



CLASSIFICATION TREES WITH NPI

REBECCA BAKER

*Department of Mathematical Sciences,
University of Durham*

- 1 The multinomial NPI model
 - Motivation
 - The probability wheel representation
- 2 Classification
 - Classification trees
 - Weka software
- 3 Finding the maximum entropy distribution for NPI
 - The approximate algorithm
 - The exact algorithm
 - Comparison of NPI-M and A-NPI-M
- 4 Future work



The multinomial NPI model

Model for learning from multinomial data

- inferences about a future observation
- in form of a probability interval
- based entirely on past observations

Have observed Y_1, \dots, Y_n , want to find out about Y_{n+1}

K categories in total: C_1, \dots, C_K

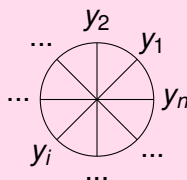
Event of interest is $(Y_{n+1} \in E)$ where E is a subset of the K categories



The probability wheel representation

Represent data on a **probability wheel**

- Y_{n+1} has probability $\frac{1}{n}$ of being in each slice



- Slice bordered by two observations in the same category is assigned to this category
- Slice bordered by two observations in different categories may be assigned to any available category

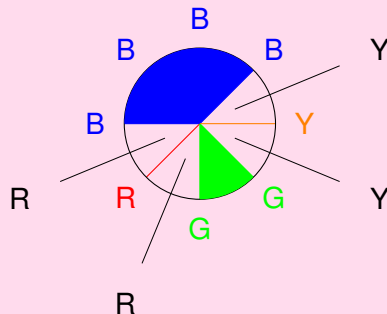
Note: Each category may only be represented by a single segment of the wheel.



The probability wheel representation

Deriving lower probabilities

- Possible categories are blue (B), green (G), red (R), yellow (Y), pink (P) and orange (O)
- Event $E = \{B, G, P\}$



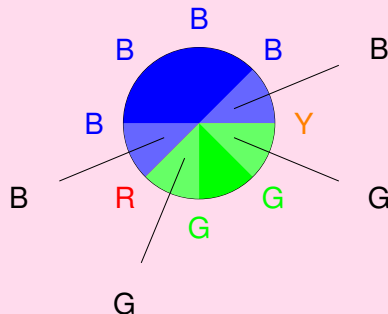
- $\underline{P}(Y_{n+1} \in E) = \frac{4}{8}$



The probability wheel representation

Deriving upper probabilities

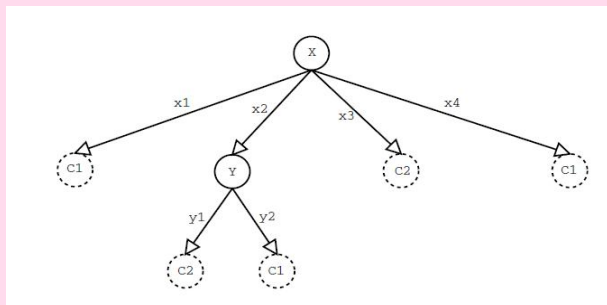
- Possible categories are blue (B), green (G), red (R), yellow (Y), pink (P) and orange (O)
- Event $E = \{B, G, P\}$



- $\bar{P}(Y_{n+1} \in E) = 1$

Classification trees

- Hierarchical structure which defines classification rules



- Attributes at the nodes
- Category labels at the leaves



Building trees using imprecise probabilities

At each node:

- We need to select an attribute for splitting
- The generalised Shannon entropy measure S is employed, using the maximum entropy distribution p_{maxE} :

$$S = - \sum_{j=1}^K p_{maxE}(c_j) \log p_{maxE}(c_j)$$

- The information gain is measured for each attribute
- The most informative attribute is selected for splitting



Weka software

Weka software can be used to build classification trees

- One classifier can be analysed in detail
 - Multiple classifiers can be compared in a number of ways
-
- The software includes tools for pre-processing data



Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter: Choose **MultiFilter** -F "weka.filters.unsupervised.attribute.ReplaceMissingValues" -F "weka.filters.supervised.attribute.Discretize" -R first-last Apply

Current relation: nursery-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.supe...
Instances: 12960 Attributes: 9

