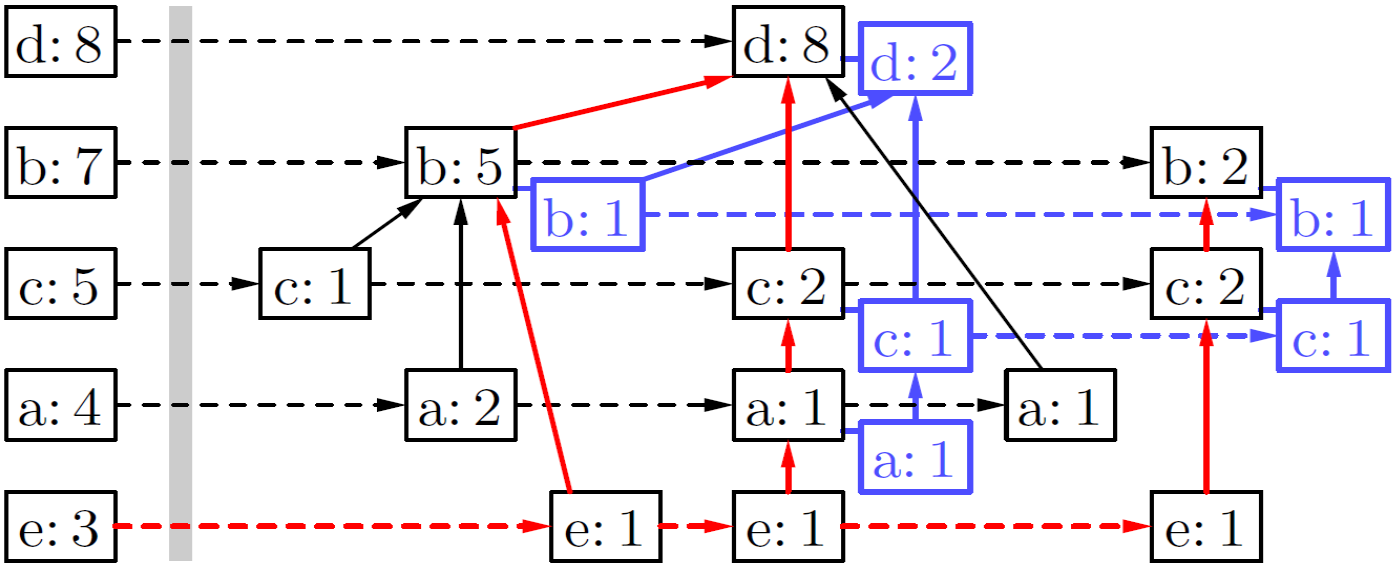
Условное FP-дерево $T' := T|f$ можно построить быстро, используя только FP-дерево T и не заглядывая в выборку.

Условное FP-дерево

Пусть FP-дерево T построено по выборке X^ℓ.

Опр. *Условное FP-дерево* (conditional FP-tree) — это FP-дерево $T' := T|f$, построенное по подвыборке $\{x_i \in X^\ell : f(x_i) = 1\}$, из которого удалены все вершины $v \in V(T', f)$ и все их потомки.

Продолжение примера: CFP-дерево $T|$"e"



Быстрое построение условного FP-дерева $T' = T|f$

Вход: FP-дерево T, признак $f \in \mathscr{F}$;
Выход: условное FP-дерево $T' = T|f$;

1: оставить в дереве только вершины на путях из вершин v признака f снизу вверх до корня v_0:
$T' := \bigcup\limits_{v \in V(T,f)} [v, v_0]$;

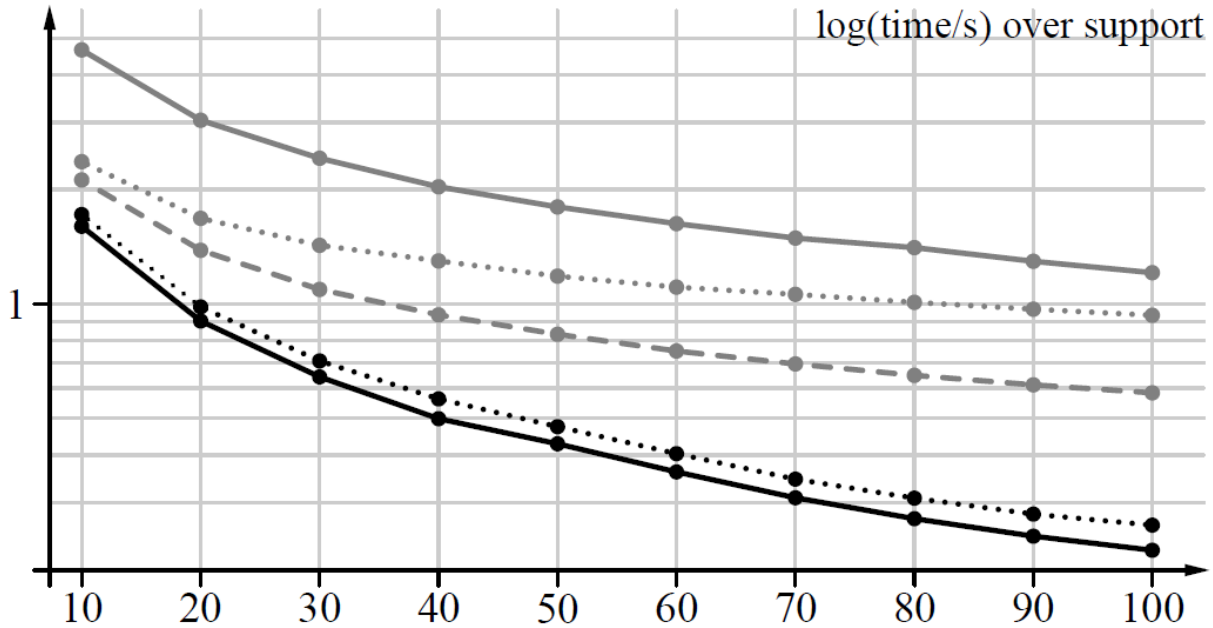
2: поднять значения счётчиков c_v от вершин $v \in V(T', f)$ снизу вверх по правилу
$c_u := \sum\limits_{w \in S_u} c_w$ для всех $u \in T'$;

3: удалить из T' все вершины признака f;
их поддеревья также не нужны и даже не создаются, т.к. в момент вызова FP-find все наборы, содержащие признаки ниже f, уже просмотрены.

Эффективность алгоритма FPGrowth

Одна из типичных зависимостей log времени работы алгоритма от `MinSupp` (на выборке данных `census`).

Нижние кривые — две разные реализации FP-growth.



Christian Borgelt. An Implementation of the FPgrowth Algorithm. 2005.

Резюме в конце лекции

- Поиск ассоциативных правил — обучение без учителя.
- Ассоциативное правило (по определению) — почти то же самое, что логическая ε, δ-закономерность.
- Простые алгоритмы типа APriory вычислительно неэффективны на больших данных.
- FP-growth — один из самых эффективных алгоритмов поиска ассоциативных правил.
- Для практических приложений часто используются его инкрементные и/или иерархические обобщения.