Rule-based Regression

Frederik Janssen

TECHNISCHE UNIVERSITÄT DARMSTADT

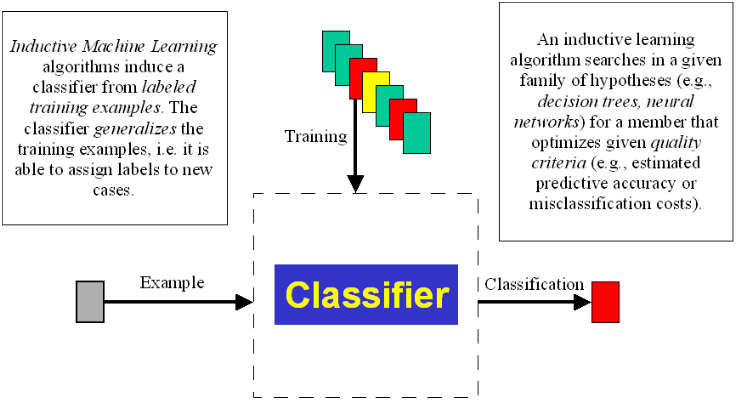
Outline

1. Machine Learning
   - Classifiers
   - Data Representation
   - Concept Representation
2. Separate-and-conquer Rule Learning
3. Regression
   - Regression by classification
   - Regression measures
4. Current implementation
5. Some results
6. Discussion

Machine Learning

- Definition (Mitchell, 1997)
  - "A computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."
- Given:
  - a task T
  - a performance measure P
  - some experience E with the task
- Goal:
  - generalize the experience in a way that allows to improve your performance on the task

Indroduction of Classifiers



Introduction of Classifiers

TECHNISCHE UNIVERSITÄT DARMSTADT

The most "popular" learning problem:

- Task:
  - learn a model that predicts the outcome of a dependent variable for a given instance
- Experience:
  - experience is given in the form of a data base of examples
  - an example describes a single previous observation
    - *instance:* a set of measurements that characterize a situation
    - *label:* the outcome that was observed in this situation
- Performance Measure:
  - compare the predicted outcome to the observed outcome
  - estimate the probability of predicting the right outcome in a new situation

Data Representation

Attribute-Value Data

- Each example is decribed with values for a fixed number of attributes (also called features)
  - **Nominal Attributes:**
    - store an unordered list of symbols (e.g., *color*)
  - **Numeric Attributes:**
    - store a number (e.g., *income*)

A sample task

TECHNISCHE UNIVERSITÄT DARMSTADT

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes
today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

A sample task

possible rules:

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

play=no $\leftarrow$ temperature $\geq$ 25.5 $\wedge$ temperature $<$ 28.5
play=no $\leftarrow$ temperature $<$ 14 $\wedge$ temperature $\geq$ 9.5
play=no $\leftarrow$ outlook=rainy $\wedge$ windy=true

A sample task

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

possible rules:

play=no $\leftarrow$ temperature $\geq$ 25.5 $\wedge$ temperature $<$ 28.5
play=no $\leftarrow$ temperature $<$ 14 $\wedge$ temperature $\geq$ 9.5
play=no $\leftarrow$ outlook=rainy $\wedge$ windy=true

but also (t=temperature):

play=no $\leftarrow$ t $<$ 26.5 $\wedge$ t $\geq$ 25.5 $\wedge$ outlook=sunny $\wedge$ humidity=high $\wedge$ windy=false
play=no $\leftarrow$ t $<$ 28.5 $\wedge$ t $\geq$ 27.5 $\wedge$ outlook=sunny $\wedge$ humidity=high $\wedge$ windy=true
...

Separate-and-conquer Rule Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- Separate-and-conquer (or Covering) paradigma (originated from the AQ algorithm (Michalski, 1969))
- still used in most Rule Learning systems (e.g., RIPPER (Cohen, 1995))

1. **Generalization:** extend the current theory by a "good" rule
2. **Separate:** remove all examples covered by this rule
3. **Conquer:** if examples left, *goto* 1.

