Hybrid Intelligent Systems

Rajendra M Sonar, Ph.D.
SJM School of Management,
Indian Institute of Technology Bombay
Mumbai-400 076
rm_sonar@iitb.ac.in

Intelligent Systems

How problems are solved?

Knowledge + Use and apply the knowledge

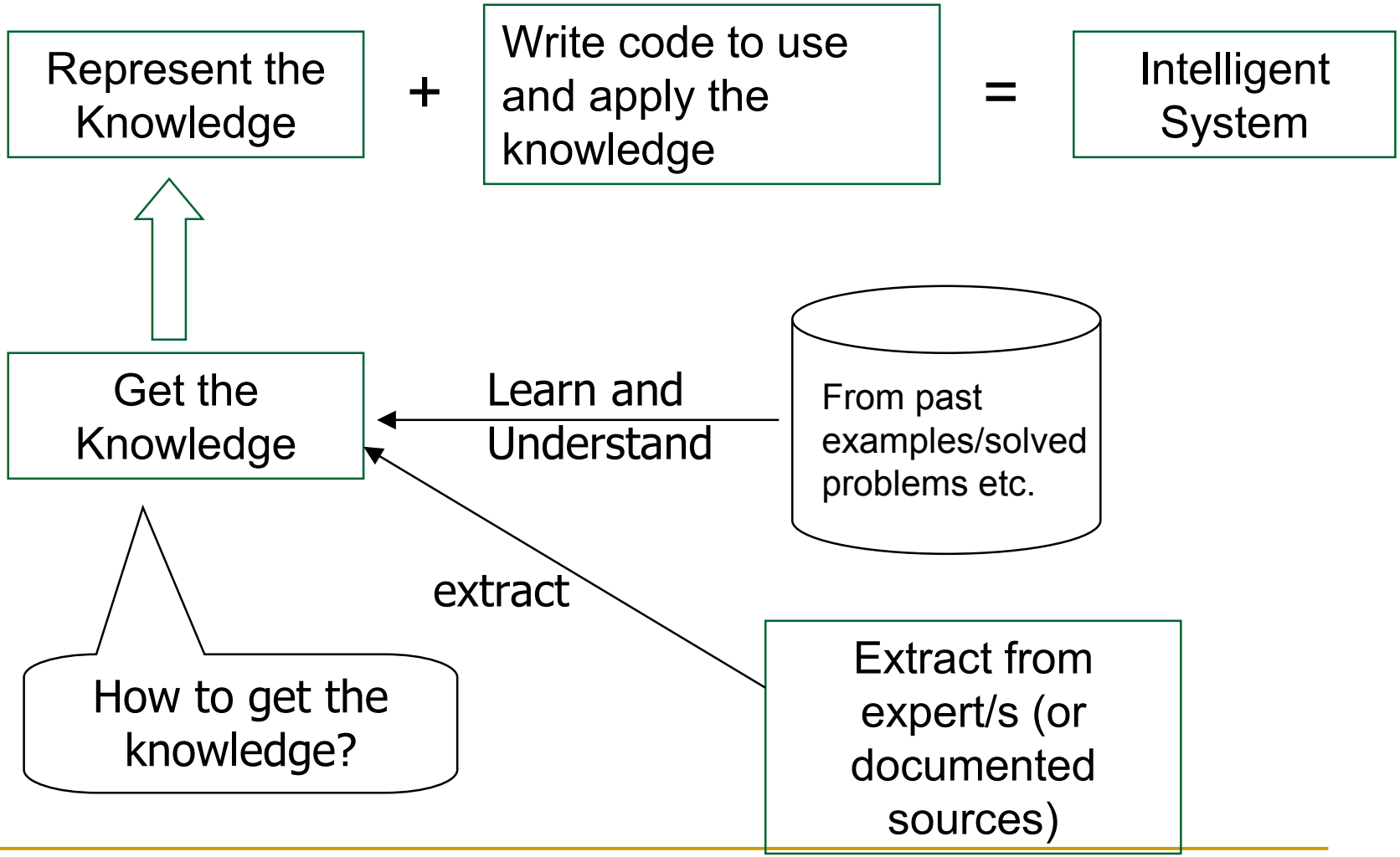
(Tools/Technique/..)

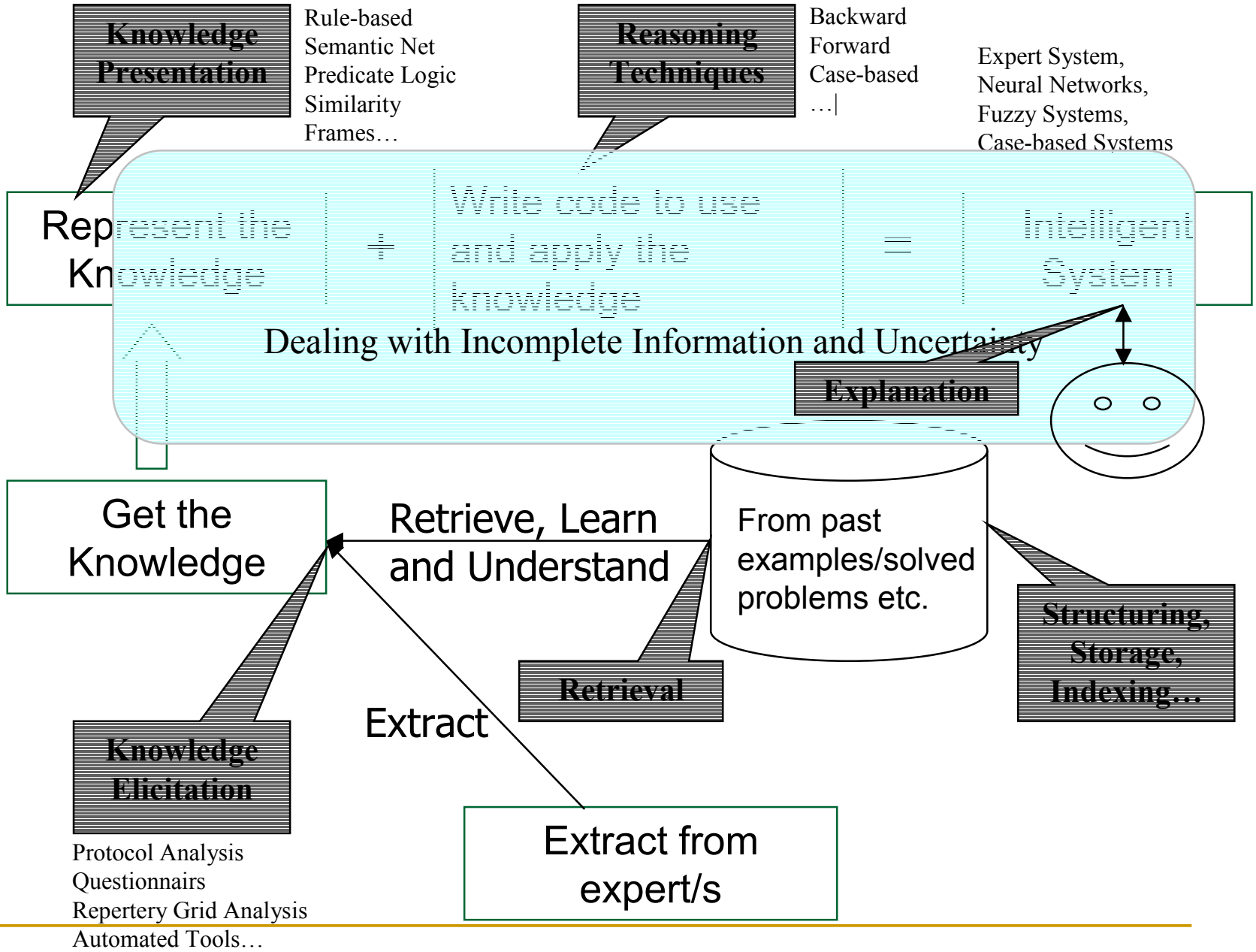
= Problem Solving

Knowledge?

- facts, ideas that have been acquired through experience, investigation and observations

How to make a machine intelligent?




Knowledge Presentation
Rule-based
Semantic Net
Predicate Logic
Similarity
Frames…
Reasoning Techniques
Backward
Forward
Case-based
…|
Expert System,
Neural Networks,
Fuzzy Systems,
Case-based Systems
Represent the Knowledge
+
Write code to use and apply the knowledge
=
Intelligent System
Dealing with Incomplete Information and Uncertainty
Explanation
Get the Knowledge
Retrieve, Learn and Understand
From past examples/solved problems etc.
Structuring, Storage, Indexing…
Retrieval
Extract
Knowledge Elicitation
Protocol Analysis
Questionnairs
Repertery Grid Analysis
Automated Tools…
Extract from expert/s

Intelligent techniques differ in ways they

- Acquire and get the knowledge
- Represent and store the knowledge
- Use and apply the knowledge
- Deal with uncertainty and incomplete information
- Deal with and adapt to new problems
- Retain/Revise the knowledge
- Explain the problem solving

Widely used intelligent techniques to develop intelligent systems

- Rule-based Reasoning
- Neural Networks
- Case-based Reasoning
- Fuzzy Systems
- Genetic Algorithms
- Model-based Reasoning

Comparing Intelligent Systems (expert system, CBR and ANN)

	Foundation
RBR	Logic
ANN	Biological nervous system
CBR	Human Reasoning

Knowledge Containers and Model complexity

RBR	Knowledge is explicit in the form of set of rules, rules are easier to understand
ANN	Knowledge is implicit (in the form neural network topology, transfer function, weights etc). Models are like black-box, less explanatory
CBR	Implicit in the form of past examples and explicit in the form of domain vocabulary, and domain knowledge. Domain knowledge is: used for indexing, retrieval and adaptation. Models are easier to understand, combination of experience and domain knowledge

Learning & Adaptability

RBR	Rules are static, do not learn, new rules needs to be explicitly added and existing to be updated.
ANN	Learn from the examples, trial-and-error, completely automated, no domain knowledge, needs to re-trained for new examples. Adapts to new examples easily.
CBR	Do not learn but find out most matching cases, reuse and adapt the similar ones based on similarity knowledge, lazy learning (defer the learning as late as possible and take care of latest experience). New cases can be easily added, Knowledge can be updated.

	Reliability
RBR	Based on experts or knowledge sources
ANN	Depends upon the data/examples.
CBR	Based on past examples/cases/data as well as domain expertise.

Matching & Generalization

RBR	The rules must be completely satisfied (matched).
ANN	Approximate and has generalization capabilities.
CBR	Approximate based on similarity measures and knowledge used.

Development Efforts

RBR	Initial efforts are high, need to convert tacit knowledge into explicit which is time consuming and costly. It is relatively easier to build general knowledge than the specific one.
ANN	Deals with only quantitative data, needs lots of preprocessing, modelling is trial and error, needs understanding of various algorithms, architectures, transfer functions etc. large and representative examples are required
CBR	Relatively easier when structured examples available, can easily be integrated with database systems in structural CBR, deals with qualitative as well quantitative data, flexible and not much of formatting and preprocessing required

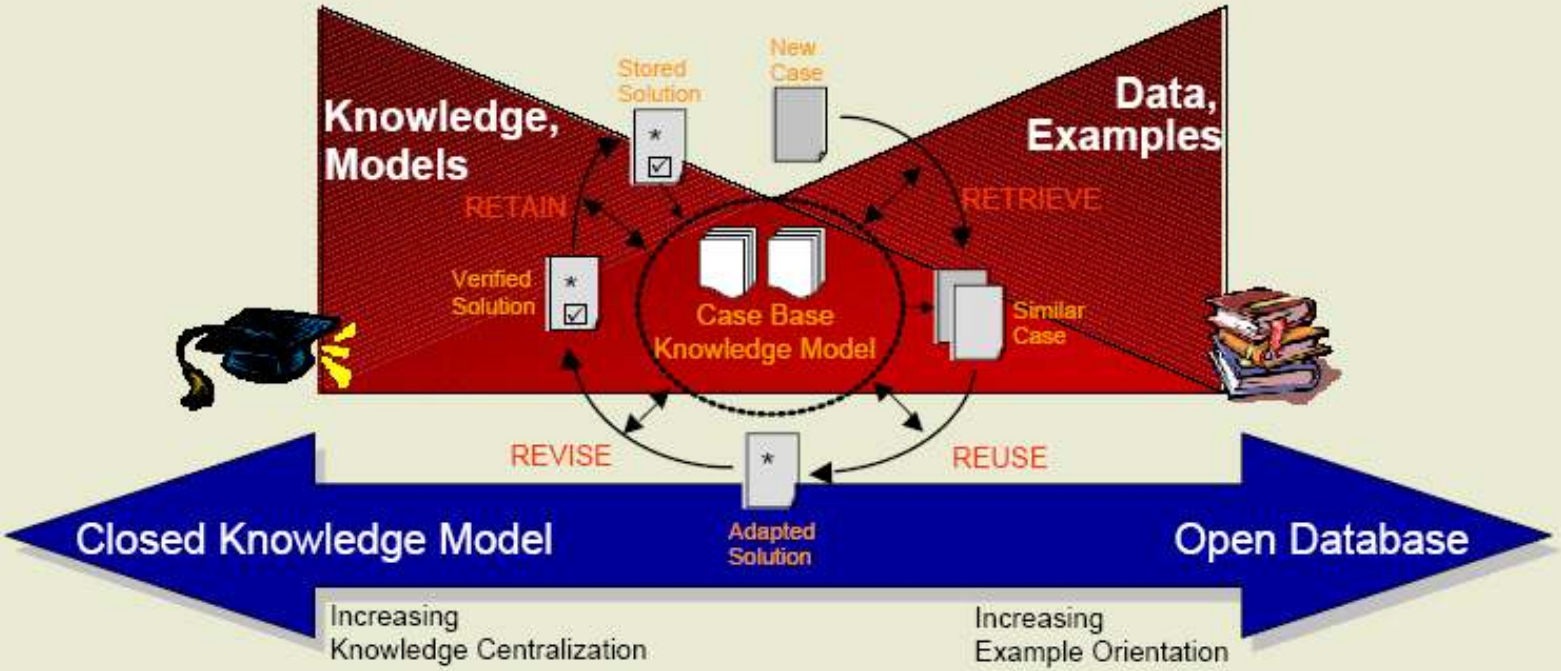
Maintenance

RBR	Modifying knowledge-base is difficult task especially when knowledge-base is large or modifications that are incremental
ANN	Relatively easily to modify ANN and retrain them or remodel again.
CBR	Easier as new cases can be added and existing cases can be revised or removed etc. Domain knowledge is less and can be managed. Case structure can be modified.

