Classification with Imprecise Probability

Cassio P. de Campos, Alessandro Antonucci, Giorgio Corani
{cassio,alessandro,giorgio}@idsia.ch

IDSIA - Switzerland

SIPTA School – Durham 2010

A companion chapter

- A companion chapter of this tutorial is available.
- It provides a complete overview of credal networks and credal classifiers.
- Have a look at:
  `www.idsia.ch/idsiareport/IDSIA-02-10.pdf`

Classification

- Fisher, 1936: determine the type of Iris on the basis of length and width of the sepal and of the petal.

(a) setosa (b) virginica (c) versicolor

- Classification: to predict the class C of a given object, on the basis of attributes (or *features*) $\mathbf{A} = \{A_1, \dots, A_k\}$.
- We assume the features to be *discrete*.

Outline

Bayesian estimation for a single variable

- Class C, with sample space $\Omega_C = \{c_1, \ldots, c_m\}$.
- $P(c_j) = \theta_j$, $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_m\}$.
- n i.i.d. observations; $\mathbf{n} = \{n_1, \ldots, n_m\}$.
- *Multinomial* likelihood:

$$L(\boldsymbol{\theta}|\mathbf{n}) \propto \prod_{j=1}^{m} \theta_j^{n_j}$$

- Max. likelihood estimator: $\hat{\theta}_j = \frac{n_j}{n}$.

Dirichlet prior

- The prior expresses the beliefs about $\boldsymbol{\theta}$, *before* analyzing the data.
- Dirichlet prior

$$\pi(\boldsymbol{\theta}) \propto \prod_{j=1}^{k} \theta_j^{st_j - 1}.$$

where

- $s > 0$ is the *equivalent sample size*, which can be regarded as a number of *hidden* instances;
- t_j is to the proportion of hidden instances in category j.
- More commonly, the Dirichlet is parameterized by $\alpha_i = st_i$.

Posterior distribution

- Obtained by multiplying likelihood and prior:

$$\pi(\boldsymbol{\theta}|\mathbf{n}) \propto \prod_{j} \theta_j^{(n_j + st_j - 1)}$$

- Dir posteriors are obtained from Dir priors (conjugacy).
- Taking expectations:

$$P(c_j|\mathbf{n}, \mathbf{t}) = E(\theta_j)|_{\pi(\theta|\mathbf{n})} = \frac{n_j + st_j}{n + s}$$

Uniform prior

- Laplace estimator : initializes to 1 each count n_j before analyzing the data.
- This corresponds to a Dirichlet prior with

$$t_j = \frac{1}{m}, \ \forall j \quad s = m$$

- The uniform prior *looks* non-informative.
- Prior and posterior probability depend on the sample space.
- Alternatively , one could set a prior which reflects domain knowledge (difficult) or his own prior beliefs (subjective).

Prior-dependent classifications

- Prior-dependent: the most probable class varies with the prior.
- If the prior is unique, prior-dependency cannot be spotted.
- Prior-dependent classifications are typically unreliable and more frequent on small data sets.
- Credal classifiers are able to systematically detects prior-dependent classifications.
- Non-Dirichlet prior are however out of scope in this talk.

Betting interpretation of the uniform prior

- A bag contains red and blue marbles; no drawings done so far.
- Uniform prior:

$$\begin{cases} P(blue) = 0.5 \\ P(red) = 0.5 \end{cases}.$$

- You assume the bag to contain an equal number of blue and red marbles.
- You are disposed to bet 0.5 on both colors, in a gamble where you win 1 if the prediction is correct and 0 if wrong.
- This is a model of prior indifference.
- But we are a priori ignorant, not indifferent.

Modelling prior-ignorance: the IDM (Walley, 1996)

- The IDM contains all the Dirichlets which satisfy:

$$\begin{cases} 0 < t_j < 1 \;\; \forall j \\ \sum_j t_j = 1 \end{cases}$$

- This is a *vacuous* model: a priori, $P(c_j) \in (0, 1)\ \forall j$.
- Yet, it learns from data:

$$\underline{P}(c_j|\mathbf{n}) = \inf_{0<t_j<1} \frac{n_j + t_j}{n+s} = \frac{n_j}{n+s}$$

$$\overline{P}(c_j|\mathbf{n}) = \sup_{0<t_j<1} \frac{n_j + t_j}{n+s} = \frac{n_j + s}{n+s}$$

- $\underline{P}(c_j|\mathbf{n})$ and $\overline{P}(c_j|\mathbf{n})$ do *not* depend on the sample space (**R.I.P.**).

IDM and the bag of marbles

- The lower probability is the *maximum* amount of money you are disposed to bet.
- Using IDM:

$$\begin{cases} \underline{P}(blue) = 0 \\ \underline{P}(red) = 0 \end{cases}$$

  (the upper probability of both colors is instead 1).
- The IDM prevents betting.
- If one is ignorant, this is more sensible than being equally disposed to bet on both colors.

Learning from data

- After drawing 43 blue marbles in 100 trials and assuming $s = 1$:

$$\overline{P}(blue) = \frac{(43+1)}{(100+1)} = 43.5\%$$

$$\underline{P}(blue) = \frac{(43)}{(100+1)} = 42.5\%$$

- *Degree of imprecision* :

$$\overline{P}(blue) - \underline{P}(blue) = \frac{s}{n+s} = \frac{1}{101}$$

- Smaller s produces faster convergence, larger s produces more imprecise inferences.

Conditional probabilities (Local IDM)

- The same can be applied for conditional probabilities. Suppose that there are two bags of marbles and the following drawing:

Bag (A)	*Marble (C)*
2	*red (or 1)*
1	*blue (or 0)*
1	*blue (or 0)*
1	*red (or 1)*
2	*red (or 1)*

- Assuming that IDM with $s = 1$ is used separately for each *bag*:

$$\overline{P}(blue|bag1) = \frac{(2+1)}{(3+1)} = 75\% \quad \underline{P}(blue|bag1) = \frac{(2)}{(3+1)} = 50\%$$

$$\overline{P}(blue|bag2) = \frac{(1)}{(2+1)} = 33.3\% \quad \underline{P}(blue|bag2) = \frac{(0)}{(2+1)} = 0\%$$

Global IDM

Bag (A)	*Marble (C)*
2	*red (or 1)*
1	*blue (or 0)*
1	*blue (or 0)*
1	*red (or 1)*
2	*red (or 1)*

- Suppose now that $s = 1$ is used on an IDM over the joint $(Bag, Marble)$. Some possible mass functions are:

	(bag1,blue)	*(bag1,red)*	*(bag2,blue)*	*(bag2,red)*
1	$\frac{2+1}{5+1} = 0.5$	$\frac{1}{5+1} = 0.16$	$\frac{0}{5+1} = 0$	$\frac{2}{5+1} = 0.33$
2	$\frac{2}{5+1} = 0.33$	$\frac{1+1}{5+1} = 0.33$	$\frac{0}{5+1} = 0$	$\frac{2}{5+1} = 0.33$
3	$\frac{2}{5+1} = 0.33$	$\frac{1}{5+1} = 0.16$	$\frac{0+1}{5+1} = 0.16$	$\frac{2}{5+1} = 0.33$
4	$\frac{2}{5+1} = 0.33$	$\frac{1}{5+1} = 0.16$	$\frac{0}{5+1} = 0$	$\frac{2+1}{5+1} = 0.5$
...	...			

