CS 681 Fall 2008

Designing Expert Systems

8. Multistrategy Rule Refinement

Gheorghe Tecuci
otecuci@gmu.edu
http://lac.gmu.edu/

Learning Agents Center
and Computer Science Department
George Mason University
Overview

Rule Refinement Problem and Method

Rule Refinement Demo and Hands On

Discussion

Hands On: Rule Learning and Refinement

Reading
Control of Modeling, Learning and Problem Solving

- Problem
- Mixed-Initiative Problem Solving
- Reasoning Tree
  - Extend Reasoning Tree
  - Reject Reasoning Steps
    - Explain Examples
  - Accept Reasoning Step
    - Explain Examples
  - Rules Refinement
- Ontology + Rules
  - Learned Rules
  - Refined Rules
  - Refined Ontology

- Rules Learning
- Explain Examples
The Rule Refinement Problem (Definition)

GIVEN:

• a plausible version space rule;

• a positive or a negative example of the rule (i.e. a correct or an incorrect problem reduction);

• a knowledge base that includes an object ontology and a set of problem reduction rules;

• an expert that understands why the example is positive or negative, and can answer agent’s questions.

DETERMINE:

• an improved rule that covers the example if it is positive, or does not cover the example if it is negative;

• an extended object ontology (if needed for rule refinement).
Rule Learning

LEARNED REDUCTION RULE

REDUCTION EXAMPLE

Assess whether John Doe is a potential PhD advisor for Bob Sharp in Artificial Intelligence.

Question
Is John Doe likely to stay on the faculty of George Mason University for the duration of Bob Sharp’s dissertation?

Answer
Yes, because John Doe has a tenured position which is a long term position.

Sub-task (1)
Assess whether John Doe would be a good PhD advisor for Bob Sharp in Artificial Intelligence.

DECOMPOSITION RULE DDR.00001 FORMAL DESCRIPTION


Q: Is ?O1 likely to stay on the faculty of ?O4 for the duration of ?O2's dissertation?

A: Yes, because ?O1 has ?O5 which is a long term position.

MAIN CONDITION

<table>
<thead>
<tr>
<th>Var</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O1</td>
<td>(PhD advisor, associate professor)</td>
<td>(person)</td>
</tr>
<tr>
<td>?O2</td>
<td>(PhD student)</td>
<td>(agent)</td>
</tr>
<tr>
<td>?O3</td>
<td>(computer science)</td>
<td>(research area)</td>
</tr>
<tr>
<td>?O4</td>
<td>(university)</td>
<td>(employer)</td>
</tr>
<tr>
<td>?O5</td>
<td>(long term position)</td>
<td>(long term position)</td>
</tr>
</tbody>
</table>

The Rule Refinement Method

Knowledge Base

Learning by Analogy and Experimentation

Failure explanation

Example of problem reductions generated by the agent

Incorrect example

Correct example

Learning from Explanations

Learning from Examples

IF we have to solve <Problem>

Main PVS Condition

Except-When PVS Condition

Except-When PVS Condition

THEN solve  
<Subproblem 1>  
...  
<Subproblem m>
Rule Generalization with a Positive Example

New positive example that satisfies the upper bound but not the lower bound

Condition corresponding to the example

- $O_1$ is Bridget Jones
- $O_4$ has as employer $O_2$
- $O_5$ has as position $O_3$
- $O_2$ is Bob Sharp
- $O_3$ is Artificial Intelligence
- $O_4$ is George Mason University
- $O_5$ is tenure position

Refined Rule

<table>
<thead>
<tr>
<th>Var</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>professor, PhD advisor</td>
<td>person</td>
</tr>
<tr>
<td>$O_2$</td>
<td>PhD student</td>
<td>agent</td>
</tr>
<tr>
<td>$O_3$</td>
<td>computer science</td>
<td>research area</td>
</tr>
<tr>
<td>$O_4$</td>
<td>university</td>
<td>employer</td>
</tr>
<tr>
<td>$O_5$</td>
<td>long term position</td>
<td>long term position</td>
</tr>
</tbody>
</table>

THEN: Assess whether $O_1$ would be a good PhD advisor for $O_2$ in $O_3$. 

