





Comparative Evaluation of Approaches to Propositionalization

Mark-A. Krogel, Otto-von-Guericke-Universität Magdeburg

Simon Rawles, University of Bristol

Filip Zelezný, Czech Technical University and University of Wisconsin, Madison

Peter A. Flach, University of Bristol

Nada Lavrač, Institute Jozef Stefan, Ljubljana

Stefan Wrobel, Friedrich-Wilhelms-Universität Bonn and Fraunhofer-Institut AiS

Introduction

- Propositionalization:
  largely automatic transformation of relational data
  into a single-table representation
  and application of propositional learners

- In principle less powerful than searching
  full first-order hypothesis space

- In practice often sufficient, efficient, and flexible

- Here: first comparative study using representatives
  of logic-oriented approaches (RSD, SINUS)
  and database-oriented approaches (RELAGGS)

Propositionalization

- An ILP learning task:
  given ground facts of target predicate (examples) and
  clauses of background predicates,
  find hypothesis to explain together with background theory
  some properties of examples

- Complete vs. partial approches,
  general-purpose vs. special-purpose approaches

- Clauses constructed from relational background knowledge
  and structural properties of individuals,
  calls of clauses for individuals produce feature values

RSD

- Declarative bias similar to Progol/Aleph, e.g.

  :-modeb(3,hasCar(+train,-car).

- Step 1: identification of all **closed** feature definitions (Prolog queries) corresponding to declarations

  hasCar(Train,Car), shape(Car,Shape), instantiate(Shape)

- Step 2: instantiation of variables plus feature filtering, e.g.

  hasCar(Train,Shape), shape(Shape,bucket)

- Step 3: creation of propositionalized representation

RSD: Constraints & Pruning

- Language
  - argument modes & types, predicate recall
  - max feature length & variable depth
  - **undecomposability**: *f1 <> f2 & f3*
- Evaluation
  - non-triviality: |*cov(f)*| < |*Data*|
  - relevance: |*cov(f)*| > *min*
  - uniqueness: if *cov(f1)* = *cov(f2)* then discard the longer
- Pruning:
  - large subspaces identified containing only decomposable f.
    - eg. EW Trains: *SearchTime -> +inf as MaxLength -> +inf*
    - with pruning: *SearchTime -> const as MaxLength -> +inf*
  - if |cov(f)| < min then don't refine f

SINUS: Overview

- Developed from LINUS and its feature generation extension
- A modular transformational ILP experimentation platform
  - Automated type construction
  - Feature reduction
  - Invocation of learner and back-translation of induced theory to first-order form.
- Data as flattened Prolog facts + data definition
  - Declarative bias similar to 1BC, e.g.
    train 1 train cwa
    train2car 2 1:train *:#car * cwa
    cshape 2 car #shape * cwa

SINUS: Step by step

- Step 1: construction of instantiated feature definitions, e.g. f_aaaa(A) :- train(A), hasCar(A,B),shape(B,bucket). Recursive left-to-right considering current variable types and bindings.
  - Constraining maximum literals, variable, values in a type and the nature of variable reuse.
- Step 2: feature set reduction (REDUCE)
- Step 3: creation of propositionalized representation
- After learning: result transformation into first-order hypothesis

RELAGGS

- Declarative bias from foreign key relationships
  in relational database schema

- After example identifier propagation to non-target relations:

- Step 1: summarize each non-target relation by example id,
  avg, max, min, sum, stdev, range, quartiles for numeric data,
  count possible values for nominal attributes,
  plus some two-column aggregates

- Step 2: creation of propositionalized representation by
  concatenating aggregate function values to target relation

