Bayesian Network model of NHTSA: FARS Data

Joseph Miller

Rutgers University

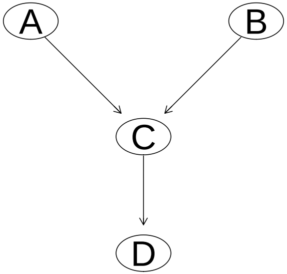
arch1190@gmail.com

June 2, 2016

Overview

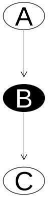
Introduction to Belief Networks

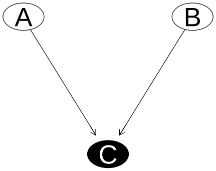
- Encodes random variables and their conditional (in)dependencies. In general, $\Pr(X|\text{par}(X))$
- Can dramatically reduce the number of parameters (example: 15 vs. 8 if binary data) and model size.
- All BNs are DAGs.
- Different DAGs can correspond to the same factorization.



$$\Pr(A, B, C, D) = \Pr(D|C)\Pr(C|A, B)\Pr(A)\Pr(B)$$

Examples





In a causal chain, when B is known, C is independent of A: information "stops" at B.

$$\Pr(C|B)\Pr(B|A)\Pr(A) \Rightarrow$$
$$\Pr(C,A|B) =$$
$$\Pr(C|B)\Pr(A) \times \left[\frac{\Pr(B|A)}{\Pr(B)}\right]$$

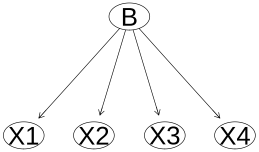
When C is known, A and B may be dependent: information "flows" through C.

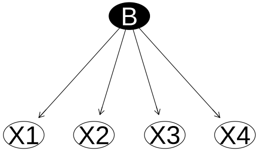
$$\Pr(A)\Pr(B)\Pr(C|A,B) \Rightarrow$$
$$\Pr(A,B|C) =$$
$$\Pr(B)\Pr(A|B) \times \left[\frac{\Pr(C|B)}{\Pr(C)}\right]$$

Coin Flip Model

- Consider an IID coin flip experiment with $\Pr(\mathrm{H}) = \theta$.
- B is a latent variable that we can model hierarchically where $X_i \sim \mathrm{Bern}(\theta)$.
- Is X_4 independent of the event $(X_1, X_2, X_3) =$(H,H,H)?
- What about if $\theta = 1/2$?

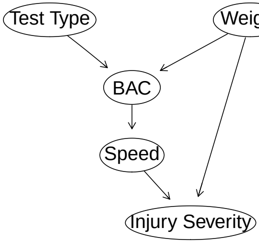
BNs generalize hierarchical modeling



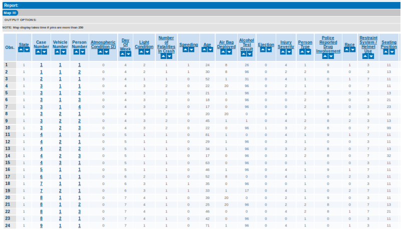


BN Sketch

- Plausible hierarchical structure in the data.
- BAC only acts on Injury Severity through Speed and correlates with it through Weight.
- Number of passengers (not shown) influences number injured.
- Passengers influence driver through distraction (not shown).



BN Sketch



Report:

Map it!

OUTPUT OPTIONS:

NOTE: Map display takes time if pins are more than 250

Obs.	State	Case Number	Vehicle Number	Person Number	Atmospheric Condition (2)	Day Of Week	Light Condition	Number of Fatalities In Crash	Speeding	Age	Air Bag Deployed	Alcohol Test Result	Ejection	Injury Severity	Person Type	Police Reported Drug Involvement	Race	Restraint System / Helmet Use	Seating Position
1	1	1	1	1	0	4	2	1	1	24	8	26	0	4	1	9	1	3	11
2	1	1	1	2	0	4	2	1	1	30	8	96	0	2	2	8	0	3	13
3	1	2	1	1	0	4	1	1	0	52	1	31	0	4	1	0	1	7	11
4	1	3	1	1	0	4	3	2	0	22	20	96	0	2	1	9	0	7	11
5	1	3	1	2	0	4	3	2	0	21	1	96	0	0	2	8	0	3	13
6	1	3	1	3	0	4	3	2	0	18	0	96	0	0	2	8	0	3	21
7	1	3	1	4	0	4	3	2	0	17	0	96	0	0	2	8	0	3	23
8	1	3	2	1	0	4	3	2	0	20	20	0	0	4	1	9	2	3	11
9	1	3	2	2	0	4	3	2	0	45	1	1	0	4	2	8	2	3	13
10	1	3	2	3	0	4	3	2	0	22	0	96	1	3	2	8	0	7	99
11	1	4	1	1	0	5	1	1	0	81	1	0	0	4	1	9	1	7	11
12	1	4	2	1	0	5	1	1	0	29	1	96	0	3	1	0	0	3	11
13	1	4	2	2	0	5	1	1	0	34	1	96	0	3	2	8	0	7	13
14	1	4	2	3	0	5	1	1	0	17	0	96	0	3	2	8	0	7	32
15	1	4	3	1	0	5	1	1	0	63	0	96	0	0	1	0	0	3	11
16	1	5	1	1	0	5	1	1	0	46	1	96	0	4	1	9	1	7	11
17	1	6	1	1	0	6	2	1	0	52	8	0	0	4	1	0	2	3	11
18	1	7	1	1	0	6	3	1	1	35	0	96	0	0	1	0	0	3	11
19	1	7	2	1	0	6	3	1	1	33	1	17	0	4	1	0	2	7	11
20	1	8	1	1	0	7	4	1	0	39	20	0	0	2	1	9	0	3	11
21	1	8	1	2	0	7	4	1	0	25	20	96	0	2	2	8	0	7	13
22	1	8	1	3	0	7	4	1	0	46	0	0	0	4	2	8	1	7	21
23	1	8	2	1	0	7	4	1	0	42	0	96	0	0	1	0	0	3	11
24	1	9	1	1	0	7	1	1	0	71	1	96	0	4	1	0	1	3	11

Goal:

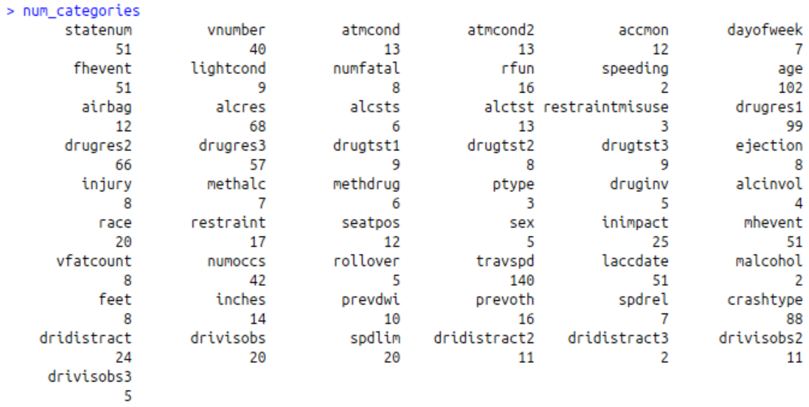
Build a model to predict VFATCOUNT (vehicle fatality count).

1. Prepare Data
2. Structure Learning
3. Parameter Learning
4. Evaluate Model

Difficulties:

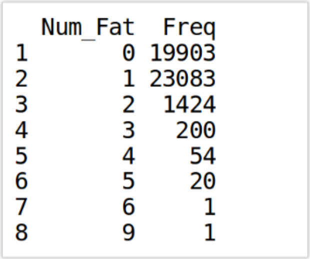
- Mixed categorical and continuous
- Categorical data with many levels and few observations in some levels
- Many different missing value codes
- $>$ 100 variables in the entire dataset
- Observations with multiple codes

