Intelligent Systems

Rule based systems 1
Aims of session

To understand

- Basic principles
- Forward chaining
- Problems with rule-based systems
- Backward chaining
- Expert systems in a bit more detail
- Notes: Based around Johnson and Picton (1995)
During construction

- Taken from Johnson and Picton (1995)
- Knowledge-base: Contains facts and rules
- Inference engine: input information is combined with rules and facts to make decisions and construct new information.
Usually come humans
Process of getting the knowledge in suitable form is knowledge elicitation.
- Often trying capture human expertise
- Not always explicit
During Use

- Taken from Johnson and Picton (1995)
Difference

- Knowledge is kept separate from control structures of programs
- Advantage of knowledge can be added or removed relatively easily
- Should have methods to explain conclusions or reasoning.
Drawbacks

- Tend to be used in narrowly defined applications.
- Need to updated regularly to maintain the validity of the knowledge used.
- Ability to add knowledge as you go along means inconsistencies can arise as can complex sets of rules slowing system down.
Knowledge base

- Facts and rules – domain of a problem
- Proposition or Predicates (if variables included) can be used.
- The part of predicate asserted to be true is the clause.
Deep and Surface Knowledge

- **Deep Knowledge**
  - Basic key principles that are assumed not to change.

- **Surface Knowledge**
  - Known to work from experience but are possibly going to change.
Production rules

- If-Then
- If $T(\text{Cold})$ Then $T(\text{Heating ON})$

What is the problem with this?
Cold needs to be more precisely defined

- $T(\text{Cold}) = \text{TRUE if sensor } < 10^\circ\text{C}$
- $T(\text{Cold}) = \text{FALSE if sensor } \geq 10^\circ\text{C}$

- $T(\text{Heating ON}) = \text{TRUE if heating is switched ON}$
- $T(\text{Heating ON}) = \text{FALSE if heating is switched OFF}$
- If $T(\text{Cold})$ Then $T(\text{Heating ON})$
- The LHS of the rule that contains the condition is the ANTECEDENT.
- The RHS of the rule that contains what happens if the antecedent is TRUE is the CONSEQUENCE.
- This is called modus ponens
  - From previous lecture rules of inference
  - If $(A \rightarrow B)$ is TRUE AND $A$ is TRUE then $B$ is TRUE.
Triggered rules 1

- A rule is triggered if antecedent is TRUE.
  - If a triggered rule goes on to be used it is FIRED.
  - If a triggered rule does not fire it FAILS, due to antecedent being FALSE/UNKNOWN or rule not being selected to fire.
Often more than one rule is triggered at the same time.

Need a strategy for selecting a rule to fire.

Inference engine is in charge of this. Two basic approaches

- Forward chaining
- Backward chaining
In this approach the inference engine works in cycles, updating information:

- Inputted
- Deduced since last cycle

Next all rules whose antecedents are satisfied are triggered

- Conflict set – needs to be resolved.
Security Example (Johnson and Picton 1995, pp222-223)

- **Rule Database**
  1. IF T(image contains a face) AND T(face recognised) THEN T(open door) AND \( \neg T \) (face recognised) AND \( \neg T \) (image contains a face)
  2. IF T(image contains a face) AND \( \neg T \) (face recognised) THEN T(alert security guard) AND \( \neg T \) (image contains face)
  3. IF \( \neg T \) (image contains a face) THEN \( \neg T \) (open door) AND \( \neg T \) (alert security guard)
Fact Database

- $\neg T(\text{image contains a face})$
- $\neg T(\text{face recognised})$
- $\neg T(\text{open door})$
- $\neg T(\text{alert security guard})$
Initially

- Starting at rule 1 inference engine finds it is not triggered and then moves on to 2 which is also not triggered.
- Rule 3 is triggered, so it fires Rules 3 setting
  - \( \neg T(\text{open door}) \) and \( T(\text{alert security guard}) \)
  - These were already set.
- Then goes back to rule 1.
- The system has software that can determine the truth value of $T(\text{face recognised})$ automatically.
- True if the face is matched in an image database.
- The system can also detect if a face is in an image, setting $T(\text{image contains a face})$. 
Scenario 1

- Visitor comes to the door $T$ (image contains a face) is set to TRUE, but face is not recognised $\neg T$ (face recognised).
  - In forward chaining, start with rule 1, first antecedent is true but second is false. Does not trigger.
  - Rule 2 both antecedents true is triggered.
  - Rule 3 antecedent false does not trigger
- Only rule 2 is triggered, so it fires
  - $\neg T$ (alert security guard) becomes $T$ (alert security guard)
  - $T$ (image contains face) becomes $\neg T$ (image contains face)
- System beginnings again at rule 1.
- If person is still visible $T(\text{image contains face})$ and Rule 2 fires again.
- When person is dealt with $\neg T(\text{image contains a face})$ in the fact database, Rules 1 and 2 will not be triggered, but rule 3 is triggered causing the alarm to be turned off and the door will be locked.
If a face is recognised then \( T(\text{image contains a face}) \) and \( T(\text{face is recognised}) \) in the Fact Database.

