Week 6: Chapter 9

Using Past History Explicitly as Knowledge: Case-based Reasoning Systems

Chapter Objectives

- Introduce the student to the concept of using explicit historical occurrences to solve current problems.
  - Explained in the context of rule-based systems that also use past experience to solve current problems
- Introduce case-based reasoning.
- Introduce how case-based systems can learn from their own experience

Weaknesses of rule-based systems

- **Experts** may not be able to identify specifically when they learned a particular bit of knowledge that permits them to solve the current problem.
- **Experts** may not be able to externalize their experience into clean bits of knowledge that can be encoded into rules
  - Their knowledge is an accumulation and a combination of years of being exposed to many instances of similar problems (and their subsequent solutions)

Weaknesses of rule-based systems

- To manage the knowledge of experts, we must:
  - Elicit it from the expert
  - Represent or formalize it in a form suitable for computing
  - Validate and verify the knowledge
- All these contain pitfalls for the rule-based systems approach

Q. Why may rule-based systems have difficulty with eliciting, representing, and validating knowledge from the expert?

A1. Results vary depending on which expert

- It is the experts’ personal interpretation of the domain
- Some experts are very knowledgeable, others only minimally so
- Different experts may see the same domain from different perspectives –in fact, they may all be correct in some way

Q. Why may rule-based systems have difficulty with eliciting, representing, and validating knowledge from the expert?

A2. Transferring the codified knowledge can be difficult and error-prone

- Experts can provide erroneous knowledge if the KE’s question is ill-posed
- The KE can misinterpret an expert’s correct answer
- Developers can misrepresent correct knowledge in the system code (or rules)

Q. Why may rule-based systems have difficulty with eliciting, representing, and validating knowledge from the expert?

A3. Too many rules may be needed to properly represent one domain.

- Eg: A medium rule-based system can include up to 10,000 rules when initially deployed
- Disadvantage 1: The rules have to be coded, verified, validated, and maintained
  - Complexity of validation and maintenance can grow exponentially with the number of rules, due to rule interaction!
- Disadvantage 2: They have to be executed by the inference engine
  - Computational cost can become infeasible

Case-Based Reasoning (CBR)

CBR is an alternative to rule-based systems…

- Keep all the concrete cases that might have led to the learning by the experts
- Stick to recording the concrete details of each case, without generalizing experience into rules
- Avoid the personal influence of individual experts

Case-Based Reasoning (CBR)

CBR is an alternative to rule-based systems…

- Bypass the expert and look directly at the information that allowed them to learn and acquire their expertise
- We no longer need the experts’ interpretation as we are relying on the same source from which the experts learn, and thus avoid the associated drawbacks (One approach to avoid the problem of knowledge acquisition and maintenance)

Case-Based Reasoning (CBR) Main concepts

- The CBR technique originates from Schank’s [1982] concept of **remindings**:
  - When people are thinking (eg, solving problems), they are merely recalling past experiences that somehow remind them of the current situation
  - If the current and historical situations are sufficiently similar, then it can be inferred that the solutions to both situations are the same
  - I.e., people apply solutions of past problems to current problems that are similar in nature

Case-Based Reasoning (CBR) Examples

- Example 1 [Klein 1985]
    - Fire ground commander coordinating his crew, while fighting a **fire** at a low-rise apartment building
        - *Low-rise building: a low-rise building does not have many storeys (floors or levels)*
    - Notices billboards on the building’s roof
        - *Billboard: a large sign used for advertising*
    - Recalls earlier incident where flames burned through the wooden billboard supports, causing them to crash to the street below
    - Orders that spectators be moved farther back to prevent injury from falling billboards
        - *Spectator: someone who is watching an event or game*
- Note: Highly unlikely that such a rule would have already been included in a KBS
    - Unless spectators had already previously been injured by falling billboards –but that would be too late!

Case-Based Reasoning (CBR) Examples

- Example 2 [Klein 1985]
  - Fire ground commander notices some peculiar properties in a cloud of smoke at a fire
  - Recalls an incident in which toxic smoke had been given off showing the same features of density, color, and heaviness
  - Orders his crew to use breathing support systems
- Note: Highly unlikely that such a rule would have already been included in a KBS
  - Unless fire crew had already previously been injured by this specific kind of toxic smoke –but that would be too late!