Profiling the Data



```
> num_categories
     statenum         vnumber         atmcond        atmcond2          accmon       dayofweek
           51              40              13              13              12               7
      fhevent       lightcond        numfatal            rfun        speeding             age
           51               9               8              16               2             102
       airbag          alcres          alcsts          alctst restraintmisuse        drugres1
           12              68               6              13               3              99
     drugres2        drugres3        drugtst1        drugtst2        drugtst3        ejection
           66              57               9               8               9               8
       injury         methalc        methdrug           ptype         druginv        alcinvol
            8               7               6               3               5               4
         race       restraint         seatpos             sex        inimpact         mhevent
           20              17              12               5              25              51
    vfatcount         numoccs        rollover         travspd        laccdate        malcohol
            8              42               5             140              51               2
         feet          inches         prevdwi         prevoth          spdrel       crashtype
            8              14              10              16               7              88
  dridistract        drivisobs          spdlim    dridistract2    dridistract3       drivisobs2
           24              20              20              11               2              11
   drivisobs3
            5
```

Vehicle Fatality Count



```
  Num_Fat  Freq
1       0 19903
2       1 23083
3       2  1424
4       3   200
5       4    54
6       5    20
7       6     1
8       9     1
```

- Variable: VFATCOUNT
- High entropy
- Will bin higher values

Cleaning Data

```
strings <- strsplit(as.character(fars$drivisobs),
split = " ")

l <- vector(mode="integer", length=length(strings))
for (i in 1:length(strings)) { # Get vector with
lengths of strings.
  l[i] <- length(strings[[i]])
}
table(l) # Histogram of number of values in each entry
```

```
l
    1     2     3
44650    30     6
```

Transformations

New Variables

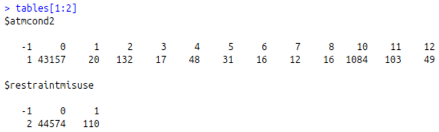
- Four new columns, two each from DRIVISOBS and DRIDISTRACT.
- Transformed INCHES (0-12) and feet (2-7) to continuous variable HEIGHT.

Remaining Work

1. Eliminate highly unbalanced features.
2. Merging categories (will handle missing values).
3. Structure/Parameter Learning.
4. Evaluate Models.

Low Variability Features

```r
library(caret);
library(dplyr)
cat_data <- select(fars,
-age, -height, -spdlim,
-travspd)

x <- nzv(cat_data, names
= T)
tables <- list()
for (i in x) {
  tables[[i]]
<-table(cat_data[,i])
}
```

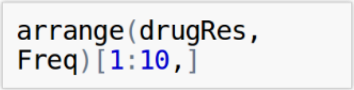


```
> tables[1:2]
$atmcond2

   -1     0     1     2     3     4     5     6     7     8    10    11    12
    1 43157    20   132    17    48    31    16    12    16  1084   103    49

$restraintmisuse

   -1     0     1
    2 44574   110
```

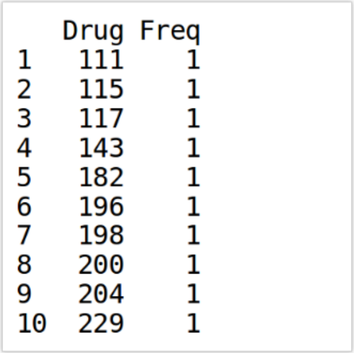
Heading

- 13/54 variables with near-zero-variability:
  - $<$ 10% unique values.
  - Ratio of frequency of most common to second most common category $> 20:1$.
- Parameter estimates too variables.
- Will handle metric variables differently.

