

Schedule

This Week

Today:

Autoclass

Context specific independence and local structure.

Friday:

Normal-Wishart priors

Progress report #2

You should be 80% done with the technical portion of your project.

23 April: Papers due (No extensions)

Classification

Variables: X : *Features or Attributes* C : *Class*

Supervised

Data: $(C[m], X_1[m], \dots, X_n[m])$

Learn: $F: X \rightarrow C$

Learn a function that maps observations into a class.

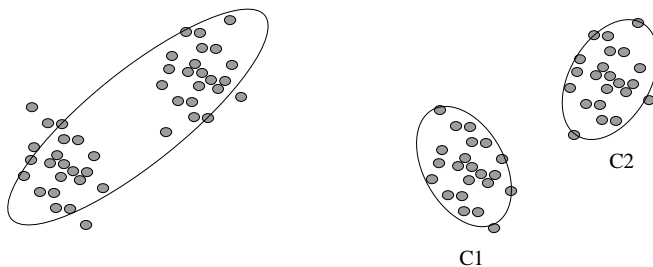
Unsupervised Classification (Clustering)

Data: $(X_1[m], \dots, X_n[m])$

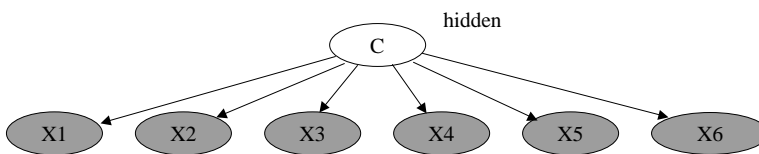
Learn: $\max_{R, \Theta} P\{X\} = \prod_m \sum_{j=1}^R P\{X[m] | C_j[m]\} P\{C_j[m]\}$

Learn a hidden variable that explains "clusters" in the data.

Clustering



Autoclass Assumptions



$$P\{\Theta_C\} = K \prod_{j=1}^{|C|} \theta_j^{1/|C|}$$

Discrete:
$$P\{\Theta_{X|C}\} = K \prod_{k=1}^{|X|} \theta_{k|C}^{1/|L_k|}$$

$$\hat{\theta}_{X_i=C=j} = \frac{E(N_{ijk})}{E(N_{ij})}$$

Gaussian (unknown mean and variance):

$$P\{\mu_{X|C}\} = \frac{1}{\mu_{X|C,\max} - \mu_{X|C,\min}}$$

$$\hat{\mu}_{X_i=C=j} = E(X_{ij})$$

$$P\{\sigma_{X|C}\} = \sigma_{X|C}^{-1} \left[\log \frac{\sigma_{X|C,\max}}{\sigma_{X|C,\min}} \right]$$

$$\sigma_{X_i=C=j}^2 = \frac{N_j}{N_j + 1} E((X_{ij} - \mu_{ij})^2)$$

Autoclass C: Application and Distribution Types

Web address:

<http://ic-www.arc.nasa.gov/ic/projects/bayes-group/autoclass/autoclass-c-program.html>

Variable types

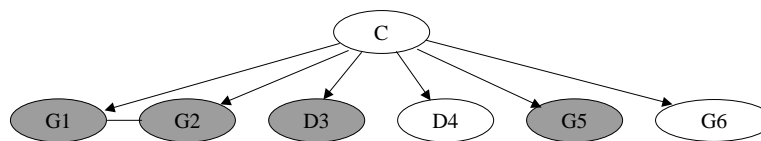
single_multinomial - discrete attribute multinomial model, including missing values.

single_normal - real valued attribute model with no missing values.

single_normal_missing - real valued attribute model with missing values.

multi_normal - real valued covariant normal model with no missing values.

Autoclass C: Distribution Types Illustrated



normal variable combinations
need to be specified in advance.

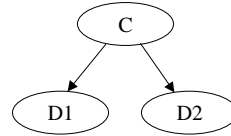
Autoclass-C Experience

Dependencies in discrete variables

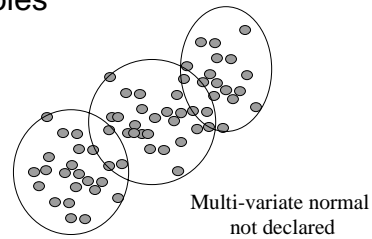
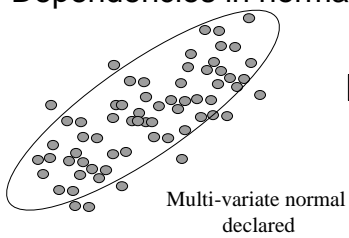
Discrete/Discrete dependency
in "generator"



Explained through
classification node.