Problem domains and capabilities

RBR	Work in well understood and narrow domain. Expertise needs to be available. Good for diagnostics, monitoring, control, data analysis, advisory applications
ANN	Can address complex problems right from classification to clustering
CBR	Addresses wide range of problems from diagnostics, collaborative filtering to knowledge-management.

Positioning the CBR System



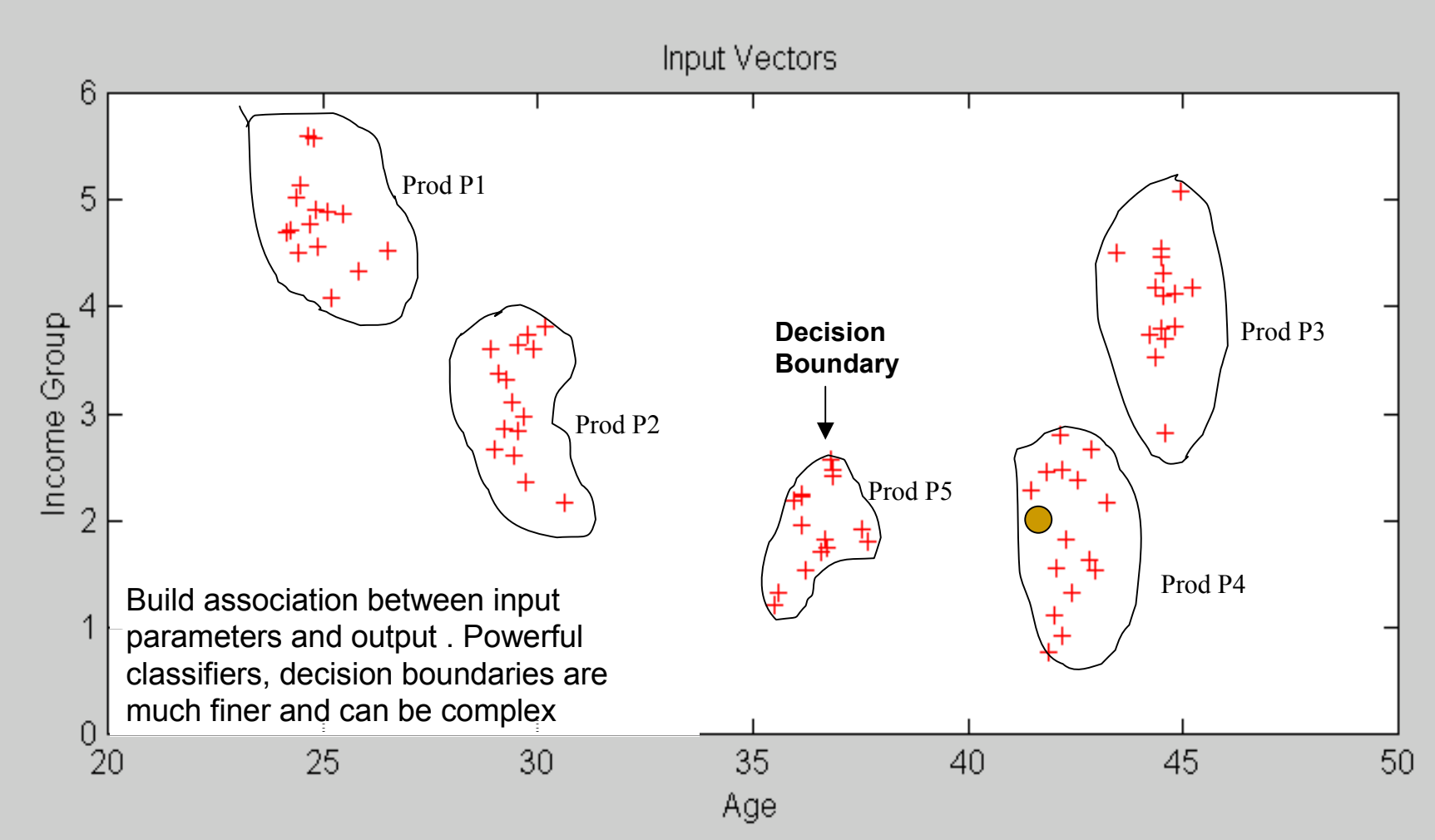
Expert Systems **CBR Systems** **Databases**

FACHBEREICH INFORMATIK

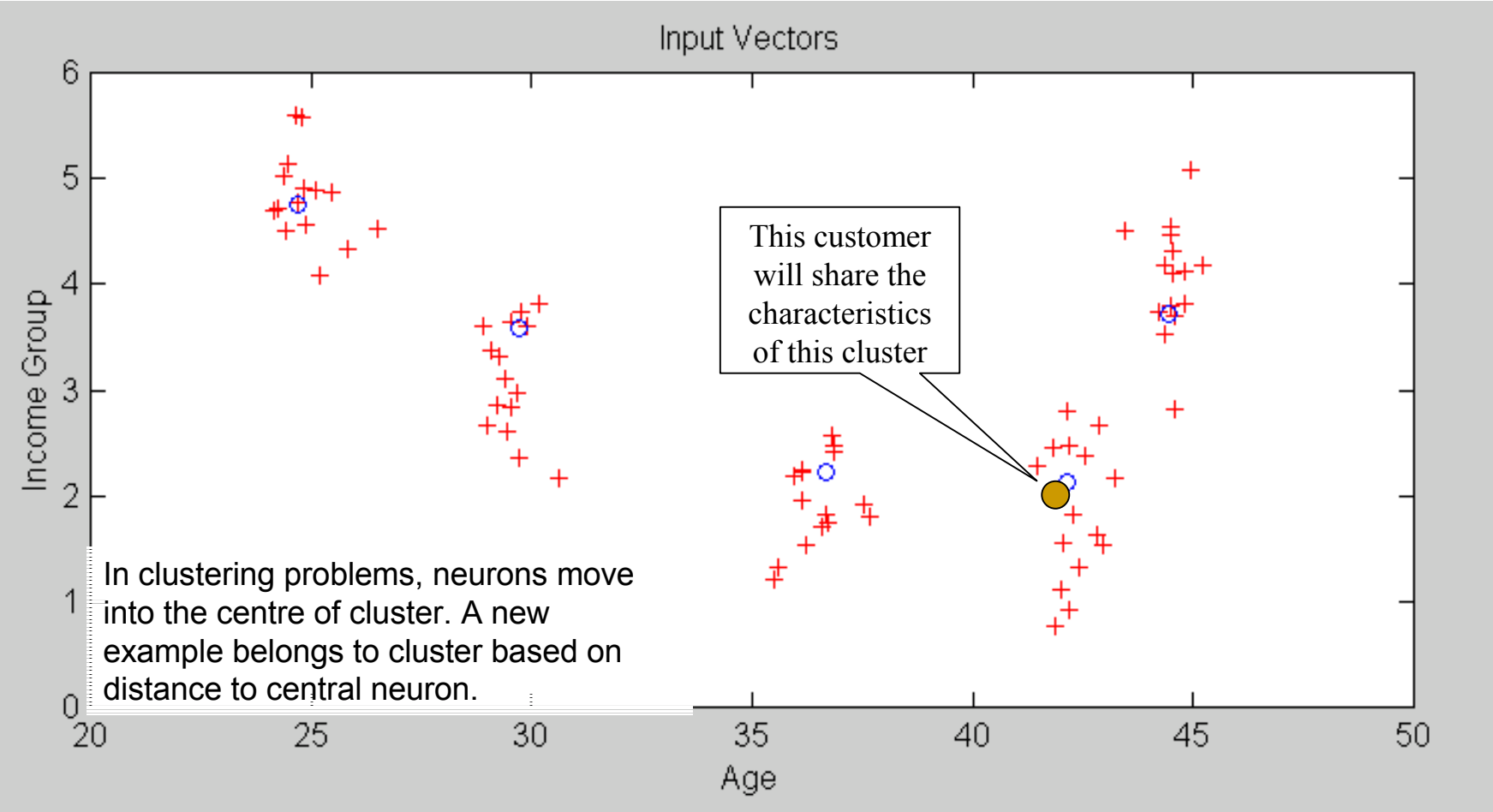
INRECA

- Problem solving approaches using Intelligent Systems:
  - The next slides contain illustrations are based on very simple problem: input parameters are customer age and income and output parameter what product the person will buy (in case classification problem solving which product customer is likely to buy)