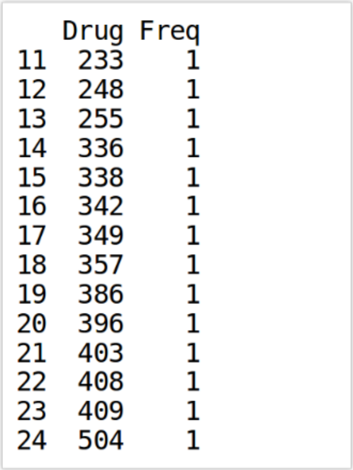
Merging Categories



```
arrange(drugRes,
Freq)[1:10,]
```



```
   Drug Freq
1   111    1
2   115    1
3   117    1
4   143    1
5   182    1
6   196    1
7   198    1
8   200    1
9   204    1
10  229    1
```

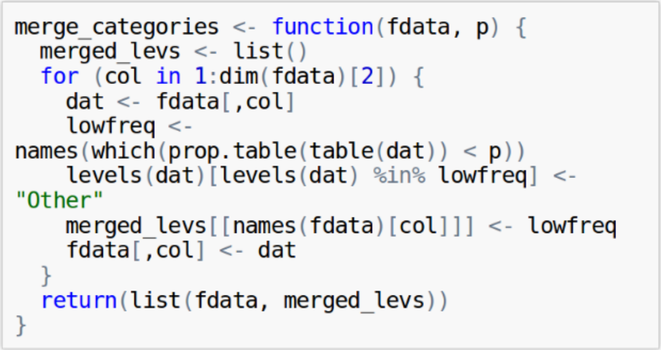


```
   Drug Freq
11  233    1
12  248    1
13  255    1
14  336    1
15  338    1
16  342    1
17  349    1
18  357    1
19  386    1
20  396    1
21  403    1
22  408    1
23  409    1
24  504    1
```

Merging Categories

- Drug Test Results has 99 observed categories, 150+ possible. Others bad, too.
- Want:
  - Combine levels of cat. variables that are most similar in relation to target variable.
    - Depenent on learned structure of BN, which is dependent on how the categories are merged.
    - Data is not ordinal (merge categories).
    - Reminiscent of cluster analysis.
- Am not aware of an appropriate/efficient algorithm.
- Forfeit my ability to predict elements within these categories.

Merging Categories Below Cutoff Freq



```
merge_categories <- function(fdata, p) {
  merged_levs <- list()
  for (col in 1:dim(fdata)[2]) {
    dat <- fdata[,col]
    lowfreq <-
names(which(prop.table(table(dat)) < p))
    levels(dat)[levels(dat) %in% lowfreq] <-
"Other"
    merged_levs[[names(fdata)[col]]] <- lowfreq
    fdata[,col] <- dat
  }
  return(list(fdata, merged_levs))
}
```

Merging Categories Below Cutoff Freq

```
out <-
merge_categories(fdata,
0.01)
fdata <- out[[1]];
merged_levs <- out[[2]]
```

- Smallest allowed factor is now 0.01 × 44686.
- Much room for improvement exists!

```
arrange(drugRes,
Freq)[1:10,]
```

```
      Drug   Freq
1      600    471
2      999    860
3      605    879
4      996    974
5       95   1036
6      997   1237
7       -1   1426
8    Other   4065
9        1   9953
10       0  23785
```

Discretizing Metric Data

```
mdata <- discretize(data
= temp, method =
"hartemink", breaks = 7,
ibreaks = 30,
idisc="quantile")
```

- Cuts numerical data into 30 quantiles, combines intervals in a way that reduces pairwise mutual information minimally.
- Final intervals depend on other variables in data frame: made 6-variable drame with 5 metric variables and the target variable.

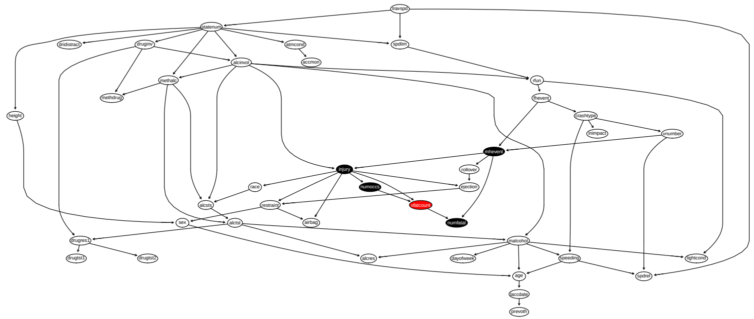
```
levels(mdata$height)
```

```
[1] "(33.9998,65.0003]"
"(65.0003,68.9999]"
"(68.9999,75.0007]"
[4] "(75.0007,83.9991]"
"(83.9991,84.001]"
```