Extreme Global IDM

	(bag1,blue)	*(bag1,red)*	*(bag2,blue)*	*(bag2,red)*
1	$\frac{2+1}{5+1} = 0.5$	$\frac{1}{5+1} = 0.16$	$\frac{0}{5+1} = 0$	$\frac{2}{5+1} = 0.33$
2	$\frac{2}{5+1} = 0.33$	$\frac{1+1}{5+1} = 0.33$	$\frac{0}{5+1} = 0$	$\frac{2}{5+1} = 0.33$
3	$\frac{2}{5+1} = 0.33$	$\frac{1}{5+1} = 0.16$	$\frac{0+1}{5+1} = 0.16$	$\frac{2}{5+1} = 0.33$
4	$\frac{2}{5+1} = 0.33$	$\frac{1}{5+1} = 0.16$	$\frac{0}{5+1} = 0$	$\frac{2+1}{5+1} = 0.5$

- These four mass functions are not all possible functions in the global IDM. Any way to split the $s = 1$ is valid:

(bag1,blue)	*(bag1,red)*	*(bag2,blue)*	*(bag2,red)*
$\frac{2+0.5}{5+1}$	$\frac{1+0.2}{5+1}$	$\frac{0+0.3}{5+1}$	$\frac{2}{5+1}$
...			

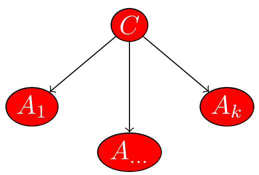
- In spite of that, the extreme global IDM (Cano et al, ISIPTA'10) has devised to consider only the extremal functions.

Credal classifiers and IDM

- Credal classifiers specify a set of priors (IDM) rather than a single prior.
- This allows to represent prior-ignorance and to spot prior-dependent instances.
- When faced with a prior-dependent instance, credal classifiers return a set of classes (indeterminate classification).
- This allows to robustly deal with small data sets.

Outline

Naive Bayes (NBC)



- *Naively* assumes the features to be independent given the class.
- This causes NBC to be excessively confident in its predictions: it often returns probability ≈ 1 for the most probable class.
- Thus, NBC computes biased probabilities.

NBC (II)

- Yet, NBC performs well under 0-1 loss (Domingos & Pazzani, 1997), namely it produces good *ranks*.
- Bias-variance decomposition of the misclassification error (JH Friedman,1997): NBC has high bias, but this is remediated by low variance.
- Low bias is more important for performance than low variance, if the data set is not very large.
- Attempts to improve NBC include feature selection (Fleuret, 2004) and TAN (Friedman et al., 1997).

Joint prior

- $\theta_{c,\mathbf{a}}$: the unknown joint probability of class and features, which we want to estimate.
- Under naive assumption and Dirichlet prior, the joint prior is:

$$P(\theta_{c,\mathbf{a}}) \propto \prod_{c \in \Omega_C} \theta_c^{st(c)} \prod_{i=1}^{k} \prod_{a \in \Omega_{A_i}} \theta_{(a|c)}^{st(a,c)}.$$

  where $t(a,c)$ is the proportion of hidden instances with $C = c$ and $A_i = a$.
- Let vector $\mathbf{t}$ collect all the parameters $t(c)$ and $t(a,c)$.
- Thus, a joint prior is specified by $\mathbf{t}$.

Likelihood and posterior

- The likelihood is like the prior, with coefficients $st(\cdot)$ replaced by the $n(\cdot)$.

$$L(\theta|\mathbf{n}) \propto \prod_{c \in \mathcal{C}} \left[\theta_c^{n(c)} \prod_{i=1}^{k} \prod_{a \in \mathcal{A}_i} \theta_{(a|c)}^{n(a,c)} \right].$$

- The joint posterior $P(\theta_{c,\mathbf{a}}|\mathbf{n},\mathbf{t})$ is like the likelihood, with coefficients $n(\cdot)$ replaced by $st(\cdot) + n(\cdot)$.
- Once $P(\theta_{c,\mathbf{a}}|\mathbf{n},\mathbf{t})$ is available, the classifier is *trained*.

Issuing a classification

- The value of the features is specified as $\mathbf{a} = (a_i, \ldots, a_k)$.

$$P(c,\mathbf{a}|\mathbf{n},\mathbf{t}) = E\left[\theta_{c,\mathbf{a}}|\mathbf{n},\mathbf{t}\right] = P(c|\mathbf{n},\mathbf{t}) \prod_{i=1}^{k} P(a_i|c,\mathbf{n},\mathbf{t})$$

  where

$$P(c|\mathbf{n},\mathbf{t}) = \tfrac{n(c)+st(c)}{n+s}$$
$$P(a_i|c,\mathbf{n},\mathbf{t}) = \tfrac{n(a_i,c)+st(a_i,c)}{n(c_+ st(c)}.$$

- Under 0-1 loss, NBC selects the class with highest probability.
- A classification is prior-dependent if the most probable class varies with $\mathbf{t}$.

Next

From Naive Bayes to Naive Credal Classifier:

Naive Credal Classifier

IDM over the naive topology

We consider a *set* of joint priors, defined by:

$$P(\theta_C)\begin{cases}0 < t(c) < 1 & \forall c \in \Omega_c\\ \sum_c t(c) = 1\end{cases}$$

- A priori, $0 < P(c_j) < 1, \;\forall j.$

$$P(\theta_{A|C})\begin{cases}\sum_a t(a,c) = t(c) & \forall a, c\\ 0 < t(a,c) < t(c) & \forall a, c\end{cases}$$

- A priori, $0 < P(a|c) < 1, \;\forall a, c.$

- Such constraints define polytope $\mathcal{T}$, within which $\mathbf{t}$ varies.
- $P(c|\mathbf{a}, \mathbf{n}, \mathbf{t})$ becomes an interval, because $\mathbf{t}$ is not fixed.

Classification with naive credal Classifier (NCC)

- NCC returns the *non-dominated* classes.