ANN Approach: Classification

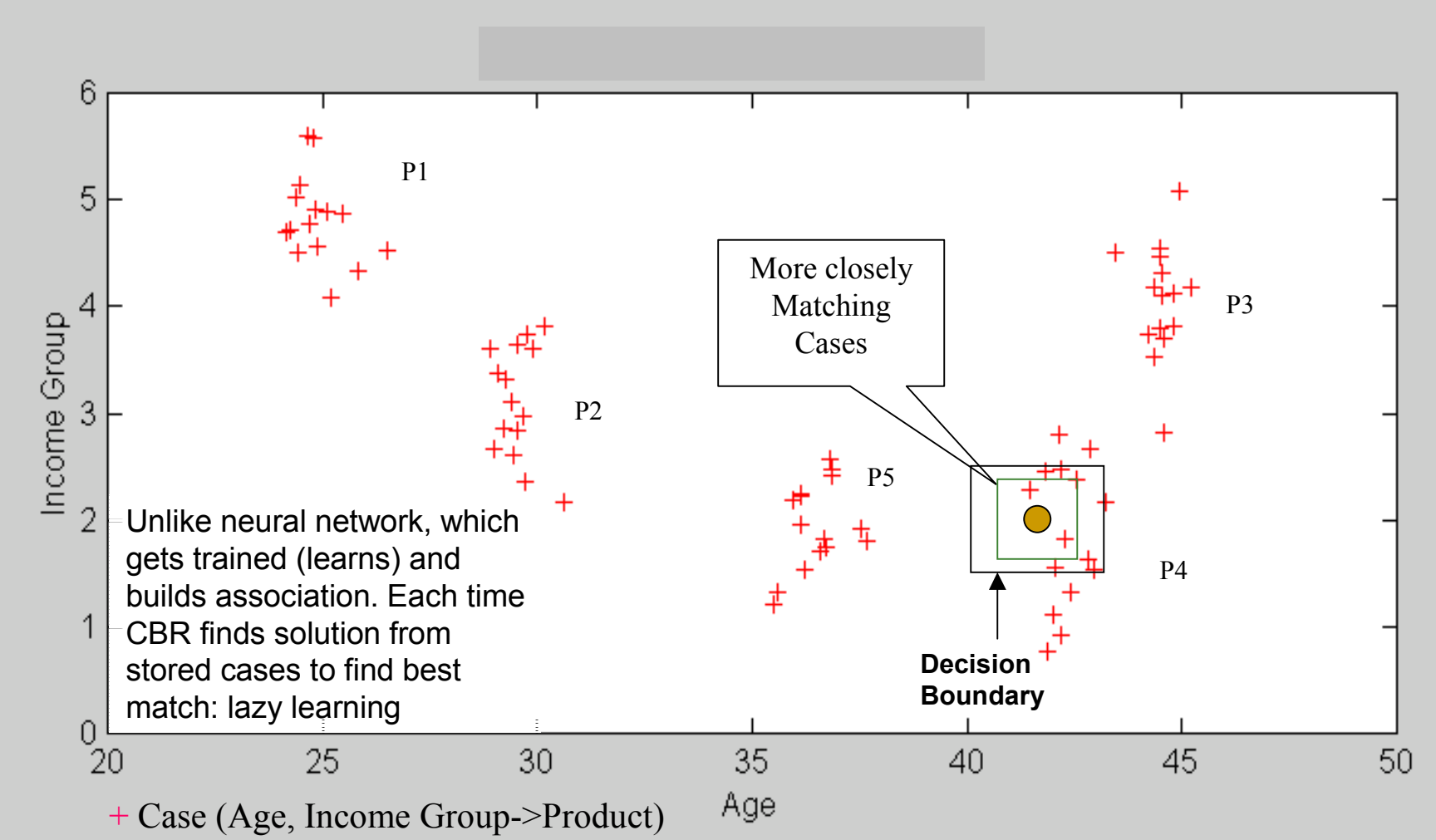


ANN Approach: Clustering

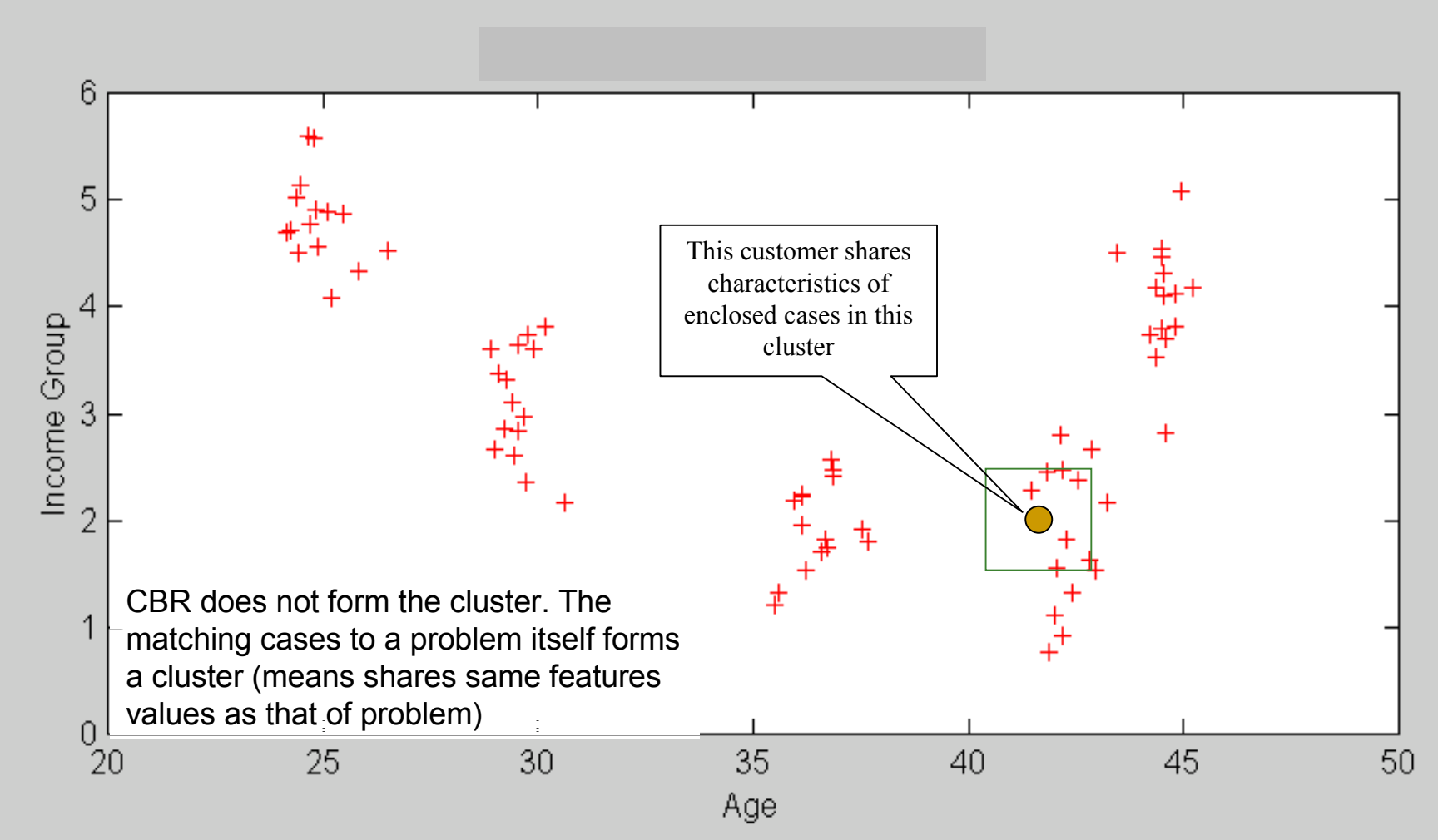


In clustering problems, neurons move into the centre of cluster. A new example belongs to cluster based on distance to central neuron.

CBR Approach: Classification

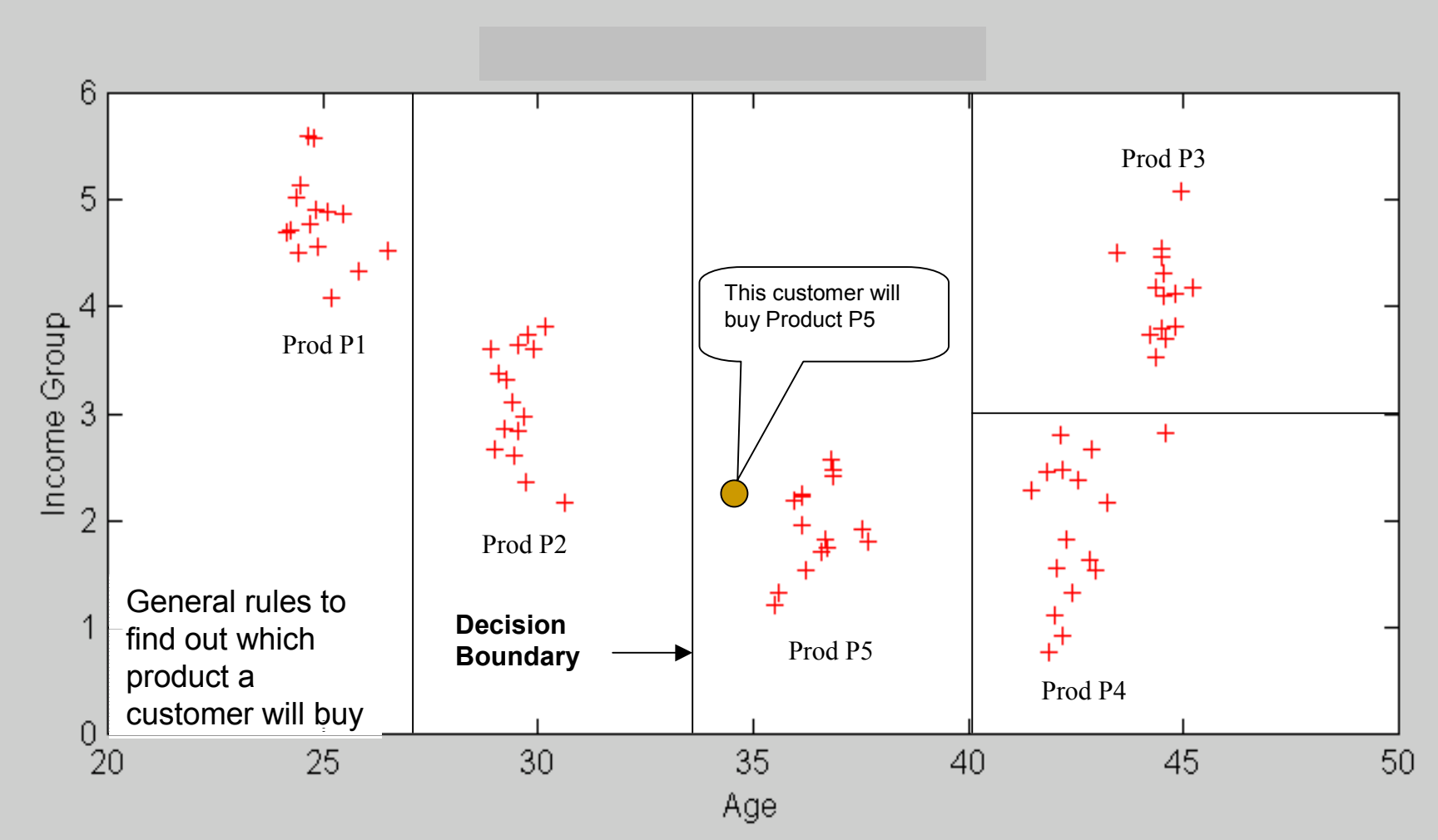


CBR Approach: Clustering

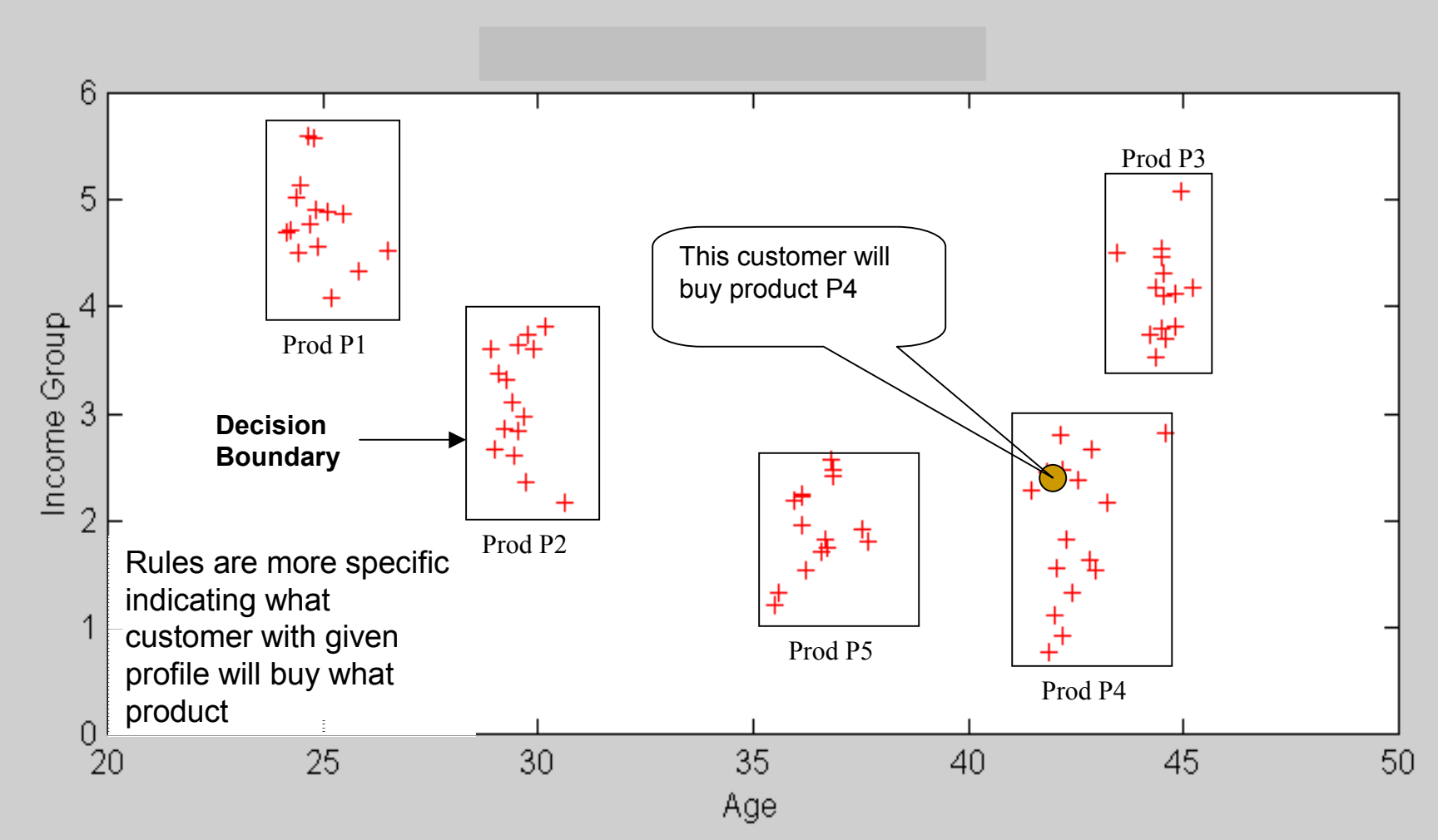


CBR does not form the cluster. The matching cases to a problem itself forms a cluster (means shares same features values as that of problem)

RBR: General Knowledge



RBR: More deeper knowledge



Features of Intelligent Systems

Intelligence Tool	Response Time	Scalability	Flexibility	Ease of Use	Embeddability	Processing Overhead	Expert Dependence	Tolerance for "Dirty" Data	Implementation Speed	Tolerance for Complexity	Accuracy
Genetic algorithm	H	L	L	L	H	H	L	—	H	H	H
Neural networks	H	M	L	L	M	L	L	H	M	H	H
Fuzzy logic systems	H	L	M	H	M	H	H	H	H	M	M
Rule-based systems	X	L	M	H	H	M	H	L	H	M	H
Case-based reasoning	L	H	L	M	M	H	L	H	M	H	H

Legend: H = High; L = Low; M = Medium; X = Deteriorates as the number of active rules grows.