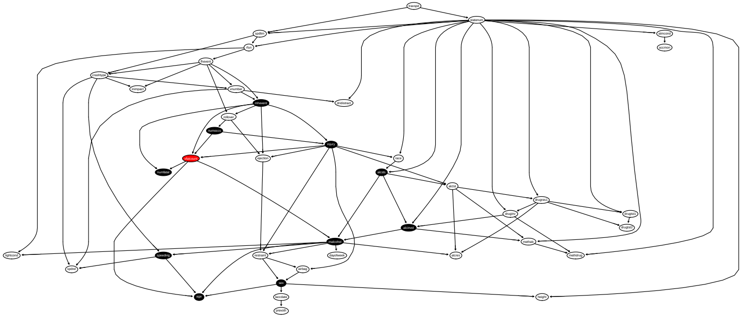
Structure Learning

- Efficient algorithms exist to learn tree structures. Arbitrary structures: search space is too big.
- Two notable methods for comparing structures:
  - Independence tests: reduce search space by restricting to learning trees.
  - Scoring criterion: can more efficiently search structures. No guarantee of convergence. Must choose search algorithm.
- Chose BIC/MDL and Bayesian Dirichlet Method. Space of DAGs must still be searched.
  - Greedy Search with perturbations and restarts.

BIC Optimized Model

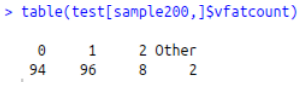


Bayesian Model



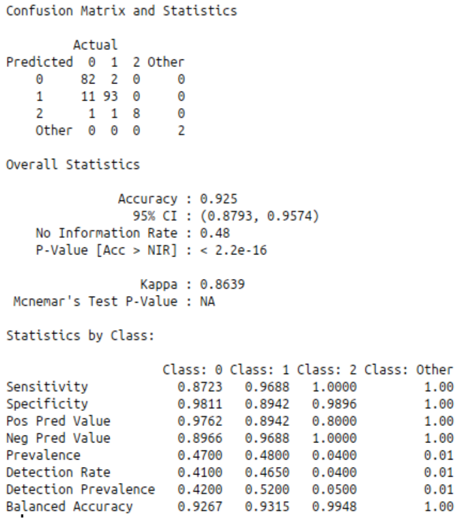
Parameter Learning and Classification

- Parameter learning is straightforward: MAP estimates with uniform Dirichlet prior on all parameters.
- Start with predictions given Markov Blanket (black nodes) on target variable: number of fatalities in driver's vehicle.
- BIC Model: 185/200 correct classifications, 42 seconds.
- Bayesian Model: 194/200 correct classifications, 191 seconds.
- 200 samples, low SE.



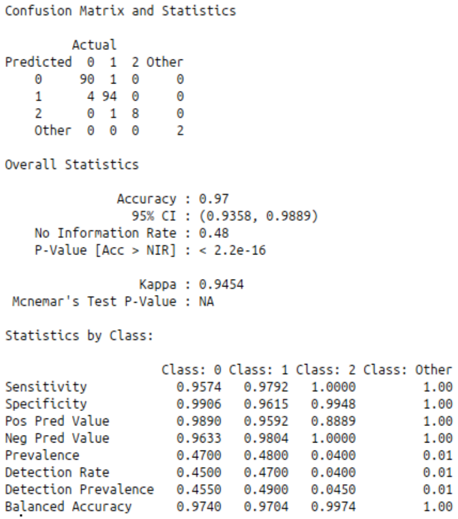
```
> table(test[sample200,]$vfatcount)

    0     1     2 Other
   94    96     8     2
```

