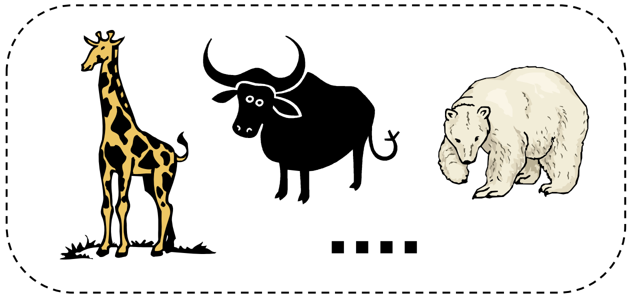
RP-growth: Top-k mining of relevant patterns with minimum support raising

Yoshitaka Kameya and Taisuke Sato (Tokyo Tech)

Background

- Inconvenience in frequent pattern mining:
  - Flood of common, uninformative patterns
  - Difficulty in finding an appropriate min-sup
- Remedies:
  - Top-k mining
  - Discriminative pattern mining
    - Subgroup discovery
    - Contrast set mining
    - Emerging pattern mining
    - Supervised descriptive rule discovery
    - Cluster grouping
    - …



C^+:Positive class C^-:Negative class

Discriminative pattern x

milk=True $\wedge$ aquatic=False

➔ c^+

Class c of interest

Relevance scores

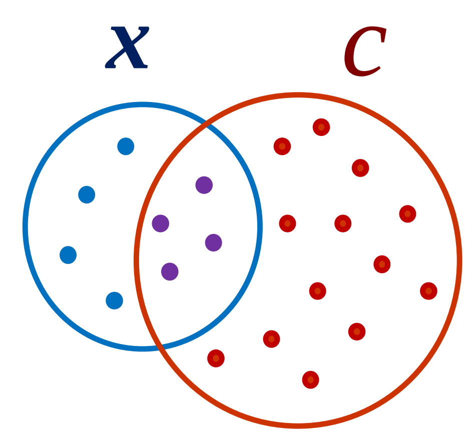
Relevance $R_c(x)$ between class c of interest and pattern x

- Positive support (Recall) $p(\boldsymbol{x} \mid c)$
- Confidence (Precision) $p(c \mid \boldsymbol{x}) \propto \dfrac{p(\boldsymbol{x} \mid c)}{p(\boldsymbol{x})}$
- F-score $\mathrm{F}_c(\boldsymbol{x}) = \dfrac{2p(c \mid \boldsymbol{x})p(\boldsymbol{x} \mid c)}{p(c \mid \boldsymbol{x}) + p(\boldsymbol{x} \mid c)}$
- χ^2-score $\chi^2_c(\boldsymbol{x}) = N \sum_{c' \in \{c, \neg c\},\, \boldsymbol{x}' \in \{\boldsymbol{x}, \neg \boldsymbol{x}\}} \frac{(p(c', \boldsymbol{x}') - p(c')p(\boldsymbol{x}'))^2}{p(c')p(\boldsymbol{x}')}$
- Support difference $\mathrm{SupDiff}_c(\boldsymbol{x}) = p(\boldsymbol{x} \mid c) - p(\boldsymbol{x} \mid \neg c)$

- Most of these relevance scores do not satisfy anti-monotonicity
- Branch and bound strategy:
  - Computes an upper bound $\overline{R}_c(\boldsymbol{x})$ of $R_c(\boldsymbol{x})$
  - Prunes the search space based on $\overline{R}_c(\boldsymbol{x})$
- Previous methods:
  - Subgroup discovery [Wrobel 97], AprioriSMP [Morishita & Sese 00], CG algorithm [Zimmermann & De Raedt 09]

RP-growth (our proposal)

- Finds top-k pattern $\boldsymbol{x}$'s according to R_c for class c of interest under the constraints:
  - Support $p(\boldsymbol{x} \mid c) \geq \sigma_{\min}$ (default: $\sigma_{\min} = 1/|D|$)
  - Confidence $p(c \mid \boldsymbol{x}) \geq \beta_{\min}$ (default: $\beta_{\min} = 0.5$ or $p(c)$)
  - $\boldsymbol{x}$ and $\boldsymbol{x}'$ are not weaker than each other



