

## Data Mining : Frequent Itemset Mining Algorithm

情報知識ネットワーク特論/IKN Special Lecture  
Data Mining 2b: Subset enumeration  
データマイニング2b: 補足: 部分集合の列挙

有村 博紀, 喜田拓也

北海道大学大学院 情報科学研究科 コンピュータサイエンス専攻

email: {arim,kida}@ist.hokudai.ac.jp

<http://www-ikn.ist.hokudai.ac.jp/~arim>

Slide: <http://www-ikn.ist.hokudai.ac.jp/ikn-tokuron/>

## 2回(補足): 部分集合の列挙

- 前回の演習問題
- 集合の基本
- 部分集合列挙
- アルゴリズム
  - 整数との1対1対応
  - 二分割法(分割統治法)
  - 逆探索法(バックトラック法)

# 前回の演習

- 練習1. 集合を $S = \{1, 2, 3, 4\}$ とする. 次の問題に答えよ.  
Let  $S = \{1, 2, 3, 4\}$  be a finite set of four elements.
  1. 空集合 $\varnothing$ は $S$ の部分集合か? Is the empty set  $\varnothing$  a subset of  $S$ ?
  2.  $S$ のすべての部分集合を書き出せ. Write/List all subsets of  $S$ .
  3. それらの総数はいくつか?  
How many are they?
  4. オプション問題) 非負整数  $n$  を入力としてもらい, 集合 $S = \{1, \dots, n\}$ のすべての部分集合を書きだすプログラムを書け. プログラミング言語は何を使ってもよい. 日本語や英語の疑似コードでもかまわない.  
Optional Problem) Write a computer program (algorithm) that receives a nonnegative number  $n$ , and prints all subsets of the set  $S = \{1, \dots, n\}$ . You can use any programming languages as well as pseudo code written in Japanese/English.

# 集合の基本

- 集合 (set) とは モノの集まり  $A$  で, 任意の要素  $x$  と集合  $A$  に対して, 所属関係 " $x \in A$ " が YES か NO に決定する
- 空集合  $\Phi$ : 任意の  $x$  に対して " $x \notin \Phi$ " となる集合
- 例 :
  - $\Phi = \{ \}$  (空集合)
  - $A = \{1, 2, 3\}$  (外延記法)
  - $E = \{ x \mid x \text{を} 2 \text{で割った余りが} 0 \}$  (内包記法)

# 集合の基本

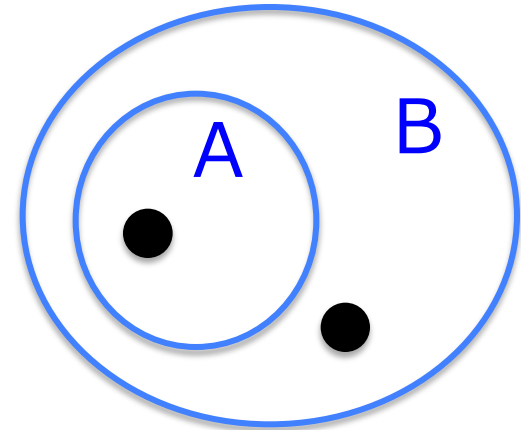
- 定義：AがBの部分集合である：

$$A \subseteq B$$

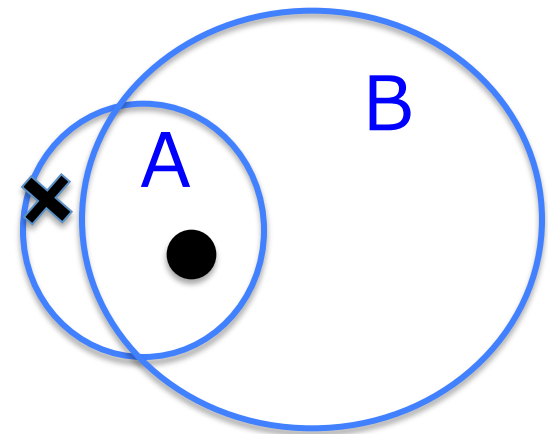
$$\Leftrightarrow \forall x [x \in A \Rightarrow x \in B]$$

$\Leftrightarrow$  「すべての要素xに対して、xが集合Aに入るならば集合Bにも入る」

$$A \subseteq B$$



$$A \subseteq B$$



# 集合Sの部分集合

- $S = \{1, 2, 3\}$
- $2^S = \{X \mid X \subseteq S\}$ : 集合Sの全集合の族

$\emptyset$

1, 2, 3

12, 13, 23

123

# 集合Sの部分集合

- $S = \{1, 2, 3, 4\}$
- $2^S = \{ X \mid X \subseteq S \}$

$\Phi$

1, 2, 3, 4

12, 13, 14, 23, 24, 34

123, 124, 234

1234

# 部分集合列挙1：数との1対1対応

数	ビットベクトル (123の順)	部分集合
0	000	$\phi$
1	001	3
2	010	2
3	011	2,3
4	100	1
5	101	1,3
6	110	1,2
7	111	1,2,3

- $S = \{1, \dots, n\}$ の部分集合全体は,
- 0から $2^n - 1$ の整数全体と対応する



# 部分集合列挙2：二分分割法

ビットベクトル (123の順)	部分集合
<u>0</u> 00	$\Phi$
<u>0</u> 01	3
<u>0</u> 10	2
<u>0</u> 11	2,3
<u>1</u> 00	1
<u>1</u> 01	1,3
<u>1</u> 10	1,2
<u>1</u> 11	1,2,3

