Possibilistic Graphical Models and How to Learn Them from Data

Christian Borgelt

Dept. of Knowledge Processing and Language Engineering
Otto-von-Guericke-University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
E-mail: borgelt@iws.cs.uni-magdeburg.de
Contents

- Possibility Theory
  - Axiomatic Approach
  - Semantical Considerations

- Graphical Models / Inference Networks
  - relational
  - probabilistic
  - possibilistic

- Learning Possibilistic Graphical Models from Data
  - Computing Maximum Projections
  - Naive Possibilistic Classifiers
  - Learning the Structure of Graphical Models

- Summary
Definition: Let $\Omega$ be a (finite) sample space.

A **possibility measure** $\Pi$ on $\Omega$ is a function $\Pi : 2^\Omega \rightarrow [0, 1]$ satisfying

1. $\Pi(\emptyset) = 0$ and
2. $\forall E_1, E_2 \subseteq \Omega : \Pi(E_1 \cup E_2) = \max\{\Pi(E_1), \Pi(E_2)\}$.

- Similar to Kolmogorov’s axioms of probability theory.
- From the axioms follows $\Pi(E_1 \cap E_2) \leq \min\{\Pi(E_1), \Pi(E_2)\}$.
- Attributes are introduced as random variables (as in probability theory).
- $\Pi(A = a)$ is an abbreviation of $\Pi(\{\omega \in \Omega : A(\omega) = a\})$
- If an event $E$ is possible without restriction, then $\Pi(E) = 1$.
  If an event $E$ is impossible, then $\Pi(E) = 0$. 
Interpretation of Degrees of Possibility [Gebhardt and Kruse 1993]

- Let $\Omega$ be the (nonempty) set of all possible states of the world, $\omega_0$ the actual (but unknown) state.

- Let $C = \{c_1, \ldots, c_n\}$ be a set of contexts (observers, frame conditions etc.) and $(C, 2^C, P)$ a finite probability space (context weights).

- Let $\Gamma : C \rightarrow 2^\Omega$ be a set-valued mapping, which assigns to each context the most specific correct set-valued specification of $\omega_0$. The sets $\Gamma(c)$ are called the focal sets of $\Gamma$.

- $\Gamma$ is a random set (i.e., a set-valued random variable) [Nguyen 1978]. The basic possibility assignment induced by $\Gamma$ is the mapping

$$
\pi : \Omega \rightarrow [0, 1] \\
\pi(\omega) \mapsto P(\{c \in C \mid \omega \in \Gamma(c)\}).
$$
Example: Dice and Shakers

<table>
<thead>
<tr>
<th>shaker 1</th>
<th>shaker 2</th>
<th>shaker 3</th>
<th>shaker 4</th>
<th>shaker 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>tetrahedron</td>
<td>hexahedron</td>
<td>octahedron</td>
<td>icosahedron</td>
<td>dodecahedron</td>
</tr>
<tr>
<td>1 – 4</td>
<td>1 – 6</td>
<td>1 – 8</td>
<td>1 – 10</td>
<td>1 – 12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>numbers</th>
<th>degree of possibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 4</td>
<td>$\frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} = 1$</td>
</tr>
<tr>
<td>5 – 6</td>
<td>$\frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} = \frac{4}{5}$</td>
</tr>
<tr>
<td>7 – 8</td>
<td>$\frac{1}{5} + \frac{1}{5} + \frac{1}{5} = \frac{3}{5}$</td>
</tr>
<tr>
<td>9 – 10</td>
<td>$\frac{1}{5} + \frac{1}{5} = \frac{2}{5}$</td>
</tr>
<tr>
<td>11 – 12</td>
<td>$\frac{1}{5} = \frac{1}{5}$</td>
</tr>
</tbody>
</table>
From the Context Model to Possibility Measures

Definition: Let $\Gamma : C \rightarrow 2^\Omega$ be a random set. The possibility measure induced by $\Gamma$ is the mapping

$$\Pi : 2^\Omega \rightarrow [0, 1],$$
$$E \mapsto P(\{c \in C \mid E \cap \Gamma(c) \neq \emptyset\}).$$

Problem: From the given interpretation it follows only:

$$\forall E \subseteq \Omega : \max_{\omega \in E} \pi(\omega) \leq \Pi(E) \leq \min \left\{1, \sum_{\omega \in E} \pi(\omega)\right\}.$$
Attempts to solve the indicated problem:

- Require the focal sets to be **consonant**:

  **Definition:** Let $\Gamma : C \rightarrow 2^\Omega$ be a random set with $C = \{c_1, \ldots, c_n\}$. The focal sets $\Gamma(c_i)$, $1 \leq i \leq n$, are called **consonant**, iff there exists a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_n}$, $1 \leq i_1, \ldots, i_n \leq n$, $\forall 1 \leq j < k \leq n : i_j \neq i_k$, so that

  $\Gamma(c_{i_1}) \subseteq \Gamma(c_{i_2}) \subseteq \ldots \subseteq \Gamma(c_{i_n})$.

  $\rightarrow$ mass assignment theory [Baldwin et al. 1995]

  **Problem:** The “voting model” is not sufficient to justify consonance.

- Use the lower bound as the “most pessimistic” choice. [Gebhardt 1997]

  **Problem:** Basic possibility assignments represent negative information, the lower bound is actually the **most optimistic** choice.

- Justify the lower bound from decision making purposes. [Borgelt 1995, Borgelt 2000]
• Assume that in the end we have to decide on a single event.
• Each event is described by the values of a set of attributes.
• Then it can be useful to assign to a set of events the degree of possibility of the “most possible” event in the set.