$\boldsymbol{x}'$ is weaker than $\boldsymbol{x}$ $\Leftrightarrow$

$\boldsymbol{x} \subset \boldsymbol{x}'$ but $R_c(\boldsymbol{x}) \geq R_c(\boldsymbol{x}')$

$R_c(\{\text{A}\}) = 0.6$

$R_c(\{\text{B}\}) = 0.8$

not weaker than

$R_c(\{\text{A, B}\}) = 0.9$

not weaker than

weaker than

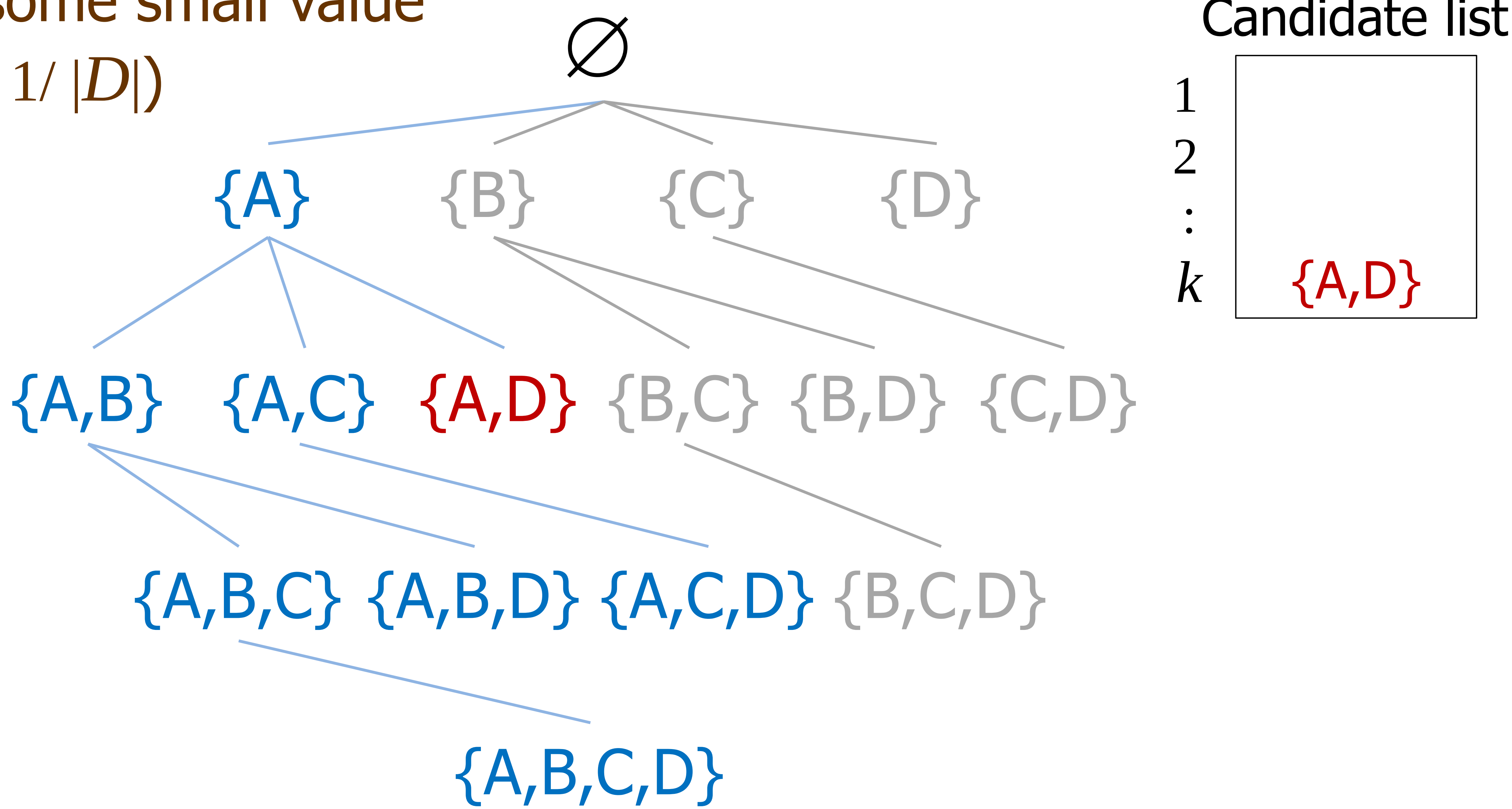
:

$R_c(\{\text{A, B, C}\}) = 0.7$

Top-k frequent pattern mining

- **Base strategy**: Depth-first search + Minimum support raising

$\sigma_{\min}$:= some small value
(typically $1/|D|$)



Minimum support raising: $\sigma_{\min} := p(\{A, D\} \mid c)$

B&B pruning translated into min-sup raising

- Definition of the F-score: $\mathrm{F}_c(\boldsymbol{x}) = \dfrac{2p(\boldsymbol{x} \mid c)p(c \mid \boldsymbol{x})}{p(\boldsymbol{x} \mid c) + p(c \mid \boldsymbol{x})}$
- An *anti-monotonic upper bound* of $\mathrm{F}_c(\boldsymbol{x})$ by substituting $p(c \mid \boldsymbol{x}) := 1$ (or substituting $p(\boldsymbol{x} \mid \neg c) := 0$, etc.)

$$\overline{\mathrm{F}}_c(\boldsymbol{x}) = \frac{2p(\boldsymbol{x} \mid c)}{p(\boldsymbol{x} \mid c) + 1}$$

- Pruning: Patterns including $\boldsymbol{x}$ will never remain in the candidate list if:

$$\mathrm{F}_c(\boldsymbol{z}) > \overline{\mathrm{F}}_c(\boldsymbol{x}) = \frac{2p(\boldsymbol{x} \mid c)}{p(\boldsymbol{x} \mid c) + 1} \quad \overset{\text{iff}}{\longleftrightarrow} \quad p(\boldsymbol{x} \mid c) < \frac{\mathrm{F}_c(\boldsymbol{z})}{2 - \mathrm{F}_c(\boldsymbol{z})}$$

