Due to the unknown mode, there tends to be an exponential number of diagnoses.

Fault Models don’t help.
Due to the unknown mode, there tends to be an exponential number of diagnoses.

Fault Models don’t help.

But “unknown” diagnoses represent a small fraction of the probability density space.

Most of the density space may be approximated by enumerating the few most likely diagnoses.

**Sequential Model-based Diagnosis**

Input:
- Set of component mode variables M, with finite domains.
- Set of observables X, with finite domains.
- Device model \( \Phi \) over M and X, in propositional logic.
- Prior distribution \( P(M_i) \) of mode assignments for each component \( i \).
- Observation sequence \( X_{1,n} = x_{1,n} \) provided dynamically.

Output:
- \( P(M) \) Prior Probability of Failure
- \( P(M \mid X_{1,n} = x_{1,n}) \) Posterior Given Observation updated after each observation is received.

Assume:
- Independence of component mode prior distribution.
- Conditional independence of observations given candidate (Naïve Bayes).
- Uniform distribution of observables, given candidate.
Mode Estimation Example

Inverter(i):
- G(i): Out(i) = not(In(i))
- S1(i): Out(i) = 1
- S0(i): Out(i) = 0
- U(i): • Isolates surprises
  • Explains

Nominal, Fault and Unknown Modes

Candidate (Prior) Initial Probabilities

\[
P(M) = \prod_{i} P(G_i) P(S1_i) P(S0_i) P(U_i)
\]

Assume Independence Of Initial Mode

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(G)</td>
<td>.99</td>
<td>.99</td>
<td>.99</td>
</tr>
<tr>
<td>P(S1)</td>
<td>.008</td>
<td>.008</td>
<td>.001</td>
</tr>
<tr>
<td>P(S0)</td>
<td>.001</td>
<td>.001</td>
<td>.008</td>
</tr>
<tr>
<td>P(U)</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
</tr>
</tbody>
</table>

P(G(A),G(B),G(C)) = .97
P(S1(A),G(B),G(C)) = .008
P(S1(A),G(B),S0(C)) = .00006
P(S1(A),S1(B),S0(C)) = .0000005
Posterior Probability, after Observations $X_{1,n} = x_{1,n}$

$$P(\mathcal{M}|x_1) = \frac{P(\mathcal{M}|\mathcal{P}M)}{P(\mathcal{P})} = \frac{\mathcal{P}(\mathcal{M}|\mathcal{P}M)}{\mathcal{P}(\mathcal{M})}$$

Bayes’ Rule

For $n > 1$:

$$P(\mathcal{M}|x_{1,n}) = 2$$

Observations are conditionally independent

$$= \mathcal{P}(\mathcal{M}|\mathcal{P}M)$$

Estimating the Observation Probability $P(x_i | M)$

Assumption: All consistent observations for $X_i$ are equally likely

$P(x_i | M)$ is estimated using model, $\Phi$, according to:

- **If** previous observations $X_{i,j-1} = x_{i,j-1}$, $M$ and $\Phi$ entails $X_i = x_i$
  - **Then** $P(x_i | M) = 1$

- **If** previous observations $X_{i,j-1} = x_{i,j-1}$, $M$ and $\Phi$ entails $X_i \neq v_i$
  - **Then** $P(x_i | c) = 0$

- **Otherwise**, Assume all consistent assignments to $X_i$ are equally likely observations:
  - let $D_c = \{x_c \in D_{X_i} | c, \Phi \text{ is consistent with } X_i = x_c \}$
  - **Then** $P(x_i | M) = 1/|D_c|$
Observe out = 1:
- \( m = <G(A), G(B), G(C)> \)
- Prior: \( P(m) = .97 \)
- \( P(out = 1 | m) = ? \)
- \( = 1 \)
- \( P(m | out = 0) = ? \)
- \( = 1 \times .97 \times \alpha \)

Observe out = 0:
- \( m = <G(A), G(B), G(C)> \)
- \( P(m) = .97 \)
- \( P(out = 0 | m) = ? \)
- \( = 0 \)
- \( P(m | out = 0) = ? \)
- \( = 0 \times .97 \times \alpha = 0 \)
Example: Tracking Single Faults

- Which are eliminated?
- Which are predict observations?
- Which are agnostic?

Priors for Single Fault Diagnoses:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(S1)</td>
<td>.006</td>
<td>.008</td>
<td>.001</td>
</tr>
<tr>
<td>P(S0)</td>
<td>.001</td>
<td>.001</td>
<td>.008</td>
</tr>
<tr>
<td>P(U)</td>
<td>.001</td>
<td>.001</td>
<td>.001</td>
</tr>
</tbody>
</table>

Leading diagnoses before output observed
Due to the unknown mode, there tends to be an exponential number of diagnoses.

But these diagnoses represent a small fraction of the probability density space.

Most of the density space may be represented by enumerating the few most likely diagnoses.
Optimal CSP

\[\text{OCSP}= \langle Y, g, \text{CSP} \rangle\]
- Decision variables \(Y\) with domain \(D_Y\)
- Utility function \(g(Y): D_Y \rightarrow \mathbb{R}\)
- CSP is over variables \(\langle X, Y \rangle\)

Find Leading arg max \(g(Y)\)
\[Y \in D_Y\]
\(\text{s.t. } \exists X \in D_X \text{s.t. } C(X,Y) \text{ is True}\)

- Encode \(C\) in propositional state logic
- \(g()\) is a multi-attribute utility function that is preferentially independent.

Outline

- Self-Repairing Agents
- Formulating Diagnosis
- Diagnosis from Conflicts
- Single Fault Diagnosis
- Extracting Conflicts
Symptom:
F is observed 0, but should be 1 if A1, A2 and X1 are okay.

Conflict: {A1=G, A2=G, X1=G} is inconsistent.

→ At least A1=U, A2=U or X1=U
Find Symptom Using Unit Propagation while Maintaining Support for Propagation

Extract Conflict by Tracing Support

Symptom: F observed 0 but predicted 1.
Conflict: \{A1=G, A2=G, X1=G\}.
Extract Conflict by Tracing Support

**procedure** `Conflict(C)`

**Input:** an inconsistent clause C.

**Output:** A conflict of C.

```plaintext
for each literal I in C
    union Support-Conflict(I, support(I))
end Conflict
```

**procedure** `Support-Conflict(l, S)`

**Input:** l is a literal and S is the support clause of l.

**Output:** A set of mode assignments supporting l.

```plaintext
If unit-clause?(C)
    If mode-assignment?(literal(C))
        Then {literal(C)}
        Else {} 
    Else for each literal I1 in C, other than l
        Union Support-Conflict(I1, support(I1))
end Support-Conflict
```

**procedure** `Test_Candidate(c, M, obs)`

**Input:** Candidate c, Model M, Observation Obs.

**Output:** Consistent or a conflict.

```plaintext
Assert candidate assignment c;
Propagate obs through model M using unit propagation;
If propagate results in an inconsistent clause
    Return Conflict(c);
Else
    Search for satisfying solution using DPLL;
    If inconsistent
        Return c as a conflict;
    Else
        Return consistent;
End Test_Candidate
```