Credal dominance

- Class c' dominates c'' iff:

$$P(c'|\mathbf{a}, \mathbf{n}, \mathbf{t}) > P(c''|\mathbf{a}, \mathbf{n}, \mathbf{t})$$

$$\forall\; \mathbf{t} \in T$$

- This criterion is called *maximality*.

Computing maximality

- Comparing c' and c'' through maximality requires to solve:

$$\min_{\mathbf{t}\in T} \frac{P(c'|\mathbf{a}, \mathbf{n}, \mathbf{t})}{P(c''|\mathbf{a}, \mathbf{n}, \mathbf{t})} > 1$$

where the $\min_{\mathbf{t}\in T}$ implies the constraints:

$$\begin{aligned} 0 < t(c) < 1 \quad & \forall c\\ \sum_c t(c) = 1 \quad & \\ 0 < t(a,c) < t(c) \quad & \forall a, c\\ \sum_a t(a,c) = t(c) \quad & \forall c \end{aligned}$$

- An exact minimization procedure is given in Zaffalon, (2001).

Identification of the non-dominated classes

- The non-dominated classes are identified by pairwise tests.

Procedure

NonDominatedClasses := Ω_C;
for $c' \in \Omega_C$ $\{$
  for $c'' \in \Omega_C,\, c'' \neq c'$ $\{$
    if (c' dominates c'' $\{$
      drop c'' from NonDominatedClasses;
    $\}$
  $\}$
return NonDominatedClasses;

NCC and prior-dependent instances

- If the instance is prior-dependent, NCC detects and returns *more* non-dominated classes.
- In this case, NCC draws a less informative but more robust conclusion than NBC.

The next applications shows that

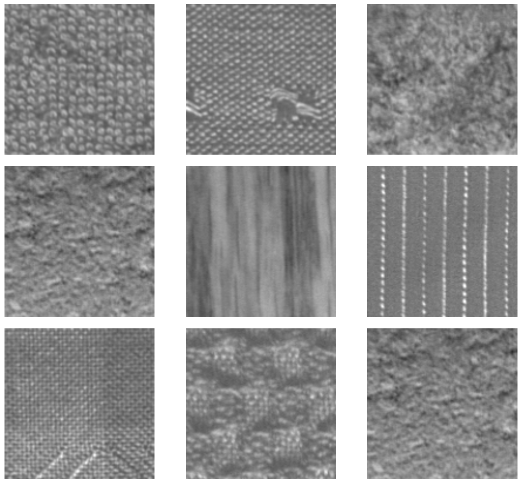
- prior-dependent instances are present also on large data sets.
- NBC is unreliable on prior-dependent instances, even if trained by a large data set.

Next

From Naive Bayes to Naive Credal Classifier:

Prior-dependent classifications in a real application

Texture recognition

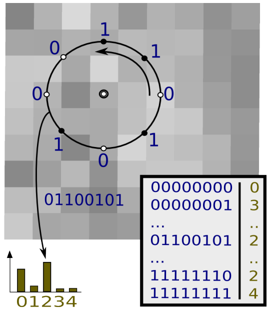


- The goal: to assign an image to the correct class.
- The classes include textiles, carpets, woods etc.

Comparing NBC and NCC

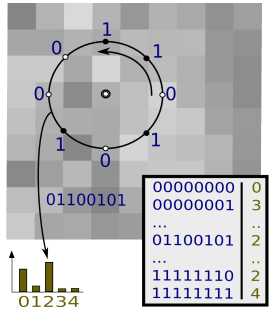
- The OUTEX data sets (Ojama, 2002): 4500 images, 24 classes.
- No missing data.
- We aim at comparing NBC and NCC.

Local Binary Patterns (Ojama, 2002)



- The gray level of each pixel is compared with that of its neighbors.
- This produces a binary judgment (more intense/ less intense) for each neighbor.
- The binary judgments constitute a 0-1 string.
- A string is associated to each pixel.

Local Binary Patterns (2)



- Each string is then assigned to a single category.
- The categories group similar strings: e.g., 00001111 is in the same category of 11110000 for rotational invariance.
- There are 18 categories.
- For each image there are 18 features: the $\%$ of pixels assigned to each category.

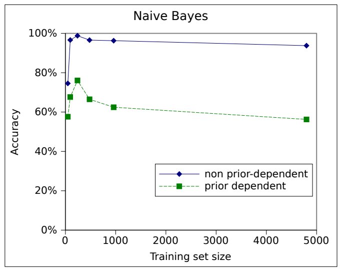
Results

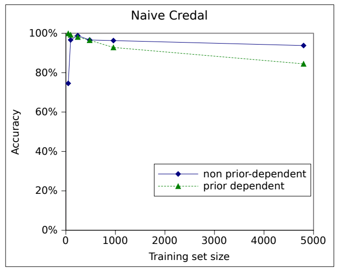
- 10 folds cross-validation; supervised discretization of features.
- Accuracy of NBC: 92% (SVMs: 92.5%).
- But NBC drops to 56% on prior-dependent instances!
- Half of prior-dependent instances are classified by NBC with probability $> 90\%$.

	Non prior-dependent	*Prior-dependent*
Amount%	95%	5%
NBC: accuracy	94%	56%
NCC: accuracy	94%	85%
NCC: non-dom. classes	1	2.4

Sensitivity on n

- Smaller training sets generated by stratified downsampling.





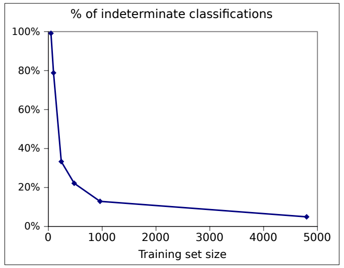
At any sample size

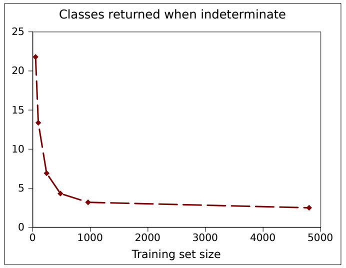
- the accuracy of NBC drops on prior-dependent instances;
- indeterminate classifications preserves the reliability of NCC.

Different training set sizes (II)

As n grows:

- the % of indet. classification decreases;
- the avg. number of classes returned when indeterminate decreases.





- Qualitatively similar results are also obtained on the UCI data sets (Corani and Zaffalon, 2008).

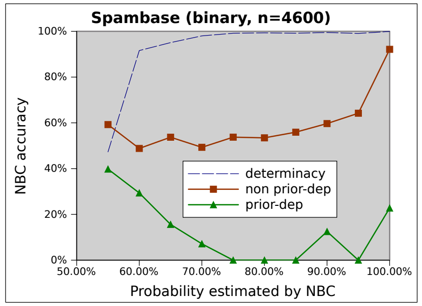
Other approaches to suspend the judgment

- Rejection rule : refuses to classify an instance, if the probability of the most probable class is below a threshold.
- Constant risk : returns the minimum number of best classes so that the accumulated probability exceeds the threshold.
- Non-deterministic classifiers : look for the subset of classes which maximizes the F-measure (Del Coz, 2009).
- All such approaches consider a single posterior distribution.
- Yet, they can be deceived if the posterior is not reliable.