$\boldsymbol{z}$: k-th pattern

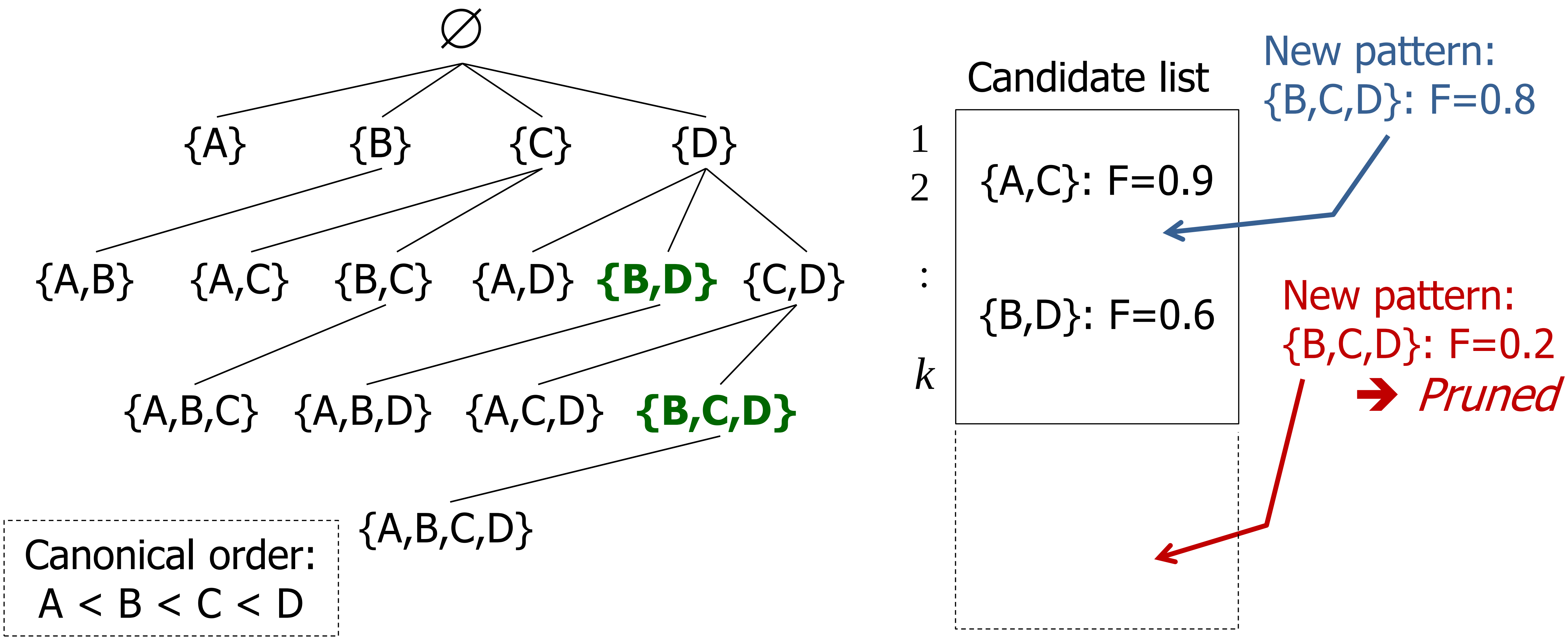
- Min-sup raising:

$$\sigma_{\min} := \frac{\mathrm{F}_c(\boldsymbol{z})}{2 - \mathrm{F}_c(\boldsymbol{z})}$$

- Applicable to non-convex relevance scores such as F-score
- Applicable to (sequence|tree|graph) mining
- Can benefit from FP-growth's dynamic shrinking of conditional databases

