Building Classification Trees With Entropy Ranges

Ric Crossman, Frank Coolen, Joaquin Abellan, Thomas Augustin

University of Warwick

ISIPTA '11, Innsbruck

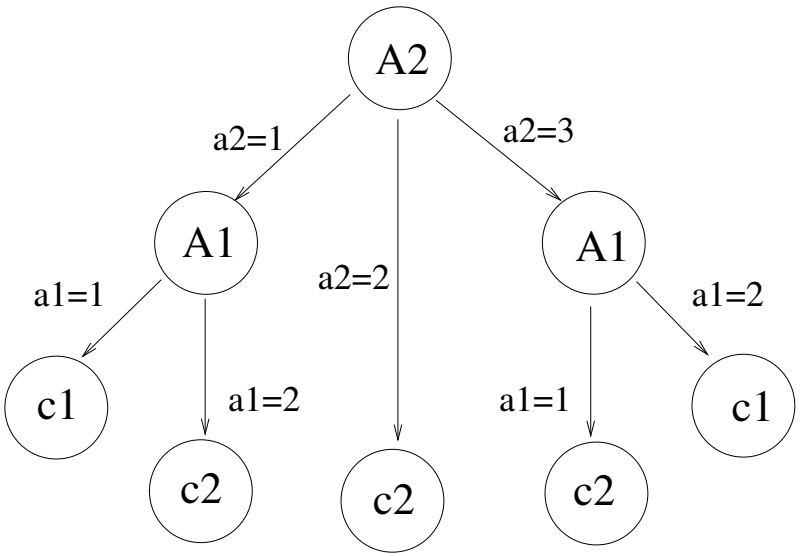
Classification

Each object in a population described by *m attribute variables* X_i (taking values from $\mathcal{X}_i$), and a *category variable* C (taking values from $\mathcal{C}$ with $|\mathcal{C}| = K$). Each object is associated with a vector $(x_1, \ldots, x_m, c) = (\mathbf{x}, c)$.

Challenge: to create a method by which a new value of $\mathbf{x}$ can be assigned the correct category.

Classification trees

These are trees in which each parent node is an attribute variable, and each leaf is a category.



Building A Tree

Building a tree requires an *impurity measure*. How much we do we expect to reduce uncertainty by if we split using a given variable?

E.g. info gain (Quinlan, 1986), based on Shannon's entropy:

$$H(\mathbf{p}) := -\sum_{j=1}^{K} p(c_j) \log[p(c_j)]$$

for K categories. For data set R, define $\mathcal{P}^R$ as structure of probability distribution for categories, across data set.

Define $\mathcal{P}^{a_i}$ as structure of probability distributions of categories, conditional on $X_i = a_i$.

Building A Tree II

For data set R, the impurity measure is defined as

$$IM(R, X_i) = H(\mathbf{p}^R) - \sum_{a_i \in \mathcal{X}_i} p(X_i = a_i) H(\mathbf{p}^{a_i}) := H(\mathbf{p}^R) - I(R, X_i)$$

(in imprecise case, $\mathbf{p}^R \in \mathcal{P}^R$ and $\mathbf{p}^{a_i} \in \mathcal{P}^{a_i}$.)

Entropy of the data set minus the expected entropy of the data set when conditioned on X_i. Larger $IM(R, X_i) \Rightarrow$ more gain from choosing X_i.

Structures result in multiple values of $IM(R, X_i)$. For each X_i, we choose the maximum values of $H(\mathbf{p}^R)$ and $I(R, X_i)$ ($H(\hat{\mathbf{p}})^R$ invariant over X_i, so choose variable minimising the maximum expected entropy.)

Iterative process.