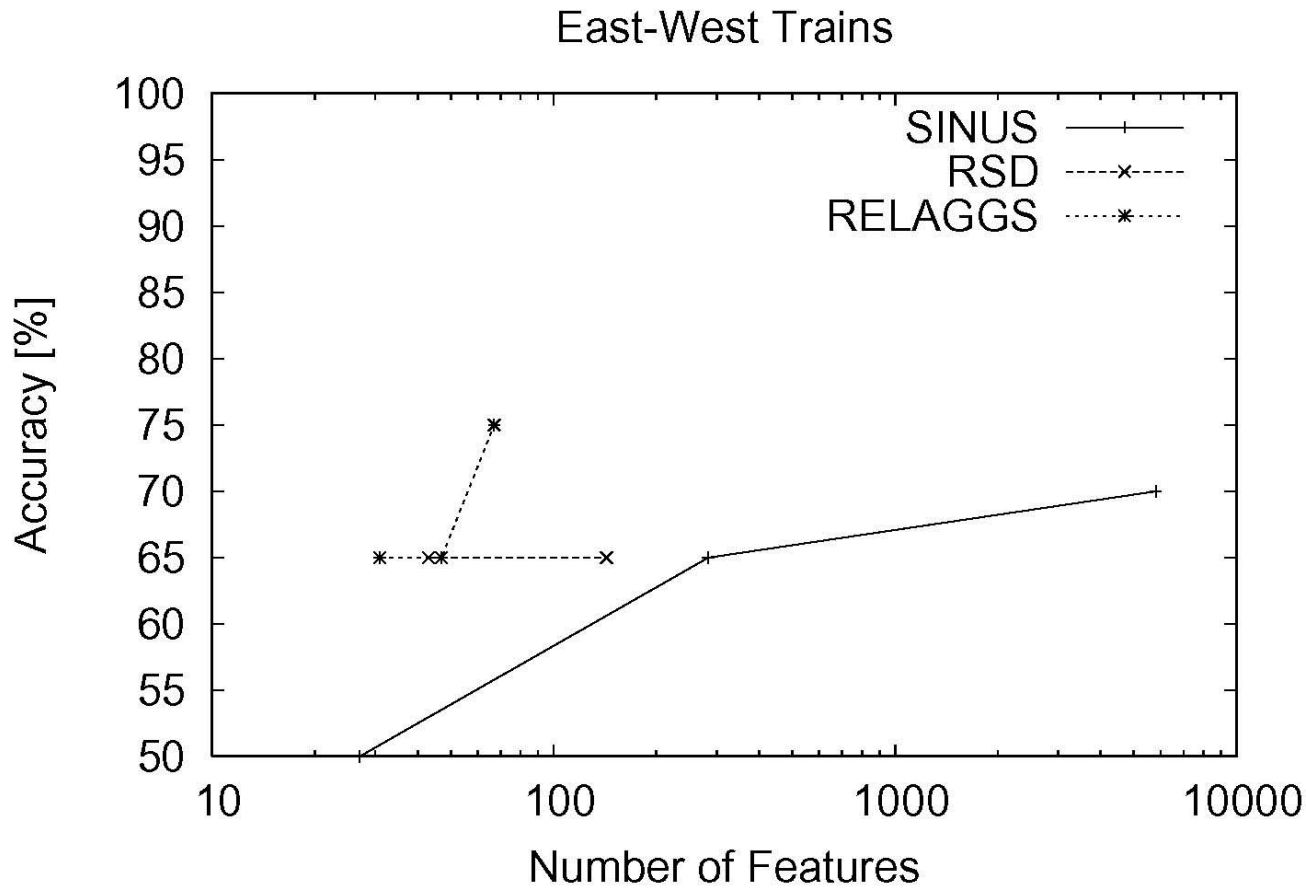
Learning Tasks

- Trains: 20 trains east- or west-bound?
- King-Rook-King: 1000 board states legal or not?
- Mutagenesis: 188 molecules mutagenic or not?
- PKDD Challenges 1999/2000: 682 loans problematic or not?
- KDD Cup 2001: 862 genes/proteins with certain function or not and with certain localization or not?
- Numbers of predicates/relations depend on modeling issues.