Handling weakness

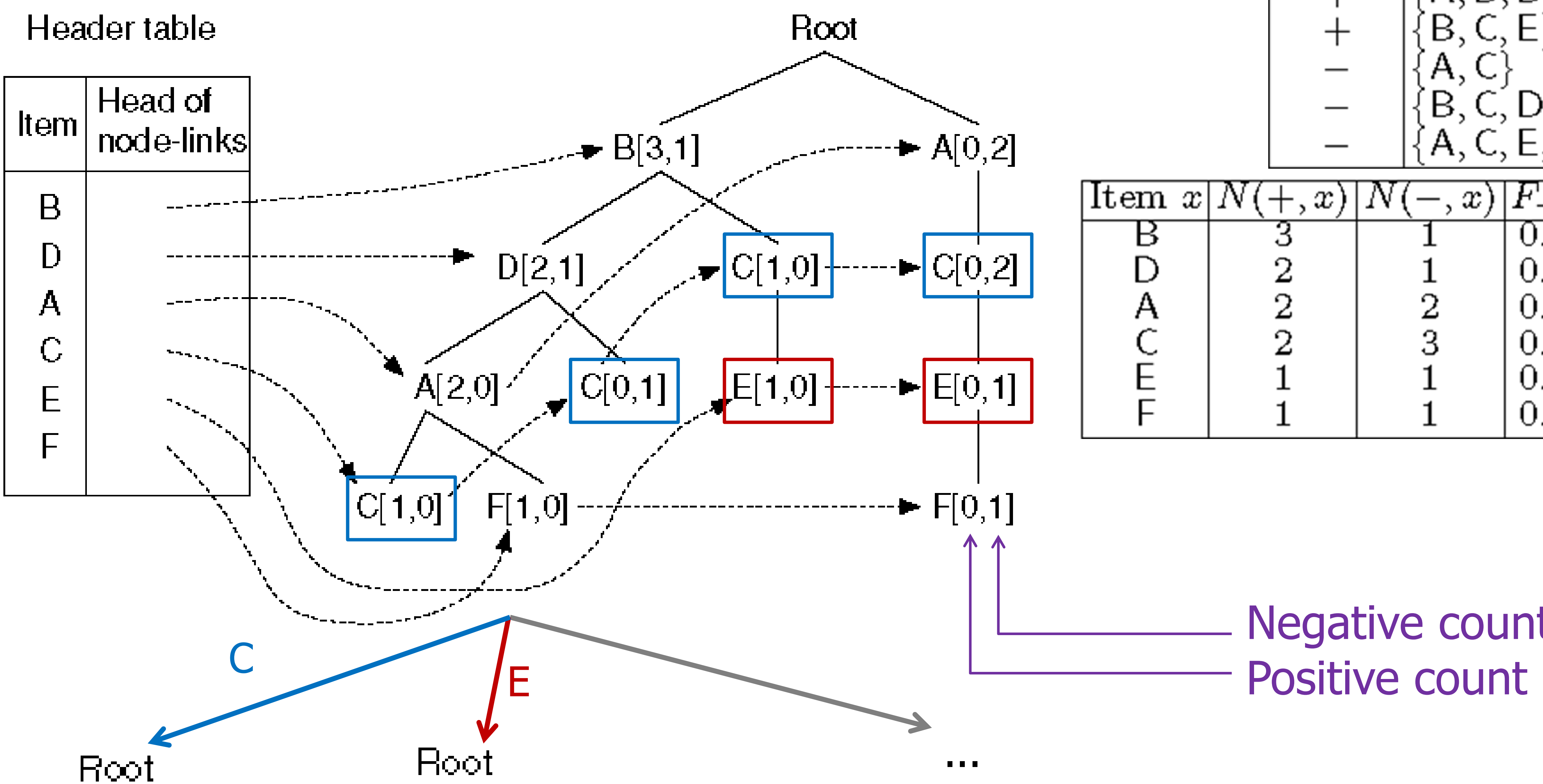
- **Key point:** Use of suffix enumeration trees
  - "When visiting $\boldsymbol{x}$, any sub-pattern $\boldsymbol{x}'$ of $\boldsymbol{x}$ has already been visited"
  - FP-growth (implicitly) uses a suffix enumeration tree
  - When $\overline{R}_c(\boldsymbol{x}') \leq R_c(\boldsymbol{x})$, all patterns including $\boldsymbol{x}'$ are guaranteed to be weaker than $\boldsymbol{x}$ ➔ Patterns below $\boldsymbol{x}'$ are prunable

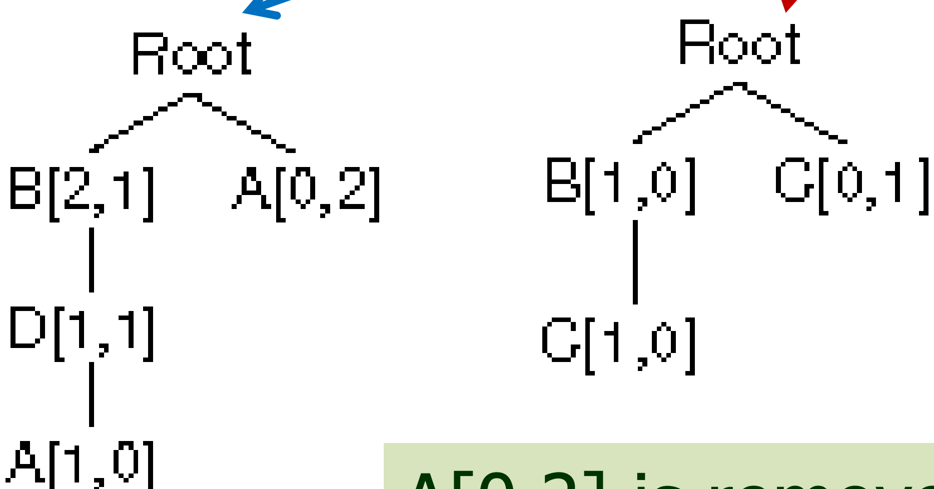


RP-trees: Extension of FP-trees

Class c	Transaction
$+$	$\{A, B, C, D\}$
$+$	$\{A, B, D, F\}$
$+$	$\{B, C, E\}$
$-$	$\{A, C\}$
$-$	$\{B, C, D\}$
$-$	$\{A, C, E, F\}$