Selected attribute: Name: parents Missing: 0 (0%) Distinct: 3 Type: Nominal Unique: 0 (0%)

Label	Count
usual	4320
pretentious	4320
great_pret	4320

Attributes: All None Invert Pattern

No.	Name
1	parents
2	has_nurs
3	form
4	children
5	housing
6	finance
7	social
8	health
9	class

Remove

Class: class (Nom) Visualize All

Status: OK Log

Windows: TeXnicCenter - [...], weka-3.5.7 - Ne..., 2 Java(TM) Pla..., slides4munich.p... EN

System tray: 16:08

Figure: Weka Explorer: Pre-process tab



Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier: **NPIDecisionTrees -SM NPI_M**

Test options

- Use training set
- Supplied test set (Set...)
- Cross-validation (Folds: 10, %: 66)
- Percentage split (%: 66)

More options...

(Nom) class: [dropdown]

Start Stop

Result list (right-click for options)

11:50:00 - trees.NPIDecisionTrees

Classifier output

```

Time taken to build model: 0.45 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      12332           95.1543 %
Incorrectly Classified Instances    628             4.8457 %
Kappa statistic                    0.9285
Mean absolute error                 0.0264
Root mean squared error             0.1198
Relative absolute error             9.6734 %
Root relative squared error         32.421 %
Total Number of Instances          12960

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
1        0        1          1        1          1         1     not_recom
0        0        0          0        0          0         0.498 recommend
0.259   0.002   0.78      0.259   0.389     0.971   very_recom
  
```

Status: OK

Log [icon] x0

Windows taskbar: start, TeXnicCen..., weka-3.5..., 2 Java(T..., slides4mu..., untitled..., EN, 16:11

Figure: Weka Explorer: Classify tab



Weka software

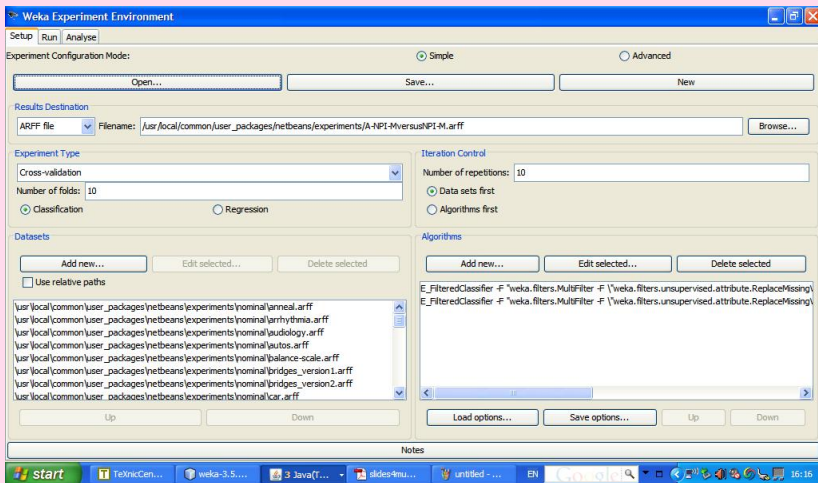


Figure: Weka Experimenter: Setup tab



Weka Experiment Environment

Setup Run Analyse

Source

Got 8000 results

File... Database... Experiment

Configure test

Testing with: Paired T-Tester (corrected)

Row: Select

Column: Select

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Test base: Select

Displayed Columns: Select

Show std. deviations:

Output Format: Select

Perform test Save output

Result list

16:17:59 - Percent_correct - meta.E_FilteredClassif

Test output

Tester: weka.experiment.PairedCorrectedTTester
Analysing: Percent_correct
Datasets: 40
Resultsets: 2
Confidence: 0.05 (two tailed)
Sorted by: -
Date: 09/09/09 16:17

Dataset	(1) meta.E_F	(2) meta.
anneal	(100) 99.09	99.09
arrhythmia	(100) 67.88	68.06
audiology	(100) 85.04	85.04
autos	(100) 78.45	78.25
balance-scale	(100) 69.59	69.59
bridges-version1-weka.fil	(100) 67.74	67.74
bridges-version2-weka.fil	(100) 64.15	63.87
car	(100) 90.13	90.13
cmc	(100) 48.98	48.98
dermatology	(100) 93.43	93.46
ecoli	(100) 80.19	80.19
flags	(100) 59.12	59.27

start | TeXnicCen... | weka-3.5... | 3 Java(T... | slides4mu... | untitled -... | EN | 16:18