Case-Based Reasoning (CBR)

- ***CBR*** is reasoning from relevant past cases in a manner similar to humans' use of past experiences to arrive at conclusions or solve new problems
- Goal is to bring up the most similar *historical cases* (experiences) that match the current (new) case
- Assumes that
  - problems recur and
  - similar problems have similar solutions
- Provides a natural (though simplistic) form of learning by merely adding any newly solved problem to its database of past cases

Group Discussion

- In 20 minutes, For each group:
  - Discuss a situation that you faced which needs to recall past experiences that somehow remind you of the current situation. The current and historical situations are sufficiently similar such tha it can be inferred that the solutions to both situations are the same. Show how you could apply solutions of the past problems to current problems that are similar in nature.
- In 10 minutes, write your situations and solution.
- Each group will introduce themselves and pitch their ideas in 10 minutes.

Case-Based Reasoning (CBR)

- Simplest, most basic form of CBR:
  - A repository of historical cases called the **case library**
  - A means to find and retrieve a similar case from the case library, and use its solution to solve the current problem
  - A means to add the newly solved problem and solution to the case library as a new case

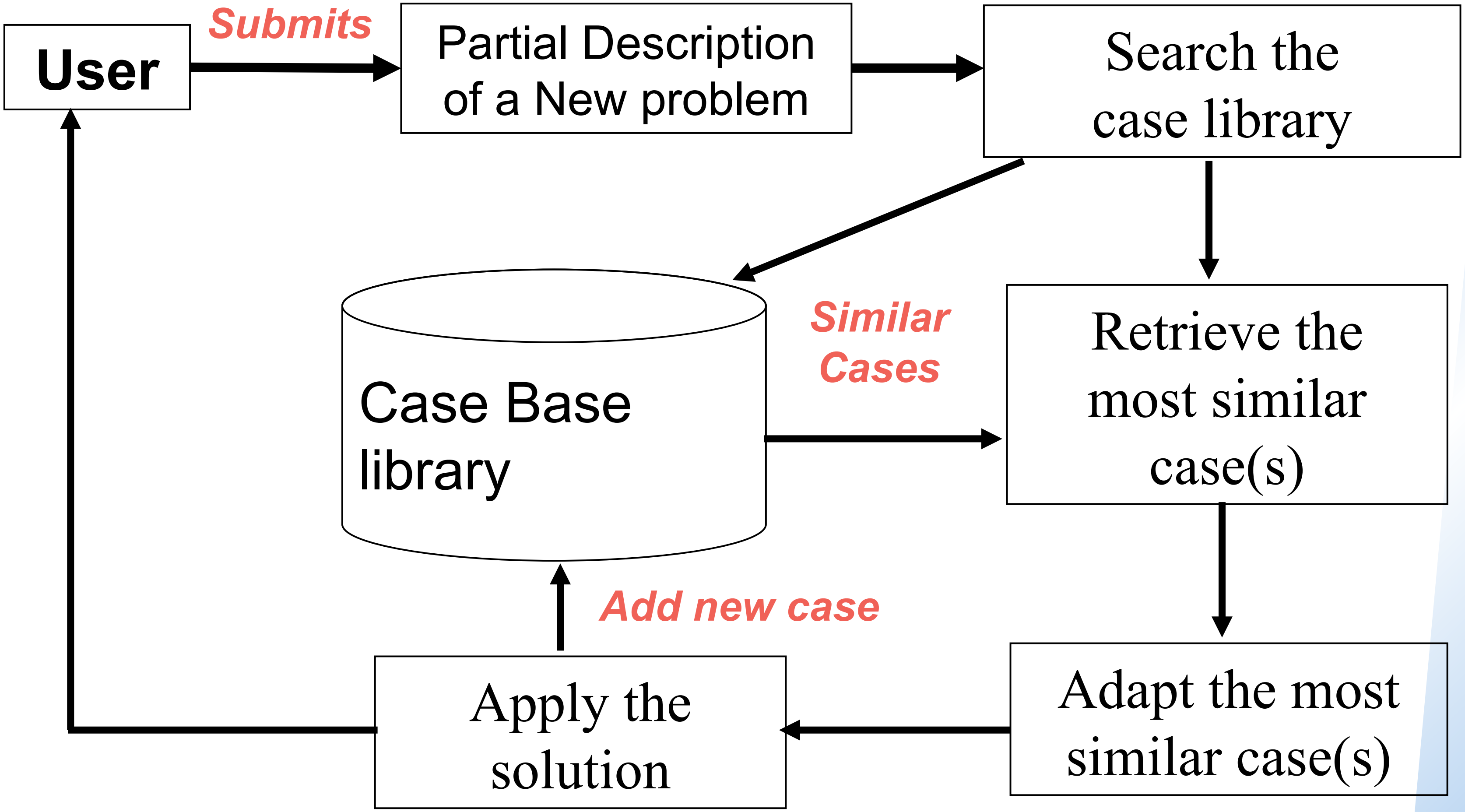
Case-Based Reasoning (CBR): Adaptation

- But: What happens when the most similar case is not judged similar enough to the current problem?
- In such circumstances, the solution(s) of the most similar case(s) should be **adapted** to the current problem.