- $S$  の部分集合全体が答え全体
- 答え全体を二つに分けて, 分割統治法で列挙する
- 各要素  $i \in S$  に対し, 答え全体を  $i$  を含むものと含まないもの二つずつに分けて再帰的に列挙する

## 2. 二分分割法のプログラム

```
#subset2.rb
#二分分割法でS={1,...,n}の部分集合を出力する
#再帰てつづき
def enum(x, n)
  if n == 0 then
    print "{x}'¥n" #集合xを出力
  else
    a = n.to_s
    enum(x+a, n-1)
    #=> nを含む部分集合をすべて出力
    enum(x, n-1)
    #=> #nを含まない部分集合をすべて出力
  end
end

#メイン文：入力nは集合サイズ
enum("", n)

#EOF
```

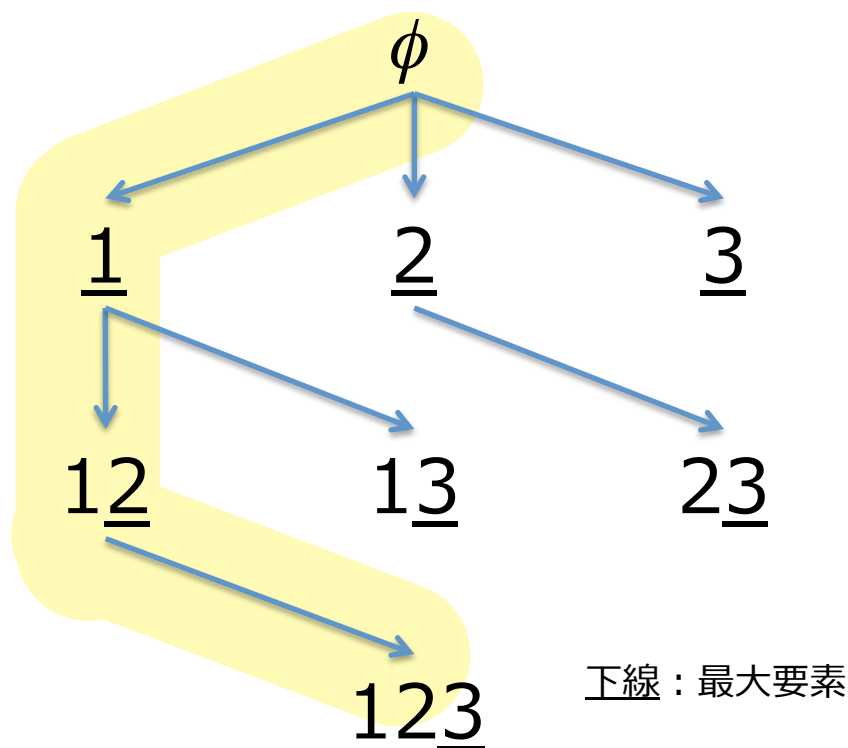
```
### 実行例
```

```
$ ruby subset2.rb 3
```

```
subset2.rb: printing all
subsets of S={1,...,3}
'321'
'32'
'31'
'3'
'21'
'2'
'1'
''
```

# 部分集合列挙3：逆探索法

基本アイデア：部分集合がなす空間上に、木の形をした探索路をつくり、その上を探索する。



概要：

- 空でない全ての部分集合 $X$ に対して、その一意な「親」 $P(X)$ を一つ定める。
  - $P(X) := X - \{\max(X)\}$
- 親ポイント全体は、根 $\phi$ をもつ全部分集合上の根付き木を定める。
- アルゴリズム：この木を、根からスタートして、再帰でバックトラックしながら探索する。

探索では、現在の集合 $X$ に、その最大要素  $k = \max(X)$  より大きな要素  $i > k$  を一つ加えたものを「子」 $Y := X \cup \{i\}$  として再帰的に空間を探索する

# 3 逆探索法のプログラム

```
#subset3.rb
#逆探索法でS={1,...,n}の全ての部分集合を出力する
#再帰てつづき
def enum(x, k, n)
  print "{x}" % n      #集合xを出力
  if k == n then
    return
  end
  for i in k+1..n do
    a = i.to_s        #文字に変換
    enum(x+a, i, n)
  end
end

#メイン文：入力nは集合サイズ
enum("", 0, n)

#EOF
```

### 実行例

```
$ ruby subset3.rb 3
```

```
subset3.rb: printing all
subsets of S={1,...,3}
```

```
"
'1'
'12'
'123'
'13'
'2'
'23'
'3'
```

Python 2.7.9. Mac OS X, 0.10.5

# 補足：部分集合列挙：共通ヘッダー

```
# coding: utf-8
# S = {1,...,n}の全ての部分集合を出力する

#入力nの読み込み
if ARGV.size() < 1 then
  print "usage: #{ $0 } n¥n"
  exit 1;
end
n = ARGV[0].to_i
print "#{ $0 }: printing all subsets of S={1,...,#{n}}¥n"

#このあとにそれぞれの手続き定義
```

- 上記のプログラムを実行するための共通ヘッダー
- 入力nのコマンドラインからの読み込みを行う。
- 各プログラムの先頭に、左のソースコードを貼り付けて、ソースコードとして、シェル上でrubyコマンドで実行する。

Python言語の表記:

- 行頭が'#'文字の行はコメント
- ARGV => コマンド引数 ARGV[0], ARGV[1], ...の配列
- var.to\_i => 整数に変換.
- var.to\_s => 文字列に変換
- #{var} => 変数varの値を文字列中に展開

## 2回(補足): 部分集合の列挙

- 集合の基本
- 部分集合列挙
- アルゴリズム
  - 整数との1対1対応
  - 二分割法(分割統治法)
  - 逆探索法(バックトラック法)