Making a Case for Case-based Reasoning

Qiang Yang
Simon Fraser University
http://www.cs.sfu.ca/~qyang
The Setting

- The mission: NSERC industry chair program
- The problem
- Elevator test
- The methodology
  - Application domain
  - Research problem
The Problem

- Rogers cable-TV has hundreds of customer service representatives (CSR’s) who solve customers’ cable-TV and internet problems on the phone (call center).
- If a problem cannot be solved, Rogers must send out a truck to customer’s site --> truck roll.
- Truck rolls, and training, are expensive!
Problem Resolution Example

- **Customer:** “my VCR is not working”
- **CSR** “do you have a recording problem”
- **Customer:** “yes. I cannot record channel 13”
- **CSR:** “first, turn your TV to channel 3. Now tell me what you see on your TV screen”
- **Customer:** “I see the music channel”
- **CSR:** “OK, now change to channel 13 through the remote..., finally, unplug and then plug the TV”
- **Customer:** “OK, problem solved”
Domain Problem

- **Problem**: cache and re-use the knowledge through small and focused databases and interactive retrieval
- **Requirements**: no formal domain model, knowledge change at fast rate, knowledge highly typical
- **Solution**: case based reasoning
Case Representation

- **Case name:** VCR not taping required channels
- **Description:** most likely, VCR hookup problems
  - Questions: “Does direct hookup of VCR help solve the prob?”
- **Solution:**
  1. Check that account is enabled for required channels
  2. Check that sub has required equipment, and is following correct recording procedures
  3. If problem continues, advise that the VCR is faulty and should be examined
- **Multimedia attachment**
Case Based Reasoning Cycle

- Create
- Maintain
- Retrieve
- Revise??
System Demo

CaseAdvisor is available at
http://www.cs.sfu.ca/~isa/isaresearch.html#systems
Problem 1: Unstructured Cases

Much of knowledge is stored in flat files (Text, Html, Etc)
Semi-structured Cases

- In help desk applications, knowledge is distributed among different data sources
  - User manuals
  - Database records
  - HTML files
- Cases are in semi-structured format:
  `<attributes, problem, solution, links...>`
- Changes are often incremental
Two Types of Cases

- **Structured Cases**
  - **Case Id:** 10056
  - **Make:** Honda
  - **Model:** Civic
  - **Year:** 1997
  - **Price:** $17 000
  - **Number of Doors:** 2
  - **Engine Location:** Rear
  - **Engine Size:** 420EL
  - **Problem:** Engine stalling
  - **Validation:** Condition of fuel injector.
  - **Solution:** Clean fuel injector.

- **Unstructured Cases**
  - **Case Name:** Income Funds
  - **Case Solution:** Income funds can be considered a core holding for almost all mutual fund investors. These mutual funds provide investors with a regular stream of income, plus the potential for long-term growth. These are also known as “fixed income” funds. They include government bonds, corporate bonds and mortgages. The funds can also hold very short-term securities known as money market instruments. Because bonds pay interest, value tied to interest rates.
Information Retrieval

- *Task*: detect cases that are similar in content
- *Information Retrieval (IR):*
  - remove stop words
  - stem remaining terms
  - collapse terms using thesaurus
  - build inverted index
  - extract key words - build key word index
  - extract key phrases - build key phrase index

![Diagram](Case base ➔ Keyword Extraction ➔ Redundancy Detection)
Keyword and Feature Classification

- Case Notation \((P, Q, S\) are sets of keywords)\n  - Problem Descriptions: \(P\)
  - Solution Qualifications: \(Q\)
  - Solutions: \(S\)

- Case\(<P, Q, S>\) means
  \(\text{given}(Q)\ and\ \text{do}(S)\ \Rightarrow\ \text{solved}(P)\)
Subsumption Rules

- Case 1 *subsumes* Case 2 if
  - Rule: $P_1 \geq P_2$, $Q_1 \leq Q_2$, $S_1 \leq S_2$
  - Case 1 can solve all problems that Case 2 solves
  - Case 1 requires fewer preconditions and is more efficient

- Removing Case 2 does not affect the coverage of the case base!
Subsumption Example

• Case 1
  **Problem:** fever, headache
  **Qualification:** adult
  **Solution:** take 2 Tylenol

• Case 2
  **Problem:** fever
  **Qualification:** adult
  **Solution:** take 2 Tylenol, 2 aspirin

► Case 1 subsumes case 2
- Case 2 may be redundant, a candidate for removal
Empirical Testing
CaseAdvisor Redundancy Detection Module

- 210 cases generated from cable-TV domain
- 5 separate authors
Problem 1: Unstructured Cases

With Kersti Racine, MSc.