- rules are combined in a decision list
  - sorted list of rules
  - the first rule that "covers" the example is used to classify the example
  - if no rule covers the example the last rule is used as a default rule (predicts the majority class)

Searching for a single rule

TECHNISCHE UNIVERSITÄT DARMSTADT

- generate the first rule that covers all examples
- generate all refinements of the current rule by creating all attribute-value pairs from the data
  - nominal attributes: use equality tests (i.e., $=$)
  - numerical attributes: use inequality tests (i.e., $\geq$ and $<$)
- add each refinement to the current rule and test which is the best for a given (heuristic) criterion
- if a new best is found store it
- if the error of the rule is 0 stop the process and return the best rule that was found during this process

Combining rules in a decision list

- if a rule is found add the rule to the sorted list of rules
- remove all the examples that are covered by the rule
- if all but the remaining n examples are covered stop inducing rules (currently $n = 1$)
- else: search for the next rule on the remaining examples
- as last rule add a default rule that predicts the majority class

Rule Learning Heuristics

- Rule Learning Heuristics implement the criterion for evaluating rules
- many Rule Learning Heuristics for classification are known (based on positive and negative examples)
- Parametrized trade-off between
  - *Consistency*: $(1 - \textit{error})$ of the rule and
  - *Coverage*: how many examples are covered by the rule
- Heuristics for Regression (positive and negative examples are not known here) rely on
  - the current error/loss (Consistency in classification) of the rule
  - the coverage of the rule
- Regression Heuristics may also feature a parameter that trades off between the error and the Coverage of the rule

From Classification to Regression

- instead of predicting a discrete outcome in Regression the outcome is continuous
- 2 ways to deal with this:
  1. discretize numeric outcome and use standard classification algorithms
     - problem: number of classes has to be known in advance
     - algorithm used to discretize: P-CLASS (Weiss and Indurkhya, 1995))
  2. adapt the algorithm to Regression tasks
- example for an adaption in Rule Learning
  - either predict a certain value (*Median* or *Mean*) in the head of the rule directly (like we did)
  - or use a (linear) model in the head to predict the value (algorithm M5RULES (Holmes, Hall, and Frank, 1999), (Quinlan, 1992))

Regression measures

- Mean Absolute Error $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \bar{y}_i|$
- Mean Squared Error $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2$
- Deviation from Mean $def = \frac{1}{n}\sum_{i=1}^{n}(y_i - y')^2$
- Normalized Mean Squared Error $NMSE = {}^{MSE}/_{def}$
- Relative Coverage $RC = {}^{\text{Coverage}(r)}/_{n}$
- Relative Cost Measure $h_{rcm} = c \cdot (1 - NMSE) + (1 - c) \cdot RC$

where $n = \#$ of examples left, $y_i =$ true value, $\bar{y}_i =$ predicted value, $y' =$ mean of all instances, $r =$ the current rule

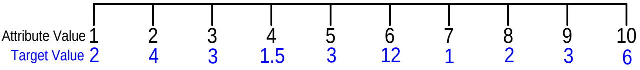
Current implementation

- numerical and nominal attributes, numerical target variable
- covering paradigma
- interchangeable heuristics and splitpoint computing methods
- parameters:
  - parameter of the heuristic
  - parameter for splitpoint computation
    - to reduce the number of splitpoints for a numerical attribute a clustering was used
    - the parameter determines how many clusters are computed
  - percentage of coverage of ruleset (for inducing the default rule)
    - currently all but the last remaining example has to be covered

Handling of numerical attributes

TECHNISCHE
UNIVERSITÄT
DARMSTADT

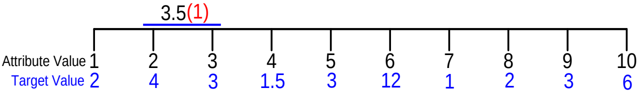
- if all possible splitpoints (those between 2 instances) for all numeric attributes are used the search space explodes
- remedy: do not create all splitpoints but cluster examples together that minimize some error criterion
- and use only the splitpoints between these clusters (currently about 5-10)
- Algorithm:
  - sort the examples of the attribute in ascending order
  - remove duplicates by setting the mean over all duplicates as target value
  - merge examples that minimize the mean absolute error



KE

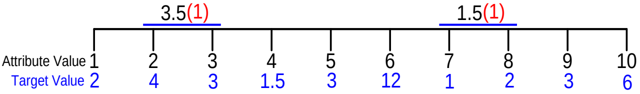
Handling of numerical attributes

- if all possible splitpoints (those between 2 instances) for all numeric attributes are used the search space explodes
- remedy: do not create all splitpoints but cluster examples together that minimize some error criterion
- and use only the splitpoints between these clusters (currently about 5-10)
- Algorithm:
  - sort the examples of the attribute in ascending order
  - remove duplicates by setting the mean over all duplicates as target value
  - merge examples that minimize the mean absolute error