Item x	$N(+, x)$	$N(-, x)$	$F_+(x)$
B	3	1	0.857
D	2	1	0.667
A	2	2	0.571
C	2	3	0.500
E	1	1	0.400
F	1	1	0.400





Conditional pattern bases are shrinked dynamically (recursively)
→ The remaining search is accelerated

A[0,2] is removed due to min-sup

Experiments: 20 news group dataset

- Preprocessed data: 17,930 articles consisting of 5,666 words
- Top-25 non-weak relevant patterns:

comp.graphics

Pattern $\boldsymbol{x}$	$p(c \mid \boldsymbol{x})$	$p(\boldsymbol{x} \mid c)$	$\mathrm{F}_c(\boldsymbol{x})$
{graphic, program}	0.537	0.136	0.217
{gif}	0.552	0.119	0.196
{graphic, imag}	0.642	0.108	0.185
{imag, program}	0.516	0.110	0.181
{imag, file}	0.531	0.105	0.175
{graphic, find}	0.578	0.087	0.151
{imag, bit}	0.514	0.083	0.144
{graphic, code}	0.613	0.081	0.143
{graphic, bit}	0.545	0.080	0.140
{graphic, packag}	0.591	0.076	0.134
{format, convert}	0.588	0.075	0.132
{graphic, comp}	0.730	0.072	0.132
{imag, format}	0.613	0.072	0.129
{graphic, point}	0.573	0.070	0.125
{graphic, format}	0.670	0.068	0.123
{imag, convert}	0.596	0.066	0.118
{polygon}	0.915	0.060	0.113
{imag, softwar}	0.500	0.062	0.111
{graphic, ftp}	0.500	0.061	0.109
{graphic, algorithm}	0.852	0.058	0.108
{jpeg}	0.825	0.058	0.108
{graphic, group}	0.514	0.060	0.108
{graphic, site}	0.530	0.059	0.106
{graphic, comput, articl}	0.525	0.059	0.106
{code, algorithm}	0.500	0.059	0.105

rec.sport.hockey

Pattern $\boldsymbol{x}$	$p(c \mid \boldsymbol{x})$	$p(\boldsymbol{x} \mid c)$	$\mathrm{F}_c(\boldsymbol{x})$
{hockei}	0.943	0.377	0.538
{team}	0.519	0.473	0.495
{playoff}	0.943	0.277	0.428
{game, plai}	0.506	0.273	0.354
{nhl}	0.990	0.206	0.341
{cup}	0.584	0.195	0.292
{player, plai}	0.575	0.190	0.286
{score}	0.510	0.194	0.281
{game, player}	0.561	0.186	0.280
{game, goal}	0.899	0.157	0.267
{game, win}	0.517	0.174	0.260
{game, fan}	0.622	0.164	0.260
{plai, goal}	0.852	0.144	0.246
{wing}	0.515	0.156	0.240
{leaf}	0.894	0.132	0.230
{bruin}	1.000	0.130	0.230
{pittsburgh}	0.567	0.142	0.226
{game, watch}	0.621	0.136	0.224
{detroit}	0.733	0.131	0.222
{penguin}	0.871	0.127	0.222
{game, season}	0.539	0.137	0.219
{game, night}	0.660	0.129	0.216
{ranger}	0.629	0.129	0.214
{plai, win}	0.529	0.134	0.214
{plai, fan}	0.603	0.128	0.211