Example:

\[
\begin{array}{cccc}
\sum & 36 & 18 & 18 & 28 \\
28 & 0 & 0 & 0 & 28 & 28 \\
18 & 18 & 0 & 0 & 0 & 18 \\
18 & 18 & 0 & 0 & 0 & 18 \\
36 & 0 & 18 & 18 & 0 & 18 \\
& 18 & 18 & 18 & 28 & \text{max}
\end{array}
\]

\[
\begin{array}{cccc}
& 0 & 40 & 0 \\
& 0 & 0 & 20 \\
& 40 & 40 & 20 & \text{max}
\end{array}
\]
Possibility Distributions

**Definition:** Let $X = \{A_1, \ldots, A_n\}$ be a set of attributes defined on a (finite) sample space $\Omega$ with respective domains $\text{dom}(A_i), i = 1, \ldots, n$. A **possibility distribution** $\pi_X$ over $X$ is the restriction of a possibility measure $\Pi$ on $\Omega$ to the set of all events that can be defined by stating values for all attributes in $X$. That is, $\pi_X = \Pi|_{\mathcal{E}_X}$, where

$$\mathcal{E}_X = \left\{ E \in 2^\Omega \mid \exists a_1 \in \text{dom}(A_1) : \ldots \exists a_n \in \text{dom}(A_n) : E \equiv \bigwedge_{A_j \in X} A_j = a_j \right\}$$

$$= \left\{ E \in 2^\Omega \mid \exists a_1 \in \text{dom}(A_1) : \ldots \exists a_n \in \text{dom}(A_n) : E = \left\{ \omega \in \Omega \mid \bigwedge_{A_j \in X} A_j(\omega) = a_j \right\} \right\}.$$

- Corresponds to the notion of a probability distribution.
- Advantage of this formalization: No index transformation functions are needed for projections, there are just fewer terms in the conjunctions.
**Conditional Possibility and Independence**

**Definition:** Let $\Omega$ be a (finite) sample space, $\Pi$ a possibility measure on $\Omega$, and $E_1, E_2 \subseteq \Omega$ events. Then

$$\Pi(E_1 \mid E_2) = \Pi(E_1 \cap E_2)$$

is called the **conditional possibility** of $E_1$ given $E_2$.

**Definition:** Let $\Omega$ be a (finite) sample space, $\Pi$ a possibility measure on $\Omega$, and $A$, $B$, and $C$ attributes with respective domains $\text{dom}(A)$, $\text{dom}(B)$, and $\text{dom}(C)$. $A$ and $B$ are called **conditionally possibilistically independent** given $C$, written $A \perp_{\Pi} B \mid C$, iff

$$\forall a \in \text{dom}(A) : \forall b \in \text{dom}(B) : \forall c \in \text{dom}(C) :$$

$$\Pi(A = a, C = c \mid B = b) = \min\{\Pi(A = a \mid B = b), \Pi(C = c \mid B = b)\}.$$

- Similar to the corresponding notions of probability theory.
**Decomposition:** Under certain conditions a distribution $\delta$ (e.g. a probability distribution) on a multi-dimensional domain, which encodes *prior* or *generic knowledge* about this domain, can be decomposed into a set $\{\delta_1, \ldots, \delta_s\}$ of (overlapping) distributions on lower-dimensional subspaces.

**Simplified Reasoning:** If such a decomposition is possible, it is sufficient to know the distributions on the subspaces to draw all inferences in the domain under consideration that can be drawn using the original distribution $\delta$.

Since such a decomposition is usually represented as a network and since it is used to draw inferences, it can be called an **inference network**. The edges of the network indicate the paths along which evidence has to be propagated.

Another popular name is **graphical model**, where “graphical” indicates that it is based on a *graph* in the sense of graph theory.
A Simple Example

Example World

Relation

<table>
<thead>
<tr>
<th>color</th>
<th>shape</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>■</td>
<td>○</td>
<td>small</td>
</tr>
<tr>
<td>■</td>
<td>○</td>
<td>medium</td>
</tr>
<tr>
<td>■</td>
<td>○</td>
<td>small</td>
</tr>
<tr>
<td>■</td>
<td>△</td>
<td>medium</td>
</tr>
<tr>
<td>■</td>
<td>△</td>
<td>medium</td>
</tr>
<tr>
<td>■</td>
<td>△</td>
<td>large</td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>medium</td>
</tr>
<tr>
<td>■</td>
<td>△</td>
<td>medium</td>
</tr>
<tr>
<td>■</td>
<td>△</td>
<td>large</td>
</tr>
</tbody>
</table>

- 10 simple geometric objects, 3 attributes
- One object is chosen at random and examined.
- Inferences are drawn about the unobserved attributes.
The Reasoning Space

Relation

<table>
<thead>
<tr>
<th>color</th>
<th>shape</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>■</td>
<td>○</td>
<td>small</td>
</tr>
<tr>
<td>■</td>
<td>○</td>
<td>medium</td>
</tr>
<tr>
<td>■</td>
<td>○</td>
<td>small</td>
</tr>
<tr>
<td>■</td>
<td>○</td>
<td>medium</td>
</tr>
<tr>
<td>■</td>
<td>△</td>
<td>medium</td>
</tr>
<tr>
<td>■</td>
<td>△</td>
<td>large</td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>medium</td>
</tr>
<tr>
<td>□</td>
<td>□</td>
<td>medium</td>
</tr>
<tr>
<td>□</td>
<td>△</td>
<td>medium</td>
</tr>
<tr>
<td>□</td>
<td>△</td>
<td>large</td>
</tr>
</tbody>
</table>

Geometric Interpretation

Each cube represents one tuple.
Reasoning

- Let it be known (e.g. from an observation) that the given object is green. This information considerably reduces the space of possible value combinations.