© 2008, Learning Agents Center
The left side of the previous slide shows an example generated by the agent. This example is generated because it satisfies the plausible upper bound condition of the rule.

This example is accepted as correct by the expert. Therefore the plausible lower bound condition is generalized to cover it as shown in the following slide.
Minimal Generalization of Plausible Lower Bound

Plausible Lower Bound Condition

?O1 is \{PhD advisor, professor\}
has as employer ?O4
has as position ?O5

?O2 is PhD student

?O3 is computer science

?O4 is university

?O5 is long term position

Condition corresponding to the example

?O1 is Bridget Jones
has as employer ?O4
has as position ?O5

?O2 is Bob Sharp
?O3 is Artificial Intelligence
?O4 is George Mason University
?O5 is tenure position

Plausible Lower Bound Condition

?O1 is \{PhD advisor, associate professor\}
has as employer ?O4
has as position ?O5

?O2 is PhD student

?O3 is computer science

?O4 is university

?O5 is long term position
Rule Specialization with a Negative Example

Negative Example

Assess whether Dan Smith is a potential PhD advisor for Bob Sharp in Information Security.

Question
Is Dan Smith likely to stay on the faculty of George Mason University for the duration of Bob Sharp’s dissertation?

Answer
Yes, because Dan Smith has a tenured position which is a long term position.

Sub-task
Assess whether Dan Smith would be a good PhD advisor for Bob Sharp in Information Security.

Failure Explanation
Dan Smith plans to retire from George Mason University

Rewrite as
Except When Condition 1
?O4 is George Mason University
?O1 is Dan Smith
plans to retire from ?O4

Most specific generalization
Most general generalization
Rule Specialization with another Negative Example

**Negative Example**

*Task*
Assess whether Jane Austin is a potential PhD advisor for Bob Sharp in Information Security.

*Question*
Is Jane Austin likely to stay on the faculty of George Mason University for the duration of Bob Sharp's dissertation?

*Answer*
Yes, because Jane Austin has a tenured position which is a long term position.

*Sub-task*
Assess whether Jane Austin would be a good PhD advisor for Bob Sharp in Information Security.

**Failure Explanation**
Jane Austin plans to move to Indiana University

Rewrite as
Except When Condition 2

?O6 is Indiana University
?O1 is Jane Austin
plans to move to ?O4

*Main Condition*

<table>
<thead>
<tr>
<th>Var</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O1</td>
<td>(professor, PhD advisor)</td>
<td>(person)</td>
</tr>
<tr>
<td>?O2</td>
<td>(PhD student)</td>
<td>(agent)</td>
</tr>
<tr>
<td>?O3</td>
<td>(computer science)</td>
<td>(research area)</td>
</tr>
<tr>
<td>?O4</td>
<td>(university)</td>
<td>(employer)</td>
</tr>
<tr>
<td>?O5</td>
<td>(long term position)</td>
<td>(long term position)</td>
</tr>
</tbody>
</table>

*Except When Condition 1*

<table>
<thead>
<tr>
<th>Var</th>
<th>Relationship</th>
<th>Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O1</td>
<td>has as employer</td>
<td>?O4</td>
</tr>
<tr>
<td>?O1</td>
<td>has as position</td>
<td>?O5</td>
</tr>
</tbody>
</table>

*Except When Condition 2*

<table>
<thead>
<tr>
<th>Var</th>
<th>Relationship</th>
<th>Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O6</td>
<td>(university)</td>
<td></td>
</tr>
<tr>
<td>?O1</td>
<td>(PhD advisor, full professor)</td>
<td></td>
</tr>
</tbody>
</table>

*Then*
Assess whether ?O1 would be a good PhD advisor for ?O2 in ?O3.
Solving, Modeling, and Learning