Summary: BIC-scored Model



```
Confusion Matrix and Statistics

         Actual
Predicted  0  1  2 Other
    0     82  2  0     0
    1     11 93  0     0
    2      1  1  8     0
    Other  0  0  0     2

Overall Statistics

               Accuracy : 0.925
                 95% CI : (0.8793, 0.9574)
    No Information Rate : 0.48
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8639
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: Other
Sensitivity            0.8723   0.9688   1.0000         1.00
Specificity            0.9811   0.8942   0.9896         1.00
Pos Pred Value         0.9762   0.8942   0.8000         1.00
Neg Pred Value         0.8966   0.9688   1.0000         1.00
Prevalence             0.4700   0.4800   0.0400         0.01
Detection Rate         0.4100   0.4650   0.0400         0.01
Detection Prevalence   0.4200   0.5200   0.0500         0.01
Balanced Accuracy      0.9267   0.9315   0.9948         1.00
```

Summary: Bayesian Model



```
Confusion Matrix and Statistics

         Actual
Predicted  0  1  2 Other
    0     90  1  0     0
    1      4 94  0     0
    2      0  1  8     0
    Other  0  0  0     2

Overall Statistics

               Accuracy : 0.97
                 95% CI : (0.9358, 0.9889)
    No Information Rate : 0.48
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9454
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: Other
Sensitivity            0.9574   0.9792   1.0000         1.00
Specificity            0.9906   0.9615   0.9948         1.00
Pos Pred Value         0.9890   0.9592   0.8889         1.00
Neg Pred Value         0.9633   0.9804   1.0000         1.00
Prevalence             0.4700   0.4800   0.0400         0.01
Detection Rate         0.4500   0.4700   0.0400         0.01
Detection Prevalence   0.4550   0.4900   0.0450         0.01
Balanced Accuracy      0.9740   0.9704   0.9974         1.00
```

Evaluating Forecasts

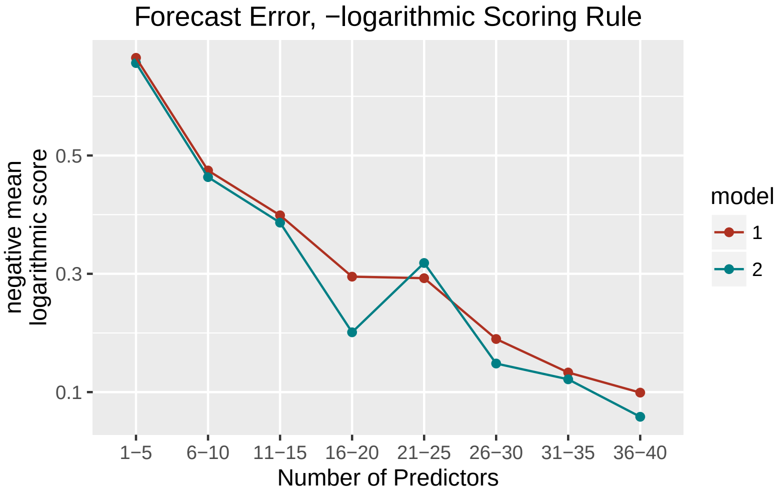
Algorithm:

1. N, picked uniformly from 1 to 40 (41 total variables).
2. Choose N PREDICTORS, picked without replacement from the columns.
3. Predict VFATCOUNT with each model.
4. Repeated (1-3) 1000 times.
5. Calculate average $\log(p)$ of predicted class, a proper scoring rule.

Results:

- Correct classification rate:
  - 85.8%
  - 86.2%
- average $\log(p)$
  - -0.3226594
  - -0.2969937

Error Trend


Forecast Error, −logarithmic Scoring Rule
negative mean logarithmic score
0.5
0.3
0.1
1−5
6−10
11−15
16−20
21−25
26−30
31−35
36−40
Number of Predictors
model
1
2

References

- *bnlearn* R package and documentation, Marco Scutari
- *gRain* R package and documentation, Soren Hojsgaard
- *ggplot2* R package, Hadley Wickham
- *dplyr* R package, Hadley Wickham

The End