Hybrid Intelligent Systems

Hybrid Intelligent Systems

- The systems in which more than one intelligent system has been used are called as *hybrid intelligent systems* or *intelligent hybrid systems*.
- Why?
  - The intelligent systems collectively have features like learning ability, adaptation to changes, explanation capability and flexibility in dealing with imprecise and incomplete information, etc.
  - No single intelligent system has all the features.
  - The limitations and strengths of individual systems is the central driving force behind the hybrid intelligent systems.
  - By integrating the systems their strengths can be increased and weaknesses can be reduced.

Reasons for creating Hybrid Systems

[Suran Goonatilake, sukhdev Khebbal, 1995]

- *Technique enhancement*: This is the integration of different techniques to overcome the limitations of each individual technique. Here the aim is to take a technique that has weakness in a particular property and combine it with a technique that has strength in that same property.

- *Multiplicity of application tasks*: When no single technique is available to the many sub-problems of a given application then this hybrid system is used.

- *Realizing multi-functionality*: These hybrid systems can exhibit multiple information processing capabilities within one architecture.

Types of Hybrids

[Suran Goonatilake, sukhdev Khebbal, 1995]

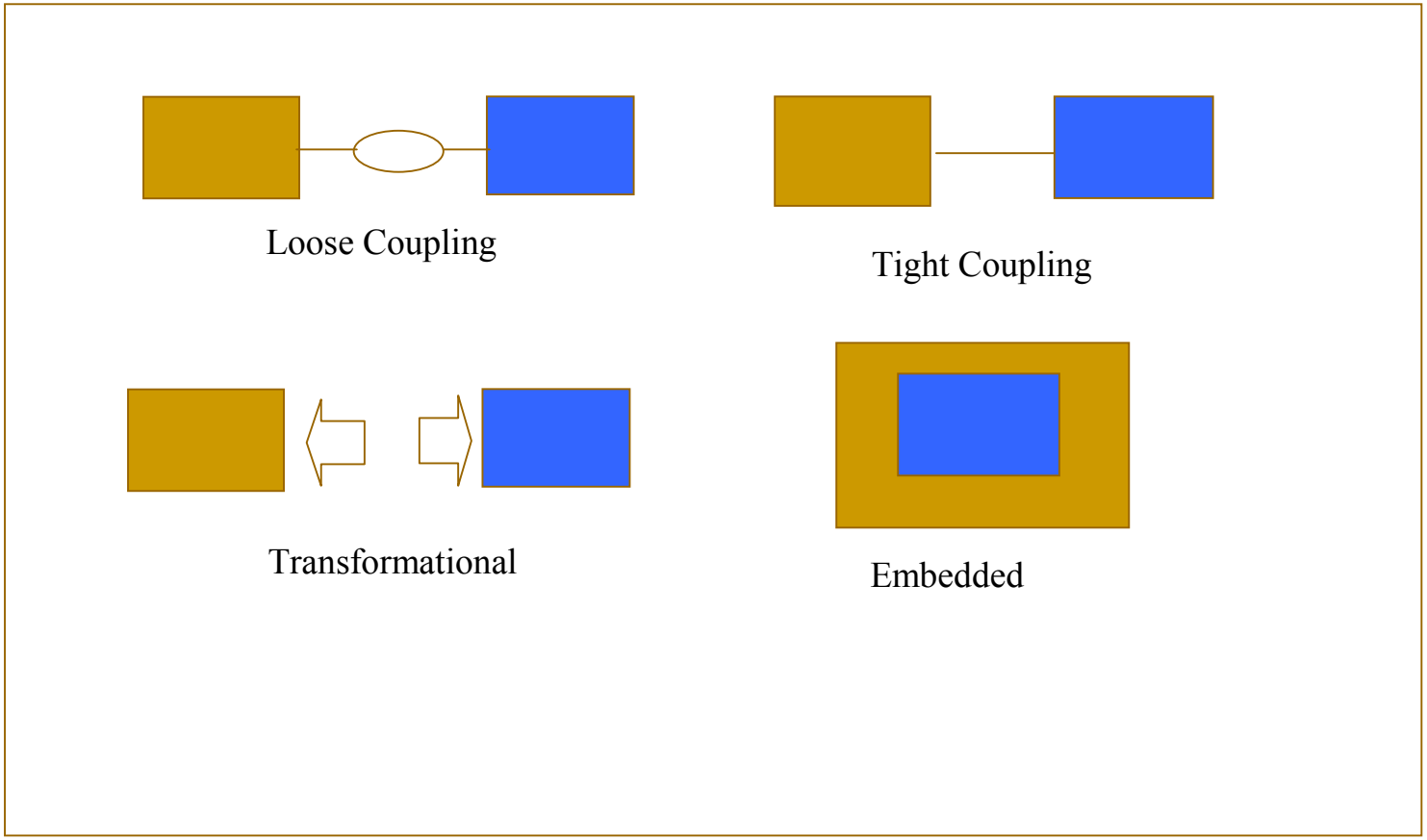
- *Function-Replacing hybrids* – In this system, a principal function of the given technique is replaced by another intelligent processing technique. It is done for either to increase execution speed or enhance reliability. The motivation for this approach is the *technique enhancement*.

- *Intercommunicating hybrids* – These are *independent*, self-contained, intelligent processing modules that exchange information and perform separate functions to generate solutions. If a problem can be subdivided into distinct processing tasks, then different independent intelligent modules can be used to solve the parts of the problem, which they are best. These independent modules, which collectively solve the given task, are coordinated by a *control mechanism*. This approach is motivated by multiplicity of application tasks.

- *Polymorphic hybrids* – These systems use a *single processing architecture* to achieve the functionality of *different* intelligent processing techniques. The broad motivation for these hybrid systems is *realizing multi-functionality* within particular computational architectures. These systems can *functionally mimic* or emulate different processing techniques. This is appropriate in situations whrethe desired functionality dynamically changes, this required the ability to a switch from one style of processing to another.

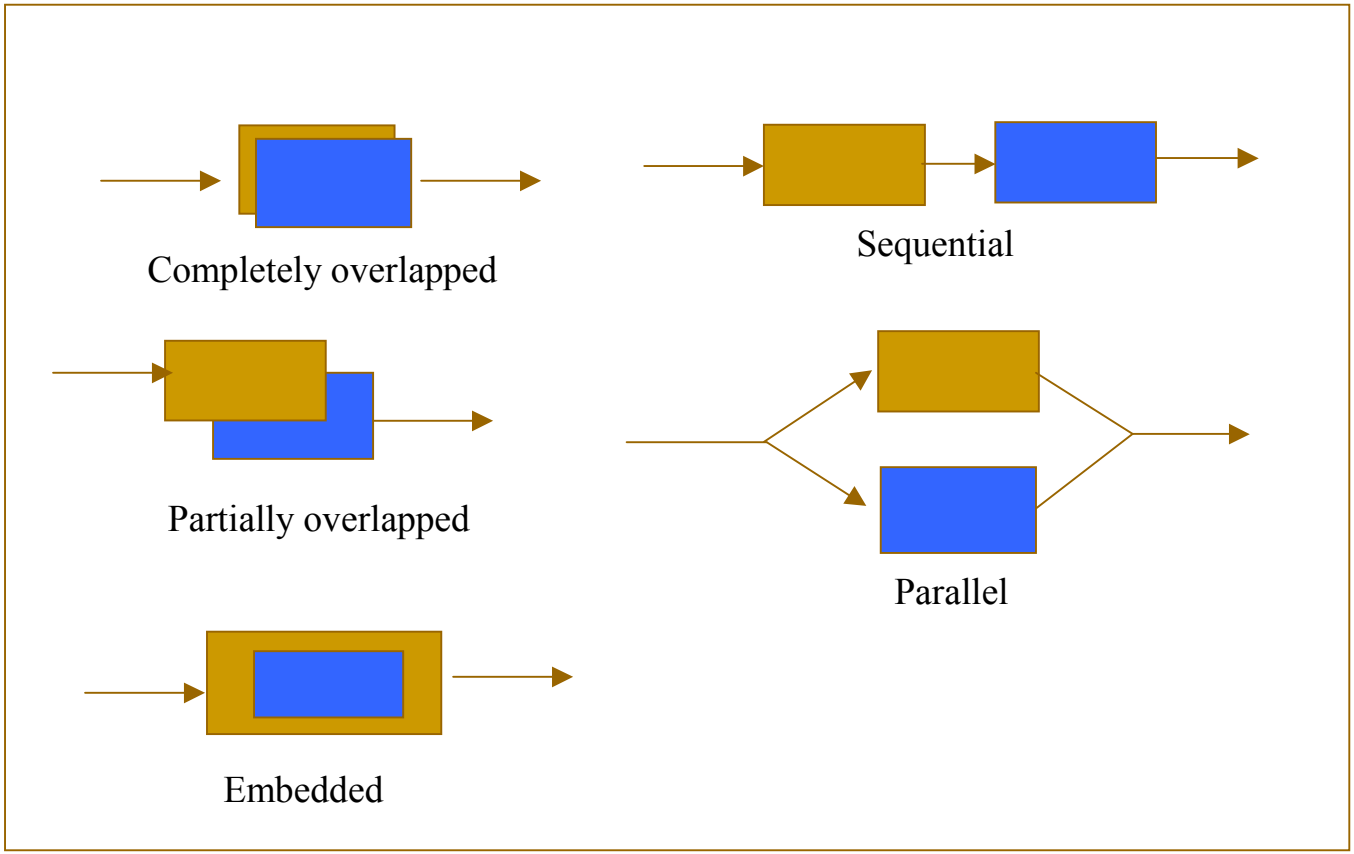
Hybrids architectures/models

[Medsker and Liebowitz, 1994]



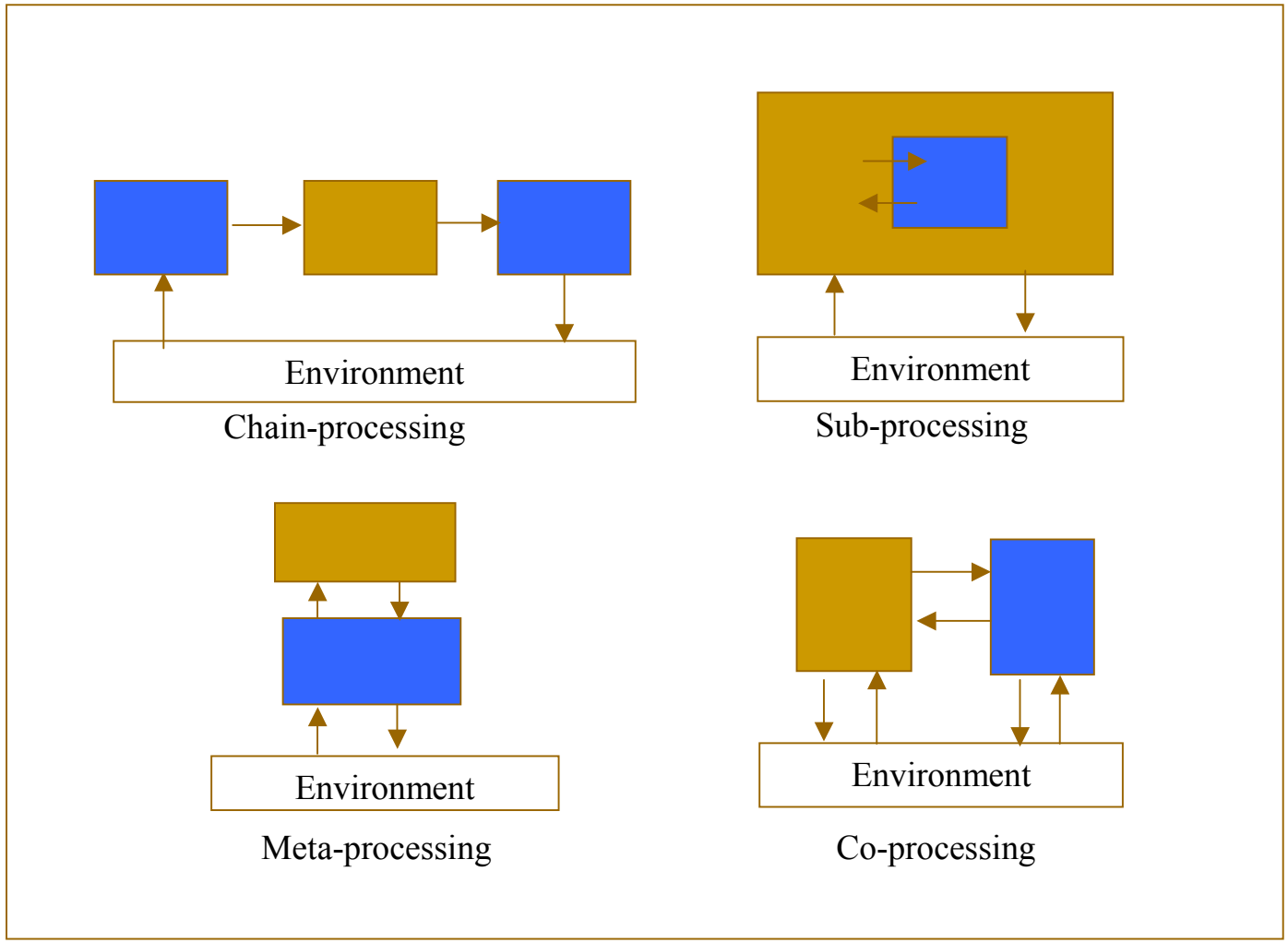
Hybrids architectures/models

[Fu, L.M., 1994]