talk.politics.guns

Pattern $\boldsymbol{x}$	$p(c \mid \boldsymbol{x})$	$p(\boldsymbol{x} \mid c)$	$\mathrm{F}_c(\boldsymbol{x})$
{gun}	0.540	0.414	0.469
{weapon}	0.528	0.253	0.342
{fbi}	0.506	0.246	0.331
{firearm}	0.884	0.196	0.321
{batf}	0.662	0.155	0.252
{waco}	0.543	0.154	0.240
{assault}	0.587	0.124	0.205
{cdt, sw}	0.933	0.110	0.196
{cdt, stratu}	0.916	0.110	0.196
{handgun}	0.818	0.111	0.195
{cdt}	0.817	0.110	0.193
{stratu, sw}	0.700	0.110	0.190
{fire, compound}	0.698	0.109	0.188
{stratu}	0.570	0.110	0.184
{bd}	0.530	0.110	0.182
{sw}	0.521	0.110	0.181
{atf}	0.692	0.101	0.176
{arm, law}	0.527	0.086	0.148
{compound, dai}	0.598	0.082	0.144
{nra}	0.696	0.079	0.143
{rocket, special}	0.750	0.077	0.140
{rocket, speak}	0.840	0.076	0.139
{rocket, vo}	0.918	0.075	0.139
{vo, investor}	0.918	0.075	0.139
{vo, speak, todai}	0.918	0.075	0.139

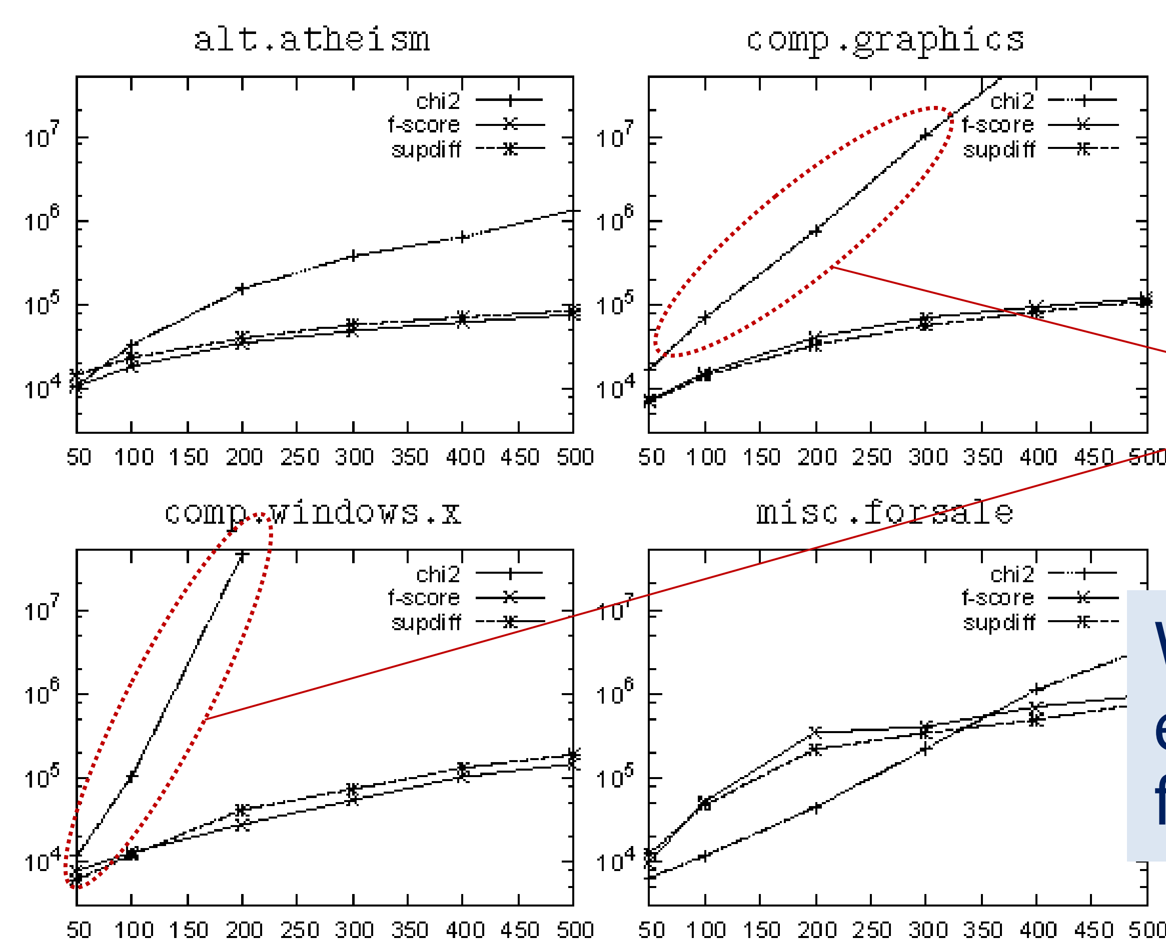
- Relevance score: F-score
- Constraint: $p(c \mid x) \geq 0.5$