- ICCBR’97
- IEEE TKDE 2001
Problem 2: Case-base Coverage Problem

Lots of cases are repetitive, small variations of one anther
Maintenance Policies

- Given:
  - a large data base $Z$ of (prob,sol) pairs
  - a constant $K$, the final size of a case base
  - a similarity metric defined by adaptation costs.
  - a frequency of problem occurrences
- Find a case base of size $K$ with good competence
- Optimal solution is NP-complete
- Want: good approximate algorithm
Coverage of Cases

- Coverage(case) =\{case'| Adaptable(case,case')\}
- Cases are classified into several classes:
  - **Pivotal**: not contained in the coverage of any other cases in the case base
  - **Auxiliary**: its coverage is contained in the coverage of some other case in the case base

![Diagram showing relationships between cases a, b, c, and z.]

2003/8/8  Problem 2: Coverage  20
Smyth and Keane’s Case Deletion Policy (IJCAI-95)

- Deletion Policy:
  - Delete auxiliary cases first
  - Delete support and spanning cases
  - Delete pivotal cases

Until case base size is K (user defined size).

However, deletion-based policy can lose almost all coverage
(set K=1, case-base={Z} coverage=1/(n+1)
Our Case-Addition Policy

1. Find the coverage $N(x)$ of every problem $x$ in database $Z$; case base $X=\{}$;
3. Select a case from $Z-X$ with the **maximal benefit** with respect to $N(X)$ and add it to $X$
4. Repeat step 3 until $N(Z)-N(X)$ is empty or $X$ has $K$ elements
Case-Addition Policy
Competence Preserving Claim

- **Theorem:** The case-addition policy produces a case base $X$ such that the coverage of $X$ is no less than 63% of the coverage of an optimal case base

- Proof based on set-covering, also similar to one given by [Harinarayan, Rajaraman and Ullman 96] for data cube construction
How many cases are enough?

- Let the size of database be $n$; size of case base be $k$;
- Let $r = k/n$ be the ratio;
- Suppose when adding cases into a case, the benefits decreases *linearly*;
- Then:
  \[ \text{coverage} = r(2-r) \]
How to compute case-coverage?

- Count the number of adaptation steps needed,
- State-based similarity metric for path planning:
  \[ \text{Dist}(x, y) = \min \text{ # of steps added/deleted from } x \text{ to } y \]
Problem 2: Case-base Coverage Problem

Jun Zhu, MSc.

- IJCAI '99
- Computational Intelligence Journal
Experts pay attention to some problem features more than others
Maintaining Indexes

- Weights to question-answers set by domain expert may be inaccurate, change over time

- Adjust weights to refine case associations based on usage patterns
  - close the feedback loop

- Different type of users have different preferences, usage behavior
  - agents vs. customers visiting web site
Architectural Changes

- Two layer case base
  - Feature-value layer
  - Problem-Solution Layer
  - Weights

- Three Layer case base
  - Feature-value layer
  - Problem Context/Types
  - Solution Layer
A Video Rental Domain Ex

- Actor = A1
- Director = D2
- Music = M12

- Comedy
  - Independence Day
- Science Fic
  - Star Trek
- Action
  - Titanic
Problem Resolution and Learning

Case List
Prob1
Prob2
Prob3
Prob4
Prob5

Confirm Problem
Prob. Desc:
Confirm
Disappr.
Cancel

Confirm Solution
Possible Sol’n:
Sol’n 1:
Sol’n 2:
...........
Confirm
Disappr.
Cancel

Web browser
Case Name:
Problem:
Solution:
...........
.............

Neural Network Learning algorithm

Problem 3: Feature Weight Learning
Step 1:
\[
S_j = \frac{1}{1 + e^{-\sum w_{2ij}p_i}} \\
P_j = \frac{1}{1 + e^{-\sum w_{1ij}(Q,A)_i}}
\]

Step 2:
\[
\delta_{2j} = S_j(1 - S_j)(y_j - S_j) \\
\delta_{1j} = P_j(1 - P_j) \sum_i \delta_{2i} \cdot w_{2ji}
\]

Step 3:
\[
\Delta w_{2ij} = \eta \cdot \delta_{2j} \cdot P_i \\
\Delta w_{1ij} = \eta \cdot \delta_{1j} \cdot (Q,A)_i
\]

\(y_j\): the target output
Test the Index Learning Module:

- Rogers Cable-TV Case Base (30 Q/A)
- Video Rental Case Base (25 Q/A)
- UCI Data
Test Results

Error Convergence Chart for 150 Highest Cases in 150 Queries

Problem 3: Feature Weight Learning
Training time: quadratic with CB-size

![Graph showing average running time for training solutions of individual cases.](image)

- **Problem 3: Feature Weight Learning**

2003/8/8
Problem 3: Feature Weight Learning

Zhong Zhang, Msc.