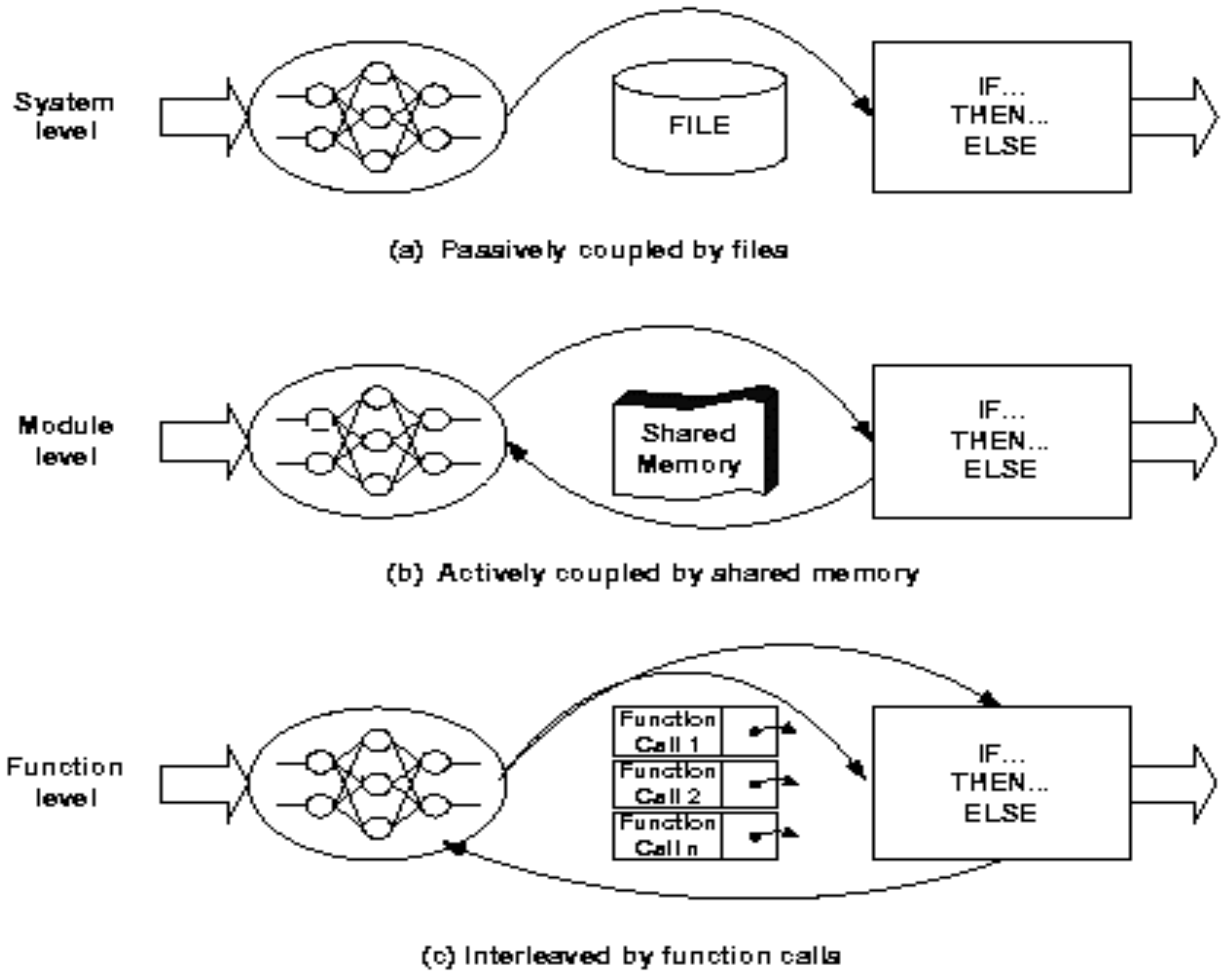
Hybrids architectures/models

[Hilario, et al 1994]

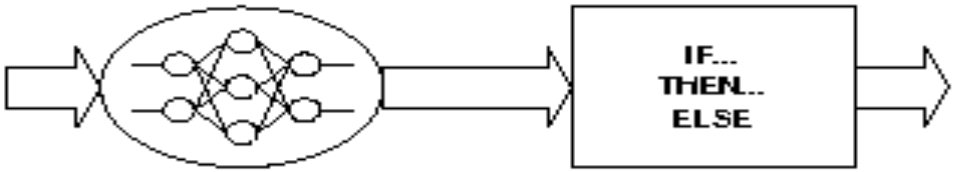


ES+ANN hybrids

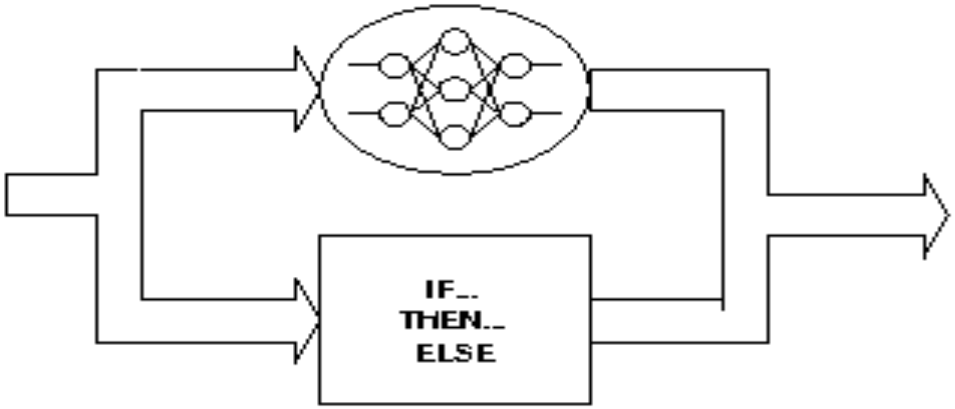
[Wermter and Sun, 2000]



ES+ANN hybrids

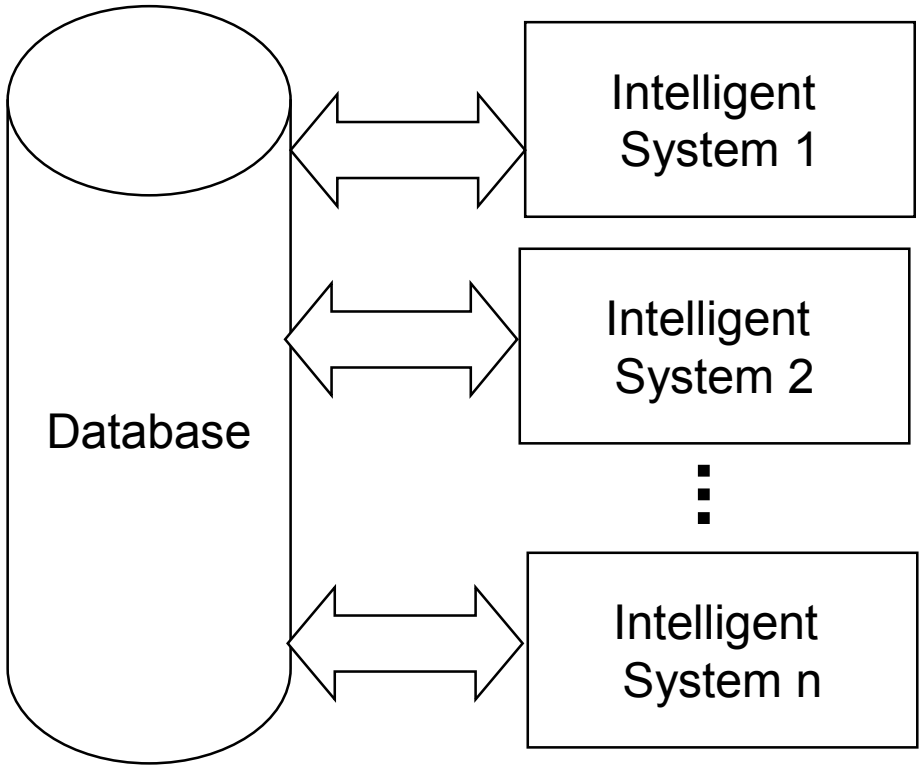


(a) Sequential Processing

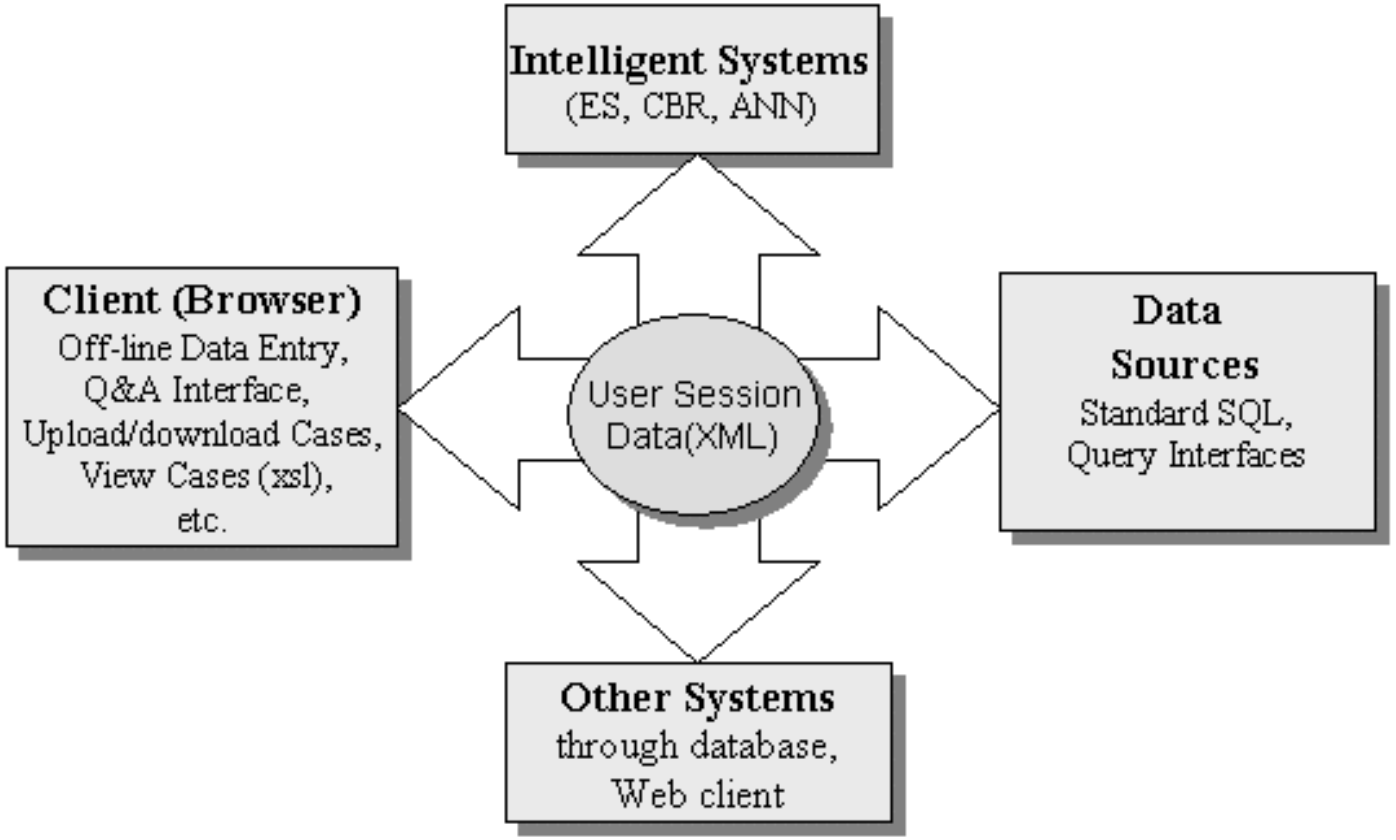


(b) Parallel Processing

Database coupling: database as an integrating and coordinating mechanism [Sonar, 1999, 2001]