1. Solving
   Applies learned rules to solve new problems

2. Modeling
   Extends the reasoning tree

3. Learning
   Learns a new rule
   Rule 1

- Solving
  - Assess whether Jill Knox is a potential PhD advisor for Bob Sharp.
    - Is Bob Sharp interested in the area of expertise of Jill Knox?
      - Yes, because Bob Sharp is interested in Information Security which is the area of expertise of Jill Knox.
    - Assess whether Jill Knox is a potential PhD advisor for Bob Sharp in Information Security.
    - Is Bob Sharp interested in the area of expertise of Jill Knox?
      - Yes, because Bob Sharp is interested in Artificial Intelligence which is the area of expertise of Jill Knox.
    - Assess whether Jill Knox is a potential PhD advisor for Bob Sharp in Artificial Intelligence.
      - Is Jill Knox likely to stay on the faculty of George Mason University for the duration of Bob Sharp’s dissertation?
        - Yes, because Jill Knox has a tenure track position and it is almost certain to get a tenured position.
    - Assess whether Jill Knox would be a good PhD advisor for Bob Sharp in Artificial Intelligence.
Rule Learning

**Example 1**

**Task**: Assess whether Jill Knox is a potential PhD advisor for Bob Sharp in Artificial Intelligence.

**Question**: Is Jill Knox likely to stay on the faculty of George Mason University for the duration of Bob Sharp's dissertation?

**Sub-task**: Assess whether Jill Knox would be a good PhD advisor for Bob Sharp in Artificial Intelligence.

**Answer**: Yes, because Jill Knox has a tenure track position and it is **almost certain** to get a tenured position.

**Variable**

<table>
<thead>
<tr>
<th>Var</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi_1$</td>
<td>assistant professor</td>
<td>professor</td>
</tr>
<tr>
<td>$\Phi_2$</td>
<td>PhD student</td>
<td>agent</td>
</tr>
<tr>
<td>$\Phi_3$</td>
<td>computer science</td>
<td>research area</td>
</tr>
<tr>
<td>$\Phi_4$</td>
<td>university</td>
<td>employer</td>
</tr>
<tr>
<td>$\Phi_5$</td>
<td>faculty position</td>
<td>position</td>
</tr>
<tr>
<td>$\Phi_6$</td>
<td>long term position</td>
<td></td>
</tr>
<tr>
<td>$\Phi_7$</td>
<td>almost certain</td>
<td>almost certain</td>
</tr>
<tr>
<td>$\Phi_8$</td>
<td>no evidence</td>
<td>almost certain</td>
</tr>
</tbody>
</table>

**Relationships**

<table>
<thead>
<tr>
<th>Var</th>
<th>Relationship</th>
<th>Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi_1$</td>
<td>has as position</td>
<td>$\Phi_5$</td>
</tr>
<tr>
<td>$\Phi_1$</td>
<td>likelihood of tenure</td>
<td>$\Phi_7$</td>
</tr>
<tr>
<td>$\Phi_1$</td>
<td>has as employer</td>
<td>$\Phi_4$</td>
</tr>
</tbody>
</table>

**Main Condition**

**IF**: Assess whether $\Phi_1$ is a potential PhD advisor for $\Phi_2$ in $\Phi_3$.

**Q**: Is $\Phi_1$ likely to stay on the faculty of $\Phi_4$ for the duration of $\Phi_2$'s dissertation?

**A**: Yes, because $\Phi_1$ has $\Phi_5$ and it is $\Phi_7$ to get $\Phi_6$. 
Rule Refinement with a Positive Example: Details

1. If the positive example E is covered by ML and is not covered by XU (case 1), then the rule does not need to be refined because the example is correctly classified as positive by the current rule.

2. If E is covered by MU, but it is not covered by ML and XU (case 2), then minimally generalize ML to cover E and remain less general than MU. Remove also from MU the elements that do not cover E.

3. If E is not covered by MU (cases 3, 4, and 5), or if E is covered by XL (cases 5, 6, and 7), then keep E as a positive exception of the rule.
4. If E is covered by ML and XU, but it is not covered by XL (case 8), then interact with the expert to find an explanation of the form: “The problem reduction step is correct because Ii is Ci,” where Ii is an instance from the example E and Ci is a concept from the ontology. If such an explanation is found, then XU is minimally specialized to no longer cover Ci. Otherwise, E is kept as a positive exception.