NCC vs. rejection rule

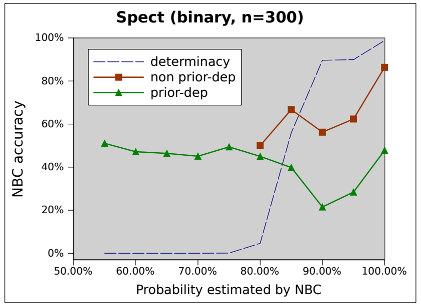
- The indeterminacy of the rejection rule is almost independent from the sample size.
- Rejection rule is not effective with NBC, because NBC associates high probability to the most probable class (even on small data sets).
- In texture classification, half of the prior-dependent instances is classified by NCC with probability $> 90\%$.

Indeterminate classifications vs. NBC probabilities (I)



- About half of the instances classified with probability $< 55\%$ by NBC are *not* prior-dependent.
- NCC does not get indeterminate only because NBC computes a small margin for the most probable class!
- NBC is little accurate on prior-dependent instances.

Indeterminate classifications vs. NBC probabilities (II)



- All instances classified with probability $< 75\%$ by NBC are prior-dependent. Here, NBC is almost random guessing.
- NCC is indeterminate also on some instances confidently classified by NBC, and over which NBC is nevertheless unreliable.
- NCC has a more complex behavior than the rejection rule.

An open problem: comparing credal and traditional classifiers

- This has been done so far by assessing the drop NBC on the instances indeterminately classified by NCC.
- This drop is considerable in most cases.
- Yet, is it better the credal or the traditional classifier?
- E.g., is it better 85% accuracy returning two classes, or 65% returning a single class?
- Comparing credal and traditional classifiers implies modelling a trade-off between informativeness and robustness.

Next

From Naive Bayes to Naive Credal Classifier:

Counter-intuitive behaviors of NCC

Non-dominated classes (refresh)

- Recall:
  - $\mathbf{t}$: vector containing the $t(c)$ and $t(\mathbf{a}, c)$ of NCC.
  - T: polytope containing all the admissible $\mathbf{t}$.
- Class c' *dominates* c'' iff:

$$\min_{\mathbf{t} \in T} \frac{P(c'|\mathbf{a}, \mathbf{n}, \mathbf{t})}{P(c''|\mathbf{a}, \mathbf{n}, \mathbf{t})} > 1$$

- NCC detects the *non-dominated* classes by pairwise comparing all classes.
- Min(ratio) $\approx$ min(numerator) + max(denominator).
- But unexpected behaviors can appear while minimizing the ratio.

Minimizing numerator: *feature problem*

- Naive assumption (not showing for simplicity the cond. on $\mathbf{n}, \mathbf{t}$):

$$\min_{\mathbf{t}\in T} \frac{P(c'|\mathbf{a})}{P(c''|\mathbf{a})} = \min_{\mathbf{t}\in T} \frac{P(c',\mathbf{a})}{P(c'',\mathbf{a})} = \min_{\mathbf{t}\in T} \frac{P(c')}{P(c'')} \prod_i \frac{P(a_i|c')}{P(a_i|c'')}$$

- To compute $\min_{\mathbf{t}\in T} P(c'|\mathbf{a})$ we need to minimize each $P(a_i|c')$.

$$\min_{\mathbf{t}\in T} P(a_i|c') = \frac{n(a_i,c')}{n(a_i,c') + st(c')}$$

- Even a single feature with $n(c', a_i) = 0$ implies $P(a_i, c') = 0$.
- Thus $P(c', \mathbf{a}) = 0$, regardless the remaining features.
- A single feature thus prevents c' from dominating any other class.
- One idea is to allow $P(a|c')$ to become very small, but not 0.

Maximizing denominator: *class problem*

- Problems arise when $n(c'') = 0$.
- Considering the naive assumption, we maximize each $P(a|c'')$ as:

$$\max_{\mathbf{t}\in \mathcal{T}} P(a|c'') = \frac{\overbrace{n(a,c'')}^{=0} + st(a,c'')}{\underbrace{n(c'')}_{=0} + st(c'')} = \frac{st(c'')}{st(c'')} = 1$$

- This repeats on each feature; eventually $P(\mathbf{a}|c'') >> P(\mathbf{a}|c')$.
- This often allows c'' to be non-dominated and to appear in the output of the classifier, despite never being observed.

Consequences of the feature and the class problem

- High indeterminacy of NCC.
- Surprisingly, NBC can be accurate on the instances indeterminately classified by NCC because of the feature or the class problem.
- Such instances are thus not really difficult to classify.
- As these problems are mainly due to the extreme distributions of the IDM, we present two approaches to restrict the set of priors.

NCC$_\epsilon$: an ϵ-contamination of NCC and NBC.

- The set of priors of NCC$_\epsilon$ is an ϵ-contamination of uniform prior (NBC) and IDM.
- NCC$_\epsilon$ violates the R.I.P., being contaminated with NBC.

The contamined set of priors is:

$$\mathcal{T}_c := \begin{cases} \sum_{c\in\mathcal{C}} t(c) = 1, \quad t(c) \in \left[\epsilon_0 \frac{1}{|\mathcal{C}|}, \epsilon_0 \frac{1}{|\mathcal{C}|} + (1-\epsilon_0)\right], \\ \sum_{a_i\in\mathcal{A}_i} t(a_i,c) = t(c), \quad t(a_i,c) \in \left[\epsilon_i \frac{t(c)}{|\mathcal{A}_i|}, \epsilon_i \frac{t(c)}{|\mathcal{A}_i|} + (1-\epsilon_i)t(c)\right], \ \forall (i,c) \end{cases}$$

where the ϵ_0 refers to C, while a different $\epsilon_i \in (0,1)$ can be specified for each A_i.

- By setting $\epsilon_0 = \epsilon_i = 1$, NCC$_\epsilon$ is equal to NBC.

Feature problem: squash-stored data set

- Features: taste of squash after different periods of storage; class: a final measure of acceptability of the fruit.
- A feature has many $n(a, c) = 0$.
- $\epsilon_0 = \epsilon_i = 0.05$.