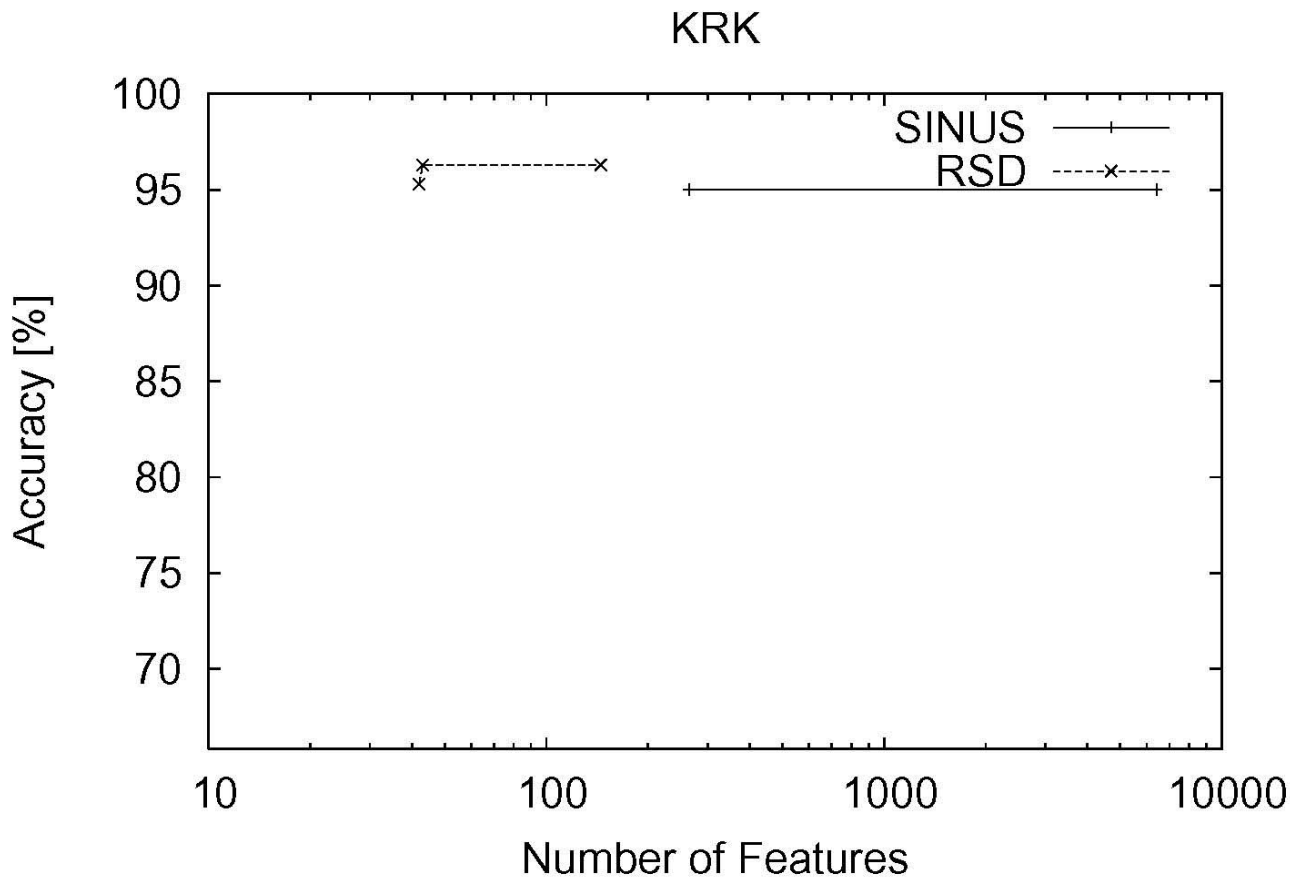
Procedure

- Mostly starting point: Prolog representation of target predicate facts and background predicate definitions, SQL scripts generated from those if necessary
- Manual construction of declarations, propagation of id‘s if necessary
- Application of RSD, SINUS, and RELAGGS to produce single-table representations of relational input data, with different parameter settings to produce feature sets of different sizes
- Application of WEKA‘s J48 (10-fold stratified cross-validation) to those tables