5. If E is covered by MU and XU, but it is not covered by ML and XL (case 9), then minimally generalize ML to cover E and remain less general than MU. Also remove from MU the elements that do not cover E. Then continue as in step 4.
Rule Refinement with a Negative Example: Details

1. If the negative example E is covered by ML and it is not covered by XU (case 1), then interact with the subject matter expert to find an explanation of why E is a wrong problem reduction step. If an explanation EX is found, then generate a new Except When plausible version space condition and add it to the rule. Otherwise, keep E as a negative exception.

2. If E is covered by MU but it is not covered by ML and by XU (case 2) then interact with the expert to find an explanation of why E is a wrong problem reduction step. If an explanation EX is found and it has the form “li is not a Ci,” where Ci is a concept covered by MU, then specialize MU to be covered by Ci. Otherwise, if another type of explanation EX is found then learn a new Except When condition based on it, and add this condition to the rule.
3. If E is not covered by MU (cases 3, 4, 5), or it is covered by XL (cases 5, 6, 7), then the rule does not need to be refined because the example is correctly classified as negative by the current rule.

4. If E is covered by ML and XU but it is not covered by XL (case 8), or E is covered by MU and XU but it is not covered by ML and XL (case 9), then minimally generalize XL to cover E and specialize XU to no longer include the concepts that do not cover E.
Rule Refinement with a Positive Example: Details

1. If the positive example E is covered by ML and is not covered by XU (case 1), then the rule does not need to be refined because the example is correctly classified as positive by the current rule.
2. If E is covered by MU, but it is not covered by ML and XU (case 2), then minimally generalize ML to cover E and remain less general than MU. Remove also from MU the elements that do not cover E.
3. If E is not covered by MU (cases 3, 4, and 5), or if E is covered by XL (cases 5, 6, and 7), then keep E as a positive exception of the rule.
4. If E is covered by ML and XU, but it is not covered by XL (case 8), then interact with the expert to find an explanation of the form: “The problem reduction step is correct because Ii is Ci,” where Ii is an instance from the example E and Ci is a concept from the ontology. If such an explanation is found, then XU is minimally specialized to no longer cover Ci. Otherwise, E is kept as a positive exception.
5. If $E$ is covered by $MU$ and $XU$, but it is not covered by $ML$ and $XL$ (case 9), then minimally generalize $ML$ to cover $E$ and remain less general than $MU$. Also remove from $MU$ the elements that do not cover $E$. Then continue as in step 4.

“$li$ is $Ci$”
1. If the negative example E is covered by ML and it is not covered by XU (case 1), then interact with the subject matter expert to find an explanation of why E is a wrong problem reduction step. If an explanation EX is found, then generate a new Except When plausible version space condition and add it to the rule. Otherwise, keep E as a negative exception.
Rule Refinement with a Negative Example: Details

2. If E is covered by MU but it is not covered by ML and by XU (case 2) then interact with the expert to find an explanation of why E is a wrong problem reduction step. If an explanation EX is found and it has the form “li is not a Ci,” where Ci is a concept covered by MU, then specialize MU to be covered by Ci. Otherwise, if another type of explanation EX is found then learn a new Except When condition based on it, and add this condition to the rule.
3. If E is not covered by MU (cases 3, 4, 5), or it is covered by XL (cases 5, 6, 7), then the rule does not need to be refined because the example is correctly classified as negative by the current rule.
4. If E is covered by ML and XU but it is not covered by XL (case 8), or E is covered by MU and XU but it is not covered by ML and XL (case 9), then minimally generalize XL to cover E and specialize XU to no longer include the concepts that do not cover E.
Summary: Rule Refinement with Negative Example

Rule Condition C

- Learn Except When Condition (C, Ex)
- Keep as Negative Exception (C, Ex)
- Generalize Lower Bound of Except When Condition (C, Ex)
- Specialize Upper Bound of Main Cond (C, Ex)
- Learn Except When Condition (C, Ex)
- Keep as Negative Exception (C, Ex)

- Keep as Negative Example (C, Ex)
- Generalize Lower Bound of Except When Condition (C, Ex)
- Specialize Upper Bound of Main Cond (C, Ex)
- Learn Except When Condition (C, Ex)
- Keep as Negative Exception (C, Ex)
Overview

- Rule Refinement Problem and Method
- Rule Refinement Demo and Hands On
- Discussion
- Hands On: Rule Learning and Refinement
- Reading
Hands On: Rule Learning and Refinement