- IJCAI ’99
- International Journal of Information Systems, Kluwer
Problem 4: Interactive Retrieval

In case-retrieval, experts usually ask a small number of key questions to find problems.
Retrieval Issues:
- **Given** a set of candidate clusters that may share attributes
- **Find**: A small set of attributes that can distinguish the clusters
- **Problem**: similar to decision-tree construction
Information Theory

- **Information (Entropy):** Given a probability distribution \( P = \{P_1, P_2, ..., P_n\} \), information conveyed by this distribution is

\[
Info(P) = -(p_1 \log(p_1) + p_2 \log(p_2) + ... + p_n \log(p_n))
\]

- **Gain:**

\[
Gain(X, T) = Info(T) - Info(X, T)
\]

where

\[
Info(X, T) = -\sum_{i=1}^{m} \frac{T_i}{T} Info(T_i)
\]
Cluster Retrieval Example

### Attribute 1: b
- CBP1, CBP2, CBP4
- CBP1, CBP2

### Attribute 3: a
- CBP5, CBP1, CBP2, CBP4

#### CBC ID Information Gain Ratio

<table>
<thead>
<tr>
<th>CBC ID</th>
<th>Information Gain Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Attribute 2
- CBP3, CBP4
- CBP1, CBP2

#### For CBP1 and CBP2
System Process

1. Initial user query
2. Loaded cases
3. CBSCAN
4. Select Attributes using BuildDFoutext
5. Interactively Answer Questions (Attributes)
6. User Select Clusters
7. One Cluster Left?
   - Yes: CBR Retrieval Tool (CaseAdvisor I)
   - No: Repeat steps 5 and 6
Ablation Study Evaluation

- **Precision** = \((1 - n/10)\)
  - if we set 10 to be the number of cases shown

- **Interactive Efficiency** = 
  \[
  1 - \frac{Q_c}{Q_{all}}
  \]
## Experimental Results

<table>
<thead>
<tr>
<th>UCI Thyroid CB</th>
<th>CA</th>
<th>Cluster</th>
<th>Info Gain</th>
<th>Cluster+Info Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0%</td>
<td>0%</td>
<td>45%</td>
<td>44%</td>
</tr>
<tr>
<td>Interactive Efficiency</td>
<td>56%</td>
<td>58%</td>
<td>97%</td>
<td>96%</td>
</tr>
<tr>
<td>Time (CPU sec)</td>
<td>448</td>
<td>4.3</td>
<td>62.3</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCI Mushroom</th>
<th>CA</th>
<th>Cluster</th>
<th>Info Gain</th>
<th>Cluster+Info Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>6%</td>
<td>83%</td>
<td>92%</td>
<td>92%</td>
</tr>
<tr>
<td>Interactive Efficiency</td>
<td>59%</td>
<td>56%</td>
<td>92%</td>
<td>89%</td>
</tr>
<tr>
<td>Time (CPU sec)</td>
<td>5374</td>
<td>29</td>
<td>201</td>
<td>10</td>
</tr>
</tbody>
</table>
Problem 4: Interactive Retrieval

Jing Wu, MSc.

- Canadian AI 2000
- Applied Intelligence Journal, 2001
Problem 5: Information Gathering and ActiveCBR

Lots of answers are available in various databases already
Thus, no need to ask customers again!
A Typical Interactive-CBR Scenario

1. **Agent**: “What is your name and address?”
   **Customer**: “John, 9004 Lyra Place…”
2. **Agent**: “What is the nature of your problem?”
   **Customer**: “Fuzzy picture on Ch. 3”
3. **Agent**: “Let me check your payment status…OK, you are a paid customer.”
4. **Agent**: “Let me check if there is an outage in your area…”
5. **Agent**: “Has the problem occurred before?”
   **Customer**: “Yes, but I can’t remember how it was fixed.”
6. **Agent**: “No outage. How many outlets do you have…”
A Typical Interactive-CBR Scenario

1. **Agent**: “What is your name and address?”
   **Customer**: “John, 9004 Lyra Place…”

2. **Agent**: “What is the nature of your problem?”
   **Customer**: “Fuzzy picture on Ch. 3”

3. **Agent**: “Let me check your payment status…OK, you are a paid customer.”

4. **Agent**: “Let me check if there is an outage in your area…”

5. **Agent**: “Has the problem occurred before?”
   **Customer**: “Yes, but I can’t remember how it was fixed.”