- From the prior knowledge it follows that the given object must be
  - either a triangle or a square and
  - either medium or large.
Prior Knowledge and Its Projections

Christian Borgelt
Possibilistic Graphical Models and How to Learn Them from Data 15
Intersecting the cylindrical extensions of the projection to the subspace formed by color and shape and of the projection to the subspace formed by shape and size yields the original three-dimensional relation.
Reasoning with Projections

The same result can be obtained using only the projections to the subspaces without reconstructing the original three-dimensional space:

This justifies a network representation:
Is Decomposition Always Possible?
The numbers state the probability of the corresponding value combination.
Reasoning

all numbers in parts per 1000

- Using the information that the given object is green.
Probabilistic Decomposition

- As for relational networks, the three-dimensional probability distribution can be decomposed into projections to subspaces, namely:
  - the marginal distribution on the subspace color × shape and
  - the marginal distribution on the subspace shape × size.

- It can be reconstructed using the following formula:

\[
\forall i, j, k : P(\omega_i^{(color)}, \omega_j^{(shape)}, \omega_k^{(size)}) = P(\omega_i^{(color)}, \omega_j^{(shape)}) \cdot P(\omega_k^{(size)} | \omega_j^{(shape)})
\]

\[
= P(\omega_i^{(color)}, \omega_j^{(shape)}) \cdot \frac{P(\omega_j^{(color)}, \omega_k^{(size)})}{P(\omega_j^{(shape)})}
\]

- This formula expresses the **conditional independence** of the attributes *color* and *size* given the attribute *shape*, since they only hold if

\[
\forall i, j, k : P(\omega_k^{(size)} | \omega_j^{(shape)}) = P(\omega_k^{(size)} | \omega_i^{(color)}, \omega_j^{(shape)})
\]
Again the same result can be obtained using only projections to subspaces (marginal distributions):

This justifies a network representation:
Probabilistic Evidence Propagation, Step 1

\[ P(B = b \mid A = a_{\text{obs}}) \]
\[ = P \left( \bigvee_{a \in \text{dom}(A)} A = a, B = b, \bigvee_{c \in \text{dom}(C)} C = c \mid A = a_{\text{obs}} \right) \]
\[ \overset{(1)}{=} \sum_{a \in \text{dom}(A)} \sum_{c \in \text{dom}(C)} P(A = a, B = b, C = c \mid A = a_{\text{obs}}) \]
\[ \overset{(2)}{=} \sum_{a \in \text{dom}(A)} \sum_{c \in \text{dom}(C)} P(A = a, B = b, C = c) \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)} \]
\[ \overset{(3)}{=} \sum_{a \in \text{dom}(A)} \sum_{c \in \text{dom}(C)} P(A = a, B = b) \cdot P(B = b, C = c) \cdot P(A = a \mid A = a_{\text{obs}}) \]
\[ \quad \cdot \frac{P(A = a)}{P(B = b)} \sum_{c \in \text{dom}(C)} P(C = c \mid B = b) \]
\[ = \sum_{a \in \text{dom}(A)} P(A = a, B = b) \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)} \cdot \sum_{c \in \text{dom}(C)} P(C = c \mid B = b) \]
\[ = \sum_{a \in \text{dom}(A)} P(A = a, B = b) \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)} \cdot 1 \]
A Possibility Distribution

The numbers state the degrees of possibility of the corresponding value combination.

- All numbers in parts per 1000.
Using the information that the given object is green.
As for relational and probabilistic networks, the three-dimensional possibility distribution can be decomposed into projections to subspaces, namely:

- the maximum projection to the subspace color $\times$ shape and
- the maximum projection to the subspace shape $\times$ size.

It can be reconstructed using the following formula:

$$
\forall i, j, k : \pi(\omega_i^{(\text{color})}, \omega_j^{(\text{shape})}, \omega_k^{(\text{size})})
= \min \left\{ \pi(\omega_i^{(\text{color})}, \omega_j^{(\text{shape})}), \pi(\omega_j^{(\text{shape})}, \omega_k^{(\text{size})}) \right\}
= \min \left\{ \max_k \pi(\omega_i^{(\text{color})}, \omega_j^{(\text{shape})}, \omega_k^{(\text{size})}),
\max_i \pi(\omega_i^{(\text{color})}, \omega_j^{(\text{shape})}, \omega_k^{(\text{size})}) \right\}
$$
Reasoning with Projections

Again the same result can be obtained using only projections to subspaces (maximal degrees of possibility):

<table>
<thead>
<tr>
<th>old color</th>
<th>new color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 70</td>
<td></td>
</tr>
<tr>
<td>80 90 70 70</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>old shape</th>
<th>new shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 80 10 70</td>
<td></td>
</tr>
<tr>
<td>30 10 70 60</td>
<td></td>
</tr>
<tr>
<td>80 90 20 10</td>
<td></td>
</tr>
<tr>
<td>80 90 20 10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>old size</th>
<th>new size</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 80 70</td>
<td></td>
</tr>
<tr>
<td>40 70 70</td>
<td></td>
</tr>
</tbody>
</table>

This justifies a network representation:

```
color -> shape -> size
```
Possibilistic Evidence Propagation, Step 1

\[ \pi(B = b \mid A = a_{\text{obs}}) \]

\[ = \pi \left( \bigvee_{a \in \text{dom}(A)} A = a, B = b, \bigvee_{c \in \text{dom}(C')} C = c \mid A = a_{\text{obs}} \right) \]

\[ \overset{(1)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C')} \{ \pi(A = a, B = b, C = c \mid A = a_{\text{obs}}) \} \right\} \]

\[ \overset{(2)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C')} \{ \min \{ \pi(A = a, B = b, C = c), \pi(A = a \mid A = a_{\text{obs}}) \} \} \right\} \]

\[ \overset{(3)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C')} \{ \min \{ \pi(A = a, B = b), \pi(B = b, C = c), \pi(A = a \mid A = a_{\text{obs}}) \} \} \right\} \]

\[ = \max_{a \in \text{dom}(A)} \min \{ \pi(A = a, B = b), \pi(A = a \mid A = a_{\text{obs}}), \max_{c \in \text{dom}(C')} \{ \pi(B = b, C = c) \} \} \]

\[ = \pi(B = b) \geq \pi(A = a, B = b) \]

\[ = \max_{a \in \text{dom}(A)} \min \{ \pi(A = a, B = b), \pi(A = a \mid A = a_{\text{obs}}) \} \]
Graphs and Decompositions

