

Goals

- What is an Expert System?
- What problems are they useful for?
- How are they built? used?
- General Architecture
 - Domain Knowledge
 - Inference Engine
 - Explanation Facility
- Inference Process
 - Reasonable not absolute,
e.g. you are ics 171 students.
 - Forward Chaining
 - Backward Chaining

What is an Expert System

- Expert Systems make decisions about limited domain.
e.g. medical diagnosis, computer configuration, data interpretation.
- Requires human expertise. (no free lunch)
- Specific, explicit knowledge, representing as rules.
if ACA occurs at codon 53 of RTgene then don't give cyclovir.
- Domain knowledge acquired by interviewing experts.
- Cannot not deal with common sense knowledge.
common sense knowledge = knowledge not in encyclopedia.
- Warning: does not always work
e.g. does not work for chess, othello, programming...
- Warning: Expert System is not as good as expert

Why Bother

- Few experts: spread knowledge
- Experts die: record knowledge
- Consortium: combine expertise
more difficult then first thought
- Instruction device
more difficult then first thought
- What can computers do?
trajectory of missiles?
accounting?
mathematics?

Examples

1965	DENDRAL	Stanford	analyze mass spectrometry
1965	MacSyma	MIT	symbolic mathematics
1972*	MYCIN	Stanford	Diagnosis of blood diseases
1975	Cadeceus	U Pitt.	Internal Medicine
1978	Digitalis	MIT	Digitalis therapy
1979	PUFF	Stanford	pulmonary diseases
1980	R1	CMU	Computer configuration
1982	XCON	DEC	Computer configuration
1984	FAITH	JPL	Spacecraft fault diagnosis
1986	ACES	Aerospace	satellite diagnosis
1986	Delta	GE	diagnosis of diesel locomotives
1992	Max	NYNEX	Telephone network troubleshooting

What's been done

- Classification (predict class)
 - Interpretation of data
 - Prediction of future
 - Diagnosis: where is the fault
 - Monitoring: is patient/machine still working?
 - Repair: suggest the fix
- Design
 - configure objects to meet constraints
- Optimization
 - improve design
- Planning
 - decide on actions
- Control
 - e.g. control production in chemical plant
- Instruction
 - diagnose and debug students

MYCIN

- 50-500 rules
- Example Rule:
if the stain of organism is gramneg and
the morphology is rod and
aerobicity is aerobic
then strongly suggestive evidence (.8) that
class of organism is enterocabateriaceae
- Example Data
Organism-1:
gram = gramneg 1.0
morp = rod .8, coccus .2
air = aerobic .6, facul .4
- Restrictive Domain
- Rules match knowledge in Domain
similar to medical literature

Problem Selection for Expert System

- No algorithmic solution
don't use for sorting
- People can do it.
- Cooperative expert is available
- Knowledge must be static
- Knowledge must be expressible
we are all experts at seeing, but...
- No common sense knowledge
- Recurring problem.

Architecture

- Expert
talks to Knowledge Engineering
- Knowledge Engineering
adds/corrects rules
- Knowledge Base
contains facts of case plus rules
- Inference Engine
uses facts and rules to draw conclusions
- User
provides facts and questions to system
- User Interface
natural language and/or graphical
links user to system
- Explanation Facility
explains conclusions and non-conclusions

Rules are Heuristics

Natural Rules

- If temperature above 105, you have a high fever.
- If high fever, go to doctor.
- If animal has large, pointed teeth, then it is a carnivore.
- If carnivore has orange and black stripes, then it is a tiger

Mycin Rules

- If patient has high cerebro-spinal fluid glucose and low blood glucose, then patient has viral meningitis.
- If patient has blood glucose measurement less than 118, the patient has low blood glucose.
- If patient has cerebro-spinal fluid glucose level above 40, then patient has high cerebro-spinal fluid glucose.

Expert System Shells

- Provides mechanism for making inferences, storing and editing facts and rules, interacting with user.
- Does not provide specific rules or facts.
- Why not use logic?
 - No explanation of reasoning.
 - Doesn't interact (ask questions) of user.
 - Doesn't handle contradictory information.
 - Doesn't allow best guesses.
 - Doesn't provide confidence level in conclusion.

Forward Chaining

- Starts with the facts and apply the rules
- Continue until no new inference can be made.
- General useful when you don't have a specific goal and all the facts.

- Three Facts:
 - F1: X gives milk
 - F2: X eats meat
 - F3: X has hoofs
- Three Rules
 - R1: if X gives milk then it is a mammal
 - R2: if X is a mammal and eats meat, then it is a carnivore
 - R3: if X is a carnivore and has hoofs, then it is a ungulate.
- What can we conclude?
- Processing
 - From fact F1 by rule R1, conclude X is a mammal (F4).
 - From facts F2 and F4, by rule R2 conclude X is carnivore (F5).
 - From facts F3 and F5, by rule R3 conclude X is an ungulate (F6).

