

Lazy Naive Credal Classifier

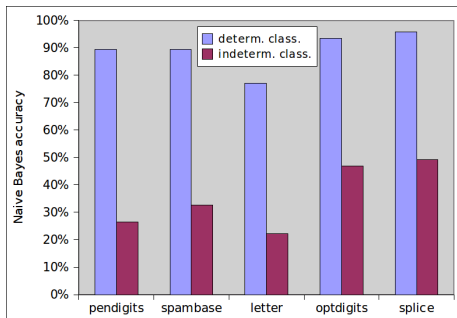
G. Corani M. Zaffalon

IDSIA
Switzerland
`giorgio{zaffalon}@idsia.ch`

U' 09 - Knowledge discovery from uncertain data

Naive Credal Classifier (NCC) (Corani and Zaffalon, JMLR 2008)

- NCC extends naive Bayes to imprecise probabilities.
- NCC returns *indeterminate* classifications (more than one class) on *hard* instances, i.e., for which the evidence does not smooth the effect of the chosen prior (e.g., small samples);
- Naive Bayes is unreliable on such hard instances.



Two issues of NCC

- The naive assumption (statistical independence of the features given the class) can be unrealistic in some domains.
- A sometimes excessive indeterminacy.
- Lazy Naive Credal Classifier (LNCC) is designed to overcome both issues.

Lazy Classifiers

Lazy classifiers do *not* learn until there is an instance to classify (*query*).

Then:

- 1 they rank the instances of the training set according to the distance from the query;
- 2 a local classifier is trained on the k closest instances (k is named *bandwidth*) and issues the classification;
- 3 the local classifier is discarded, while the training set is kept in memory to answer future queries.

Open question: how to select the bandwidth?

Bandwidth selection

The bandwidth is increased until LNCC is determinate, i.e., until the evidence smooths the effect of the choice of the prior.

Pseudo-code

- $k=25$;
- `lncc.train(k)`;
- `while (lncc indeterminate OR $k == \text{trainingSet.size}$)`
 - $k=k+20$;
 - `lncc.train(k)`;
- **end**

Why a lazy NCC (LNCC)?

- working locally *reduces the bias* due to the naive assumption (J. Friedman, 1997; Frank et al., UAI 2003);
- to *increase determinacy*, thanks the design of the *bandwidth selector*.

Comparing credal classifiers: d-acc

A classifier is *accurate* on a certain instance if its output includes the correct class.

- Discounted accuracy (borrowed from multi-label classification):

$$\text{d-acc} = \frac{1}{N} \sum_{i=1}^N \frac{(\text{accurate})_i}{|Z_i|}$$

where $|Z_i|$ is the number of classes returned on the i -th instance.

- d-acc entails some arbitrariness: why not discounting on $|Z_i|^2$?
- The d-acc of two credal classifiers can be compared via t -test.

Comparing credal classifier: a new rank test

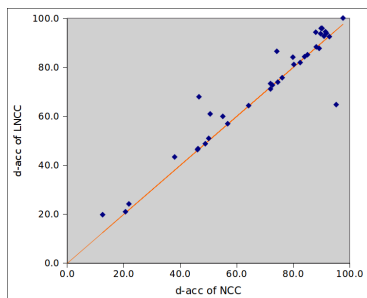
On each instance, we rank the two credal classifiers CR_1 and CR_2 :

CR1	CR2	winner
accurate	not accurate	CR1
accurate	accurate $ Z_i _{CR2} < Z_i _{CR1}$	CR2
accurate	accurate $ Z_i _{CR2} = Z_i _{CR1}$	tie
inaccurate	inaccurate	tie

- Wins, ties and losses are transformed into ranks and are then analyzed via a non-parametric test.
- The rank test avoids the arbitrariness of d-acc but, using less pieces of information, can be less sensitive.

Experiments

- Comparison of LNCC and NCC on 36 data sets.



	<i>LNCC wins</i>	<i>ties</i>	<i>NCC wins</i>
<i>d-acc</i>	19	11	6
<i>rank test</i>	15	19	2
<i>cross-check</i>	15	20	1

Why does LNCC outperform NCC?

- On large data sets the improvement is due to the reduced bias.

Data set	instances	d-acc (NCC)	Δ LNCC
letter	20000	86.5	+12.1
nursery	12960	95.8	+5.6
optdigits	5620	93.9	+1.9
pendigits	10992	94.3	+6.3
waveform	5000	84.0	+4.1

- In other cases there is a considerable improvement of determinacy.

Conclusions

- Thanks to locality and to the design of the bandwidth selector, LNCC overcomes two problems of NCC:
 - the bias of the naive assumption;
 - the (sometimes) excessive indeterminacy;
- We have proposed two metrics for comparing credal classifiers;
- Experiments on 36 data sets show a clear improvement of LNCC over NCC.