Dependencies in normal variables



Autoclass C Experience

Turbine Engine Data

Engine parameters
(fan speed, exhaust gas temperature, altitude, + 65 others)

Desired behavior:

Identify differences in control systems between engines.

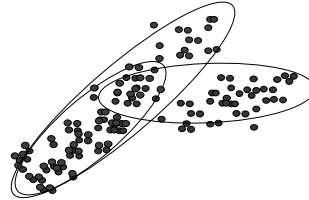
Actual behavior:

Focussed on
violations of normality assumption

Autoclass C Experience, continued

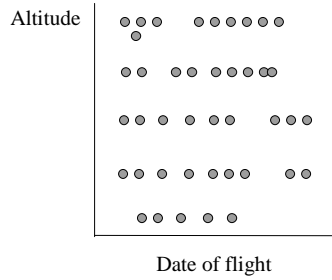
Sensitivity to Gaussian Assumption

Turbine engine model
"Desired Clustering"



Turbine engine model
"Actual Clustering"

FAA flight rules:
Flights spaced 2000 ft apart
Alternating layers of east/west
headings
Moral: Autoclass focusses on
explaining clusters in data
without regard to the perceived
importance of the clusters.



Local Structure and Learning

Context Specific Independence (CSI)

(Boutilier, et al, UAI-96)

X and Y are contextually independent given Z and context c in $\text{Val}(C)$ iff

$$P\{X | Z, c, Y\} = P\{X | Z, c\} \quad \text{whenever} \quad P\{Y, Z, c\} > 0$$

Examples:

P(D)	C=T	C=F
A=T, B=T	0.3	0.1
A=T, B=F	0.5	0.1
A=F, B=T	0.2	0.9
A=F, B=F	0.7	0.9

P(D)	C=T	C=F
A=T, B=T	0.3	0.1
A=T, B=F	0	0
A=F, B=T	0.2	0
A=F, B=F	0	0

Noisy-Or:

$$P\{cause_i | effect = F, cause_j\} = P\{cause_i | effect = F\}$$

Local Structure [Friedman+Goldszmidt, 97]

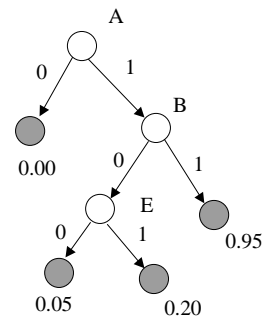
Representation for local structure (this lecture)

Default Table

Alarm Set	B	E	P(Sound)
T	T	T	0.95
T	T	F	0.95
T	F	T	0.2
T	F	F	0.05
	*		0

5 Parameters

Decision Tree



4 Parameters

Why Local Structure?

Fewer parameters required to specify each distribution

- Lower MDL score

- Less parameters implies larger # of samples per each parameter

 - BDe: Higher score.

 - More robust likelihood estimation.

Closer match to underlying distribution

- MDL Penalty is exponential in the number of parents.

- CSI can allow nodes to have a large number of parents without necessarily incurring the full exponential penalty.

Outline

MDL Score for local structure

- General (Review)

- Default tables

- Trees

BDe Score for local structure

- General (Review)

- Local Structure

Search

- Top Level Graph Search (Review)

- Top Level Graph Search (Local Structure)

- Default tables

- Trees

Reminder: MDL Score

Recall:

We would like to store the data as compactly as possible

Use a belief network to compress the data.