Now forward chaining is the only rule triggered and will fire change the predicates \( \neg T(\text{open door}) \) to \( T(\text{open door}) \) and predicates \( \neg T(\text{image contains a face}) \) and \( \neg T(\text{face is recognised}) \) are updated.
Conflict resolution strategies

- First-come, first served
- Priority – as above but the rules order as the more important a rule the earlier it is in the list.
- Prioritise data-fact are prioritised, rule so that antecedent fires-problem?
- Recency-least recently used rule fires or rule that uses most recently updated facts
- Generality – the more antecedents that are satisfied the more likely it is to be selected.
Context- split rules in groups and only one group at a time is used.

Buggins’ turn – first rule in the conflict set fires, then rules are examined, but this time starting with the one after the one that has just fired- new conflict set formed. Continues until there are no rules left in the conflict set or the only rule left is the one that just fired.
Example 2 (Johnson and Picton, 1995, pp 225-230)

Rule Database

1. IF (T(room temp<20) AND T(timer ON)) THEN T(Boiler ON)
2. IF (T(water temp<40) AND T(timer ON)) THEN T(Boiler ON)
3. IF T(Boiler ON) THEN T(Pump ON)
4. IF (T(Pump ON) AND T(room temp<20)) THEN T(valve 1 OPEN)
5. IF (T(Pump ON) AND T(water temp<40)) THEN T(valve 2 OPEN)
6. IF \( \neg T(\text{Timer ON}) \) THEN \( \neg T(\text{Boiler ON}) \)
7. IF \( \neg T(\text{room temp}< 20) \) THEN \( \neg T(\text{valve 1 OPEN}) \)
8. IF \( \neg T(\text{water temp}< 40) \) THEN \( \neg T(\text{valve 2 OPEN}) \)
9. IF \( \neg T(\text{Boiler ON}) \) THEN \( \neg T(\text{pump ON}) \)
10. IF \( \neg T(\text{water temp}< 40) \) AND \( \neg T(\text{room temp}< 20) \) THEN \( \neg T(\text{Boiler ON}) \)
- Facts Database
  - Room temp<20
  - Water temp<40
  - Timer ON
  - Valve 1 OPEN
  - Value 2 OPEN
  - Boiler ON
  - Pump ON
Scenario 2

- $T(\text{Room temp}<20)$
- $\neg T(\text{Water temp}<40)$
- $T(\text{Timer ON})$
- $\neg T(\text{Valve 1 OPEN})$
- $\neg T(\text{Valve 2 OPEN})$
- $\neg T(\text{Boiler ON})$
- $\neg T(\text{Pump ON})$
Rules 1, 8, 9 are triggered if we use first-come, first served then

- T(Room temp<20)
- ¬T(Water temp<40)
- T(Timer ON)
- ¬T(Valve 1 OPEN)
- ¬T(Valve 2 OPEN)
- T(Boiler ON)
- ¬T(Pump ON)
Assuming the external condition remain constant. Then if we used first-come, first served rules 1, 3, 8 trigger,
- rule 1 fires again we stuck with this.
If instead we use Buggin’s Turn, and take the first one of the new list of possibles (3,8) we take rule 3

- \( T(\text{Room temp}<20) \)
- \( \neg T(\text{Water temp}<40) \)
- \( T(\text{Timer ON}) \)
- \( \neg T(\text{Valve 1 OPEN}) \)
- \( \neg T(\text{Valve 2 OPEN}) \)
- \( T(\text{Boiler ON}) \)
- \( T(\text{Pump ON}) \)
Backward Chaining

- Goes the other way around to forward starting with the consequent and following the antecedents.
- Goal orientated approach.

Using the example in Artificial Intelligence (Winston 1984)
Rules 10, 5 and 2

- **Rule 10**
  - If $T(\text{animal is a carnivore})$ AND $T(\text{animal has tawny colour})$ AND $T(\text{animal has dark spots})$ THEN $T(\text{animal is a cheetah})$

- **Rule 5**
  - If $T(\text{animal is a mammal})$ AND $T(\text{animal eats meat})$ THEN $T(\text{animal is a carnivore})$

- **Rule 2**
  - If $T(\text{animal gives milk})$ THEN $T(\text{animal is a mammal})$
Is it a cheetah?