Case-Based Reasoning (CBR): Adaptation

- Automatic adaptation is a very difficult problem technically.
- Except in highly limited formalisms, there is little solid mathematical theory to support practical adaptation methods.
- In many systems, adaptation has been abandoned altogether.
  - Focus instead on improving searching and learning.

Generic CBR Process


User
Submits
Partial Description of a New problem
Search the case library
Case Base library
Similar Cases
Retrieve the most similar case(s)
Adapt the most similar case(s)
Add new case
Apply the solution

Real-world CBR Process (Requirements)

- Search the case library
  - Requires efficient indexing of the cases
- Retrieve the most similar case(s)
  - Requires quantitative similarity metrics
- Adapt the most similar case(s) if not suitably similar
  - Technically difficult, and optional
  - Sometimes impossible, eg, pre-filmed video clips
- Apply the solution to the current problem
  - Capture whether it succeeded or failed, as feedback
- Add the last case to the case library
  - Requires criteria to decide whether it is worth adding
  - Requires efficient updating of the indexing

Indexing and case library organization

- Indexing is the labeling of data items in such a way that they are easily retrieved.
- Examples:
  - Library card catalogs (usually only by title or author)
  - Library numbering system / physical organization (usually by subject)
    - Dewey Decimal System
    - Library of Congress System
  - Library electronic catalogs (more powerful searching over a wider range of attributes)
  - Library full text search engines
  - Internet full text search engines

Indexing and case library organization

- Issue: What attributes of cases are indexed?
- For real-world cases, there are typically a huge range of attributes you could imagine to index each case by.
- If attributes are defined too broadly, an unnecessarily large number of cases may be retrieved for examination.
- If attributes are defined too narrowly, there may be some truly similar cases that are overlooked during retrieval.

Indexing the case library

- Issue: What is the best way to organize a library of cases?
- The effect of retrieving an improperly matched case is often more computationally expensive than selecting the wrong rule to execute in a rule-based system.
- Inefficient searches due to poor organization can result in unacceptable performance.

Means of increasing search efficiency through indexing

- Flat library
- Shared feature networks
- Redundant shared feature networks

Flat case library

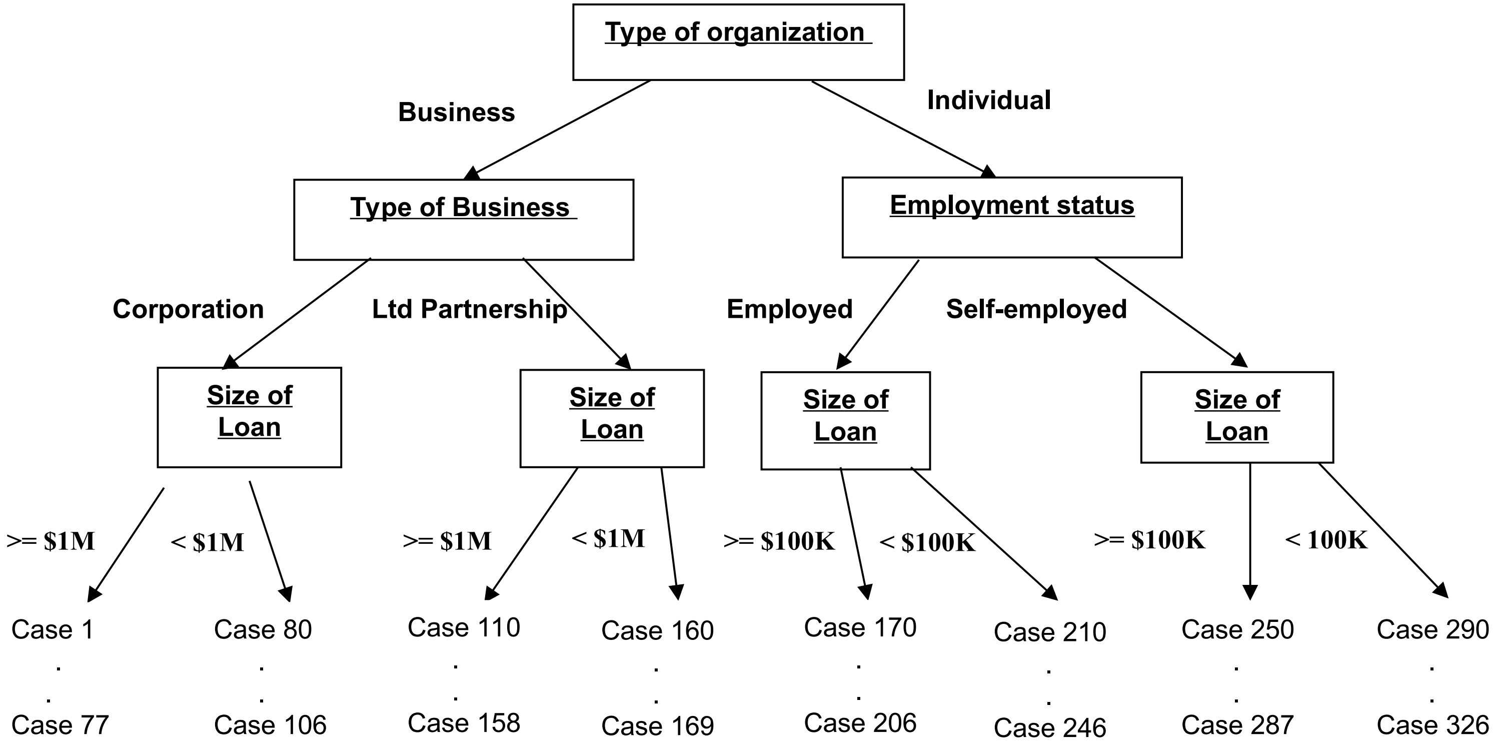
- **Flat case libraries** are the simplest
- Cases can be placed in a list, array, or file
- Some ordering may be imposed
- May be too inefficient for larger case libraries with complex cases; additional strategies:
  - Partitioning the library (eg, individual vs corporate)