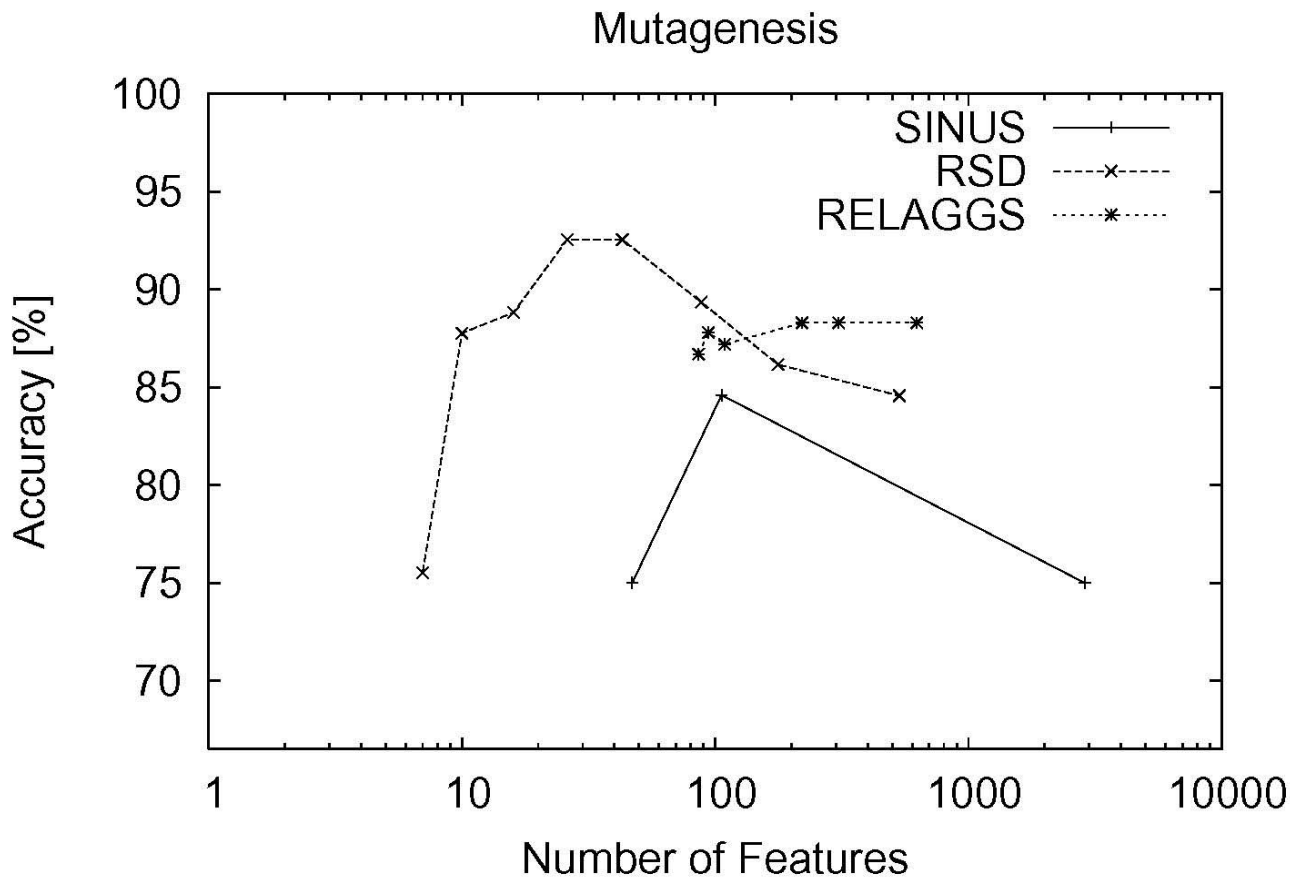
Results: Accuracies (1)