Figure: Weka Experimenter: Analyse tab



The approximate algorithm, A-NPI-M

Based on an algorithm by Abellan and Moral for finding the maximum entropy distribution within a credal set

- NPI gives set of probability intervals

$$\mathcal{L} = [l_j, u_j] = [\underline{P}(c_j), \overline{P}(c_j)]$$

- These are F-probabilities
 - The probability of any event can be defined in terms of these single-category probabilities
- The credal set associated with the NPI lower and upper probabilities can be expressed by the set \mathcal{L}



The approximate algorithm, A-NPI-M

The algorithm A-NPI-M is applied to the credal set \mathcal{L}

- For each category, $p(c_j)$ is initially set to l_j
- The remaining probability mass is shared evenly between categories, beginning with those observed least often
- At each step, probabilities are increased by $\frac{1}{n}$ until they reach the value u_j or until all probability mass has been distributed

The resulting distribution is used to build classification trees



Comparison to other methods

Classification trees using A-NPI-M were compared to 4 other methods:

- 1 Trees using IDM
 - 2 Trees with precise probabilities and IG split criterion
 - 3 Trees with precise probabilities and IGR split criterion
 - 4 More complex procedure involving pruning (J48)
-
- Experiment was carried out on 40 data sets
 - Classifiers were compared pairwise
 - Numbers of correct classifications were compared



Results

Number of Wins, Ties and Losses (W/T/L) for each classifier:

	<i>IDM</i>	<i>NPI</i>	<i>IG</i>	<i>IGR</i>	<i>J48</i>
<i>IDM</i>	-	(19/2/19)	(18/2/20)	(15/2/23)	(17/1/22)
<i>NPI</i>	(19/2/19)	-	(18/2/20)	(15/2/23)	(17/1/22)
<i>IG</i>	(20/2/18)	(20/2/18)	-	(18/3/19)	(19/1/20)
<i>IGR</i>	(23/2/15)	(23/2/15)	(19/3/18)	-	(21/1/18)
<i>J48</i>	(22/1/17)	(22/1/17)	(20/1/19)	(18/1/21)	-
W-L	15	15	-4	-20	-8

- The performance of A-NPI-M is similar to that of the IDM
- A-NPI-M performs better than the other classifiers here



Problem

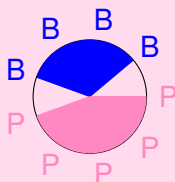
- The A-NPI-M algorithm finds the maximum entropy distribution in the credal set \mathcal{L}

- Some distributions in this set are not compatible with the probability wheel model



Example

- Possible categories $\{B, P, R, Y, O\}$ with observation counts $\{4, 5, 0, 0, 0\}$



- The credal set \mathcal{L} is $\{[\frac{3}{9}, \frac{5}{9}]; [\frac{4}{9}, \frac{6}{9}]; [0, \frac{1}{9}]; [0, \frac{1}{9}]; [0, \frac{1}{9}]\}$
- A-NPI-M gives the distribution $\{\frac{3}{9}, \frac{4}{9}, \frac{2}{27}, \frac{2}{27}, \frac{2}{27}\}$
- There is no valid configuration of the wheel that corresponds to this distribution



The exact algorithm, NPI-M

The exact algorithm finds the maximum entropy distribution consistent with the probability wheel model

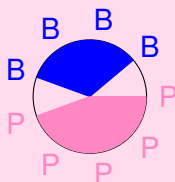
- For each category, $p(c_j)$ is initially set to l_j
- The remaining probability mass is shared **as evenly as possible** between categories, beginning with those observed least often
- At each step, probabilities are increased by $\frac{1}{n}$ until they reach the value u_j or until all probability mass has been distributed

This leads to a distribution which is as uniform as possible but still corresponds to a valid configuration of the wheel



Example

- Possible categories $\{B, P, R, Y, O\}$ with observation counts $\{4, 5, 0, 0, 0\}$