Individual:	Case 1, Case 2, … Case 200
corporate :	Case 201, Case 202, … Case 326

Shared feature networks

- Uses decision tree clustering
- Hierarchical organizations that segregate cases by what features they have in common
- Cluster cases as much as possible –given any node, segregate its cases by choosing the feature most universally shared
- Search process simply follows path through tree, matching features of the current problem

Shared feature networks for a loan application example



Type of organization

Business

Individual

Type of Business

Employment status

Corporation

Ltd Partnership

Employed

Self-employed

Size of Loan

Size of Loan

Size of Loan

Size of Loan

>= $1M

< $1M

>= $1M

< $1M

>= $100K

< $100K

>= $100K

< 100K

Case 1 . . Case 77

Case 80 . . Case 106

Case 110 . . Case 158

Case 160 . . Case 169

Case 170 . . Case 206

Case 210 . . Case 246

Case 250 . . Case 287

Case 290 . . Case 326

Redundant shared feature networks

- Problem: Often there are some unknown attributes in any new case. Can't find path through tree!
- **Redundant shared feature networks** attempt to overcome this by maintaining a number of different trees, each of which prioritizes different attributes.
- Choose the tree that gets you the longest path, i.e., closest to the leaf level, so that the roadblock comes in at the latest possible stage in the search.
  - Tries to give the narrowest subset of similar cases.

Matching and retrieval

- Retrieval is returning the most similar cases from the case library for further processing
- A **distance metric** is used to compute the distance between historical cases and current problem
- A distance metric is a function that aggregates variation over a large number of attributes
- There are always many possible aggregation functions, for example:
    - Weighted sums (weighted averages)
    - Higher-order polynomials (regression based measures)
    - Hamming distances, and more sophisticated edit distances
    - Cosine based measures
    - Ad hoc measures
    - … etc.
- which distance metric is best depends heavily on the domain

Evaluation

- How to determine whether the most similar case is similar enough?
  - Determining the adequacy of the retrieved historical solution for the current problem
- May involve implementing the solution within a simulator or in real life under test conditions
- Not always possible!

Adaptation

- The difference between the similar case and current problem may invalidate the historical solution
- The historical solution may be modified (adapted) to make it valid.
- Adaptation is difficult or not even possible. So, developer have replaced search-retrieve-adapt with **search-retrieve-propose** where the most similar case is presented to a human user for consideration.

Adaptation techniques

How to modify the most similar case when it is not sufficiently similar to the current problem?