Undirected Graphs

\[ \pi_U(A_1 = a_1, \ldots, A_6 = a_6) = \min \{ \pi_{A_1 A_2 A_3}(A_1 = a_1, A_2 = a_2, A_3 = a_3), \]
\[ \pi_{A_3 A_5 A_6}(A_3 = a_3, A_5 = a_5, A_6 = a_6), \]
\[ \pi_{A_2 A_4}(A_2 = a_2, A_4 = a_4), \]
\[ \pi_{A_4 A_6}(A_4 = a_4, A_6 = a_6) \} \]

Directed Graphs

\[ \pi_U(A_1 = a_1, \ldots, A_7 = a_7) = \min \{ \pi(A_1 = a_1), \pi(A_2 = a_2 | A_1 = a_1), \pi(A_3 = a_3), \]
\[ \pi(A_4 = a_4 | A_1 = a_1, A_2 = a_2), \]
\[ \pi(A_5 = a_5 | A_2 = a_2, A_3 = a_3), \]
\[ \pi(A_6 = a_6 | A_4 = a_4, A_5 = a_5), \]
\[ \pi(A_7 = a_7 | A_5 = a_5) \} \]
Example: Danish Jersey Cattle Blood Type Determination

21 attributes:
1 – dam correct?
2 – sire correct?
3 – stated dam ph.gr. 1
4 – stated dam ph.gr. 2
5 – stated sire ph.gr. 1
6 – stated sire ph.gr. 2
7 – true dam ph.gr. 1
8 – true dam ph.gr. 2
9 – true sire ph.gr. 1
10 – true sire ph.gr. 2
11 – offspring ph.gr. 1
12 – offspring ph.gr. 2
13 – offspring genotype
14 – factor 40
15 – factor 41
16 – factor 42
17 – factor 43
18 – lysis 40
19 – lysis 41
20 – lysis 42
21 – lysis 43

The grey nodes correspond to observable attributes.
Example: Danish Jersey Cattle Blood Type Determination

Moral Graph

Join Tree
Learning Possibilistic Graphical Models from Data

Quantitative or Parameter Learning

- Determine the parameters of the (marginal or conditional) distributions indicated by a given graph from a database of sample cases.
  - Trivial in the relational and the probabilistic case.
  - In the possibilistic case, however, this poses a problem.

Qualitative or Structural Learning

- Find a graph that describes (a good approximation of) a decomposition of the distribution underlying a database of sample cases.
  - Has been a popular area of research in recent years.
  - Several good algorithms exit for the probabilistic case.
  - Most ideas can easily be transferred to the possibilistic case.
Why is Computing Maximum Projections a Problem?

Database: 
\[
\begin{align*}
\{a_1, a_2, a_3\}, \{b_3\} : & \frac{1}{3} \\
\{a_1, a_2\}, \{b_2, b_3\} : & \frac{1}{3} \\
\{a_3, a_4\}, \{b_1\} : & \frac{1}{3}
\end{align*}
\]

There are 3 tuples (contexts), hence the weight of each is \(\frac{1}{3}\).

- Taking the maximum over all tuples containing \(a_1\) to compute \(\pi(A = a_1)\) yields a possibility degree of \(\frac{1}{3}\), but actually it is \(\frac{2}{3}\).

- Taking the sum over all tuples containing \(a_3\) to compute \(\pi(A = a_3)\) yields a possibility degree of \(\frac{2}{3}\), but actually it is \(\frac{1}{3}\).
## Computation via Support and Closure

<table>
<thead>
<tr>
<th>Database</th>
<th>Support</th>
<th>Closure</th>
</tr>
</thead>
<tbody>
<tr>
<td>( {a_1, a_2, a_3}, {b_3} ): (1/3)</td>
<td>(a_1, b_2): (1/3) (a_3, b_1): (1/3) ({a_1, a_2, a_3}, {b_3} ): (1/3)</td>
<td>({a_1, a_2, a_3}, {b_3} ): (1/3)</td>
</tr>
<tr>
<td>({a_1, a_2}, {b_2, b_3} ): (1/3)</td>
<td>(a_1, b_3): (2/3) (a_3, b_3): (1/3) ({a_1, a_2}, {b_2, b_3} ): (1/3)</td>
<td>({a_1, a_2}, {b_2, b_3} ): (1/3)</td>
</tr>
<tr>
<td>({a_3, a_4}, {b_1} ): (1/3)</td>
<td>(a_2, b_2): (1/3) (a_4, b_1): (1/3) ({a_3, a_4}, {b_1} ): (1/3)</td>
<td>({a_3, a_4}, {b_1} ): (1/3)</td>
</tr>
<tr>
<td>({a_2, b_3} ): (2/3)</td>
<td>({a_1, a_2}, {b_3} ): (2/3)</td>
<td>({a_1, a_2}, {b_3} ): (2/3)</td>
</tr>
</tbody>
</table>

3 tuples | 7 tuples | 4 tuples

Taking the maximum over compatible tuples in the support yields the same result as taking the maximum over compatible tuples in the closure [Borgelt and Kruse 1998].
Experimental Results

<table>
<thead>
<tr>
<th>dataset</th>
<th>number of cases</th>
<th>tuples in $R$</th>
<th>tuples in support($R$)</th>
<th>tuples in closure($R$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danish Jersey Cattle</td>
<td>500</td>
<td>283</td>
<td>712818</td>
<td>291</td>
</tr>
<tr>
<td>Soybean Diseases</td>
<td>683</td>
<td>631</td>
<td>n.a.</td>
<td>631</td>
</tr>
<tr>
<td>Congress Voting Data</td>
<td>435</td>
<td>342</td>
<td>98753</td>
<td>400</td>
</tr>
</tbody>
</table>

- The relation $R$ results from the dataset by removing duplicate tuples.
- The frequency information is kept in a counter associated with each tuple.
- None of these databases is a true “imprecise” database, the only imprecision results from unknown values.
- An unknown value for an attribute $A$ is interpreted as the set $\text{dom}(A)$.
- “n.a.” (not available) means that the relation is too large to be computed.
Naive Bayes Classifiers

- Try to compute $P(C = c_i | e) = P(C = c_i | A_1 = a_1, \ldots, A_n = a_n)$.
- Predict the class with the highest conditional probability.