- Rule 10 has a consequent predicate that matches what we looking for. So we have our goal.
- We now work backwards and see if the antecedents fit.
  - So lets say that it is a tawny colour animal with dark spots so two of the antecedents are true.
  - The third we need to find a definition of a carnivore (rule 5)
Is it a carnivore?

- Rule 10 left us needing to find out what is a carnivore. Is there a Rule that has a consequent that matches this question.
- Now we look at the antecedents assuming they are true then we have an intermediate goal of it is a carnivore.
- But what does a mammal mean – We need to rule 2
So now the system works backwards starting with does it give milk, eat meat, have tawny colour and darks spots.

So eventually the system finds the necessary facts in its facts database or obtains the information which is then added to the facts database.

So rules 2, then 5 and then 10 would fire.
Not a cheetah

- If the cheetah rule is not fired. The system looks another rule.

- During backward chaining get intermediate facts (e.g. mammal and carnivore) these are useful when the system does not fire.
Example 2 (Johnson and Picton, 1995, pp 225-230)

- Rule Database
  - 1. IF (T(room temp<20) AND T(timer ON)) THEN T(Boiler ON)
  - 2. IF (T(water temp<40) AND T(timer ON)) THEN T(Boiler ON)
  - 3. IF T(Boiler ON) THEN T(Pump ON)
  - 4. IF (T(Pump ON) AND T(room temp<20)) THEN T(valve 1 OPEN)
  - 5. IF (T(Pump ON) AND T(water temp<40)) THEN T(valve 2 OPEN)
6. IF \( \neg T(\text{Timer ON}) \) THEN \( \neg T(\text{Boiler ON}) \)
7. IF \( \neg T(\text{room temp}< 20) \) THEN \( \neg T(\text{valve 1 OPEN}) \)
8. IF \( \neg T(\text{water temp}< 40) \) THEN \( \neg T(\text{valve 2 OPEN}) \)
9. IF \( \neg T(\text{Boiler ON}) \) THEN \( \neg T(\text{pump ON}) \)
10. IF \( (\neg T(\text{water temp}< 40) \) AND \( (\neg T(\text{room temp}< 20)) \) THEN \( \neg T(\text{Boiler ON}) \)
Diagnostic use of backward chaining.

- Room heating is off but room is cold use backward chaining to find the problem (assuming mechanics all working)
- Possibilities
  - \( \neg T(\text{Boiler ON}) \)
  - \( \neg T(\text{Pump ON}) \)
  - \( \neg T(\text{Valve 1 OPEN}) \)
- The known fact
  - \( T(\text{Room temp}<20) \)
- Go through the possibilities
¬ T(Boiler ON)

- Need to find a rule that has this as a consequent. We have two

- Rule 6:
  - IF ¬T(Timer ON) THEN ¬T(Boiler ON)

- Rule 10:
  - IF (¬T(water temp< 40) AND (¬T(room temp< 20))) THEN ¬T(Boiler ON)
Rule 10

- The known fact is:
  - T(Room temp<20)

- Rule 10
  - IF (¬T(water temp< 40) AND (¬T(room temp< 20))) THEN ¬T(Boiler ON)

- So Rule 10 can not fire.
Rule 6

- IF \( \neg T(\text{Timer ON}) \) THEN \( \neg T(\text{Boiler ON}) \)
- We don’t what the state of the timer is so we need to look for a rule to find if the timer is on or not. No rule has \( \neg T(\text{Timer ON}) \) as a consequence.
- So we can not go any further and we can have \( \neg T(\text{Timer ON}) \) has a possible reason.
¬ T(Pump ON)

- Trying out the second possibility.
- Rule 9: IF ¬T(Boiler ON) THEN ¬T(pump ON)
  - So we need to check if boiler is not on.
  - We just done this so again a possible reason is ¬T(Timer ON)
¬ T(Valve 1 OPEN)

- Trying the third possibility.
- Rule 7
- IF ¬T(room temp< 20) THEN ¬T(valve 1 OPEN)
- So the antecedent is FALSE so this not a possibility.
Diagnostic conclusion

- One conclusion is the timer is off.
Knowledge-base: Contains facts and rules

Inference engine: input information is combined with rules and facts to make decisions and construct new information.
References