- **Reinstantiation:** instantiates an old solution with new objects that are relevant to the solution. eg, replace beef with chicken
- **Parameter adjustment:** adjust parameters in the historical solution using heuristics. eg, scaling income/credit
- **Search:** searches a knowledge structure (frame, semantic net) or a problem space for a substitute for the part of the historical solution that causes invalidity. eg, find exact location of hose leak.

Adaptation techniques

- **Case-based substitution:** for a solution that represent several independent subprocesses or values, only replace the subprocess or value that made the solution invalid
- **Transformation:** heuristics is used to build the replacement of the subprocess or value
- **Model-guided repair:** assumes a causal-model exists and may be used to repair the historic solution.

Learning

- Learning is done simply by adding new cases to the knowledge base.
- new cases should be consistent with the rest of the case library

Learning

- Adding cases progressively covers more and more of the problem space.
    - The more cases, the better the coverage of the problem domain.
- Many cases overlap.
    - Not a problem, as long as their solutions are consistent.

Learning

Issues:

- Too many cases:
  - May clog the search and retrieval process unnecessarily
- Too little diversity among the cases (even if there are many cases):
  - May leave significant gaps in coverage

Example: CBR applied to property appraisal [Gonzalez 1992]

- Use the market data method of property appraisal
  - Based on the selling price of similar properties in the market.
  - *Appraisal: a statement or opinion judging the worth, value, or condition of something*
- Features (attributes):
  - Living area in square feet
  - Number of bedrooms
  - Number of bathrooms
  - Architectural style of the house
  - Age of the house
  - Location (neighborhood)
  - Date of sale
  - Type of cooling equipment
  - Type of heating equipment
  - Type of garage
  - Site or lot size
  - Availability of swimming pool

Example: CBR applied to property appraisal [Gonzalez 1992]

- Case retrieval:
  - Retrieves 10 best cases
  - Ranks in order of decreasing similarity
- Case adaptation
  - Uses critics: heuristic rules that increase or decrease the actual sold price of a retrieved property based on differences between it and the property being appraised
    - eg, a swimming pool critic
  - Adaptation is cumulative: done for all features of the comparison
- Case evaluation
  - Too many adaptations can result in inaccuracy
  - A comfort fact or indicates which of the 10 cases was least extensively adapted
  - Top three are selected (traditional in appraisal business)
- Returns the average of the top three

Some issues in case-based systems

- It can take a much larger number of cases than rules to cover a domain to the same extent.
- Whether the assumption that similar problems have similar solutions really holds depends on the domain.
- The similarity metric depends on the domain and can greatly affect CBR systems (and sometimes there may not even be any good similarity metric).
- Adaptation is highly domain dependent (and in extreme cases may be equally or more complex than a rule-based system).

Advantages of case-based systems

1. The knowledge acquisition process is considerable simplified in many applications, especially where the case library may already exist as corporate documentation, possible even in electronic DB.
2. The knowledge maintenance process is greatly facilitated by the learning ability of CBR
3. CBR is modeled after human reasoning. There is significant evidence to believe that CBR is a cognitive problem-solving model
4. CBR performs better than RBS in so-called weak-theory-domains where experts may not exist or if they exist they do not fully understand the intricacies of the domain
5. The base of experience used be that of an entire organization, instead of that of a single individual. This can multiply the breadth of the KB in CBR

Disadvantages of case-based systems

1. Large systems can be inefficient:
   - poorly organized case libraries.
   - Cost of the matching process
2. Difficulties in distance calculations between desired and actual solution
3. Adaptation may be quiet difficult or impossible in many domains
4. Learning should carefully handle inconsistency
5. Building a case library may be as difficult as knowledge acquisition for RBS (e.g. poorly documented cases or no cases exist)

Discussion

- List five general features common to applications that would be suited for solution through case-based reasoning.

Discussion – possible answer

- Historical cases must exist explicitly and be clearly documented. The documentation should preferably be in electronic form.
- Information on the historical cases should be complete and understood by human experts.
- Historical test cases should be repeatable. In other words, the problems solved in the past are similar to problems currently faced. Problems solved that relate to obsolete problems are not useful.
- A “distance metric” that can be used to measure the similarity between current and historical problems clearly and quantifiably must be easily identified.
- If possible, the cases should be adaptable to current problems. This, however, is not an absolute requirement.

Conclusions

- The student should be familiar with:
  - The difference between how rule-based systems and case-based systems use historical knowledge.
  - The main processes of case-based reasoning:
    - Search
    - Select
    - Adapt
    - Apply
    - Learn
  - The advantages and disadvantages of case-based systems

Chapter 9

Using Past History Explicitly as Knowledge: Case-based Reasoning Systems