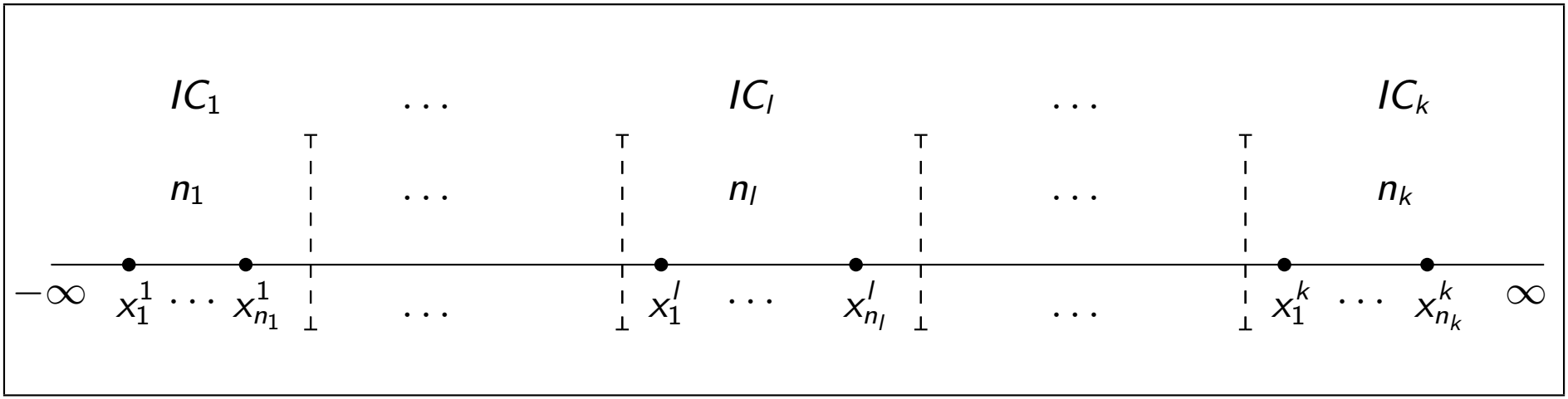
O-NPI

Assume categories 1 to K are ordered, $c_1 < c_2 < \ldots < c_K$. We can represent categories as intervals on the real line. Category c_i corresponds to interval IC_i, where $IC_i \cap IC_j = \emptyset,\ i \neq j$ and $\cup_{j=1}^{K} IC_j = \mathbb{R}$.

Use a latent variable approach: next observation is in c_j if a continuous R.V. is in IC_j.

NPI Cont.

Using $A_{(n)}$,we can find probability that next observation x_{n+1} falls in interval IC_i.



Lower probability: sum of probabilities of data intervals entirely within class interval.

Upper probabiliy: sum of probabilities of data intervals intersecting with class interval.

Results

Using our algorithm we can compare our method with the IDM method (5% level)

Data Set	O-NPI	IDM	Data Set	O-NPI	IDM
anneal	98.96	99.66 (v)	autos	71.66	79.14 (v)
balance scale	69.59	69.59	breast cancer	69.66	71.58
wisconsin breast cancer	94.55	94.72	car	84.86	91.64 (v)
horse colic	82.93	82.25	credit rating	83.65	83.75
german credit	69.08	68.99	dermatology	93.90	93.95
pima diabetes	74.26	74.20	c-14 heart disease	75.49	75.74
h-14 heart disease	78.32	78.41	heart statlog	81.85	82.30
hepatitis	78.13	79.60	ionosphere	89.91	89.72
labor	84.93	84.40	mushroom	100.00	100.00
nursery	94.72	96.28 (v)	sick	97.80	97.79
sonar	73.42	73.82	spectrometer	42.13	44.77
tae	46.78	46.78			

Potential and Guarantee

Previous approach only considers maximum values of entropy.