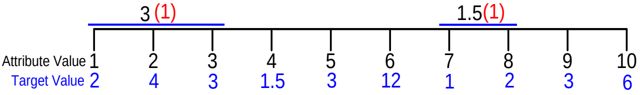


Handling of numerical attributes

- if all possible splitpoints (those between 2 instances) for all numeric attributes are used the search space explodes
- remedy: do not create all splitpoints but cluster examples together that minimize some error criterion
- and use only the splitpoints between these clusters (currently about 5-10)
- Algorithm:
  - sort the examples of the attribute in ascending order
  - remove duplicates by setting the mean over all duplicates as target value
  - merge examples that minimize the mean absolute error

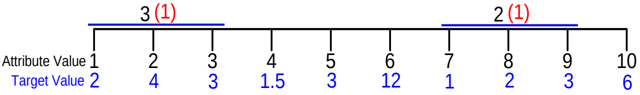


- if all possible splitpoints (those between 2 instances) for all numeric attributes are used the search space explodes
- remedy: do not create all splitpoints but cluster examples together that minimize some error criterion
- and use only the splitpoints between these clusters (currently about 5-10)
- Algorithm:
  - sort the examples of the attribute in ascending order
  - remove duplicates by setting the mean over all duplicates as target value
  - merge examples that minimize the mean absolute error



Handling of numerical attributes

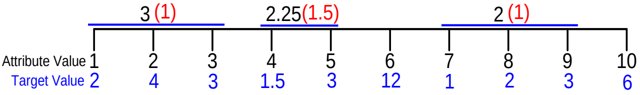
- if all possible splitpoints (those between 2 instances) for all numeric attributes are used the search space explodes
- remedy: do not create all splitpoints but cluster examples together that minimize some error criterion
- and use only the splitpoints between these clusters (currently about 5-10)
- Algorithm:
  - sort the examples of the attribute in ascending order
  - remove duplicates by setting the mean over all duplicates as target value
  - merge examples that minimize the mean absolute error



Handling of numerical attributes

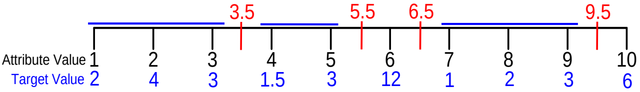
TECHNISCHE UNIVERSITÄT DARMSTADT

- if all possible splitpoints (those between 2 instances) for all numeric attributes are used the search space explodes
- remedy: do not create all splitpoints but cluster examples together that minimize some error criterion
- and use only the splitpoints between these clusters (currently about 5-10)
- Algorithm:
  - sort the examples of the attribute in ascending order
  - remove duplicates by setting the mean over all duplicates as target value
  - merge examples that minimize the mean absolute error



Handling of numerical attributes

- if all possible splitpoints (those between 2 instances) for all numeric attributes are used the search space explodes
- remedy: do not create all splitpoints but cluster examples together that minimize some error criterion
- and use only the splitpoints between these clusters (currently about 5-10)
- Algorithm:
  - sort the examples of the attribute in ascending order
  - remove duplicates by setting the mean over all duplicates as target value
  - merge examples that minimize the mean absolute error



KE

Evaluation of the models

TECHNISCHE UNIVERSITÄT DARMSTADT

- for domain-dependent evaluation we used MAE and $RMSE = \sqrt{MSE}$
- for domain-independent evaluation we used the correlation coefficient (between predicted and actual value)
- we also record model complexity by measuring the number of rules and conditions (for rule based models)
- 1x10 cross-validation with same folds for each model
- our approach was compared to M5RULES, LINEAR REGRESSION, SVMREG (all implemented in *weka* (Witten and Frank, 2005))

Results

In terms of MAE

- preliminary results ($sp = 10, c = 0.45$) for 13 datasets from the UCI-Repository (Asuncion and Newman, 2007)
- second number describes standard deviation among the 10 folds of the CV