Minimize $DL = DL(\text{BN}) + DL(\text{Data})$

$$DL(G, D) = DL_{\text{Graph}}(G) + DL_{\text{Dist}}(G, M) + DL_{\text{Data}}(D | B)$$

$$DL_{\text{Data}}(D | B) = - \sum_{m=1}^M \log P_B \{X[m]\}$$

Assuming complete data:

$$DL_{\text{Data}}(D | B) = - \sum_{j=1}^N \sum_{x_i, pa_i} N(x_i, pa_i) \log P\{x_i | pa_i\}$$

ML Estimate

$$\hat{P}(x_i | pa_i) = \frac{N(x_i, pa_i)}{N(pa_i)}$$

MDL:

$$MDL = -DL$$

MDL Score with local structure

$$DL(D, L_1, \dots, L_n, G) = DL(G) + \sum_{i=1}^N (DL_{\text{local}}(L_i) + DL_{\text{param}}(L_i)) \\ + M \sum_{i=1}^N H(X_i | \Psi_i)$$

Define

Ψ_i to be the set of *partitions* of Pa_i

...mutually-exclusive subsets of the values for the parents.

L_i is the local structure for $X_i | Pa_i$

Default Table Structure

Structure:

$$(Rows_i, \Theta_i)$$

$Rows_i(1)$	Alarm Set	B	E	P(Sound)
	T	T	T	0.95
	T	T	F	0.95
	T	F	T	0.2
$Rows_i(k)$	T	F	F	0.05
<i>Default Row:</i>		*		0

Partitions:

$$\Psi_i = Rows_i$$

MDL Score for Default Table

Define: $\|Pa_i\|$ to be the cardinality of set of parent values.

Need to encode k rows:

Encode the number of rows: $\log k$

Enumerate the combinations of k rows and index over the combinations:

$$\log \binom{\|Pa_i\|}{k}$$

DL:

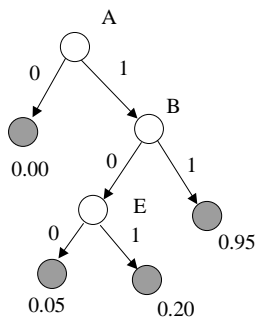
$$DL(X_i | Pa_i) = \log k + \log \binom{\|Pa_i\|}{k} + DL_{param}(i)$$

Review: DL for parameters

Given M data values:

$$DL_{Param}(X_i | Pa_i) = \frac{1}{2} (\|X_i\| - 1) \|\Psi_i\| \log M$$

MDL Score for Tree



Recursive formula:

Distinguish between leaf and internal node: 1 bit

Describe test variable

Variables can only be used once, so just need to decide between one of the k variables that haven't been tested yet: $\log k$ bits.

DL for tree:

$$DL_{\tau}(\tau, k) = \begin{cases} 1 & \text{if } \tau \text{ is a leaf.} \\ 1 + \log(k) + \sum_i DL_{\tau_i}(\tau_i, k-1) & \text{otherwise} \end{cases}$$

where τ_i is the i-th subtree of τ

Total DL: $\Psi_i = \text{Leaves}$

$$DL = DL_{\tau} + DL_{Param}$$

ML Parameter Estimate

$$\hat{P}\{x_i | v\} = \frac{N(x_i, v)}{N(v)}, \quad v \in \Psi_i$$

BDe Score (Review)

Score:

$$P\{G | D\} = \alpha P\{D | G\} P\{G\}$$

$$P\{D | G\} = \int P\{D | \Theta, G\} P\{\Theta | G\} d\Theta$$

Given a Dirichlet prior:

$$P\{D | G\} = \prod_{i=1}^N \prod_{pa_i} \frac{\Gamma(\alpha(pa_i))}{\Gamma(\alpha(pa_i) + N(pa_i))} \prod_{x_i} \frac{\Gamma(\alpha(x_i, pa_i) + N(x_i, pa_i))}{\Gamma(\alpha(x_i, pa_i))}$$

BDe for local structure

BDe:

$$P\{G, L | D\} \propto P\{D | L, G\} P\{L | G\} P\{G\}$$

Define a structure prior:

$$P\{L | G\} \propto 2^{-\sum_{i=1}^N DL_{local}(L_i)}$$

Assume parameter independence.

$$P\{D | L, G\} = \prod_{i=1}^N \prod_{v \in val(\Psi_i)} \int \prod_{x_i} \theta_{x_i|v}^{N(x_i, v)} P\{\Theta_{x_i|v} | L_i, G\} d\Theta_{x_i|v}$$