		NBC accuracy when	
	Determ. %	NCC det.	NCC indet.
NCC	32%	70%	64%
NCC$_\epsilon$	42%	80%	56%

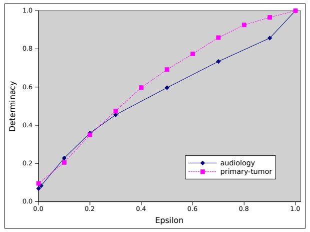
- NCC$_\epsilon$ is less indeterminate and better discriminates between eays and hard instances than NCC.

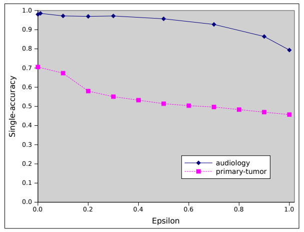
Class problem

- We now consider *audiology* and *primary tumor*, affected by the class problem.
- On both, NCC is very indeterminate, and still NBC can issue reasonable classifications.
- We evaluate the performance for different values of the ϵ

Results

- Determinacy: percentage of determinate classifications;
- Single accuracy: accuracy of the classifier when determinate;





- Increasing ϵ, determinacy increases but single accuracy deteriorates.
- The choice of ϵ should be based on a trade-off between accuracy and robustness.

Conclusions

- Feature and class problem cause NCC to be indeterminate even on instances accurately classified by NBC.
- Yet, a restricted set of priors becomes somehow informative and cannot satisfy all the properties of the original IDM.
- NCC$_\epsilon$ addresses both feature and class problem, violating R.I.P.

Next

From Naive Bayes to Naive Credal Classifier:

Conservative treatment of missing data

Ignorance from missing data

- Besides prior-ignorance, there is ignorance due to missing data.
- Usually, classifiers *ignore* missing data, assuming MAR.
- MAR: the missingness process is non-selective, i.e., the probability of an observation to be missing does not depend on its value.
- MAR cannot be tested on the incomplete data.
- A sensor breakdown generates MAR missing data.

A non-MAR example

- In a poll, the supporters of the right-wing sometimes refuse to answer.
- The probability of the answer to be missing depends on the answer itself; the missingness is selective.

Vote	Answer
L	L
L	L
L	L
R	R
R	-
R	-

- By ignoring missing data, $P(right) = 1/4$: underestimated!

Conservative treatment of missing data

- Consider each possible completion of the data (*likelihood ignorance*) and generate an interval estimate.

Answer	**D1**	**D2**	**D3**	**D4**
L	L	L	L	L
L	L	L	L	L
L	L	L	L	L
R	R	R	R	R
-	L	L	R	R
-	L	R	L	R
$P(right)$	**1/6**	**1/3**	**1/3**	**1/2**

- $P(right) \in [1/6,\ 1/2]$; this interval *includes* the real value.
- Less informative, but more robust than assuming MAR.

Conservative Updating (Zaffalon, 2005)

- Deals with a a mix of MAR and non-MAR missing data.
- MAR missing data are ignored.

Conservative treatment of non-MAR missing data

- a set of likelihoods, one for each possible completion of the training set.
- a set of virtual instances, one for each completion of the instance to be classified.
- The replacements are exponentially many, but *polynomial* algorithms are available for NCC.

NCC with conservative treatment of missing data

- The conservative treatment of missing data can generate additional indeterminacy.

More classes are returned if the most probable class depends:

- on the prior specification *or*
- on the completion of the non-MAR missing data of training set *or*
- on the completion of non-MAR missing data in the instance to be classified.

Computing maximality

- Comparing c' and c'' through maximality requires to solve:

$$\min_{\mathbf{t}\in T}\min_{\mathbf{n}\in N}\frac{P(c'|\mathbf{a},\mathbf{n},\mathbf{t})}{P(c''|\mathbf{a},\mathbf{n},\mathbf{t})}>1$$

where the $\min_{\mathbf{t}\in T}$ is processed as before, while $\min_{\mathbf{n}\in N}$ is over all possible completions of the dataset where MAR was not assumed.

Experiments

- 18 UCI complete data sets.

Generating non-MAR missing data

- For each feature, make missing with 5% probability the observations of the first half of categories;
- The class is never made missing.
- Afterwards, perform a second experiment by making missing the observations of the second half of categories.

Results with missing data

- Avg. determinacy drops from 91% (NCC, MAR) to 49% (NCC, non-MAR).
- Only half the values of A_j are possible replacements, but all values are regarded as possible replacements (too cautious).
- Yet, the avg. accuracy of NBC drops from 90% to 72% on the instances indeterminately classified because of missing data.
- In real application, a good strategy is to declare a mix of MAR and non-MAR features, based on domain knowledge.

Next

From Naive Bayes to Naive Credal Classifier:

metrics for comparing credal classifiers

Discounted-accuracy

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

where

- $accurate_i$ is a boolean, which is true if the non-dominated classes include the correct class.
- $|Z_i|$ is the number of classes returned on the i-th instance.
- Yet, *linearly* discounting on $|Z_i|$ is somehow arbitrary.

Rank test: removing the arbitrariness of d-acc

- It compares two credal classifiers CR_1 and CR_2 as:

CR1	CR2	winner
accurate	not accurate	CR1
accurate	accurate with less classes	CR2
accurate	accurate, same number of classes	tie
inaccurate	inaccurate	tie

- The ranks are then analyzed by Friedman test.
- The rank test is less arbitrary but also less sensitive than d-acc.

Comparing credal and traditional classifier via d-acc?

Looks appealing because:

- penalizes the credal when it is indeterminate, while the same instance can be accurately classified with a single class;
- penalizes the traditional classifier when it is wrong, while the credal remains reliable through indeterminate classification.

Random and vacuous classifier

- Possible diseases: $\{A,B\}$.

Disease	*Doctor A* (random)	*Doctor B* (vacuous)
A	A	$\{A, B\}$
A	B	$\{A, B\}$
B	A	$\{A, B\}$
B	B	$\{A, B\}$
d-acc	0.5	0.5

- Random and vacuous classifier are seen as equal by both d-acc and rank test.
- Yet the vacuous seems preferable: it admits to be ignorant, while the random pretends to know!

Comparing credal and traditional classifiers: an open problem

- The credal classifier cannot win, unless the traditional classifier is worse than random on the instances indeterminately classified.
- Thus, the traditional classifier is going to always win.
- To allow a more sensible comparison, the metrics could favor the vacuous over the random.
- How? By how much?
- What's about cost-sensitive problems?

Outline

Next

Further classifiers:

Lazy NCC

Two issues of NCC

- The bias due to the naive assumption.
- A sometimes excessive indeterminacy.
- Lazy NCC addresses both problems by applying *lazy learning* to NCC.

Lazy learning

- Do *not* learn until there is an instance to classify (*query*).

How gets an instance classified?

- the instances of the training set are ranked according to the distance from the query;
- a local classifier is trained on the k closest instances
- the local classifier issues the classification and then is discarded;
- the training set is kept into memory, to answer future queries.