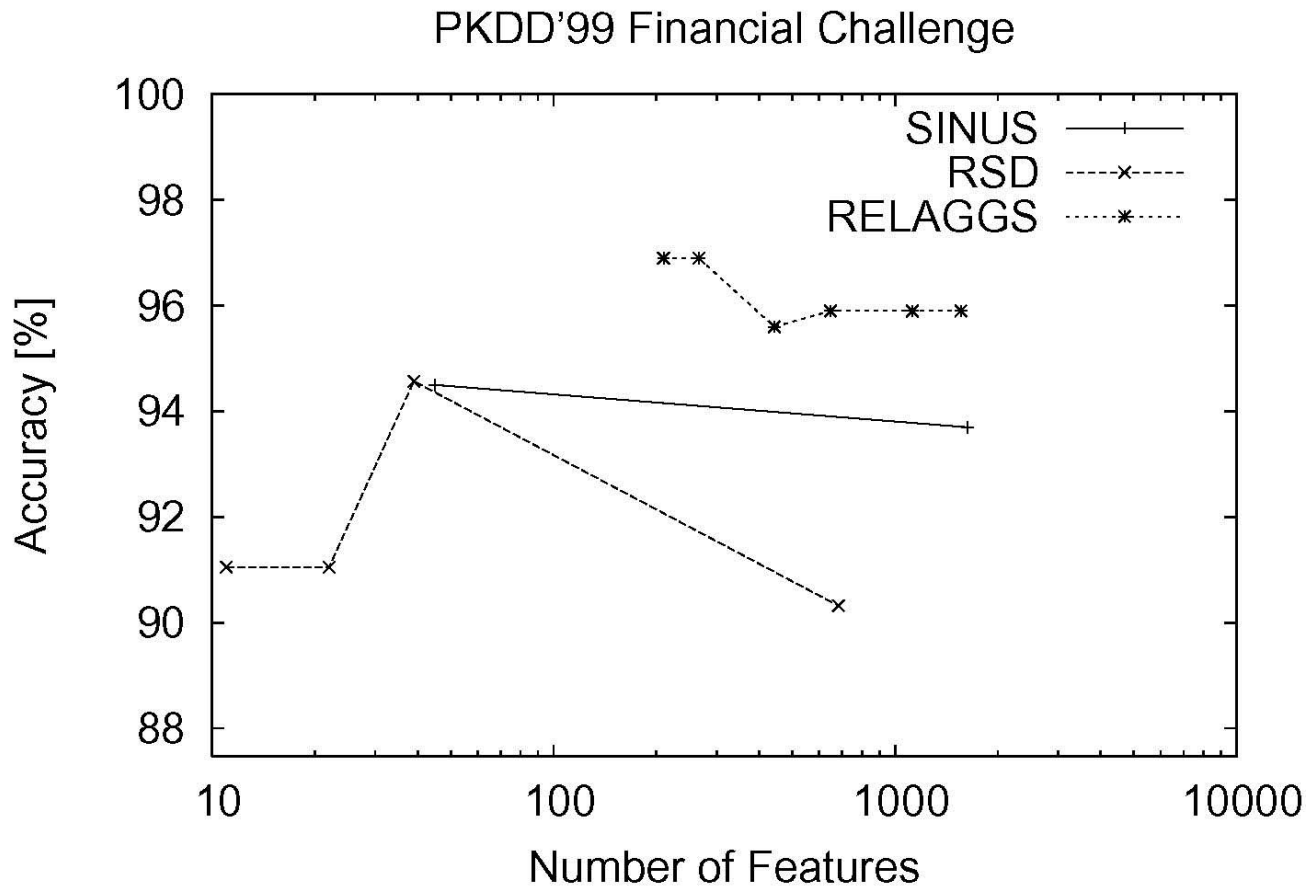
Results: Accuracies (2)