- NPI-M gives the distribution $\left\{\frac{3}{9}, \frac{4}{9}, \frac{1}{9}, \frac{1}{18}, \frac{1}{18}\right\}$
- This is as close to uniform as possible while still corresponding to a valid configuration of the wheel



Comparison of NPI-M and A-NPI-M

We implemented NPI-M for building classification trees in Weka

- Comparison of NPI-M and A-NPI-M was carried out on 40 data sets
- Numbers of correct classifications were compared



Comparison of NPI-M and A-NPI-M

Results

Percentage of correct classifications for each method:

Dataset	(1)	(2)
arrneal	99.09	99.09
arrhythmia	67.88	68.06
audiology	85.04	85.04
autos	78.45	78.25
balance-scale	69.59	69.59
bridges-version1	67.74	67.74
bridges-version2	64.15	63.87
car	90.13	90.13
cmc	48.98	48.98
dermatology	93.43	93.46
ecoli	80.19	80.19
flags	59.12	59.27
hypothyroid	99.33	99.33
iris	93.40	93.40
letter	78.77	78.77
lung-cancer	41.33	41.33
lymphography	73.68	73.68
mfeat-factors	81.71	81.68
mfeat-fourier	68.90	68.92
mfeat-karhunen	73.14	73.15
mfeat-morphological	69.78	69.78
mfeat-pixel	79.99	79.92
mfeat-zernike	64.19	64.24
nursery	95.15	94.99 ●
optdigits	78.95	78.98
page-blocks	96.08	96.10
pendigits	89.37	89.37
postoperative-patient-data	71.11	71.11
primary-tumor	39.21	39.48
segment	94.18	94.20
soybean	93.29	93.35
spectrometer	43.32	43.33
splice	93.25	93.25
sponge	94.48	94.48
tae	46.78	46.78
vehicle	69.39	69.39
vowel	75.92	75.95
waveform	73.99	73.99
wine	92.02	92.02
zoo	95.53	95.53

○, ● statistically significant improvement or degradation

- Performance is not significantly different on most data sets
- NPI-M performs significantly better on 'nursery' data set



Nursery data set

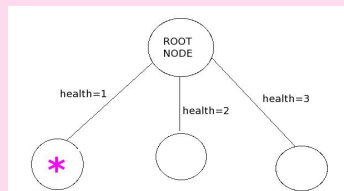
Data set taken from applications for places at a private nursery school

- Applicants classified in terms of how likely they are to be accepted
- 5 categories: c_1, c_2, c_3, c_4, c_5
- 8 attribute variables



Nursery data set

Most informative attribute is 'health'



At '*', counts in $\{c_1, c_2, c_3, c_4, c_5\}$ are $\{0, 0, 0, 1854, 2466\}$

- A-NPI-M and NPI-M both give $p_{maxE}(c_4) = \frac{1853}{4320}$ and $p_{maxE}(c_5) = \frac{2465}{4320}$
- A-NPI-M gives equal probability $\frac{1}{6480}$ to c_1, c_2 and c_3
- NPI-M gives probabilities $\{\frac{1}{4320}, \frac{1}{8640}, \frac{1}{8640}\}$ to $\{c_1, c_2, c_3\}$

In the branch of the tree beginning at '*', A-NPI-M and NPI-M will always give different distributions



Future work

- Investigation into the use of the maximum entropy algorithm for NPI with subcategories

- Study of classifiers which use NPI with various different uncertainty measures

References

- Abellán, J. and Moral, S. (2003) Maximum entropy for credal sets *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **11(5)**, 587-597.
- Augustin, T. and Coolen, F.P.A. (2004) Nonparametric predictive inference and interval probability *Journal of Statistical Planning and Inference*, **124**, 251-272.
- Coolen, F.P.A. and Augustin, T. (2005) Learning from multinomial data: a nonparametric predictive alternative to the Imprecise Dirichlet Model *ISIPTA '05*, 125-134.
- Coolen, F.P.A. (2006) On nonparametric predictive inference and objective Bayesianism. *Journal of Logic, Language and Information*, **15**, 21-47.
- Coolen, F.P.A. and Augustin, T. (2009) A nonparametric predictive alternative to the Imprecise Dirichlet Model: the case of a known number of categories *International Journal of Approximate Reasoning*, **50**, 217-230.