6. **Agent**: “No outage. How many outlets do you have…”

   - Answered from telephone number and customer database
   - Answered from Sensor Database
   - Answered from customer database
   - Answered from outage database
   - Answered from problem history database
Related Issues

- Decomposing composite questions/queries
  - Has fuzzy picture problem occurred before?
  - Find customer ID
  - Find problem ID
  - Query DB: `Select problems where`...

- Deciding on an order in which to ask questions
Our Aim: Summary

To increase *interactive efficiency* (Aha and Breslow ‘97) through automated information gathering:

- reduce the number of questions posed to customer
- answer as many questions as possible by gathering information from on-line sources
- answer first the questions which will most speed up diagnosis
Problem 5: Information Gathering

System Processes

- Extract the problem state
- Global Knowledge Space
- Problem State
- Retrieved Cases
- Solved Case
- Tested/Repaired Case
- Learned Case
- Stored Cases
- Retrieved Cases
- Solved Case
- Recycle
- Retain
- Reuse
- Retain gathered information
- Task Planer and Executor
- Plan and execute information task
- Task Selector
- Choose an information task

Problem 5: Information Gathering
Key Idea:
Key Idea:
Step 1. Initial Retrieval

- Initial retrieval by keywords in problem description
- Additional attributes focus retrieval through K-nearest neighbor search
- Retrieved cases indicate hypotheses
- Example:

  **Hypothesis:** Parental control switch on  
  **Attributes:**
  - Problem description: poor reception of the cable signal 1.0  
  - Channels affected: channel 50 0.7  
  - Uses parental control: yes 0.8  
  - Has cable box: yes 0.4  
  - Outlets concerned: 1 0.3
Step 2. Generating Queries from Retrieved Cases

- Select an attribute with high estimated utility as a query, based on the following two values:
  - Information Value
    - the number of times the question appears in the candidate cases,
    - the weights of the question in the candidate cases, and
    - the ranks of the cases containing the question
  - Cost of evaluating the attribute

- Score of the attribute is \( \frac{c(a)}{v(a)} \)

- System selects the attribute with the maximal value as the information task for subsequent planning.
### Step 2: Query Ordering

#### Signal Case

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem?</td>
<td>Poor recep</td>
<td>1.00</td>
</tr>
<tr>
<td>Channels?</td>
<td>3-10</td>
<td>0.80</td>
</tr>
<tr>
<td>Local signal?</td>
<td>clear</td>
<td>0.95</td>
</tr>
</tbody>
</table>

**Score = 80%**

#### Parental Control Case

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem?</td>
<td>Poor recep</td>
<td>0.5</td>
</tr>
<tr>
<td>Channels?</td>
<td>50-52</td>
<td>0.1</td>
</tr>
<tr>
<td>Parental control?</td>
<td>yes</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Score = 90%**

#### Information Value of attributes:

- \( V(\text{Channels}) = (0.8 \times 0.8 + 0.9 \times 0.1) = 0.73 \)
- \( V(\text{local signal}) = 0.8 \times 0.95 = 0.76 \)
- \( V(\text{parental control}) = 1.0 \times 0.9 = 0.9 \)
Step 2: Decomposing composite queries

- Given: a library of information-task schemata

  Use-parental-control :- Ask(customer)
  Use-parental-control :- Check-online
  Check-online :- Query(account) and Query-data-source
  Query-data-source :- Query(customer-profile)
  Query-data-source :- Query(work_log)

- The schema is used to expand the information task into an AND-OR Tree
Example of AND-OR Tree

Cost=13
parental control switch?

Cost=13
check on-line

get customer account number
Cost=3

query data source

query accounts
Cost=10

Cost=10

Cost Algorithm: bottom-up

Cost(AND-Node) = Max \{Cost(Children-Node)\}
Cost(OR-Node) = Min \{Cost(Children-Node)\}

Problem 5: Information Gathering
Cost Models at Leaf Nodes

- Defined or learned from database characteristics
  - propagated up the task hierarchy
- Costs include
  - time to access data source
  - reliability of source
  - intrusion (querying customer)
Problem 5: Information Gathering and Active CBR

C. Carrick, Sheng Li, I. Abi-Zeid and L. Lamontagne

- ICCBR ’99
- EWCBR ‘00
- International Journal of Knowledge and Information Systems, Kluwer
Field test

Objective?

- Real-time problem solving
- Junior CSR training
- New technology education
- Consistent answers
Status

- Rogers Cable Systems Ltd.
- Help Desks
- Educational Systems
  - Experimental testbed
  - Tool to learn about CBR
  - CBR for software requirement engineering
  - Other uses
Conclusions

- Problem-driven research methodologies
- Case-base maintenance main objective
  - Hard problem
  - CBR without maintenance???
- Case-adaptation practical?
- Future: Case mining