Bayes’ Rule:

$$P(C = c_i | e) = \frac{P(A_1 = a_1, \ldots, A_n = a_n | C = c_i) \cdot P(C = c_i)}{P(A_1 = a_1, \ldots, A_n = a_n)} \quad \leftrightarrow p_0$$

Chain Rule of Probability:

$$P(C = c_i | e) = \frac{P(C = c_i)}{p_0} \cdot \prod_{j=1}^{n} P(A_j = a_j | A_1 = a_1, \ldots, A_{j-1} = a_{j-1}, C = c_i)$$

Conditional Independence Assumptions:

$$P(C = c_i | e) = \frac{P(C = c_i)}{p_0} \cdot \prod_{j=1}^{n} P(A_j = a_j | C = c_i)$$
Star-like Probabilistic Networks

- A naive Bayes classifier is a probabilistic network with a star-like structure.
- Class attribute is the only unconditioned attribute.
- All other attributes are conditioned on the class only.

$$P(C = c_i, e) = P(C = c_i | e) \cdot p_0 = P(C = c_i) \cdot \prod_{j=1}^{n} P(A_j = a_j | C = c_i)$$
A Naive Possibilistic Classifier

- Idea: Possibilistic network with a star-like structure. [Borgelt and Gebhardt 1999].
- Class attribute is the only unconditioned attribute.
- All other attributes are conditioned on the class only.

\[
\pi(C = c_i, \mathbf{e}) = \pi(C = c_i \mid \mathbf{e}) = \min_{j=1}^{n} \pi(A_j = a_j \mid C = c_i)
\]
Naive Possibilistic Classifiers

- Try to compute \( \pi(C = c_i \mid e) = \pi(C = c_i \mid A_1 = a_1, \ldots, A_n = a_n) \).
- Predict the class with the highest conditional degree of possibility.

Analog of Bayes’ Rule:
\[
\pi(C = c_i \mid e) = \pi(A_1 = a_1, \ldots, A_n = a_n \mid C = c_i)
\]

Chain Rule of Possibility:
\[
\pi(C = c_i \mid e) = \min_{j=1}^{n} \pi(A_j = a_j \mid A_1 = a_1, \ldots, A_{j-1} = a_{j-1}, C = c_i)
\]

Conditional Independence Assumptions:
\[
\pi(C = c_i \mid e) = \min_{j=1}^{n} \pi(A_j = a_j \mid C = c_i)
\]
Experimental Results

<table>
<thead>
<tr>
<th>dataset</th>
<th>num. of tuples</th>
<th>possibilistic classifier</th>
<th>naive Bayes classifier</th>
<th>decision tree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>add. att. rem. att.</td>
<td>add. att. rem. att.</td>
<td>unpruned</td>
</tr>
<tr>
<td>audio</td>
<td>train 113</td>
<td>7( 6.2%) 2( 1.8%)</td>
<td>12(10.6%) 16(14.2%)</td>
<td>13(11.5%)</td>
</tr>
<tr>
<td></td>
<td>test 113</td>
<td>33(29.2%) 36(31.9%)</td>
<td>35(31.0%) 31(27.4%)</td>
<td>25(22.1%)</td>
</tr>
<tr>
<td></td>
<td>selected 69</td>
<td>15 21</td>
<td>9 42</td>
<td>14</td>
</tr>
<tr>
<td>bridges</td>
<td>train 54</td>
<td>8(14.8%) 8(14.8%)</td>
<td>10(18.5%) 7(13.0%)</td>
<td>9(16.7%)</td>
</tr>
<tr>
<td></td>
<td>test 54</td>
<td>23(42.6%) 23(42.6%)</td>
<td>24(44.4%) 19(35.2%)</td>
<td>24(44.4%)</td>
</tr>
<tr>
<td></td>
<td>selected 10</td>
<td>6 6</td>
<td>5 8</td>
<td>8</td>
</tr>
<tr>
<td>soybean</td>
<td>train 342</td>
<td>18( 5.3%) 20( 5.9%)</td>
<td>17( 5.0%) 14( 4.1%)</td>
<td>16( 4.7%)</td>
</tr>
<tr>
<td></td>
<td>test 341</td>
<td>59(17.3%) 57(16.7%)</td>
<td>48(14.1%) 45(13.2%)</td>
<td>47(13.8%)</td>
</tr>
<tr>
<td></td>
<td>selected 36</td>
<td>15 17</td>
<td>14 14</td>
<td>19</td>
</tr>
<tr>
<td>vote</td>
<td>train 300</td>
<td>9( 3.0%) 8( 2.7%)</td>
<td>9( 3.0%) 8( 2.7%)</td>
<td>6( 2.0%)</td>
</tr>
<tr>
<td></td>
<td>test 135</td>
<td>11( 8.2%) 10( 7.4%)</td>
<td>11( 8.2%) 8( 5.9%)</td>
<td>11( 8.2%)</td>
</tr>
<tr>
<td></td>
<td>selected 16</td>
<td>2 3</td>
<td>2 4</td>
<td>6</td>
</tr>
</tbody>
</table>

- Possibilistic classifier performs equally well or only slightly worse.
- Datasets are not well suited to show the strengths of a possibilistic approach.
Learning the Structure of Graphical Models

- **Test whether a distribution is decomposable w.r.t. a given graph.**
  
  This is the most direct approach. It is not bound to a graphical representation, but can also be carried out w.r.t. other representations of the set of subspaces to be used to compute the (candidate) decomposition of the given distribution.