- Parameter k controls the bias-variance trade off: smaller k implies lower bias but higher variance of the estimates.

Combining NBC and lazy learning

- The idea (Frank, 2003): a properly chosen k (*bandwidth*) can decrease the bias of NBC.
- Working locally also reduces dependencies between features.
- Frank et al., (2003): local NBC with weighted instances.
- More accurate than NBC and competitive with TAN.

Bandwidth selection

- Simplest approach: to choose it via cross-validation, and then answer all queries using the same optimized k;
- Better results can be obtained by tuning k query-by-query: see for instance Birattari and Bontempi, (1999) for lazy learning in regression.
- LNCC tunes k on each query, using a criterion based on imprecise probability.

Bandwidth selection with imprecise probability

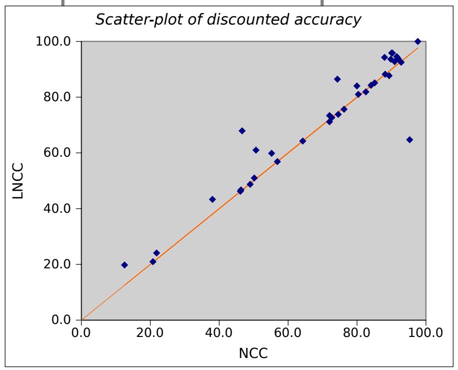
The bandwidth is increased until the classification is not prior-dependent.

Algorithm

```
k=25;
lncc.train(k);
while (lncc is indeterminate OR n = k) {
    k=k+20;
    lncc.train(k);
}
```

- LNCC is by design more determinate than NCC.

Experimental comparison



Each point refers to a specific data set.

	LNCC wins	*ties*	*NCC wins*
d-acc	19	11	6
rank test	15	19	2

LNCC and bias reduction

- Is LNCC doing better than NCC just because it is more determinate?
- On large data sets (e.g., letter: n = 20000), both NCC and LNCC are very determinate but LNCC is more accurate.
- LNCC reduces bias by selecting a k of only a few hundreds.
- LNBC: NBC learned on the bandwidth selected as shown.
- LNBC vs NBC: 16 wins of LNBC, 13 ties, 7 wins of NBC.
- This further shows the effectiveness of bias reduction.

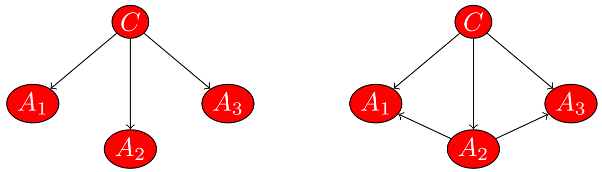
Next

Further classifiers:

Credal TAN

Credal TAN

- TAN: a feature node possibly has, besides the class parent, also a second parent, constituted by a feature.
- TAN is a compromise between Bayesian networks (BNs) and NBC.
- TAN generally outperforms both NBC and BNs (Friedman et. al, 1997; Madden, 2009).



NBC TAN

Credal TAN: TANC

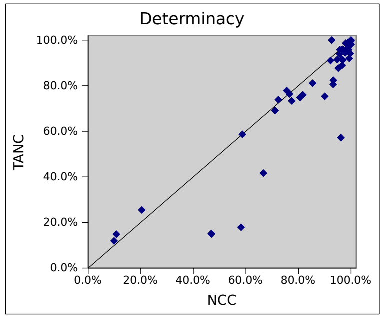
- The test of credal dominance is as usual:

$$\min_{\mathbf{t}\in T} \frac{P(c', \mathbf{a}|\mathbf{n}, \mathbf{t})}{P(c'', \mathbf{a}|\mathbf{n}, \mathbf{t})} > 1$$

  but with TAN the minimization is more difficult.
- Defining the credal set of TANC to have a feasible minimization problem: Zaffalon (2003); Corani and De Campos (2010).

TANC vs NCC

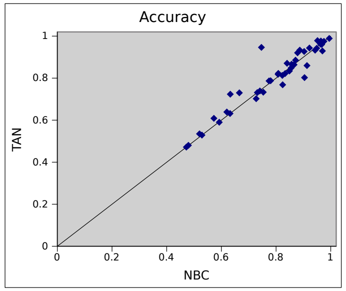
- TANC is generally less determinate than NCC.

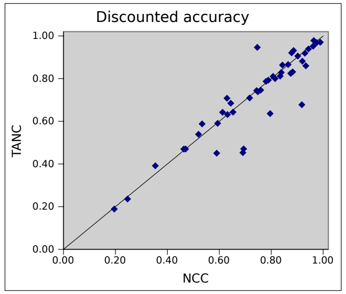


- Some features get the second parent but generate contingency table full of 0s, which causes indeterminacy of TANC.
- Open problem: learn a more parsimonious TAN, to be used with IDM.

TAN vs. Naive

Bayesian case *Credal case*



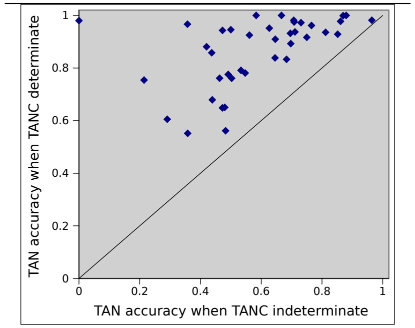


- TANC is often outperformed by NCC, unlike in the Bayesian case.
- As already seen, d-acc favors more determinate classifiers.
- Identifying more parsimonious TAN structures could fix the situation and be useful also in the Bayesian case.

Bayesian estimation vs Imprecise Dirichlet Model From Naive Bayes to Naive Credal Classifier Further classifiers Conclusions

TANC vs TAN

- Despite high indeterminacy, TANC detects instances where TAN is less reliable.



Bayesian estimation vs Imprecise Dirichlet Model From Naive Bayes to Naive Credal Classifier Further classifiers Conclusions

Next

Further classifiers:

Credal Model Averaging

Bayesian estimation vs Imprecise Dirichlet Model From Naive Bayes to Naive Credal Classifier Further classifiers Conclusions

Model uncertainty

- Let us consider NBC again.
- Given k features, we can design 2^k NBC *structures*, each with a different feature set.
- *Model uncertainty*: several structures provide good accuracy; which one do we choose?
- *Model averaging*: to average over all the 2^k models.

Bayesian Model Averaging

- Computes a weighted average of the probabilities returned by the different classifier.
- The weight of each classifier is its posterior probability.

$$P_{BMA}(C|\mathbf{n}) = \sum_{s \in \mathcal{S}} P(C|s, \mathbf{n}) P(\mathbf{n}|s) P(s)$$

where

- s is a generic structure and $\mathcal{S}$ the set of structures;
- $P(\mathbf{n}|s)$ is the marginal likelihood of structure s;
- $P(s)$: is the prior probability of s.