For a closed set of probability distributions $\mathcal{M}$ denote the *potential* of $\mathcal{M}$ as $\mathbf{v}^*$, where

$$H(\mathbf{v}^*) = \max_{\mathbf{u} \in \mathcal{M}} H(\mathbf{u})$$

and the *guarantee* of $\mathcal{M}$ as $\mathbf{v}_*$, where

$$H(\mathbf{v}_*) = \min_{\mathbf{u} \in \mathcal{M}} H(\mathbf{u})$$

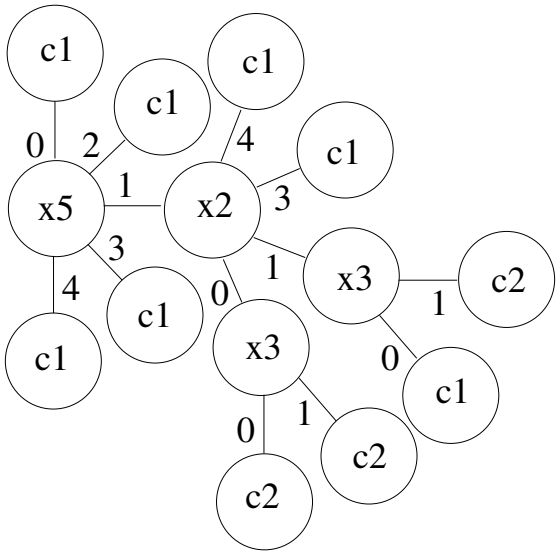
We can therefore generate an entropy interval for each attribute variable.

Entropy Intervals

$H(\mathbf{p})^R \in I_R$, and each $I(R, X_j) \in I_j$. If $I_R \cap I(R, X_j) \neq \emptyset$ for all j, we cannot say splitting on any variable is superior to not splitting, and process stops.

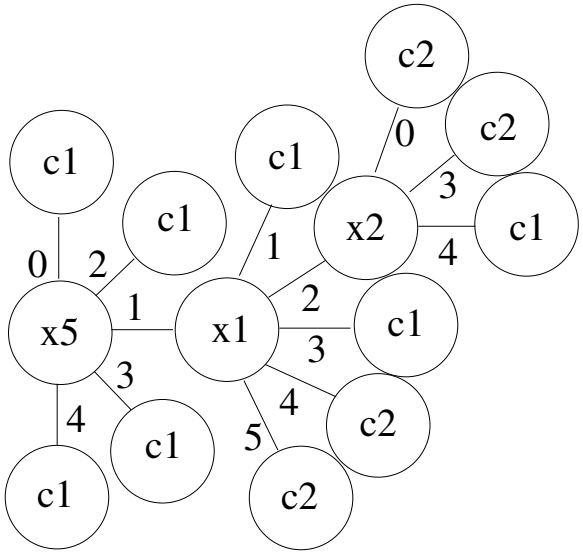
If X_j and X_k are the top two candidates for splitting, but $I_j \cap I_k \neq \emptyset$, we cannot choose between them. Instead, we draw one tree splitting on X_j, and another splitting on X_k.

Example



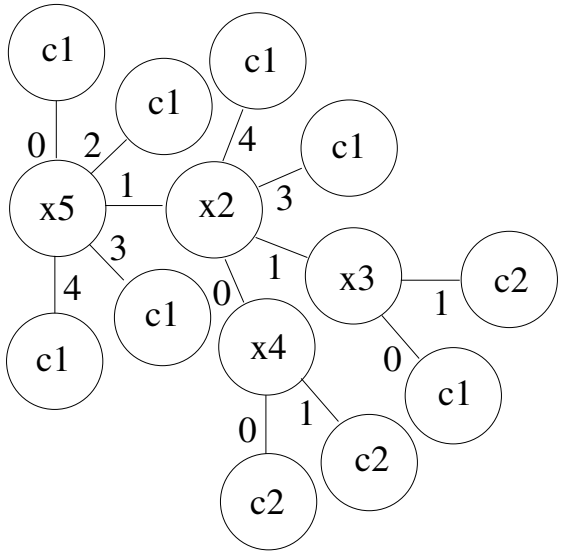
Tree 1

Example cont.



Tree 2

Example cont.



Tree 3

Voting Process

m trees means m categorisations. How do we combine them?

1. Choose most frequent category;
2. Return all categories (credal classification);
3. Choose all categories featuring at least r times.

Conclusion

- Entropy intervals allow us to create ensemble trees;
- Trees consider uncertainty at every node, not just at root;
- Can easily be defined to allow for credal classification.