- **Find an independence map by conditional independence tests.**
  
  This approach exploits the theorems that connect conditional independence graphs and graphs that represent decompositions. It has the advantage that a single conditional independence test, if it fails, can exclude several candidate graphs.

- **Find a suitable graph by measuring the strength of dependences.**
  
  This is a heuristic, but often highly successful approach, which is based on the frequently valid assumption that in a distribution that is decomposable w.r.t. a graph an attribute is more strongly dependent on adjacent attributes than on attributes that are not directly connected to them.
Learning Graphical Models from Data

1. \( \text{color} \) → \( \text{shape} \) → \( \text{size} \)

2. \( \text{color} \) → \( \text{shape} \) → \( \text{size} \)

3. \( \text{color} \) → \( \text{shape} \) → \( \text{size} \)

4. \( \text{color} \) → \( \text{shape} \) → \( \text{size} \)

5. \( \text{color} \) → \( \text{shape} \) → \( \text{size} \)

6. \( \text{color} \) → \( \text{shape} \) → \( \text{size} \)

7. \( \text{color} \) → \( \text{shape} \) → \( \text{size} \)

8. \( \text{color} \) → \( \text{shape} \) → \( \text{size} \)
**α-Cut View of Possibility Distributions**

**Definition:** Let \( \Pi \) be a possibility measure on a sample space \( \Omega \). The \( \alpha \)-cut of \( \Pi \), written \([\Pi]_\alpha\), is the function

\[
[\Pi]_\alpha : 2^\Omega \to \{0, 1\}, \quad E \mapsto \begin{cases} 
1, & \text{if } \Pi(E) \geq \alpha, \\
0, & \text{otherwise}.
\end{cases}
\]
Evaluating Approximations of Possibility Distributions

The \( \alpha \)-cut view of possibility distributions suggests the following measure for the “closeness” of an approximate decomposition to the original distribution:

\[
\text{diff}(\pi_1, \pi_2) = \int_0^1 \left( \sum_{E \in \mathcal{E}} [\pi_2]_\alpha(E) - \sum_{E \in \mathcal{E}} [\pi_1]_\alpha(E) \right) \, d\alpha,
\]

where \( \pi_1 \) is the original distribution, \( \pi_2 \) is the approximation, and \( \mathcal{E} \) is their domain of definition.

- This measure is zero if the two distributions coincide and it is the larger, the more they differ.
- This measure presupposes that
  \[
  \forall \alpha \in [0, 1] : \forall E \in \mathcal{E} : [\pi_2]_\alpha(E) \geq [\pi_1]_\alpha(E)
  \]
Specificity Divergence

**Definition:** Let $\pi$ be a possibility distribution on a set $\mathcal{E}$ of events. Then

\[
\text{nonspec}(\pi) = \int_0^{\sup_{E \in \mathcal{E}} \pi(E)} \log_2 \left( \sum_{E \in \mathcal{E}} [\pi]_\alpha(E) \right) \, \text{d}\alpha
\]

is called the **nonspecificity** of the possibility distribution $\pi$.

- $U$-uncertainty measure of nonspecificity [Higashi and Klir 1982].
- Generalization of Hartley information [Hartley 1928].

**Definition:** Let $\pi_1$ and $\pi_2$ be two possibility distributions on the same set $\mathcal{E}$ of events with $\forall E \in \mathcal{E} : \pi_2(E) \geq \pi_1(E)$. Then

\[
S_{\text{div}}(\pi_1, \pi_2) = \int_0^{\sup_{E \in \mathcal{E}} \pi_1(E)} \log_2 \left( \sum_{E \in \mathcal{E}} [\pi_2]_\alpha(E) \right) - \log_2 \left( \sum_{E \in \mathcal{E}} [\pi_1]_\alpha(E) \right) \, \text{d}\alpha
\]

is called the **specificity divergence** of $\pi_1$ and $\pi_2$. 
Direct Test for Decomposability (continued)

1. \( A \)
   \( B \quad C \)
   0.102
   72.5

2. \( A \)
   \( B \quad C \)
   0.047
   60.5

3. \( A \)
   \( B \quad C \)
   0.055
   63.2

4. \( A \)
   \( B \quad C \)
   0.076
   66.0

5. \( A \)
   \( B \quad C \)
   0
   54.6

6. \( A \)
   \( B \quad C \)
   0.028
   57.3

7. \( A \)
   \( B \quad C \)
   0.037
   60.4

8. \( A \)
   \( B \quad C \)
   0
   54.6

Upper numbers: Specificity divergence of the original distribution and its approximation.

Lower numbers: Sum of possibility degrees for an example database that induces the possibility distribution.
Transformation of the difference of two possibility distributions:

\[
\text{diff}(\pi_1, \pi_2) = \int_0^1 \left( \sum_{E \in \mathcal{E}} [\pi_2]_\alpha(E) - \sum_{E \in \mathcal{E}} [\pi_1]_\alpha(E) \right) \, d\alpha
\]

\[
= \sum_{E \in \mathcal{E}} \int_0^1 [\pi_2]_\alpha(E) \, d\alpha - \sum_{E \in \mathcal{E}} \int_0^1 [\pi_1]_\alpha(E) \, d\alpha
\]

\[
= \sum_{E \in \mathcal{E}} \pi_2(E) - \sum_{E \in \mathcal{E}} \pi_1(E).
\]

- \(\sum_{E \in \mathcal{E}} \pi_1(E)\) can be neglected, since it is the same for all decompositions.
- Restriction to the sample cases in a given database \(D = (R, w_R)\).
  \((w_R(t)\) is the weight, i.e., the number of occurrences, of a tuple \(t \in R\).)

\[
Q(G) = \sum_{t \in R} w_R(t) \cdot \pi_G(t)
\]
Problem: **Vast Search Space** (huge number of possible graphs)
- \(2^{\binom{n}{2}}\) possible undirected graphs for \(n\) attributes.
- Between \(2^{\binom{n}{2}}\) and \(3^{\binom{n}{2}}\) possible directed acyclic graphs.