Install the system from:

Load the “PAD-m2o\CS681” scenario KB.
Select question/answer pair

Click on “Reduction Rule” to see the corresponding rule
**Reduction Rule**

**IF:** Assess whether ?O1 is a potential PhD advisor for ?O2.

**Q:** Is ?O2 interested in the area of expertise of ?O1?

**A:** Yes, because ?O2 is interested in ?O3 which is the area of expertise of ?O1.

**MAIN CONDITION**

<table>
<thead>
<tr>
<th>Var</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O1</td>
<td>(PhD advisor, associate professor)</td>
<td>(person)</td>
</tr>
<tr>
<td>?O2</td>
<td>(PhD student)</td>
<td>(person)</td>
</tr>
<tr>
<td>?O3</td>
<td>(computer science)</td>
<td>(PhD research area)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Var</th>
<th>Relationship</th>
<th>Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O2</td>
<td>is interested in</td>
<td>?O3</td>
</tr>
<tr>
<td>?O1</td>
<td>is expert in</td>
<td>?O3</td>
</tr>
</tbody>
</table>

**THEN:** Assess whether ?O1 is a potential PhD advisor for ?O2 in ?O3.
Refinement with a New Problem

Close current “Mixed-Initiative Reasoner” and open it with a new problem

Select this problem

Select this problem
Rule Refinement with Positive Example

1. **Select question/answer pair**
2. **Click on “Reduction Rule” to see the corresponding rule.**
3. **Click on “Correct Reduction” to generalize the rule**
4. **Click on “Reduction Rule” to see the generalized rule**
5. **How was the rule generalized?**
Rule Refinement with Negative Example

1. Select question/answer pair.
2. Click on “Reduction Rule” to see the corresponding rule.
3. Click on “Incorrect Reduction” to specialize the rule because Dan Smith plans to retire.
4. Open “Rule Browser”, select Rule 0001 and “Formal Description” to see the refined rule.
5. Close “Rule Editor” and “Mixed-Initiative Reasoner
6. How was the rule specialized?
Close current “Mixed-Initiative Reasoner” and open it with a new problem

Select this problem

Assess whether John Doe is a potential PhD advisor for Bob Sharp.
Assess whether Bridget Jones is a potential PhD advisor for Bob Sharp.
Assess whether Dan Smith is a potential PhD advisor for Bob Sharp.
Assess whether Jane Austin is a potential PhD advisor for Bob Sharp.
Assess whether Dan Smith is a potential PhD advisor for Peter Jones.
1. Select question/answer pair.

2. Click on “Reduction Rule” to see the corresponding rule.

3. Click on “Incorrect Reduction” to specialize the rule because Jane Austin plans to move from George Mason University.

4. Open “Rule Browser”, select Rule 0001 and “Formal Description” to see the refined rule.

5. Close “Rule Editor” and “Mixed-Initiative Reasoner”.

6. How was the rule specialized?
Close current “Mixed-Initiative Reasoner” and open it with a new problem

Select this problem
Updating the Natural Language Form of a Rule

1. Select question/answer pair.
2. Click on “Reduction Rule” to see the corresponding rule.
3. Compare the condition with the question/answer pair.
4. What do you notice?
5. Select “Reasoning Step”
6. Right-click on the answer pane and select “Modify”
7. Modify the answer to reflect the rule’s condition.
8. See how the reasoning tree has been updated.
Learn with a New Problem

Close current “Mixed-Initiative Reasoner” and open it with a new problem

Click on “Problem Pattern” to assess Jill Knox
Define the Problem

1. Deselect “Filter”
2. Double click on problem pattern
3. Select instances
4. Click on “Create”
1. **Extend reasoning to indicate that Jill Knox has a tenure-track position and she is very likely to get tenure.**
   a. **Select Reasoning Step**
   b. **Select Modeling**
   c. **Select the question suggested by the Modeling assistant**
   d. **Define a new answer including the fact that the likelihood of Jill Knox getting tenure is almost certain.**
   e. **Select the subproblem suggested by the Modeling assistant.**