Using XML coupling [Sonar, 2004]



An Integration Framework [Sonar, 2007]

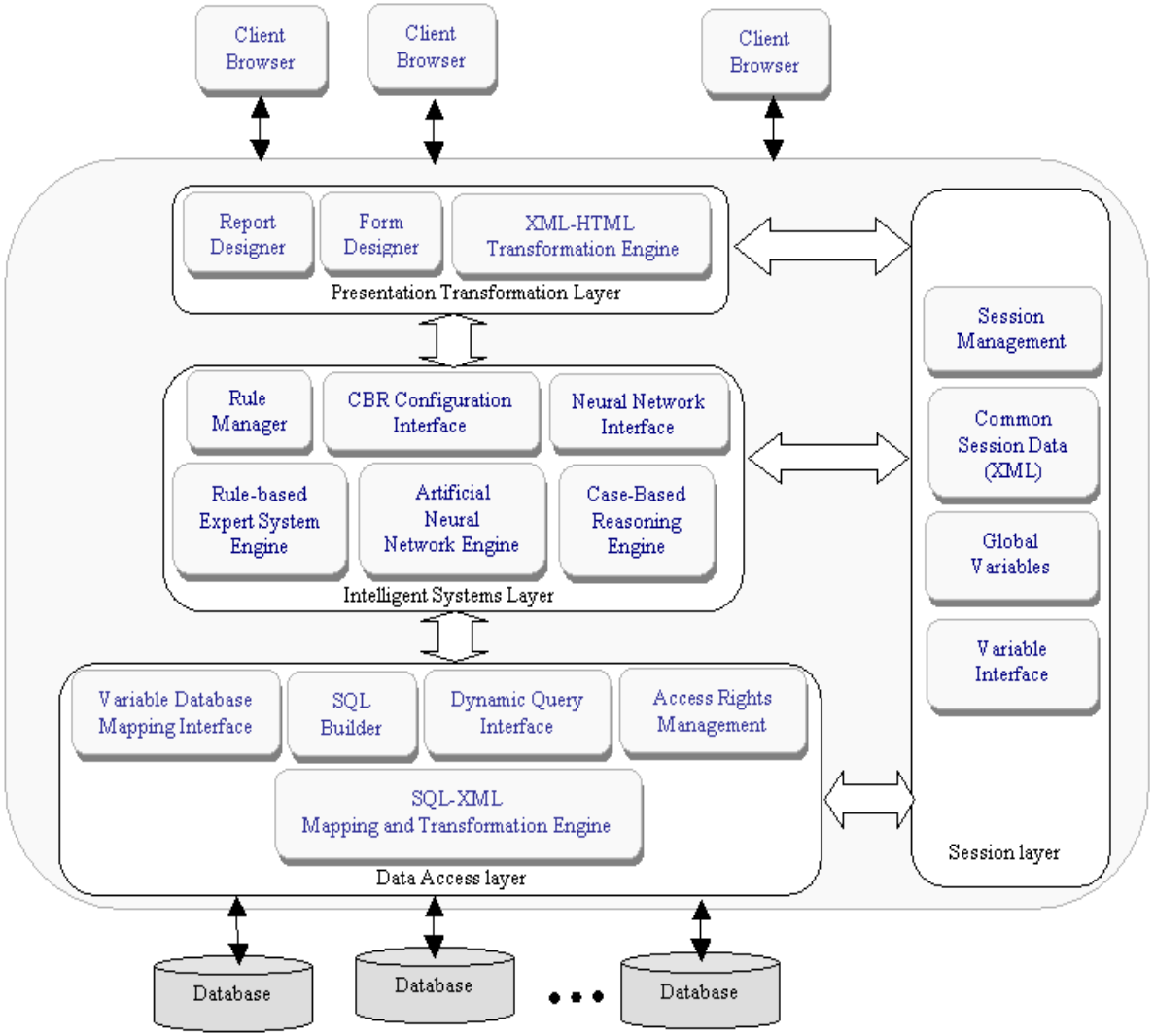


Fig. 1 Hybrid Framework

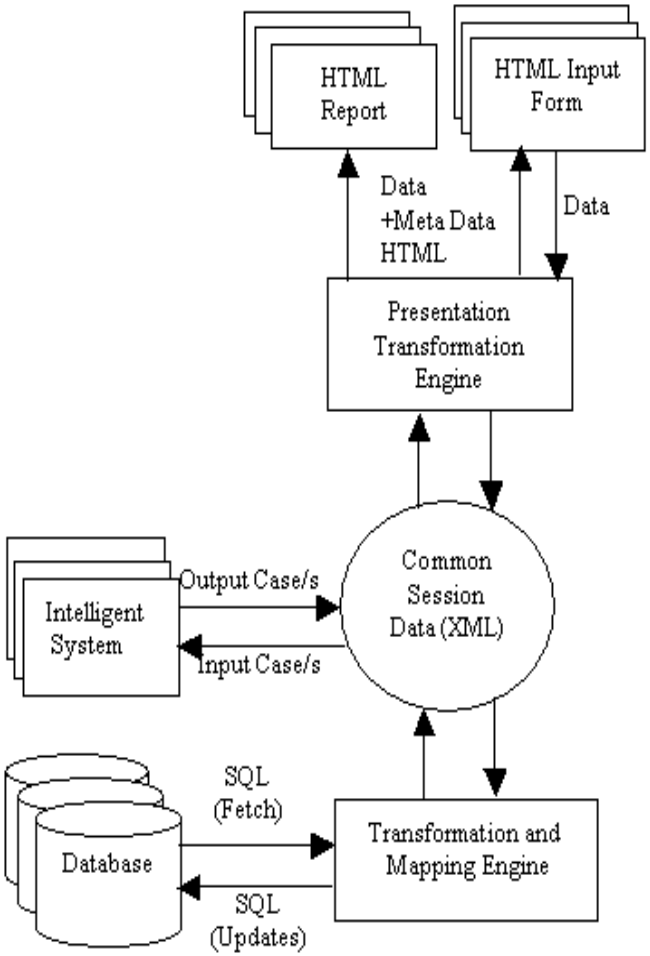


Fig. 3 Sharing Mechanism through Session Data

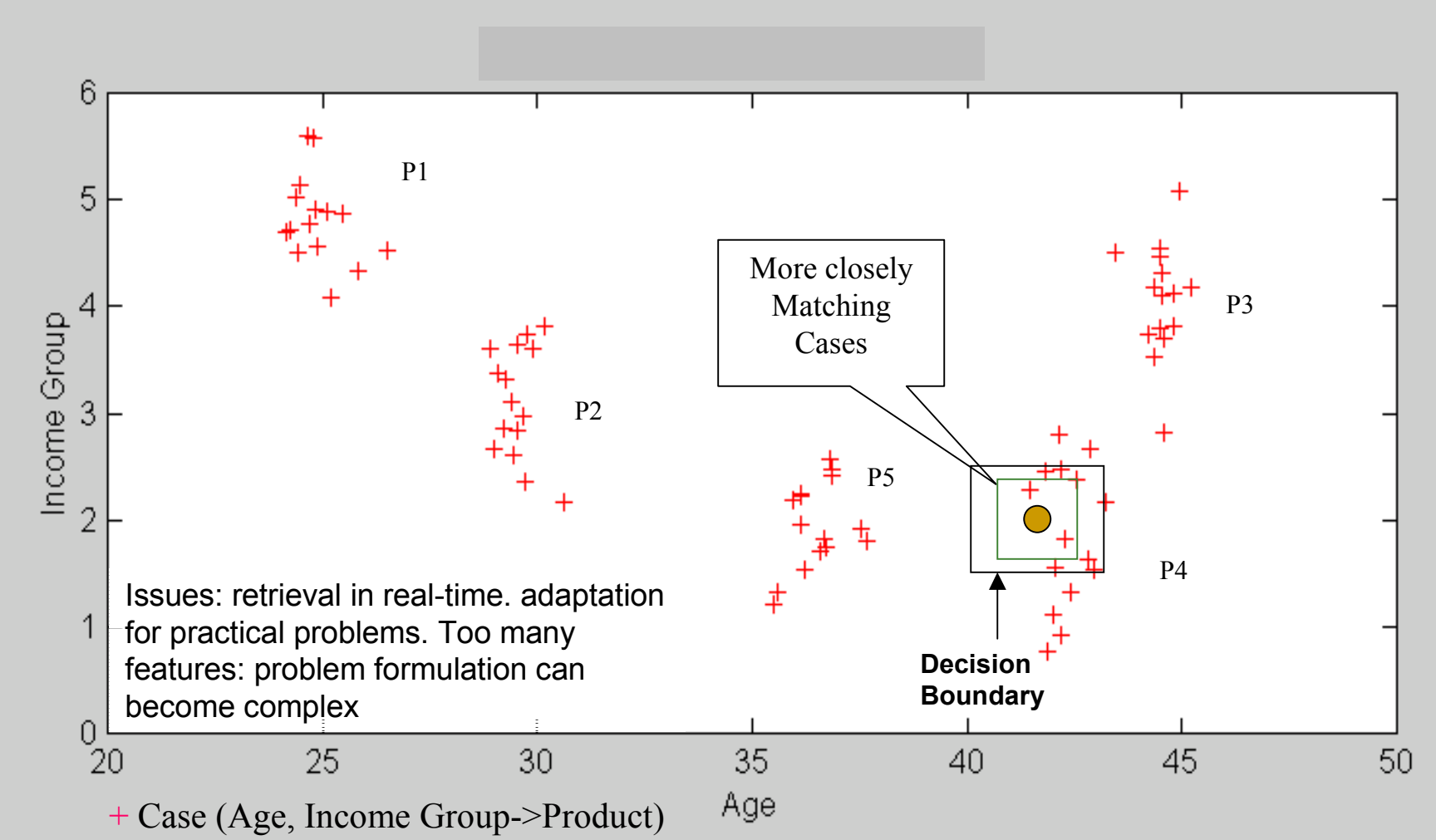
Example: Using Integrated Approach:

Expert System + CBR Approach

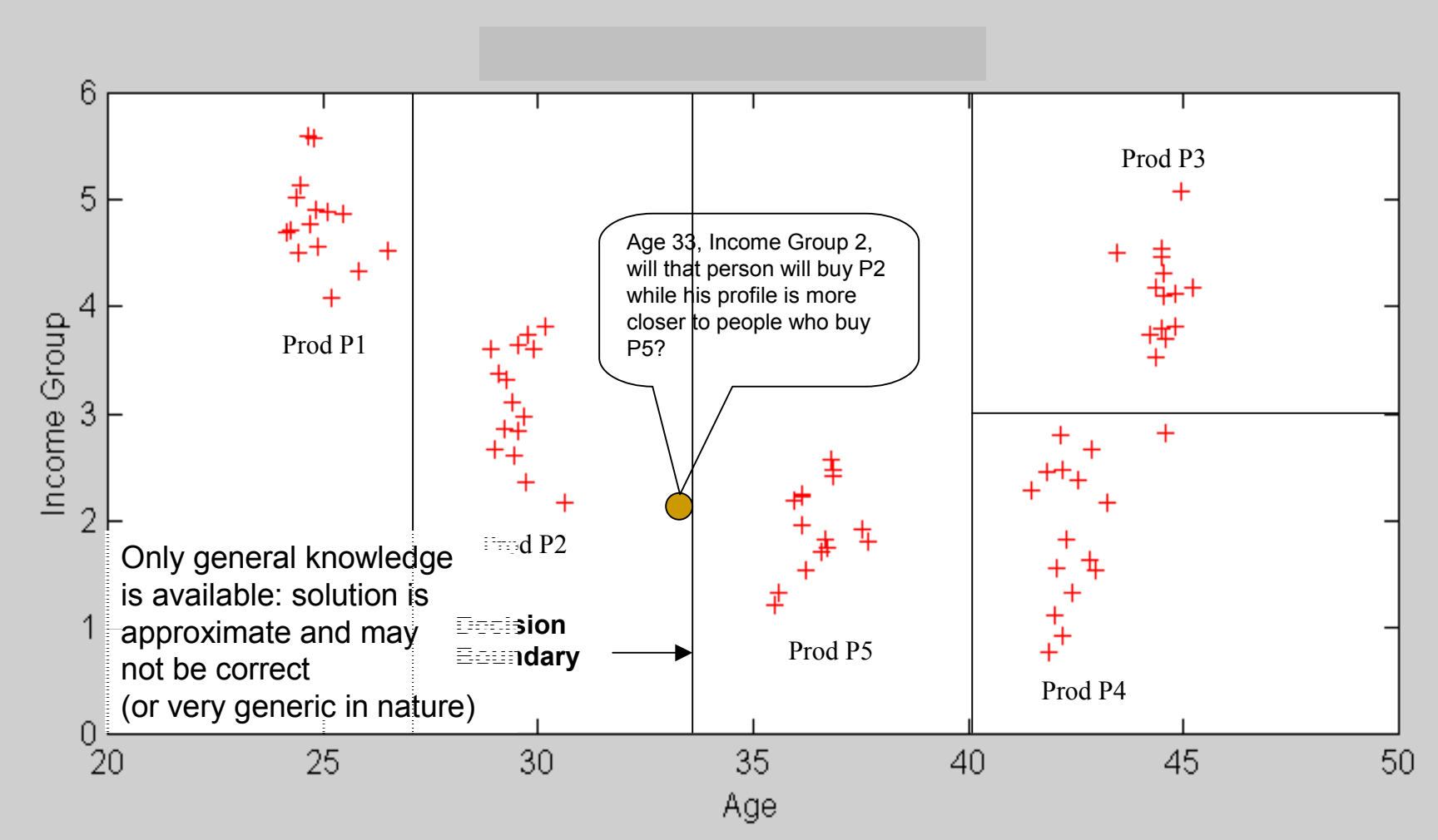
Why integrated RBR and CBR?

- RBR has explicit knowledge: expertise is automated
- CBR is more experience driven (by past examples/cases/data: implied knowledge) . Has explicit knowledge in domain vocabulary and to match cases
- RBR+CBR is combination of knowledge + experience (explicit +implied knowledge) which can address large number of problems.