Backward Chaining

- Start with the Goal(s)
- Apply the rules in backwards fashion
- Ask the user or data base for additional information
- Useful when there are few goals (e.g one of 10 diseases)

Example

- Three Facts:
F1: X gives milk
F2: X eats meat
F3: X has hoofs
- Three Rules
R1: if X gives milk then it is a mammal
R2: if X is a mammal and eats meat, then it is a carnivore
R3: if X is a carnivore and has hoofs, then it is a ungulate.
- Goal: Is X an ungulate?
- Processing
Goal: G1: Is X an ungulate?
G1 Matches conclusion of rule R3.
Set up premises of R3 as subgoals
ie. X is carnivore (G2) and X has hoofs (G3)
G3 matches Fact F3 so it is true.
G2 matches conclusion of rule R2.
Setup up premises of R2 as subgoals.
i.e. X is mammal (G4) and X eats meat (G5)
F2 matches G5 so it is true.
G4 matches conclusion of rule R1.
Set up premises of R1 as subgoals.
i.e. X gives milk (G6).
G6 matches F1.
All goals matches, so X is an ungulate.
- Machines better at bookkeeping then people.
- For medical diagnosis, can direct questions/tests.

Reasoning With Uncertainty

Why not Logical Reasoning

- Reasoning from cause to effects
 - If you type rm^* , you delete all your files.
 - If you have the flu, then your temperature is above 103.
- Want to reason from symptoms to disease
- You may have flu and your temperature is not over 103.
- You don't have all the facts at hand.

We want to make plausible conclusions.

Simple Confidence Calculus

- Different expert systems have used different ad hoc measures for computing confidences.
- Roughly you believe a conclusion is most of the evidence if for it and little is against it.
- Mostly the different approaches yield the same conclusions.
- To each fact, we assign a fact confidence between 0 and 1.
- To each rule, we assign a rule confidence between 0 and 1.
- Confidence Calculus:
 - Confidence of premise of a rule = $\text{minimum}(\text{confidence of each condition})$
 - Confidence of in conclusion of one rule = $(\text{confidence of rule premises}) * (\text{confidence in rule})$.
 - Confidence from several rules r_1, \dots, r_n with same conclusion = $\text{confidence from } r_1 \oplus r_2 \dots \oplus r_n$.
where $x \oplus y \equiv 1 - (1-x) * (1-y)$

Expert System with Confidences

- Build tree of inferences
- Works for forward or backward chaining.
- Similar to expression evaluation, with new rules.

Note new example.

- Three Facts:
F1: X gives milk (.9)
F2: X eats meat (.8)
F3: X has hoofs (.7)
- Three Rules
R1: if X gives milk then it is a mammal (.6)
R2: if X is a mammal and eats meat, then it is a carnivore (.5)
R3: if X has hoofs, then it is a carnivore (.4).
- What can we conclude?
- Processing
From fact F1 by rule R1, conclude X is a mammal (F4).
Confidence in F4 is $.9 * .6 = .54$.
From facts F2 and F4, by rule R2 conclude X is carnivore (F5).
Confidence in F5 from R2 is $\min(.54, .8) * .5 = .27$
From facts F3 by rule R3 conclude X is an carnivore (F5). Confidence in F4
from R3 is $.7 * .4 = .28$
Confidence in F4 from both R3 and R2 is
 $1 - (1 - .28) * (1 - .27) = .48$
- With logic, one proof is enough.
- With plausible reasoning, more reasons give more weight.

Why so many Expert system shells

Expert Systems differ in:

- Complexity of rules
e.g. if for all x, on(x,y) and clear(x), then ...
- Complexity of facts
e.g. for all x, married(X,Y) implies married(Y,X)
- Methods of reasoning
backwards, forwards, both,..
- User interface

C++ Propositional Expert System

```
void main()
{ Atom a("study hard");
  Atom b("do homework");
  Atom c("go to lectures");
  Atom d("get an A");
  Fact f1(a,0.9);
  Fact f2(b,0.5);
  FactList=f1+f2;
  Rule r1(a+b,d,0.8);
  Rule r2(b+c,d,0.7);
  RuleList=r1+r2;
  cout <<"Given Facts:"<<endl<<FactList<<endl;
  cout <<"Given Rules: " <<endl<<RuleList<<endl;
  for (int i=0; i< Length(RuleList) && cycle();i++);
  cout<<"Final Fact List:"<<endl<<FactList<<endl;
}
```