BDe for local structure, cont'd

Score:

$$P\{D | G\} = \prod_{i=1}^N \prod_{v \in val(\Psi_i)} \frac{\Gamma(\alpha(v))}{\Gamma(\alpha(v) + N(v))} \prod_{x_i} \frac{\Gamma(\alpha(x_i, v) + N(x_i, v))}{\Gamma(\alpha(x_i, v))}$$

Where: $\alpha(x_i, v) = \sum_{pa_i} I[v, pa_i] N(x_i, pa_i)$

$$\alpha(v) = \sum_{pa_i} I[v, pa_i] N(pa_i)$$

MAP:

$$\tilde{P}\{x_i | v\} = \frac{N(x_i, v) + \alpha(x_i, v)}{N(v) + \alpha(v)}, \quad v \in \Psi_i$$

Greedy Structure Learning (Review)

```

procedure LearnNetwork( )
  Let  $G_{current} \leftarrow G_0$ 
  do {
    Generate all successors  $S \leftarrow \{G_1, \dots, G_n\}$  of  $G_{current}$ 
     $Score_{max} \leftarrow \max_{G \in S} Score(G)$ 

    if  $Score_{max} > Score(G_{current})$  then

      Let  $G_{new} \leftarrow \arg \max_{G \in S} Score(G)$ 
  }
  while (the score is increasing)
  return  $G_{current}$ 

```

Learning Local Structure

```

procedure LearnNetworkLocal( )
  Let  $G_{current} \leftarrow G_0$ 
  do {
    Generate all successors  $S \leftarrow \{G_1, \dots, G_n\}$  of  $G_{current}$ 

     $S' \leftarrow \{\text{LearnLocal}(G_1), \dots, \text{LearnLocal}(G_n)\}$ 

     $Score_{max} \leftarrow \max_{G \in S'} Score(G)$ 
    if  $Score_{max} > Score(G_{current})$  then

      Let  $G_{new} \leftarrow \arg \max_{G \in S'} Score(G)$ 
  }
  while (the score is increasing)
  return  $G_{current}$ 

```

Learning Local Structure

Both BDe and MDL can be written as:

$$Score = K + \sum_i \sum_{v \in Val(\Psi_i)} Score(X_i | v)$$

Given the local structure type, we can optimize the local structure for each v separately. Optimize:

$$Score(X_i | v)$$

Learn Default

Learn a default table.

Greedy row choice: choose row to learn that maximizes the score.

procedure LearnDefault()

Let $Rows(L_i) \leftarrow \emptyset$

do

Let $r = \arg \max_{r \in Val(pa_i), Rows(L_i)} Score(Rows(L_i) \cup \{r\})$

if $Score(Rows(L_i) \cup \{r\}) < Score(Rows(L_i))$

return $Rows(L_i)$

$Rows(L_i) \leftarrow Rows(L_i) \cup \{r\}$

end

Learn Tree

```
procedure SimpleTree(Y)
  for  $y \in \text{Val}(Y)$  let  $l_y \leftarrow \Lambda$  (i.e., a leaf)
  return  $\langle Y, \{l_y : y \in \text{Val}(Y)\} \rangle$ 
end

procedure ExpandTree( $\Lambda$   $D$ )
  if  $D = \emptyset$  or  $X_i$  is homogeneous in  $D$  then
    return  $\Lambda$  (a leaf)
  // Growing phase
  Let  $Y_{\text{split}} = \arg \max_{Y \in \text{Pa}_i} \text{Score}(\text{SimpleTree}(Y) | D)$ 
  for  $y \in \text{Val}(Y_{\text{split}})$ 
     $D_y = \{x_i \in D : Y_{\text{split}} = y \text{ in } x_i\}$ 
     $T_y = \text{ExpandTree}(\Lambda, D_y)$ 
  //Trimming phase
  if  $\text{Score}(\Lambda) > \text{Score}(T)$  then
    return  $\Lambda$ 
  else
    return  $T$ 
end
```

Local Structure Conclusions

Simple modification to normal search algorithm

Scores for all nodes decompose given complete data.

Optimize individual scores

Default Table

Tree Learning

Noisy-Or, Noisy-Max...

Scores

MDL

BDe

Algorithms

Greedy Default Table

Greedy Tree

Next Time

Normal-Wishart priors

Progress report #2

You should be 80% done with the technical portion of your project.

Remember the project is 1/2 of your grade.