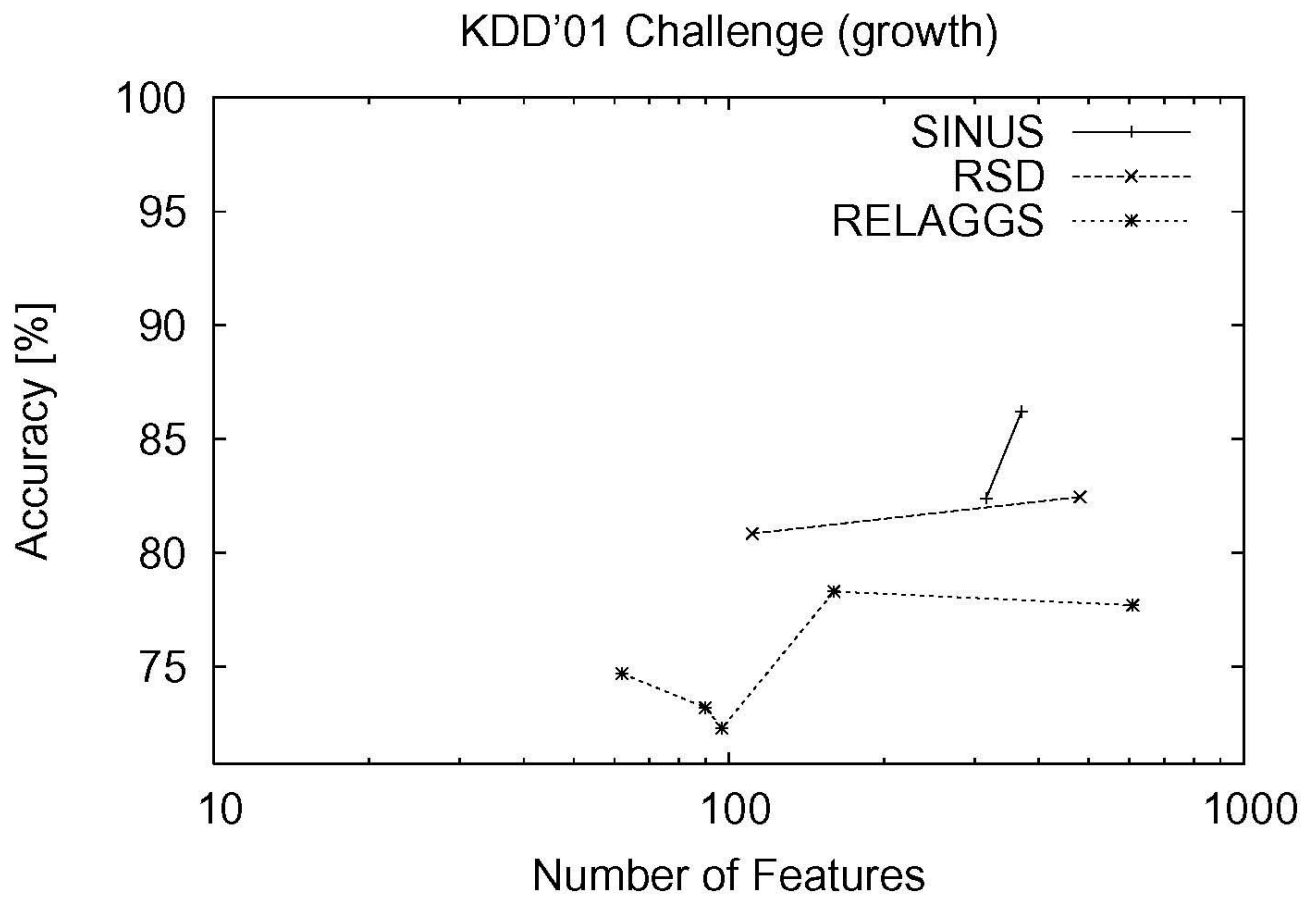
Results: Accuracies (3)