BMA for Naive Bayes (Dash & Cooper, 2002)

- Working with 2^k models is generally unfeasible.
- The algorithm by D&C computes BMA for naive networks exactly and efficiently.
- It assumes $P(S)$ to be flat.
- Yet, the classification might depend on the chosen $P(S)$, and choosing $P(S)$ is an open problem for BMA (Clyde et al., 2004).

Credal model averaging (CMA)

- A set of mass functions $P(S)$, which let vary the prior probability of each structure between ϵ and $1 - \epsilon$.
- The classification is prior-dependent if the most probable class varies with $P(S)$.
- CMA imprecisely averages over traditional classifiers.
- Imprecise averaging of credal classifiers is yet to be developed!

Results

- 31 UCI data sets.
- On average:
  - BMA accuracy decreases from 86% to 54% on prior-dependent instances.
  - CMA classifies determinately 77% of instances.
  - CMA is 90% accurate when indeterminate.
  - when indeterminate, CMA returns ≅33% of the classes.

Next

Further classifiers:

(Semi-)Imprecise Classification trees

(Semi-)Imprecise classification trees

- Based on ID3 method (Abellan and Moral 2003). The tree is build using an impurity measure. At a node R, we compute for each A_i:

$$I(R, A_i) = \sum_{a_i \in A_i} \frac{n_{R \cup a_i}}{n_R} H(p_{R \cup a_i})$$

$$H(p_{R \cup a_i}) = -\sum_j P_{R \cup a_i}(c_j) \log P_{R \cup a_i}(c_j)$$

- In the precise case, p represents a (single) distribution (can be calculated, for example, by max. likelihood: $P_{R \cup a_i}(c_j) = \frac{n_{R \cup a_i \cup c_j}}{n_{R \cup a_i}}$).

(Semi-)Imprecise classification trees

- In the (semi-)imprecise case, instead of using the imprecision throughout whole method (and possibly return multiple classes), the distribution of maximum entropy is chosen (maximum entropy is a conservative criterion).
  - $\overline{P}_{R \cup a_i}(c_j)$ and $\underline{P}_{R \cup a_i}(c_j)$ are computed (or even a more sophisticated credal set).
  - The distribution of maximum entropy is picked from the credal set and H is computed using it.
- Classification trees can be used with a non-parametric learning idea, for instance the NPI (Coolen and Augustin 2005).

Next

Further classifiers:

Classification with credal networks

Credal networks

- Topic already addressed in previous talk.
- We use inferences in credal networks to take decisions.
  - Here we focus on a maximum entropy idea to make the decision precise (same as done for classification trees just mentioned).
- Instead of looking into the mathematical formulation of the classification problem, let's go straight to examples.

Credal networks

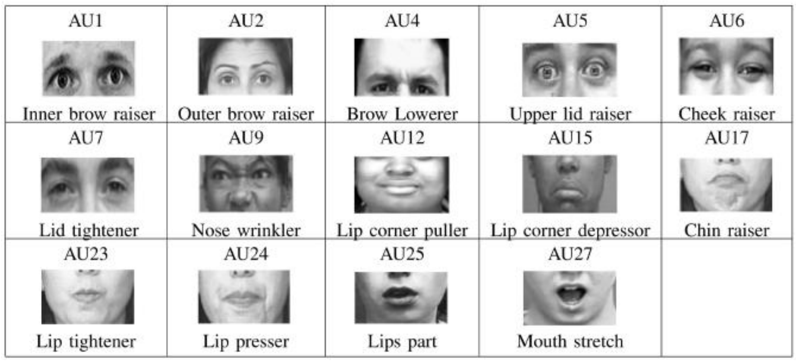
- Topic already addressed in previous talk.
- We use inferences in credal networks to take decisions.
  - Here we focus on a maximum entropy idea to make the decision precise (same as done for classification trees just mentioned).
- Instead of looking into the mathematical formulation of the classification problem, let's go straight to examples.

Facial expression recognition

- 8000 images from DFAT-504 data set.
- Facial expressions can be defined through Action Units (AUs), which represent muscle contractions.
  - AU1: inner brow raiser
  - AU2: outer brow raiser
  - AU5: upper eyelid raiser
  - AU9: nose wrinkle
  - AU17: chin raiser

Facial expression recognition

Facial action unit coding system:



Parameterization of the SQPN

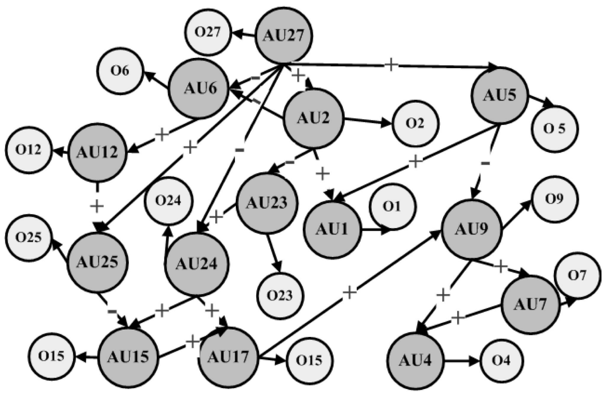
- Parameters of observed nodes are defined by the expert using the errors of the measurement technique.
- Parameters of hidden nodes are learned from data.
  - Data contains 28 columns: 14 measurements from Computer Vision techniques and 14 manually labeled AUs.
  - Prior SQPN and Imprecise Dirichlet Model are employed.

AUs have relations

- Mouth stretch increases the chance of lips apart; it decreases the chance of cheek raiser and lip presser.
- Nose wrinkle increases the chance of brow lowered and lid tightened.
- Eyelid tightened increases the chance of lip presser.
- Lip presser increases the chance of chin raiser.

Facial expression recognition

Semi-qualitative Probabilistic Network:



Inference Approaches

Two approaches are tested:

1. After learning, we perform a query in the credal network to select the distribution of maximum entropy.
   - Then standard Bayesian network belief updating is performed for each AU, given the observations: $p(AU_i|\mathcal{O})$.
   - Main advantage: performance.
2. Inference is performed directly in the credal network, and only cases with interval dominance are analyzed, that is, the maximum probability of AU occurrence (or absence) is less than the minimum of absence (or occurrence). So, we classify only if $\overline{p}(AU_i|\mathcal{O}) \leq \underline{p}(\neg AU_i|\mathcal{O})$ or $\overline{p}(\neg AU_i|\mathcal{O}) \leq \underline{p}(AU_i|\mathcal{O})$.
   - Inference algorithm is slower, but gain is greater.