Exact formula: 
\[
f(n) = \sum_{i=1}^{n} \sum_{l=0}^{n} (-1)^{i+1} \binom{n}{i} 2^i (n-i) f(n-i). 
\]

**Restriction of the Search Space**
- Fix topological order (for directed graphs)
- Declarative bias (idea from inductive logic programming)

**Heuristic Search Methods**
- Greedy Search
- Simulated Annealing
- Genetic Algorithms
A Simulated Annealing Approach

Definition: Let $G = (U, E)$ be a graph, $M$ the family of node sets that induce the maximal cliques of $G$ and $m = |M|$. $G$ is said to have hyper
tree structure iff all pairs of nodes are connected in $G$ and there is an ordering $M_1, \ldots, M_m$ of the sets in $M$, such that

$$\forall i \in \{2, \ldots, m\} : \exists k \in \{1, \ldots, i - 1\} : M_i \cap \left( \bigcup_{1 \leq j < i} M_j \right) \subseteq M_k.$$

Random construction/modification of a graph with hypertree structure by adding cliques randomly according to the following rules [Borgelt 2000]:

1. $M_i$ must contain at least one pair of nodes that are not connected in the graph represented by $\{M_1, \ldots, M_{i-1}\}$.

2. For each maximal subset $S$ of nodes of $M_i$ that are connected to each other in the graph represented by $\{M_1, \ldots, M_{i-1}\}$ there must be a set $M_k$, $1 \leq k < i$, so that $S \subseteq M_k$. 

Christian Borgelt Possibilistic Graphical Models and How to Learn Them from Data 49
Measuring the Strengths of Marginal Dependences

- **Relational networks**: Find a set of subspaces, for which the intersection of the cylindrical extensions of the projections to these subspaces contains as few additional states as possible.

- The size of the intersection depends on the sizes of the cylindrical extensions, which in turn depend on the sizes of the projections.

- Therefore it is plausible to use the relative number of occurring value combinations to assess the quality of a subspace.

<table>
<thead>
<tr>
<th>subspace</th>
<th>color × shape</th>
<th>shape × size</th>
<th>size × color</th>
</tr>
</thead>
<tbody>
<tr>
<td>possible combinations</td>
<td>12</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>occurring combinations</td>
<td>6</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>relative number</td>
<td>50%</td>
<td>56%</td>
<td>67%</td>
</tr>
</tbody>
</table>

- The relational network can be obtained by interpreting the relative numbers as edge weights and constructing the minimal weight spanning tree.
Measuring the Strengths of Marginal Dependences

Christian Borgelt
Possibilistic Graphical Models and How to Learn Them from Data 51
Hartley Information Gain

**Definition:** Let $A$ and $B$ be two attributes and $R$ a binary possibility measure with $\exists a \in \text{dom}(A) : \exists b \in \text{dom}(B) : R(A = a, B = b) = 1$. Then

$$I_{\text{gain}}^{(\text{Hartley})}(A, B) = \log_2 \left( \sum_{a \in \text{dom}(A)} R(A = a) \right) + \log_2 \left( \sum_{b \in \text{dom}(B)} R(B = b) \right) - \log_2 \left( \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} R(A = a, B = b) \right),$$

is called the **Hartley information gain** of $A$ and $B$ w.r.t. $R$.

Hartley information needed to determine coordinates:

- $\log_2 4 + \log_2 3 = \log_2 12 \approx 3.58$
- $\log_2 6 \approx 2.58$

Gain:

$$\log_2 12 - \log_2 6 = \log_2 2 = 1$$
Specificity Gain

**Definition:** Let $A$ and $B$ be two attributes and $\Pi$ a possibility measure.

\[
S_{\text{gain}}(A, B) = \int_0^{\sup \Pi} \log_2 \left( \sum_{a \in \text{dom}(A)} [\Pi]_\alpha(A = a) \right) \\
+ \log_2 \left( \sum_{b \in \text{dom}(B)} [\Pi]_\alpha(B = b) \right) \\
- \log_2 \left( \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} [\Pi]_\alpha(A = a, B = b) \right) \, d\alpha
\]

is called the **specificity gain** of $A$ and $B$ w.r.t. $\Pi$.

- Generalization of Hartley information gain on the basis of the $\alpha$-cut view of possibility distributions.
- Analogous to Shannon information gain.
Idea of Specificity Gain

- Exploiting again the $\alpha$-cut view of possibility distributions:
  Aggregate the Hartley information gain for the different $\alpha$-cuts.

\[
\begin{align*}
\log_2 1 + \log_2 1 - \log_2 1 &= 0 \\
\log_2 2 + \log_2 2 - \log_2 3 &\approx 0.42 \\
\log_2 3 + \log_2 2 - \log_2 5 &\approx 0.26 \\
\log_2 4 + \log_2 3 - \log_2 8 &\approx 0.58 \\
\log_2 4 + \log_2 3 - \log_2 12 &= 0
\end{align*}
\]
Specificity Gain in the Example

projection to subspace

minimum of marginals

specificity gain

<table>
<thead>
<tr>
<th>s</th>
<th>m</th>
<th>l</th>
</tr>
</thead>
<tbody>
<tr>
<td>△</td>
<td>20</td>
<td>80</td>
</tr>
<tr>
<td>□</td>
<td>40</td>
<td>70</td>
</tr>
<tr>
<td>○</td>
<td>90</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s</th>
<th>m</th>
<th>l</th>
</tr>
</thead>
<tbody>
<tr>
<td>△</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>□</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>○</td>
<td>80</td>
<td>90</td>
</tr>
</tbody>
</table>

