Project Proposal: Prediction by Compression

Lasse Blaauwbroek

Czech Institute for Informatics, Robotics and Cybernetics
Czech Technical University in Prague

AITP 2018

March 30, 2018

Compressor C such that $C(s)$ is the length of the compression of s

[Cilibrasi and Vitanyi 2003], [Li et al. 2004]

Compressor C such that $C(s)$ is the length of the compression of s

$$s \text{ and } t \text{ share all information} \implies C(st) \approx C(s) + b$$
$$s \text{ and } t \text{ share no information} \implies C(st) \approx C(s) + C(t)$$

[Cilibrasi and Vitanyi 2003], [Li et al. 2004]

Compressor C such that $C(s)$ is the length of the compression of s

$$s \text{ and } t \text{ share all information} \Longrightarrow C(st) \approx C(s) + b$$
$$s \text{ and } t \text{ share no information} \Longrightarrow C(st) \approx C(s) + C(t)$$

$$NCD_C(s,t) = \frac{C(st) - \min(C(s), C(t))}{\max(C(s), C(t))}$$

[Cilibrasi and Vitanyi 2003], [Li et al. 2004]

Compressor C such that $C(s)$ is the length of the compression of s

$$s \text{ and } t \text{ share all information} \implies C(st) \approx C(s) + b$$
$$s \text{ and } t \text{ share no information} \implies C(st) \approx C(s) + C(t)$$

$$NCD_C(s,t) = \frac{C(st) - \min(C(s), C(t))}{\max(C(s), C(t))}$$

Under reasonable conditions for C, NCD_c approximates a metric

[Cilibrasi and Vitanyi 2003], [Li et al. 2004]

Let P be the set of valid programs for programming language L

[Cilibrasi and Vitanyi 2003], [Li et al. 2004]

Let P be the set of valid programs for programming language L
Kolmogorov complexity K:

$$K(s) = \underset{p \in P \wedge L(p) = s}{\arg\min} |p|$$

[Cilibrasi and Vitanyi 2003], [Li et al. 2004]

Let P be the set of valid programs for programming language L
Kolmogorov complexity K:

$$K(s) = \underset{p \in P \wedge L(p) = s}{\arg\min} |p|$$

$$NCD_K(s,t) = \frac{K(st) - \min(K(s), K(t))}{\max(K(s), K(t))}$$

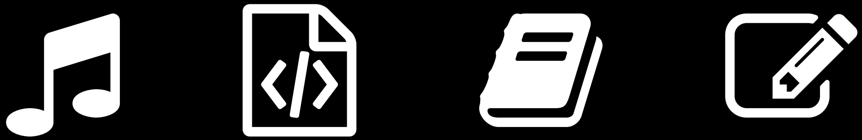
NCD_K is the distance metric:

$$\forall_{d,s,t} \ \ \text{computable}(d) \Rightarrow NCD_K(s,t) \leq d(s,t)$$

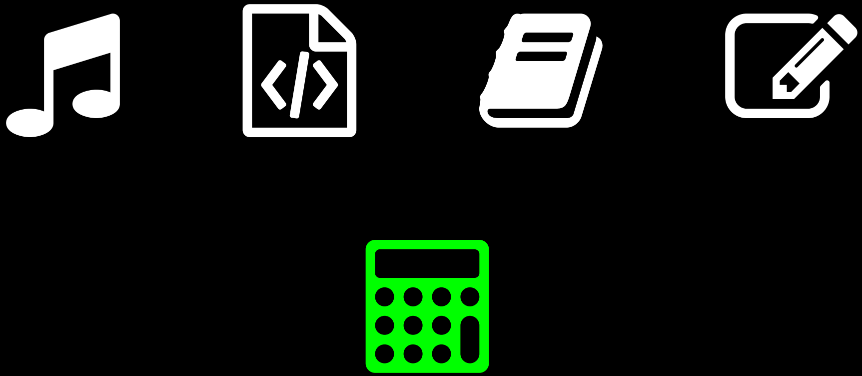
[Cilibrasi and Vitanyi 2003], [Li et al. 2004]

No domain-specific knowledge necessary!

No domain-specific knowledge necessary!



No domain-specific knowledge necessary!



Problem: Mathematical statements are short

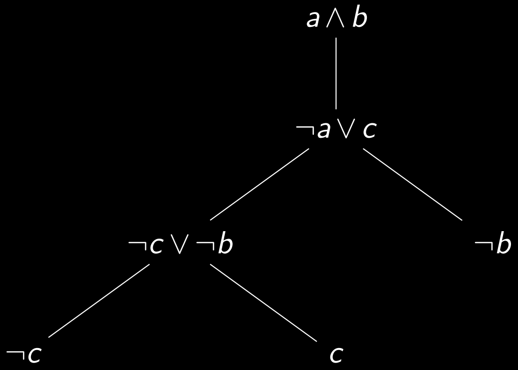
Problem: Mathematical statements are short

Compression: Prediction by Partial Matching

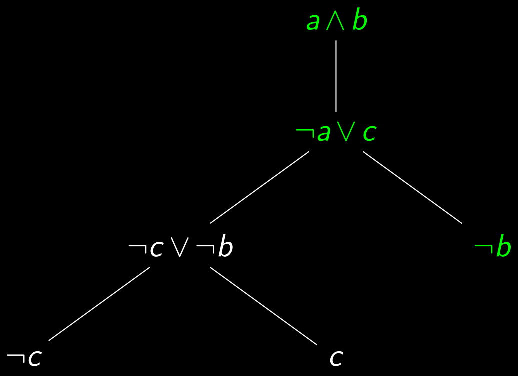
Problem: Mathematical statements are short

Compression: Prediction by Partial Matching

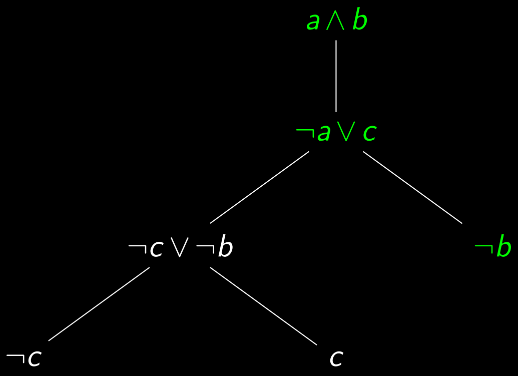
Compress entire proof states



[CKaliszyk, Urban and Vyskoci 2015]

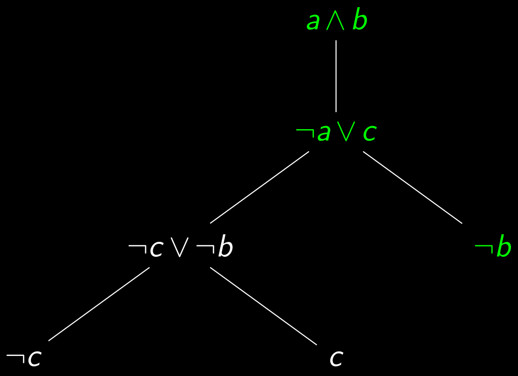


[CKaliszyk, Urban and Vyskoci 2015]



$a \quad b \quad \neg a \quad c$

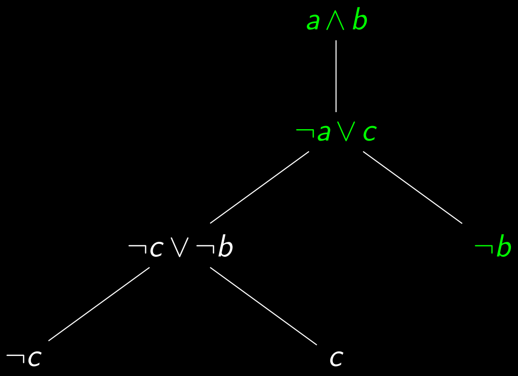
[CKaliszyk, Urban and Vyskoci 2015]



$a \quad b \quad \neg a \quad c$

"$a \wedge b \neg a \vee c \neg b a b \neg a c$"

[CKaliszyk, Urban and Vyskoci 2015]

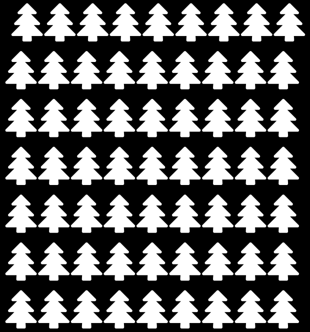


$a \quad b \quad \neg a \quad c$

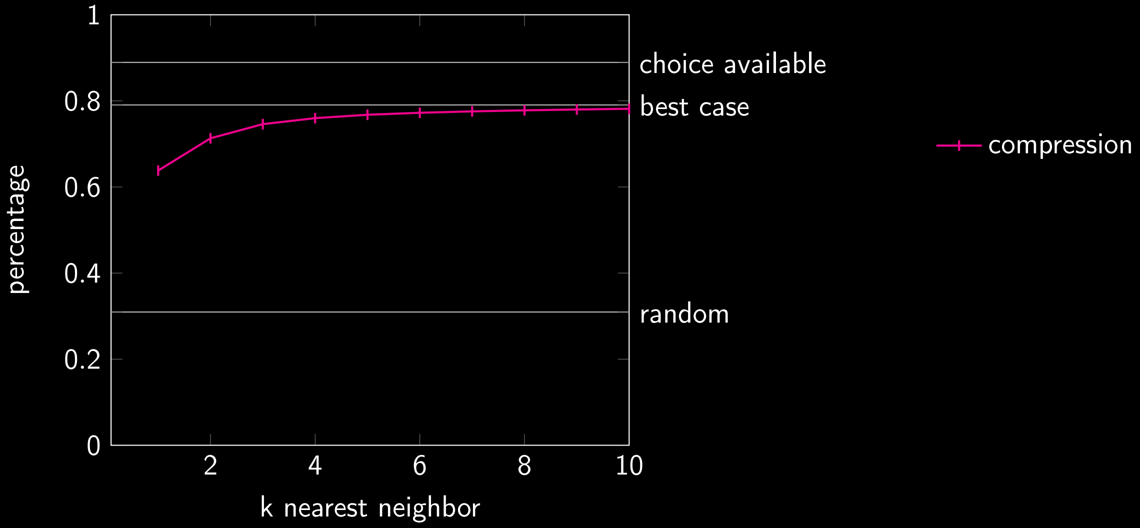
"$a \wedge b \neg a \vee c \neg bab \neg ac$"

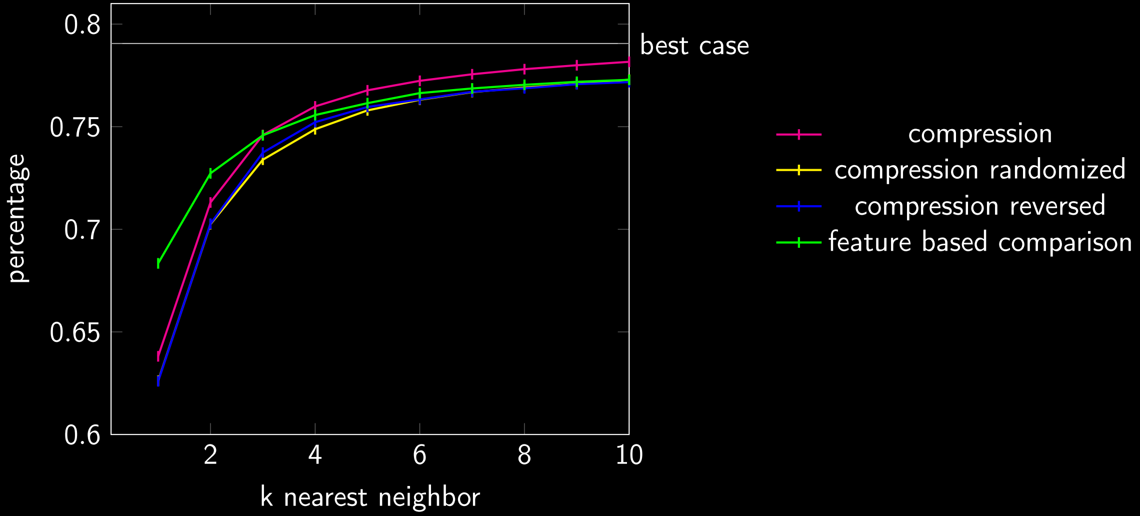
$\Leftrightarrow$

Database



[CKaliszyk, Urban and Vyskoci 2015]


1
0.8
0.6
0.4
0.2
0
percentage
2
4
6
8
10
k nearest neighbor
choice available
best case
random
compression


0.8
0.75
0.7
0.65
0.6
2
4
6
8
10
percentage
k nearest neighbor
best case
compression
compression randomized
compression reversed
feature based comparison

About 30-40 compressions per second

No vector space: n compressions per prediction

About 30-40 compressions per second

No vector space: n compressions per prediction

Idea: Impose structure through an n-dimensional lattice

$$S_n = \{X \subseteq S \mid |X| = n\}$$

$$\text{out}(s) = \arg\max_{X \in S_n} \frac{\sum_{t,u \in X} NCD(t,u)}{\sum_{t \in X} NCD(s,t)}$$

Pros

- No domain-specific knowledge required
- Predictions are competitive
- Robust against different representations of proof states

Pros

- ▷ No domain-specific knowledge required
- ▷ Predictions are competitive
- ▷ Robust against different representations of proof states

Cons

- ▷ Relatively slow
- ▷ No vector space

Pros

- No domain-specific knowledge required
- Predictions are competitive
- Robust against different representations of proof states

Cons

- Relatively slow
- No vector space

Ideas

- Adapt the PPM compressor for tree-structures
- Impose a n-dimensional lattice on the data

Pros

- No domain-specific knowledge required
- Predictions are competitive
- Robust against different representations of proof states

Cons

- Relatively slow
- No vector space

Ideas

- Adapt the PPM compressor for tree-structures
- Impose a *n*-dimensional lattice on the data

?