Facial expression recognition

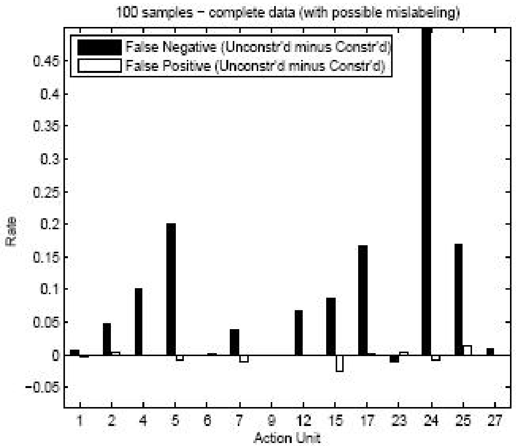


Figure: Benefits of using a prior SQPN

Facial expression recognition

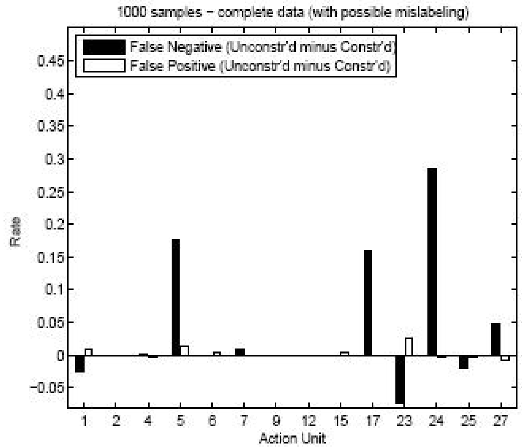


Figure: Benefits of using a prior SQPN

Facial expression recognition

Dataset Size	Maximum Entropy Positive	Maximum Entropy Negative	Interval Dominance	SQPN gain Positive	SQPN gain Negative
100	9.8%	-0.1%	49.2%	9.6%	-0.7%
1000	4.0%	0.3%	54.8%	11.4%	0.4%

Table: Percentage of improvement with Maximum entropy and SQPN+IDM approaches against standard maximum likelihood.

Image Segmentation

We worked with over-segmented images and applied Bayesian networks with imprecise probabilities to choose boundary vertices and edges using most probable explanation.

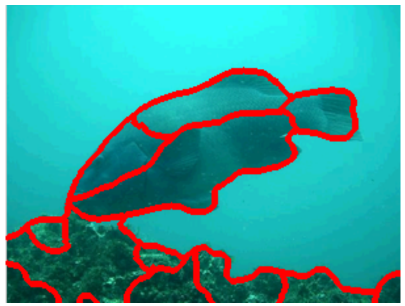
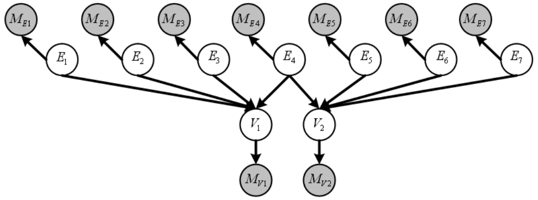


Image Segmentation



- Edges are denoted by E_j and vertices are denoted by V_t. Shadowed nodes are related to computer vision measurements.
- $p(m_{V_t}|v_t) = 0.99$ and $p(m_{V_t}|\neg v_t) = 0.1$. The same idea holds for edge measurements, but with distinct strengths.
- Border should be closed:

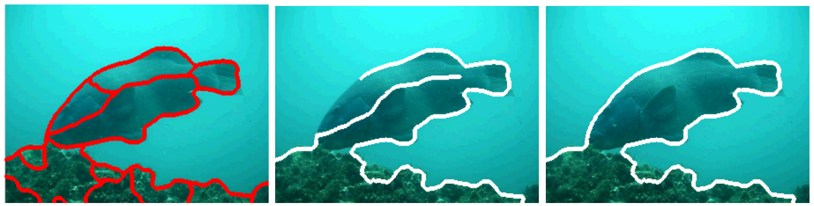
$$p(v_t|\mathit{pa}(V_t)) = \begin{cases} \geq 0.5, & \text{if exactly two parent nodes are true;} \\ 0.3, & \text{if none of the parent nodes are true;} \\ 0, & \text{otherwise.} \end{cases}$$

Inference

Given the SQPN, the goal of image segmentation is achieved by inferring the most probable categories of the variables given the observations (measurements), that is, we look for the categories of E given M_E, M_V that maximize $p(E|M_E, M_V)$. Unfortunately that is very time consuming, but it is much easier to compute categories of E, V that maximize

$$\max_p p(E, V, M_E, M_V) = \prod_t p(V_t|\mathit{pa}(V_t))p(M_{V_t}|V_t) \prod_j p(E_j)p(M_{E_j}|E_j).$$

Image segmentation



(a) Over-segmented (b) Bayesian network (c) Credal network

Image segmentation

(d) Bayesian network

(e) Credal network

Activity recognition

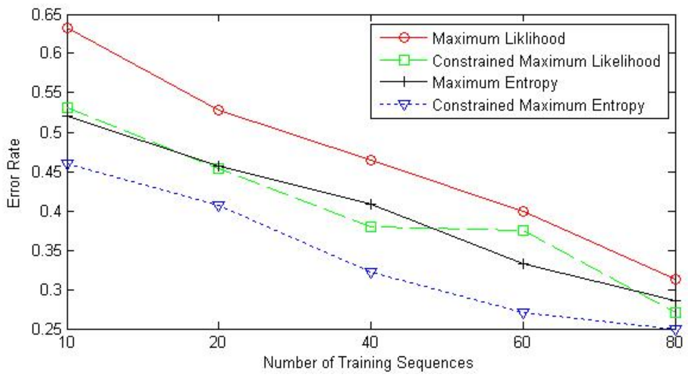


Figure: Comparison between learning ideas

Outline

Software

JNCC2

- Java implementation of NCC (command-line).
- Cross-validation and supervised discretization of features.
- `www.idsia.ch/~giorgio/jncc2.html`

TANC

- Under development.

Weka-IP

- A Weka plugin for credal classification (beta software).
- Implements NCC, LNCC, CMA, IPtree.
- GUI interface and feature selection (from WEKA).
- `http://decsai.ugr.es/~andrew/weka-ip.html`

Conclusions

- Credal classifiers prefer suspending the judgment than guessing!
- They robustly deal with the specification of the prior and with non-MAR missing data.
- Typically, the accuracy of traditional classifiers drops on the instances indeterminately classified by credal classifiers.

Some open problems

- Faster algorithms (for TAN and general nets).
- Metric for compare credal and traditional classifiers.
- Learn parsimonious structures for credal TAN, and more in general for credal networks.
- Credal model averaging of credal classifiers.