0.055 bit

<table>
<thead>
<tr>
<th>s</th>
<th>m</th>
<th>l</th>
</tr>
</thead>
<tbody>
<tr>
<td>△</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>□</td>
<td>80</td>
<td>70</td>
</tr>
<tr>
<td>○</td>
<td>90</td>
<td>70</td>
</tr>
</tbody>
</table>

0.048 bit

<table>
<thead>
<tr>
<th>s</th>
<th>m</th>
<th>l</th>
</tr>
</thead>
<tbody>
<tr>
<td>△</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>□</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>○</td>
<td>80</td>
<td>90</td>
</tr>
</tbody>
</table>

0.027 bit
Evaluation Measures / Scoring Functions

Probabilistic Graphical Models

- Mutual Information / Cross Entropy / Information Gain
- (Symmetric) Information Gain Ratio
- \( \chi^2 \)-Measure
- (Symmetric/Modified) Gini Index
- Bayesian Measures (K2 metric, BDeu metric)
- Measures based on the Minimum Description Length Principle

Possibilistic Graphical Models

- (Symmetric) Specificity Gain Ratio [Borgelt et al. 1996]
- Analog of Mutual Information [Borgelt and Kruse 1997]
- Analog of the \( \chi^2 \)-measure [Borgelt and Kruse 1997]
Two Search Methods

- **Optimum Weight Spanning Tree Construction**
  - Compute an evaluation measure on all possible edges (two-dimensional subspaces).
  - Use the Kruskal algorithm to determine an optimum weight spanning tree.

- **Greedy Parent Selection** (for directed graphs)
  - Define a topological order of the attributes (to restrict the search space).
  - Compute an evaluation measure on all single attribute hyperedges.
  - For each preceding attribute (w.r.t. the topological order):
    - add it as a candidate parent to the hyperedge and compute the evaluation measure again.
  - Greedily select a parent according to the evaluation measure.
  - Repeat the previous two steps until no improvement results from them.
# Experimental Results: Danish Jersey Cattle Data

<table>
<thead>
<tr>
<th>method</th>
<th>type</th>
<th>edges</th>
<th>params.</th>
<th>min.</th>
<th>avg.</th>
<th>max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>independent</td>
<td>0</td>
<td>80</td>
<td>10.064</td>
<td>10.160</td>
<td>11.390</td>
</tr>
<tr>
<td></td>
<td>original</td>
<td>22</td>
<td>308</td>
<td>9.888</td>
<td>9.917</td>
<td>11.318</td>
</tr>
<tr>
<td>o.w.s.t.</td>
<td>$S_{gain}$</td>
<td>20</td>
<td>438</td>
<td>8.878</td>
<td>8.990</td>
<td>10.714</td>
</tr>
<tr>
<td></td>
<td>$S_{sgr1}$</td>
<td>20</td>
<td>442</td>
<td>8.716</td>
<td>8.916</td>
<td>10.680</td>
</tr>
<tr>
<td></td>
<td>$d_{\chi^2}$</td>
<td>20</td>
<td>472</td>
<td>8.662</td>
<td>8.820</td>
<td>10.334</td>
</tr>
<tr>
<td></td>
<td>$d_{mi}$</td>
<td>20</td>
<td>404</td>
<td>8.466</td>
<td>8.598</td>
<td>10.386</td>
</tr>
<tr>
<td>greedy</td>
<td>$S_{gain}$</td>
<td>31</td>
<td>1630</td>
<td>8.524</td>
<td>8.621</td>
<td>10.292</td>
</tr>
<tr>
<td></td>
<td>$S_{gr}$</td>
<td>18</td>
<td>196</td>
<td>9.390</td>
<td>9.553</td>
<td>11.100</td>
</tr>
<tr>
<td></td>
<td>$S_{sgr1}$</td>
<td>28</td>
<td>496</td>
<td>8.946</td>
<td>9.057</td>
<td>10.740</td>
</tr>
<tr>
<td></td>
<td>$d_{\chi^2}$</td>
<td>35</td>
<td>1486</td>
<td>8.154</td>
<td>8.329</td>
<td>10.200</td>
</tr>
<tr>
<td></td>
<td>$d_{mi}$</td>
<td>33</td>
<td>774</td>
<td>8.206</td>
<td>8.344</td>
<td>10.416</td>
</tr>
<tr>
<td>sim. ann.</td>
<td>w/o penalty</td>
<td>22.6</td>
<td>787.2</td>
<td>8.013</td>
<td>8.291</td>
<td>9.981</td>
</tr>
<tr>
<td></td>
<td>with penalty</td>
<td>20.6</td>
<td>419.1</td>
<td>8.211</td>
<td>8.488</td>
<td>10.133</td>
</tr>
</tbody>
</table>
Summary

- Possibilistic networks can be seen as “fuzzyfications” of relational networks.

- Possibilistic networks are analogous to probabilistic networks:
  - probabilistic networks: sum/product decomposition
  - possibilistic networks: maximum/minimum decomposition

- Reasoning in possibilistic networks aims at finding a full description of the actual state of the world.

- Possibilistic networks can be learned from a database of sample cases.

- Quantitative/parameter learning is more difficult for possibilistic networks.

- Qualitative/structure learning is similar for probabilistic/possibilistic networks:
  - heuristic search methods are necessary
  - learning algorithms consist of a search method and an evaluation measure