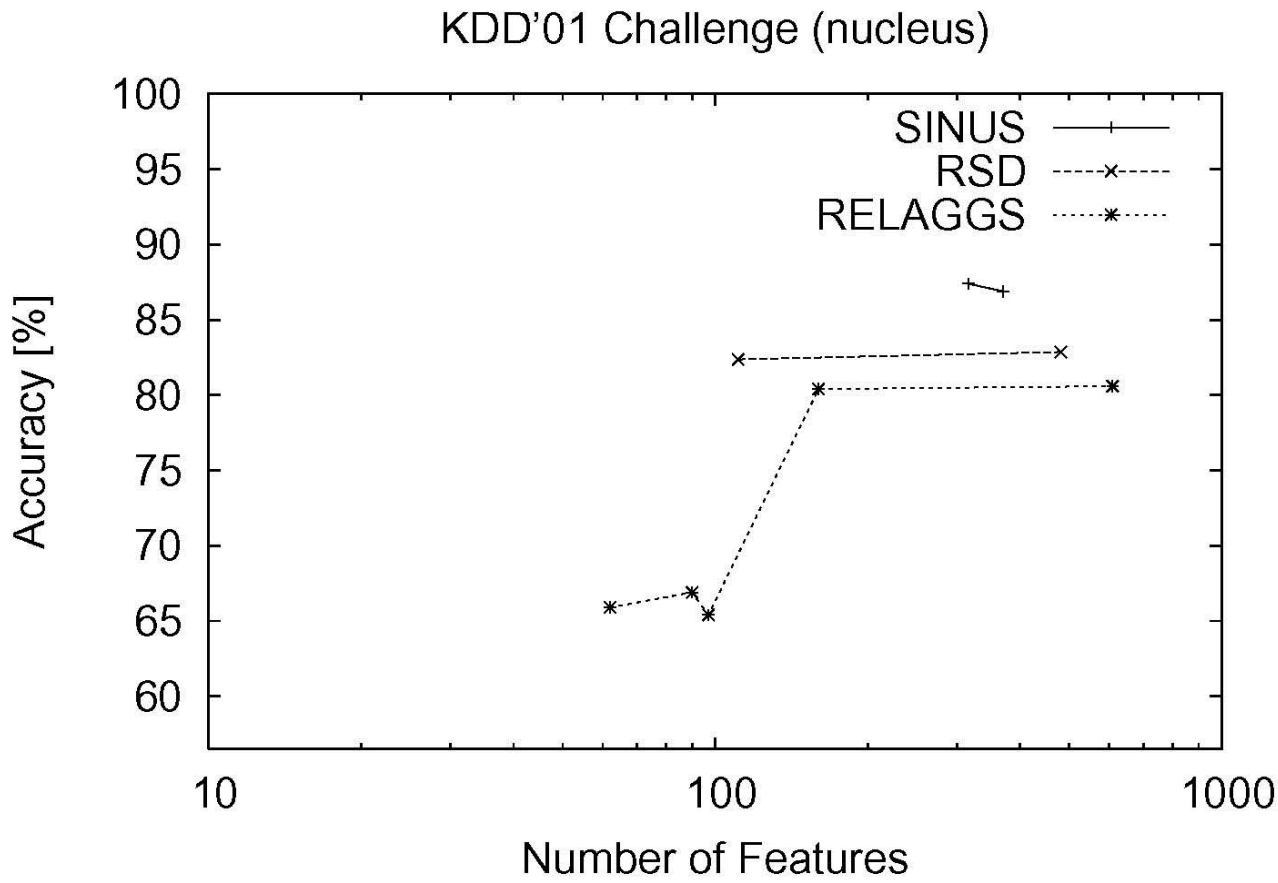
Results: Accuracies (4)



Results: Accuracies (5)



Results: Accuracies (6)



Results: Runtimes

- Different platforms, hence times only indicators
-

	RSD	SINUS	RELAGGS
Trains	< 1 sec	2 - 10 min	< 1 sec
King-Rook-King	< 1 sec	2 - 6 min	n. a.
Mutagenesis	5 min	6 - 15 min	30 sec
PKDD99-00	5 sec	2 – 30 min	30 sec
KDD01 fct	3 min	30 min	1 min
KDD01 loc	3 min	30 min	1 min

Discussion

- Not generally conclusive in favor of any approach:
  each winner on two tasks
- Aggregation strong in some domains,
  where counting features are relevant (Trains)
  or many numeric attributes exist in the original data
- Differences between RSD and SINUS mainly due
  to differences in constraining the language bias
- RELAGGS most efficient for many tasks, differences between
  RSD and SINUS possibly caused by pruning or Prolog systems

Related Work

- LINUS/DINUS (Lavrač and Džeroski 1994)
- Stochastic propositionalization (Kramer et al. 1998)
- Bottom-up propositionalization (Kramer 2000)
- Lazy propositionalization (Alphonse and Rouveirol 2000)
- ...

Future Work and Conclusion

- General:
  - Completion of formal framework
  - Comparison to other ILP approaches such as Progol and Tilde
  - Extension of feature subset selection mechanisms
  - Experiments with other propositional learners such as SVMs
  - Combination of the features produced by the approaches here

- RSD: construction of first-order hypotheses
- SINUS: improvements of feature elimination, bias control
- RELAGGS: integration with dynamic relational databases

- Promising approaches with many questions left open!