2. **Learn the corresponding rule.**

3. **Notice how the reasoning tree was extended.**
1. Extend the ontology with another faculty on a tenure-track position who has a different likelihood of getting tenure.
2. Assess that new faculty.
3. Refine the corresponding rule.
Overview

Rule Refinement Problem and Method

Rule Refinement Demo and Hands On

Discussion

Hands On: Rule Learning and Refinement

Reading
Characterization of the Learned Rule

Universe of Instances

Plausible Upper Bound

Hypothetical Exact Condition

Plausible Lower Bound
The previous slide shows the expected relationship between the plausible lower bound condition, the plausible upper bound condition, and the exact (hypothetical) condition that the agent is attempting to learn.

When the rule is learned from an example, its bounds are obtained as plausible generalizations performed in the context of an incomplete ontology. During rule learning, both the upper bound and the lower bound are generalized and specialized to converge toward one another and toward the hypothetical exact condition. This is different from the classical version space method where the upper bound is only specialized and the lower bound is only generalized.

Notice also that, as opposed to the classical version space method (where the exact condition is always between the upper and the lower bound conditions), in Disciple the exact condition may not include part of the plausible lower bound condition, and may include a part that is outside the plausible upper bound condition.

We say that the plausible lower bound is, AS AN APPROXIMATION, less general than the hypothetical exact condition. Similarly, the plausible upper bound is, AS AN APPROXIMATION, more general than the hypothetical exact condition.

These characteristics are a consequence of the incompleteness of the representation language (i.e. the incompleteness of the object ontology), of the heuristic strategies used to learn the rule, and of the fact that the object ontology may evolve during learning.
Problem Solving with Partially Learned Rules

IF
<problem>

Plausible Upper Bound Condition
<PUB condition>

Plausible Lower Bound Condition
<PLB condition>

THEN
<subproblem 1>
...
<subproblem m>

Plausible Lower Bound Condition
<PLB condition>

Correct Creative Solutions
Correct Inventive Solutions
Correct Innovative Solutions
Correct Routine Solutions
Incorrect Routine Solutions
Incorrect Innovative Solutions
Incorrect Inventive Solutions

Target Solution Space
Current Representation Space
Final Representation Space
Problem Solving with Partially Learned Rules

PVS Condition

- The rule is not applicable
- Rule’s conclusion is plausible
- Rule’s conclusion is (most likely) correct
- Rule’s conclusion is not plausible

Except-When PVS Condition

- Rule’s conclusion is (most likely) incorrect
- Rule’s conclusion is not plausible
Uses the explanation of the first positive example to generate a much smaller version space than the classical version space method.

Conducts an efficient heuristic search of the version space, guided by explanations, and by the maintenance of a single upper bound condition and a single lower bound condition.

Will always learn a rule, even in the presence of exceptions.

Learns from a few examples and an incomplete knowledge base.

Uses a form of multistrategy learning that synergistically integrates learning from examples, learning from explanations, and learning by analogy, to compensate for the incomplete knowledge.

Uses mixed-initiative reasoning to involve the expert in the learning process.

Is applicable to complex real-world domains, being able to learn within a complex representation language.
Overview

Rule Refinement Problem and Method

Rule Refinement Demo and Hands On

Discussion

Hands On: Rule Learning and Refinement

Reading
These Lecture Notes (required).

Tecuci G., Boicu M., Boicu C., Marcu D., Stanescu B., Barbulescu M.,
The Disciple-RKF Learning and Reasoning Agent, Computational
Intelligence, Volume 21, Number 4, 2005, pp 15-28 (required).

Tecuci G., Boicu M., Boicu C., Marcu D., Boicu C., Barbulescu M., Ayers
C., Cammons D., Cognitive Assistants for Analysts, 2007 (required).

Tecuci, G., Boicu, M., Marcu, D., Stanescu, B., Boicu, C., Comello, J.,
Training and Using Disciple Agents: A Case Study in the Military Center
of Gravity Analysis Domain, AI Magazine, 24, 4:51-68, AAAI Press,
Menlo Park, California, 2002 (recommended). Available at

Tecuci, Building Intelligent Agents, Academic Press, 1998, Ch. 4 pp. 79-
146 (rule learning and refinement in Disciple) (recommended).