Experiments: Feature construction in text classification

- Classifier: SVM (LIBSVM)
- The features constructed from relevant patterns give a good performance even with linear kernels

Single features		Single + combined features		
Linear kernel	RBF kernel	Linear kernel		
		χ^2	F-score	Support diff.
83.88±0.20	84.95±0.22	84.48±0.13	84.73±0.22	84.73±0.23

Experiments: Search space



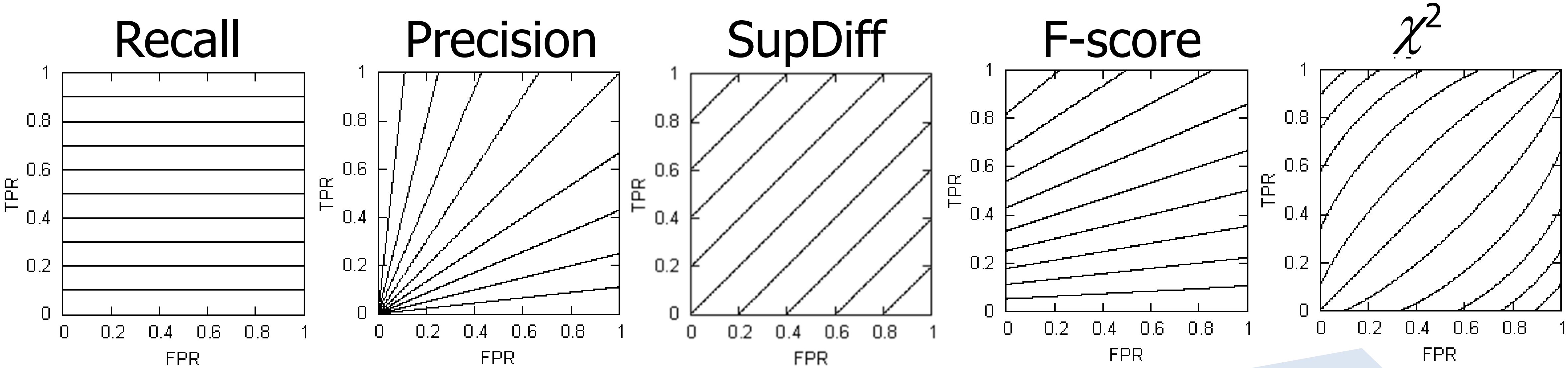
x-axis: #patterns to find
y-axis: #visited-patterns

With χ^2, the search space can be huge
(the search did not finish in 2 hours on
CPU: Core i7 2.66GHz)

With F-score, the search finished in one minute
except it takes 17 minutes
for **`comp.os.ms-windows.misc`**

Discussion: ROC analysis

Condition: $p(c) = 0.5$



$\text{TPR} = p(\mathbf{x} \mid c)$
$\text{FPR} = p(\mathbf{x} \mid \neg c)$

χ^2 prefers highly discriminative patterns
→ Our upper bound tends to be loose

Future work: Extension to sequences

- Strong points of RP-growth also apply to sequences, though projection seems to get more complicated

Enumeration tree for permutations: