Rule Learning by Modularity

Albert Nössig, Georg Moser, Tobias Hell
https://tcs-informatik.uibk.ac.at/

(0,0,0,0,0,0,0,1,0,0)

(0,1,0,0,0,0,0,0,0,0)

(0,0,0,0,1,0,0,0,0,0)

?

(0,0,0,0,0,0,0,1,0,0)

(0,1,0,0,0,0,0,0,0,0)

(0,0,0,0,1,0,0,0,0,0)

?

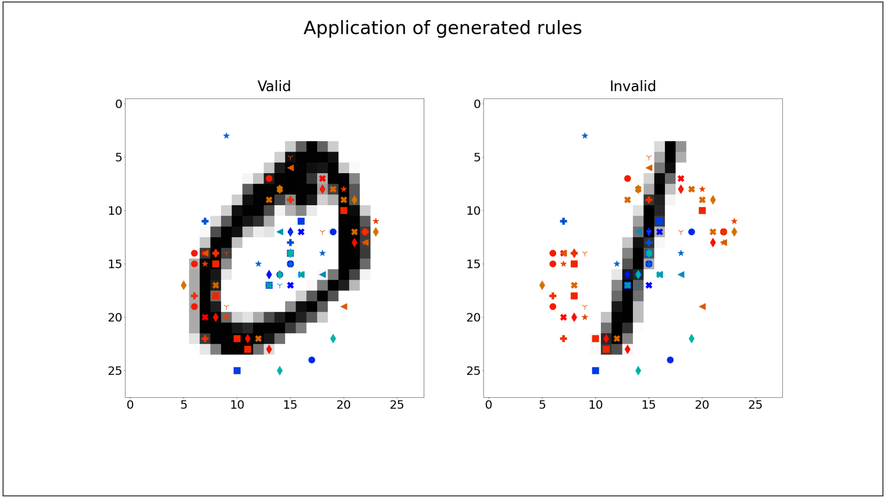
(0,0,0,0,0,0,0,1,0,0)

(0,1,0,0,0,0,0,0,0,0)

(0,0,0,0,1,0,0,0,0,0)

Rule 1	Rule 2	Rule 3	...
white_407	white_407	white_407	...
black_541	white_490	black_512	...
black_656	black_485	white_489	...
black_429	black_628	black_271	...
⋮	⋮	...	⋮

Illustration of rules



Overview

Methodology: Description of the general framework.

Clustering & Rule Learning: Some details about the crucial components of our approach and their benefits.

Evaluation: Experimental results of our approach applied on standard benchmarks as well as a use case from industries.

Related & Future Work: Remarks on pros and cons of similar approaches and ideas for further improvements of our approach.

Case Study: Dental Bills

Sehr geehrte Frau Muster,

für die erbrachten Leistungen in der Zeit vom 02.09. bis 06.09.16 erlaube ich mir folgende Beträge in Rechnung zu stellen: **EUR 257,51**

Datum	Zahn	Anz.	Nr.	Leistung	Faktor	EUR
02.09.16		1	1000	Erstellung eines Mundhygienestatus und eingehende Unterweisung zur Vorbeugung gegen Karies und parodontale Erkrankungen, Dauer mindestens 25 Minuten	1,0	11,25
02.09.16	17,16,15, 14,13,12, 11,21,22, 23,24,25, 26,27,37, 36,35,34, 33,32,31, 41,42,43, 44,45,46, 47	28	1040	Professionelle Zahnreinigung	2,0	88,20
06.09.16	27	1	2120	Präparieren einer Kavität und Restauration mit Komposit-materialien, in Adhäsivtechnik (Konditionieren), mehr als dreiflächig, ggf. einschließlich Mehrschichttechnik, einschließlich Polieren, ggf. einschließlich Verwendung von Inserts **Begründung:** Überdurchschnittlicher Schwierigkeitsgrad und Zeitaufwand wegen erschwertem Anlegen von Matrizen/Bändern durch Zahnengstand und tiefer, sehr schwer einsehbarer Approximalkavität.	3,5	151,57
	27	1		abzgl. Bema-Sachleistung Nr. 13a bis 13d		- 54,47

Zwischensumme Zahnarzt-Honorar:	251,02
abzgl. Kassenanteil nach Bema:	- 54,47
Eigenlabor: (s. beigefügte Materialrechnung)	60,96
Rechnungsbetrag:	**€ 257,51**

umsatzsteuerfrei nach § 4 Nr. 14a UStG

Case Study: Dental Bills

Sehr geehrte Frau Muster,

für die erbrachten Leistungen in der Zeit vom 02.09. bis 06.09.16 erlaube ich mir folgende Beträge in Rechnung zu stellen: **EUR 257,51**

Datum	Zahn	Anz.	Nr.	Leistung	Faktor	EUR
02.09.16		1	1000	Erstellung eines Mundhygienestatus und eingehende Unterweisung zur Vorbeugung gegen Karies und parodontale Erkrankungen, Dauer mindestens 25 Minuten	1,0	11,25
02.09.16	17,16,15, 14,13,12, 11,21,22, 23,24,25, 26,27,37, 36,35,34, 33,32,31, 41,42,43, 44,45,46, 47	28	1040	Professionelle Zahnreinigung	2,0	88,20
06.09.16	27	1	2120	Präparieren einer Kavität und Restauration mit Komposit-materialien, in Adhäsivtechnik (Konditionieren), mehr als dreiflächig, ggf. einschließlich Mehrschichttechnik, einschließlich Polieren, ggf. einschließlich Verwendung von Inserts **Begründung:** Überdurchschnittlicher Schwierigkeitsgrad und Zeitaufwand wegen erschwertem Anlegen von Matrizen/Bändern durch Zahnengstand und tiefer, sehr schwer einsehbarer Approximalkavität.	3,5	151,57
	27	1		abzgl. Bema-Sachleistung Nr. 13a bis 13d		- 54,47

Zwischensumme Zahnarzt-Honorar:	251,02
abzgl. Kassenanteil nach Bema:	- 54,47
Eigenlabor: (s. beigefügte Materialrechnung)	60,96
Rechnungsbetrag:	**€ 257,51**

umsatzsteuerfrei nach § 4 Nr. 14a UStG

⋮	⋮	...
amount 1	date 1	...
amount 2	date 2	...
amount 3	date 3	...
⋮	⋮	...

Case Study: Dental Bills

Sehr geehrte Frau Muster,

für die erbrachten Leistungen in der Zeit vom 02.09. bis 06.09.16 erlaube ich mir folgende Beträge in Rechnung zu stellen: **EUR 257,51**

Datum	Zahn	Anz.	Nr.	Leistung	Faktor	EUR
02.09.16		1	1000	Erstellung eines Mundhygienestatus und eingehende Unterweisung zur Vorbeugung gegen Karies und parodontale Erkrankungen, Dauer mindestens 25 Minuten	1,0	11,25
02.09.16	17,16,15, 14,13,12, 11,21,22, 23,24,25, 26,27,37, 36,35,34, 33,32,31, 41,42,43, 44,45,46, 47	28	1040	Professionelle Zahnreinigung	2,0	88,20
06.09.16	27	1	2120	Präparieren einer Kavität und Restauration mit Komposit-materialien, in Adhäsivtechnik (Konditionieren), mehr als dreiflächig, ggf. einschließlich Mehrschichttechnik, einschließlich Polieren, ggf. einschließlich Verwendung von Inserts **Begründung:** Überdurchschnittlicher Schwierigkeitsgrad und Zeitaufwand wegen erschwertem Anlegen von Matrizen/Bändern durch Zahnengstand und tiefer, sehr schwer einsehbarer Approximalkavität.	3,5	151,57
	27	1		abzgl. Bema-Sachleistung Nr. 13a bis 13d		- 54,47

Zwischensumme Zahnarzt-Honorar: 251,02
abzgl. Kassenanteil nach Bema: - 54,47
Eigenlabor: 60,96
(s. beigefügte Materialrechnung)
Rechnungsbetrag: **€ 257,51**
umsatzsteuerfrei nach § 4 Nr. 14a UStG

⋮	⋮	...
amount 1	date 1	...
amount 2	date 2	...
amount 3	date 3	...
⋮	⋮	...

material costs
material costs
treatment costs

Inductive Logic Programming (ILP)

- Given:

```
parent(a,b)     parent(a,c)
father(a,b)     father(a,c)
male(a)         female(c)

parent(d,b)
mother(d,b)
female(d)
```

- Sought:

```
father(X,Y) :- parent(X,Y) & male(X)
mother(X,Y) :- parent(X,Y) & female(X),
```

Rule Learning

Inductive Logic Programming (ILP)

- Given:

```
parent(a,b)     parent(a,c)
father(a,b)     father(a,c)
male(a)         female(c)

parent(d,b)
mother(d,b)
female(d)
```

- Sought:

```
father(X,Y) :- parent(X,Y) & male(X)
mother(X,Y) :- parent(X,Y) & female(X),
```

Rule Induction

- Given: Data set containing variables *Education*, *Marital Status*, *Sex*, *Has Children* and *Car*.
- Sought:

```
IF   Sex = Male
  AND Has Children = Yes
THEN  Car = Family.
```

FOIL (First Order Inductive Learner)

```
FOIL(𝒫, 𝒩)
LearnedRules <— {}
while (𝒫 not empty):
    //Learn a new rule
    NewRule <— Rule without any conditions
    CoveredNegs <— 𝒩
    while (CoveredNegs not empty):
        //Add a literal to NewRule
        PossibleLiterals <— Set of all possible literals
        BestLiteral <— argmax_{l ∈ PossibleLiterals} Gain(l)
        NewRule.append(BestLiteral)
        CoveredNegs <— Subset of CoveredNegs satisfying NewRule

    LearnedRules.append(NewRule)
    𝒫 <— Subset of 𝒫 not covered by LearnedRules

return LearnedRules
```

RIPPER (Repeated Incremental Pruning to Produce Error Reduction)

RIPPER($\mathcal{P}, \mathcal{N}, k$**)**
LearnedRules <— **GenerateRuleSet(**$\mathcal{P}, \mathcal{N}$**)**
repeat k times:
    LearnedRules <— **OptimizeRuleSet(**LearnedRules,$\mathcal{P}, \mathcal{N}$**)**

return LearnedRules

```
GenerateRuleSet(𝒫,𝒩)
LearnedRules <— {}
DL <— DescriptionLength(LearnedRules,𝒫,𝒩)
while (𝒫 not empty):
    //Grow and prune a new rule
    split (𝒫,𝒩) into (𝒫_grow,𝒩_grow) and (𝒫_prune,𝒩_prune)
    NewRule <— GrowRule(𝒫_grow,𝒩_grow)
    NewRule <— PruneRule(NewRule,𝒫_prune,𝒩_prune)
    LearnedRules.append(NewRule)
    if (DescriptionLength(LearnedRules,𝒫,𝒩) > DL + 64):
        //Prune the whole rule set and exit
        for each rule ℛ in LearnedRules in reversed order:
            if (DescriptionLength(LearnedRules\{ℛ},𝒫,𝒩) < DL):
                LearnedRules.delete(ℛ)
                DL <— DescriptionLength(LearnedRules,𝒫,𝒩)
        return LearnedRules
    DL <— DescriptionLength(LearnedRules,𝒫,𝒩)
    𝒫 <— Subset of 𝒫 not covered by LearnedRules
    𝒩 <— Subset of 𝒩 not covered by LearnedRules
return LearnedRules
```

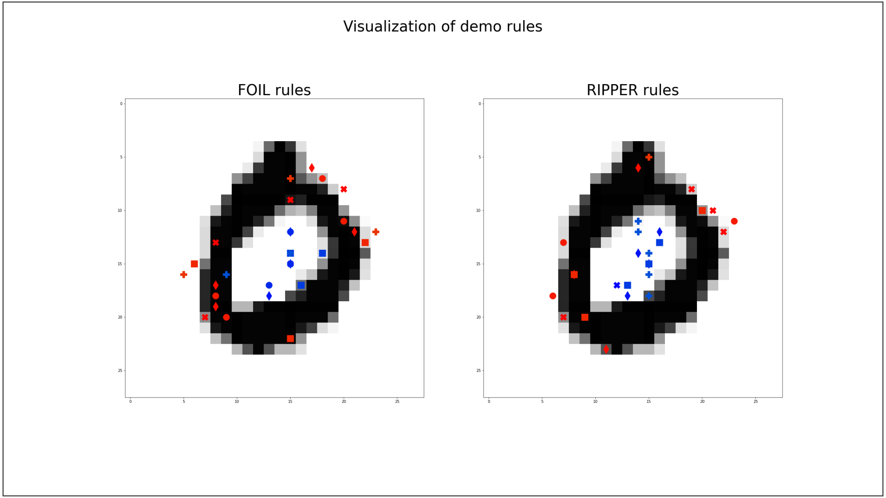
OptimizeRuleSet(LearnedRules,$\mathcal{P}$,$\mathcal{N}$**)**

```
for each rule R in LearnedRules:
    LearnedRules.delete(R)
    P_unc <— Subset of P not covered by LearnedRules
    N_unc <— Subset of N not covered by LearnedRules
    split (P_unc,N_unc) into (P_grow,N_grow) and (P_prune,N_prune)
    ReplRule <— GrowRule(P_grow,N_grow)
    ReplRule <— PruneRule(ReplRule,P_prune,N_prune)
    RevRule <— GrowRule(P_grow,N_grow,R)
    RevRule <— PruneRule(RevRule,P_prune,N_prune)
    OptimizedRule <— better of ReplRule and RevRule
    LearnedRules.append(OptimizedRule)

return LearnedRules
```

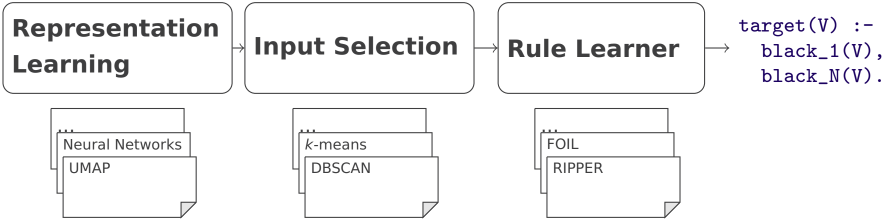
RIPPER (Repeated Incremental Pruning to Produce Error Reduction)

```
RIPPER(𝒫, 𝒩, k)
LearnedRules <— GenerateRuleSet(𝒫, 𝒩)
repeat k times:
    LearnedRules <— OptimizeRuleSet(LearnedRules, 𝒫, 𝒩)

return LearnedRules
```

Illustration of demo rules



General framework



Input Selection via Clustering

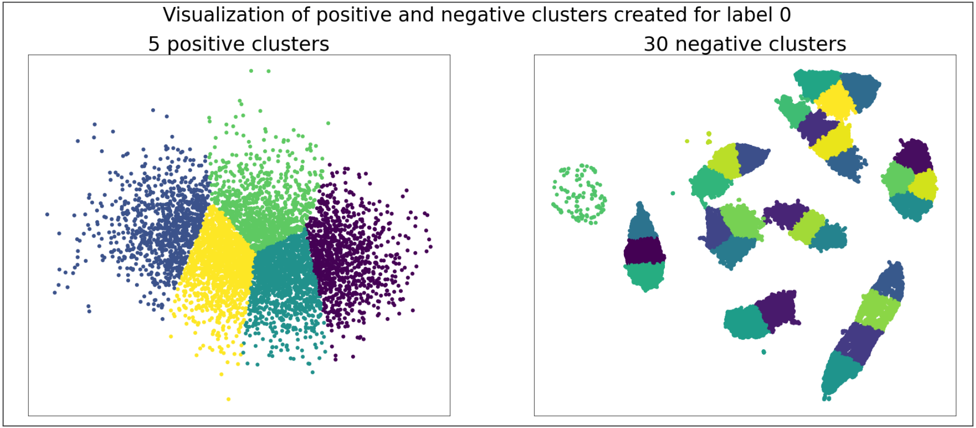
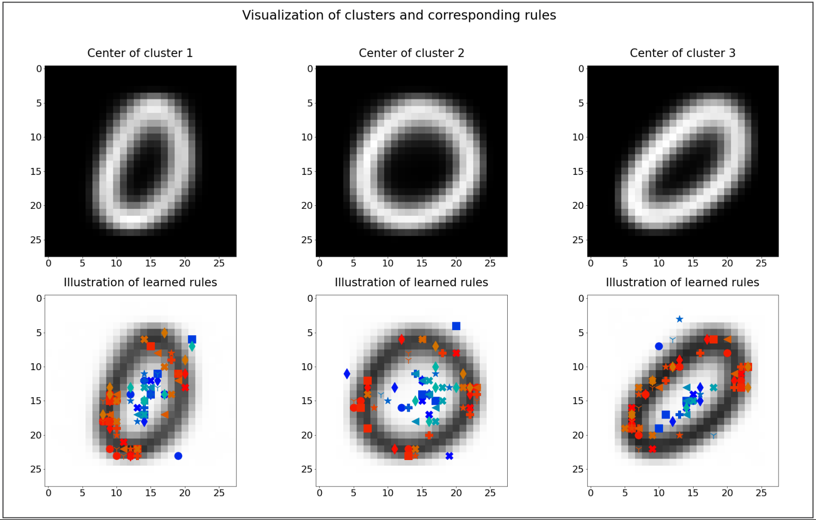


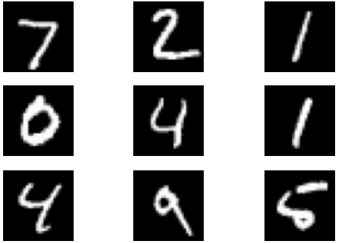
Figure: Visualization of the first and second principal components of the positive and negative clusters obtained with the example of zeros in the MNIST data set.

Advantages of Clustering



Evaluation: Benchmarks

- **MNIST**



Evaluation: Benchmarks

- MNIST
- **Fashion-MNIST**

Evaluation: Benchmarks

- MNIST
- Fashion-MNIST
- **IMDB Movie Reviews**

actor	**annoy**	**great**	...
0	1	0	...
0	0	1	...
1	1	0	...
⋮	⋮	⋮	...

Evaluation: Benchmarks

- MNIST
- Fashion-MNIST
- IMDB Movie Reviews
- **Allianz - Dental Bills**

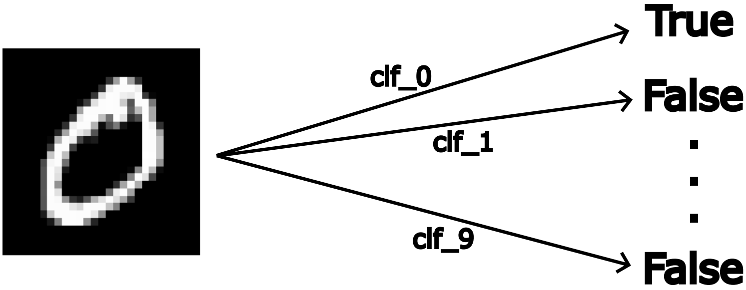
⋮	⋮	...
amount 1	date 1	...
amount 2	date 2	...
amount 3	date 3	...
⋮	⋮	...

Multiclass Classification

1. Construct binary classifiers (i.e. learn rules) for each possible label.

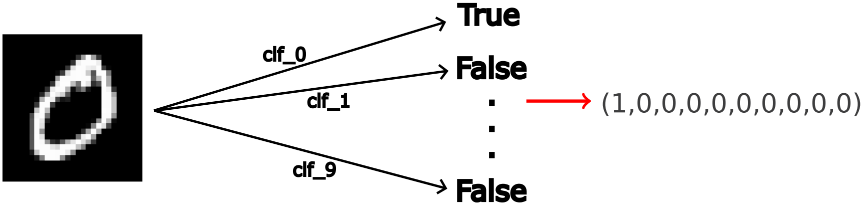
Multiclass Classification

1. Construct binary classifiers (i.e. learn rules) for each possible label.
2. Use each binary classifier to predict the class of a given input.



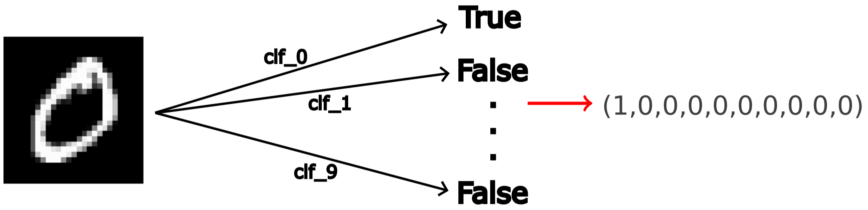
Multiclass Classification

1. Construct binary classifiers (i.e. learn rules) for each possible label.
2. Use each binary classifier to predict the class of a given input.
3. Combine the predictions to a binary vector (one hot encoding).



Multiclass Classification

1. Construct binary classifiers (i.e. learn rules) for each possible label.
2. Use each binary classifier to predict the class of a given input.
3. Combine the predictions to a binary vector (one hot encoding).



4. Return the corresponding multiclass prediction.

Experimental Results

MNIST

learner	time	speedup	accuracy	state-of-the-art
FOIL	1.960 s		94,3%	$\sim$ 99%
FOIL - modular	309 s	6,3	94,3%	$\sim$ 99%
RIPPER	8.300 s		91,65%	$\sim$ 99%
RIPPER - modular	2.770 s	3	91,56%	$\sim$ 99%

Experimental Results

Fashion-MNIST

learner	time	speedup	accuracy	state-of-the-art
FOIL	2.540 s		84,6%	$\sim$ 96%
FOIL - modular	470 s	5,4	84,6%	$\sim$ 96%
RIPPER	11.350 s		82%	$\sim$ 96%
RIPPER - modular	3.700 s	3	83,3%	$\sim$ 96%

Experimental Results

IMDB Movie Reviews

learner	time	speedup	accuracy	state-of-the-art
FOIL	2.221 s		76,1%	$\sim$ 97%
FOIL - modular	218 s	10,2	75,0%	$\sim$ 97%
RIPPER	914 s		71,2%	$\sim$ 97%
RIPPER - modular	356 s	2,6	75,2%	$\sim$ 97%

Experimental Results

Allianz - Dental Bills

learner	time	speedup	accuracy	state-of-the-art
FOIL	238.865 s		80,9%	$\sim$ 91%
FOIL - modular	5.320 s	44,9	80,2%	$\sim$ 91%
RIPPER	66.743 s		86,0%	$\sim$ 91%
RIPPER - modular	18.140 s	3,7	85,9%	$\sim$ 91%

Alternative Learners

- **Rule Induction**
  - IREP (Incremental Reduced Error Pruning): Similar results as its successor RIPPER.

Alternative Learners

- **Rule Induction**
  - IREP (Incremental Reduced Error Pruning): Similar results as its successor RIPPER.
- **Inductive Logic Programming**
  - Progol: Precision / Recall of 91,4% / 71% after more than 200 hours computation time for digit 0.

Alternative Learners

- **Rule Induction**
  - IREP (Incremental Reduced Error Pruning): Similar results as its successor RIPPER.
- **Inductive Logic Programming**
  - Progol: Precision / Recall of 91,4% / 71% after more than 200 hours computation time for digit 0.
  - Aleph: Overfitting and extensive time consumption for vector length > 15.

Alternative Learners

- **Rule Induction**
  - IREP (Incremental Reduced Error Pruning): Similar results as its successor RIPPER.
- **Inductive Logic Programming**
  - Progol: Precision / Recall of 91,4% / 71% after more than 200 hours computation time for digit 0.
  - Aleph: Overfitting and extensive time consumption for vector length > 15.
  - ∂ILP: Restriction to binary clauses.

Alternative Learners

- **Rule Induction**
  - IREP (Incremental Reduced Error Pruning): Similar results as its successor RIPPER.
- **Inductive Logic Programming**
  - Progol: Precision / Recall of 91,4% / 71% after more than 200 hours computation time for digit 0.
  - Aleph: Overfitting and extensive time consumption for vector length > 15.
  - ∂ILP: Restriction to binary clauses.
  - ILASP, FastLAS, Popper: Extensive time consumption or even unsolvable.

Current Research

- Choice of **negative representatives**.
- Application of rule learners on **text-based data**.
- **Introduction of a *Voting System*:** Whenever FOIL and RIPPER do not yield the same prediction, we let a (state-of-the-art) neural network decide. Explanation is given by the corresponding learner.

Related Work

- Mitra, A., Baral, C.: *Incremental and Iterative Learning of Answer Set Programs from Mutually Distinct Examples*. In: Theory and Practice of Logic Programming 18 (2018).
- Nguyen, H.D., Sakama, C.: *Feature learning by least generalization*. In: Inductive Logic Programming - 30th International Conference, ILP 2021.
- Burkhardt, S., Brugger, J., Wagner, N., Ahmadi, Z., Kersting, K., Kramer, S.: *Rule extraction from binary neural networks with convolutional rules for model validation.* In: Frontiers in Artificial Intelligence 4 (2021).
- Granmo, O.-C., Glimsdal, S., Jiao, L., Goodwin, M., Omlin, C.W., Berge, G.T.: *The Convolutional Tsetlin Machine* (2019).
- Eldbib, K.: *Design and analysis of rule induction systems*, University of Birmingham, UK, 2016.

Thank you for your attention!

Albert Nössig, Georg Moser, Tobias Hell

https://tcs-informatik.uibk.ac.at/