dataset	SeCo	M5Rules	Linear Regression	SVMReg
auto-horse	16.61 ± 6.35	15.85 ± 10.25	13.64 ± 3.24	13.48 ± 4.0
auto-mpg	4.44 ± 1.49	3.03 ± 0.81	2.87 ± 0.98	2.83 ± 0.98
auto-price	2526.6 ± 773.1	2157.8 ± 937.4	2450.5 ± 1084.0	2292.32 ± 1012.05
breast-tumor	8.02 ± 0.73	7.79 ± 0.74	7.9 ± 0.72	8.2 ± 0.76
cloud	0.45 ± 0.15	0.3 ± 0.12	0.27 ± 0.07	0.28 ± 0.09
cpu	36.80 ± 29.38	15.19 ± 9.17	47.7 ± 20.89	24.95 ± 23.52
echo-month	13.41 ± 2.62	8.68 ± 2.97	8.48 ± 3.15	9.08 ± 2.73
housing	5.43 ± 2.98	3.39 ± 1.44	3.99 ± 2.13	3.73 ± 2.05
meta	95.59 ± 170.29	232.52 ± 190.24	146.54 ± 148.08	96.91 ± 166.08
sensory	0.64 ± 0.13	0.73 ± 0.14	0.76 ± 0.18	0.77 ± 0.19
servo	0.54 ± 0.15	0.32 ± 0.11	0.62 ± 0.12	0.53 ± 0.17
strike	274.61 ± 116.46	287.0 ± 87.31	264.73 ± 84.0	228.49 ± 83.23
veteran	91.28 ± 59.05	92.99 ± 44.4	92.99 ± 44.4	82.58 ± 54.89
average rank	3.08	2.27	2.58	2.08

Results

In terms of different parametrizations

- the number of splitpoints are fixed to 10 but the parameter of the heuristic is varied
- lowest errors are marked blue

dataset	$c = 0.45$		$c = 0.5$		$c = 0.6$		$c = 0.7$	
	MAE	# rules	*MAE*	# rules	*MAE*	# rules	*MAE*	# rules
auto-horse	16.6	2	15.2	16	21.4	35	16.5	57
auto-mpg	4.44	1	3.92	157	3.62	184	3.64	226
auto-price	2526	6	2922	7	3104	46	2836	48
breast-tumor	8.0	0	8.5	13	10.7	209	10.4	236
cloud	0.45	7	0.46	6	0.42	12	0.39	42
cpu	36.8	5	37.8	7	38.8	9	29.3	15
echo-month	13.4	0	14.2	79	14.2	92	13.2	87
housing	5.43	5	4.7	43	4.54	369	4.47	427
meta	95.6	3	95.2	30	147.8	69	147	124
sensory	0.63	0	0.82	430	0.86	404	0.9	428
servo	0.54	4	0.39	20	0.39	22	0.38	29
strike	274	0	362	234	361	300	368	359
veteran	91	0	116	70	119	82	123	91

Discussion

- our algorithm implements a Separate-and-conquer Regression Rule Learner
- trade-off between consistency and coverage is more complex than it is in classification
  - tuning of the parameters has to be analyzed better
- but the current implementation is competetive to other rule-based implementations (that do not predict models in the head)
- a new splitpoint computing method was introduced
  - only about 10 splitpoints are sufficient for most of the datasets
  - much more faster than computing all splitpoints
  - but optimal cluster number still has to be found

Future Work

TECHNISCHE UNIVERSITÄT DARMSTADT

- this is work-in-progress so there are many ways to improve the algorithm
  - by determine a suitable setting of the cluster parameter
  - by systematically tune the parameter of the heuristic
    - previously we tuned the parameters of 5 heuristics for classification
    - we also want to find the best parameter for regression
  - by avoiding overfitting by leaving more examples uncovered
- predict (linear) models in the head of the rule
- try to visualize the behaviour of the different heuristics in a space similar to *Coverage Spaces*
- include domain-independent comparison with $\text{RRMSE} = \sqrt{\frac{MSE}{def}}$

References

- (Weiss and Indurkhya, 1995): S. M. Weiss and N. Indurkhya. Rule-based Machine Learning Methods for Functional Prediction. *Journal of Artificial Intelligence Research*, 3:383-403, 1995.
- (Holmes, Hall, and Frank, 1999): G.Holmes, M.Hall, and E.Frank. Generating Rule Sets from Model Trees. In *Proceedings of the 12th Australian Joint Conference on Artificial Intelligence*, pages 1-12, Sydney, Australia, 1999.
- (Quinlan, 1992): R.J. Quinlan. Learning with Continuous Classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, pages 343-348, Hobart, Australia, 1992.
- (Michalski, 1969): R.S. Michalski. On the Quasi-Minimal Solution of the Covering Problem. In *Proceedings of the 5th International Symposium on Information Processing*, Volume A3, pages 125-128, Bled, Yugoslavia, 1969.
- (Cohen, 1995): W.W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pages 115-123, Tahoe City, USA, 1995.
- (Mitchell, 1997): T.M.Mitchell. Machine Learning, McGraw-Hill, 1997
- (Asuncion and Newman, 2007): A.Asuncion and D.Newman. UCI machine learning repository, 2007. URL: `http://archive.ics.uci.edu/ml/datasets.html`
- (Witten and Frank, 2005): I.H.Witten and E.Frank. Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, 2nd edition, 2005.