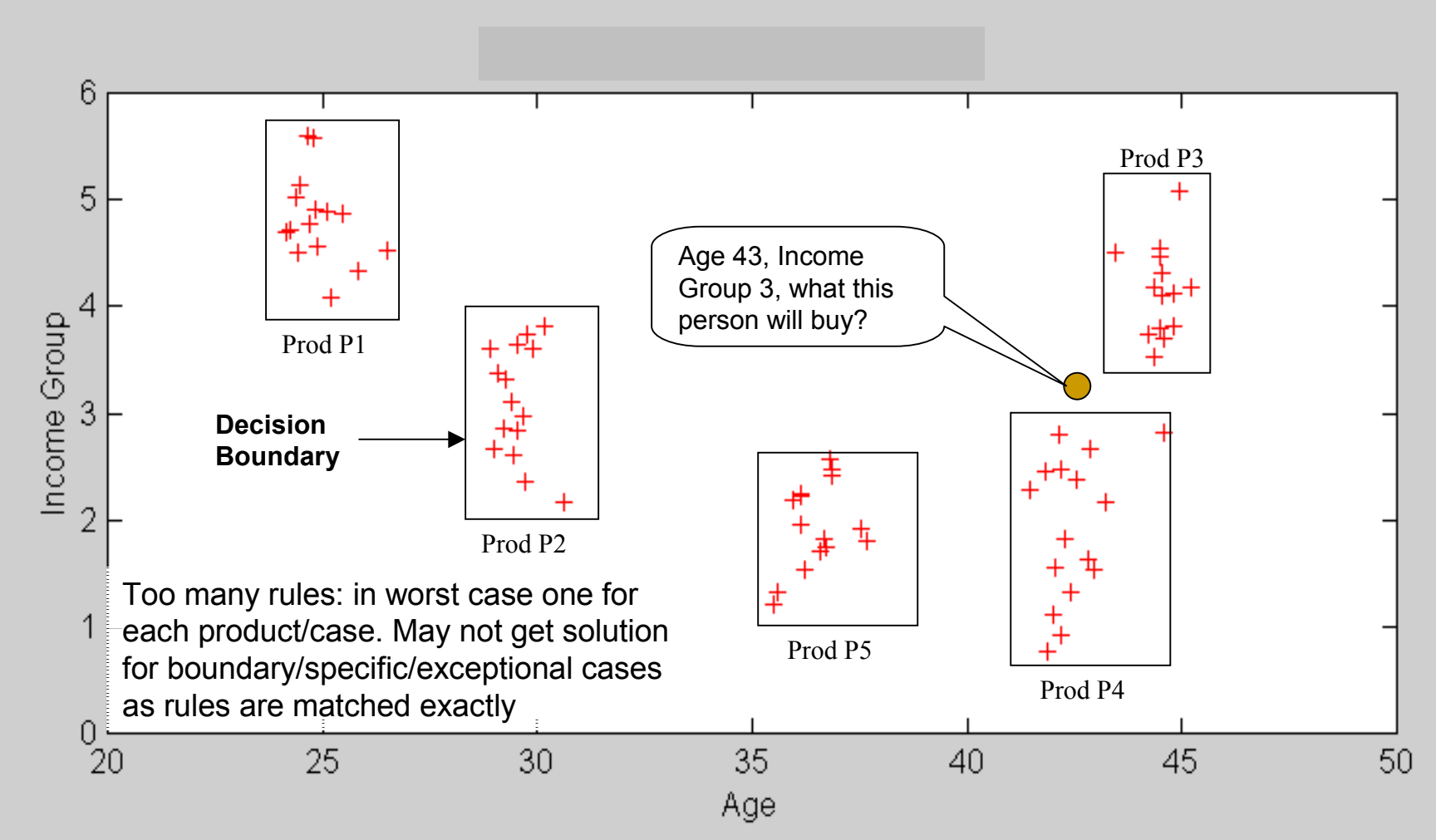
Only CBR Approach



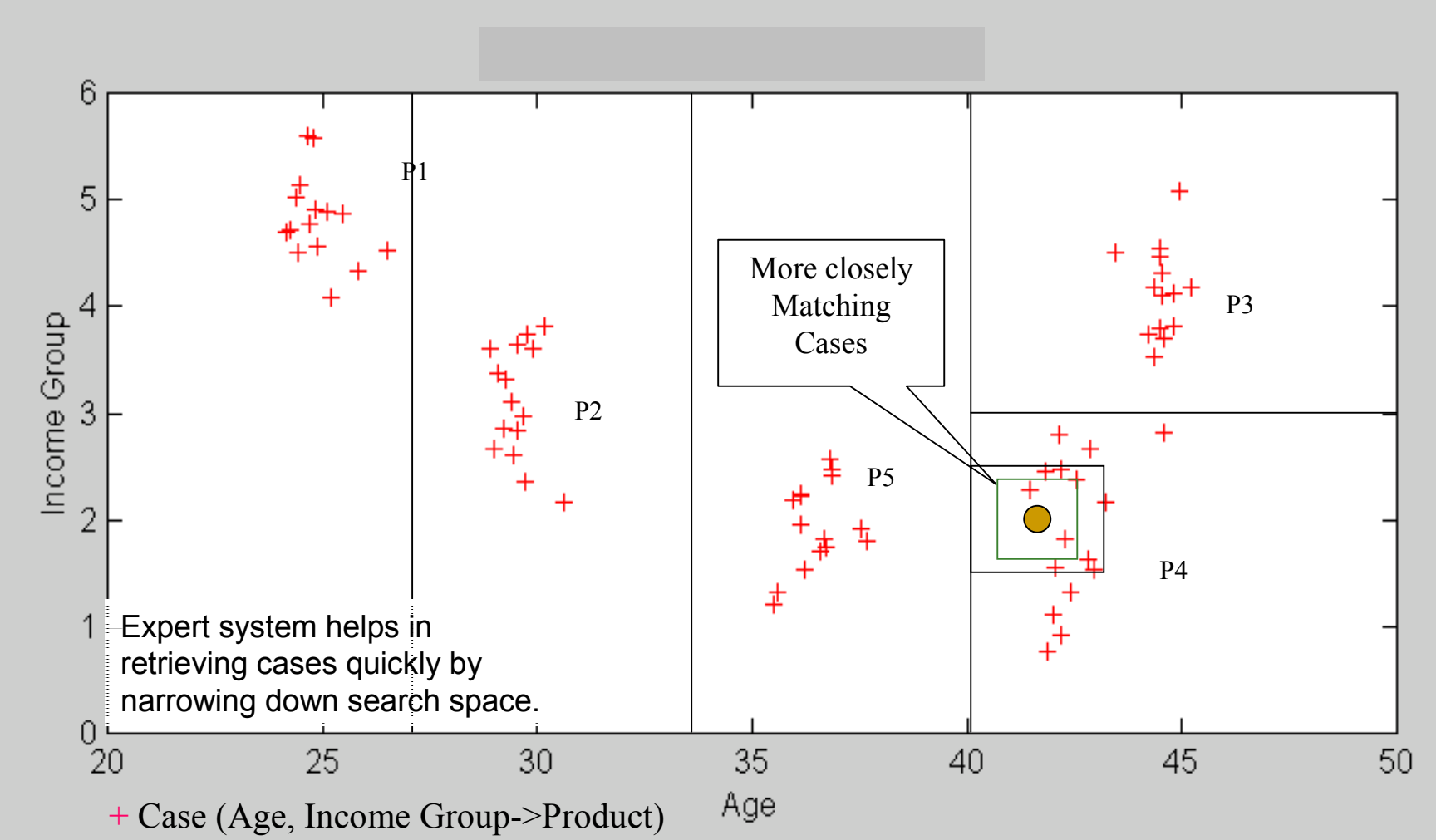
Only RBR: General Knowledge



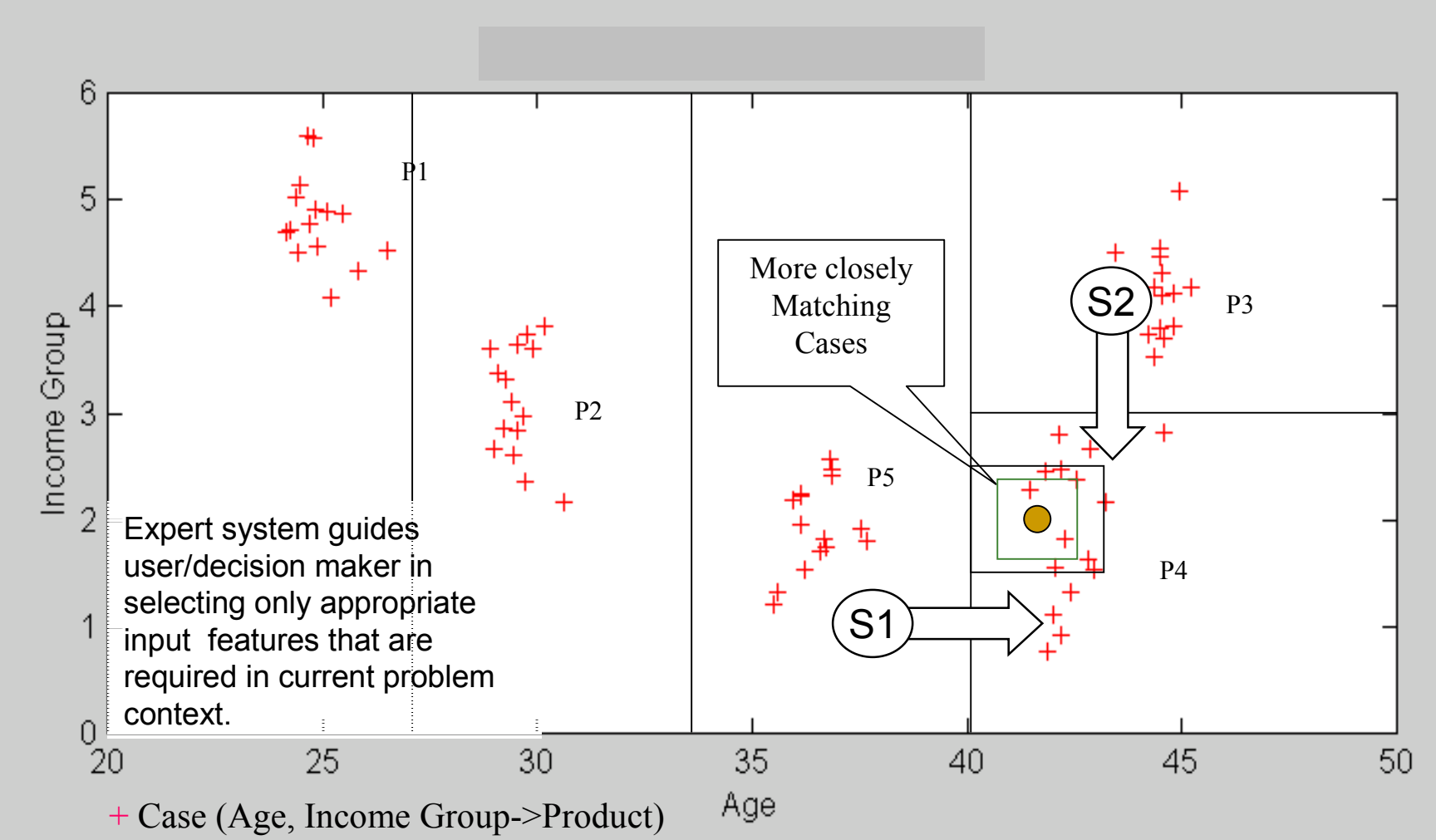
Only RBR: More deeper knowledge



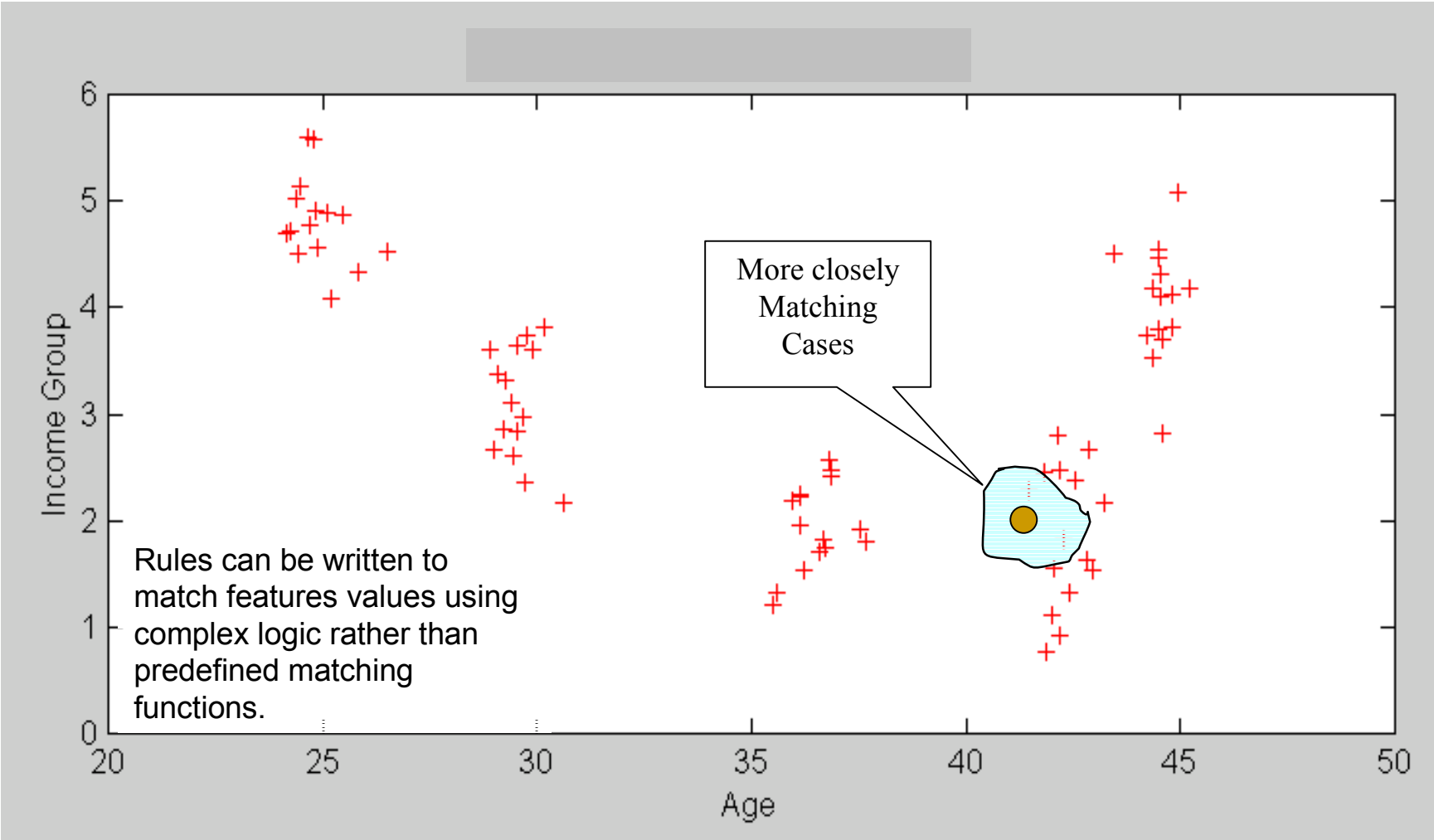
RBR (Indexing) + CBR



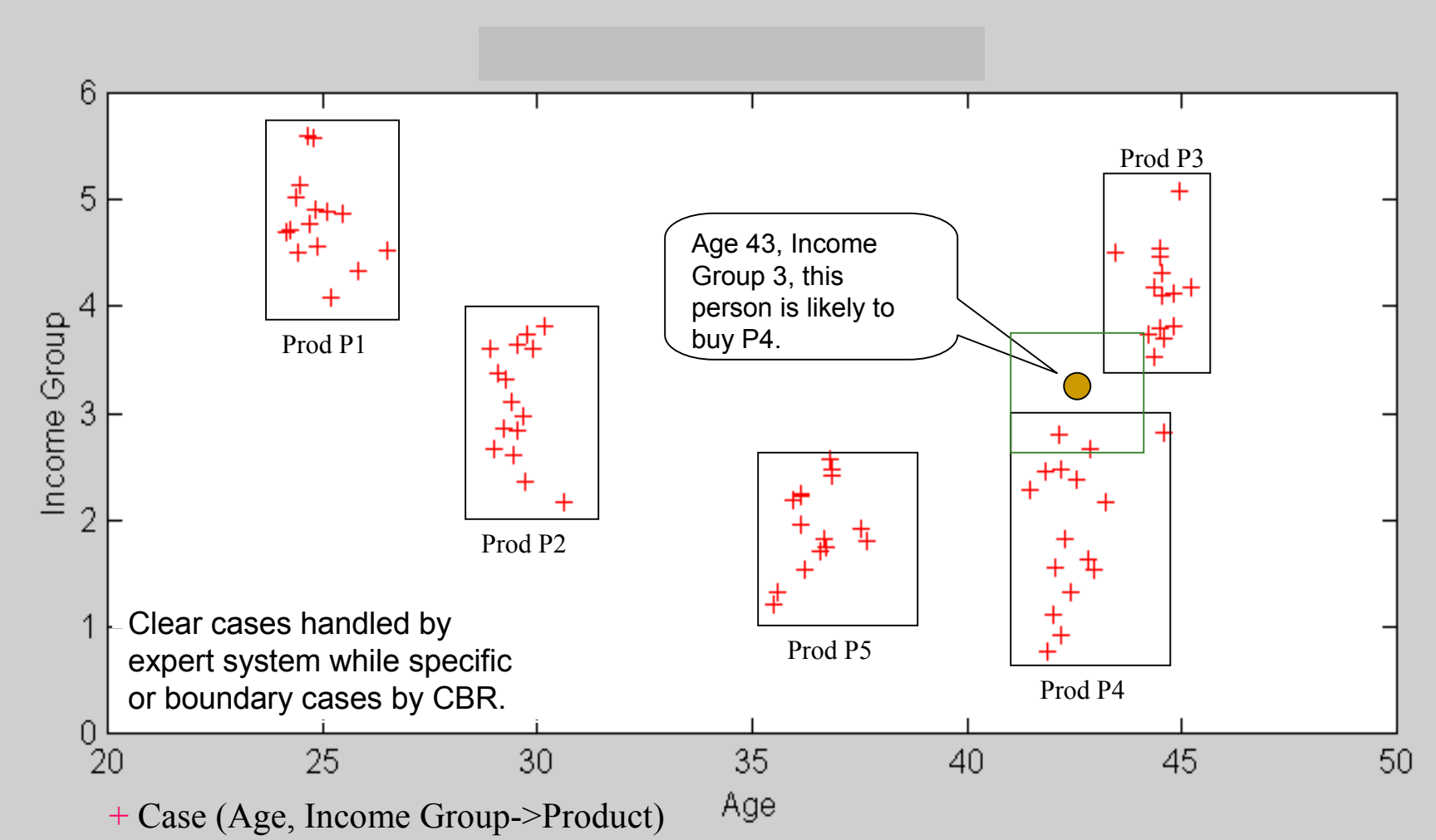
RBR (Path/Guide in Feature Selection) + CBR



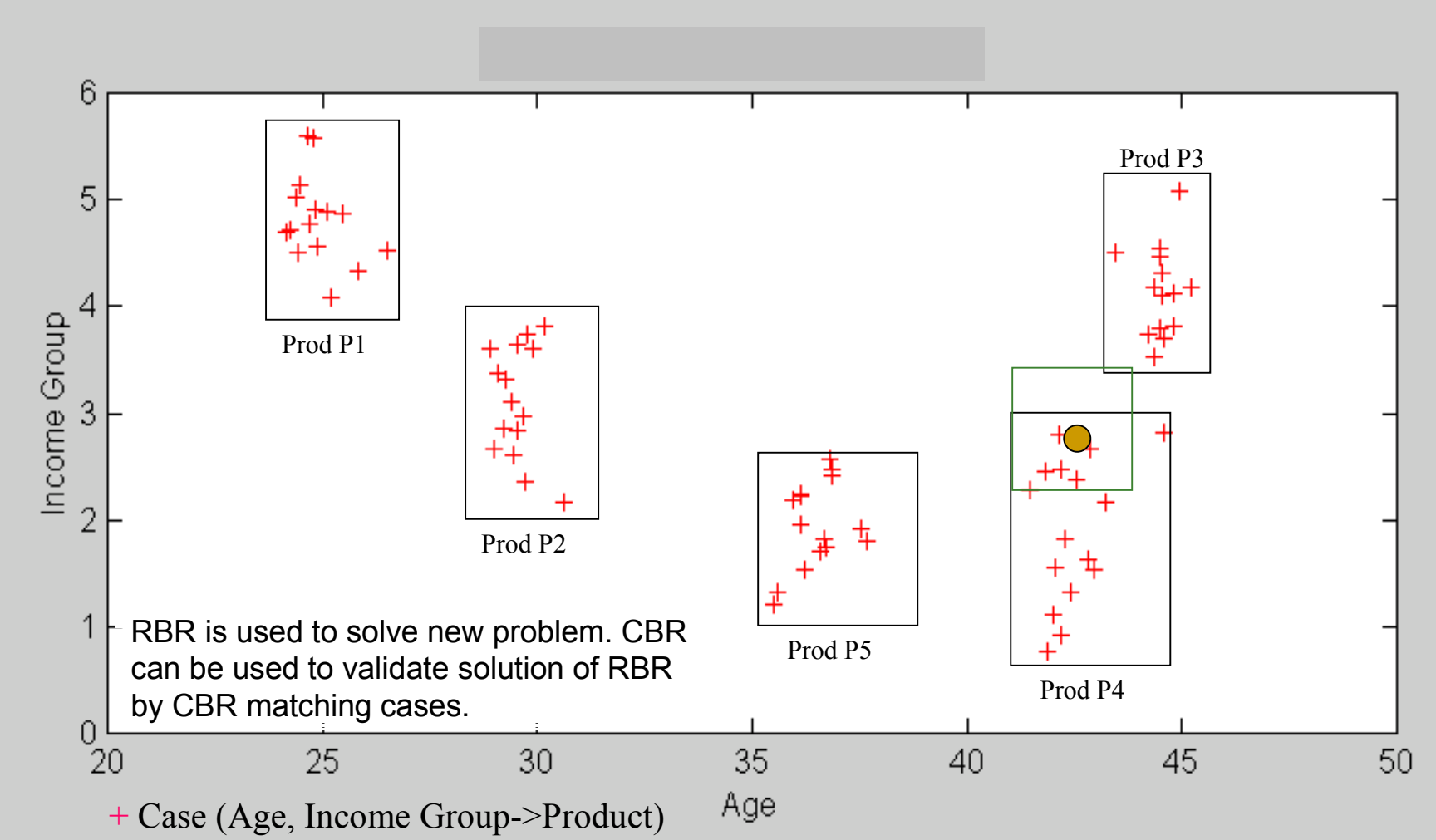
CBR+RBR (Rule-based Feature Matching)



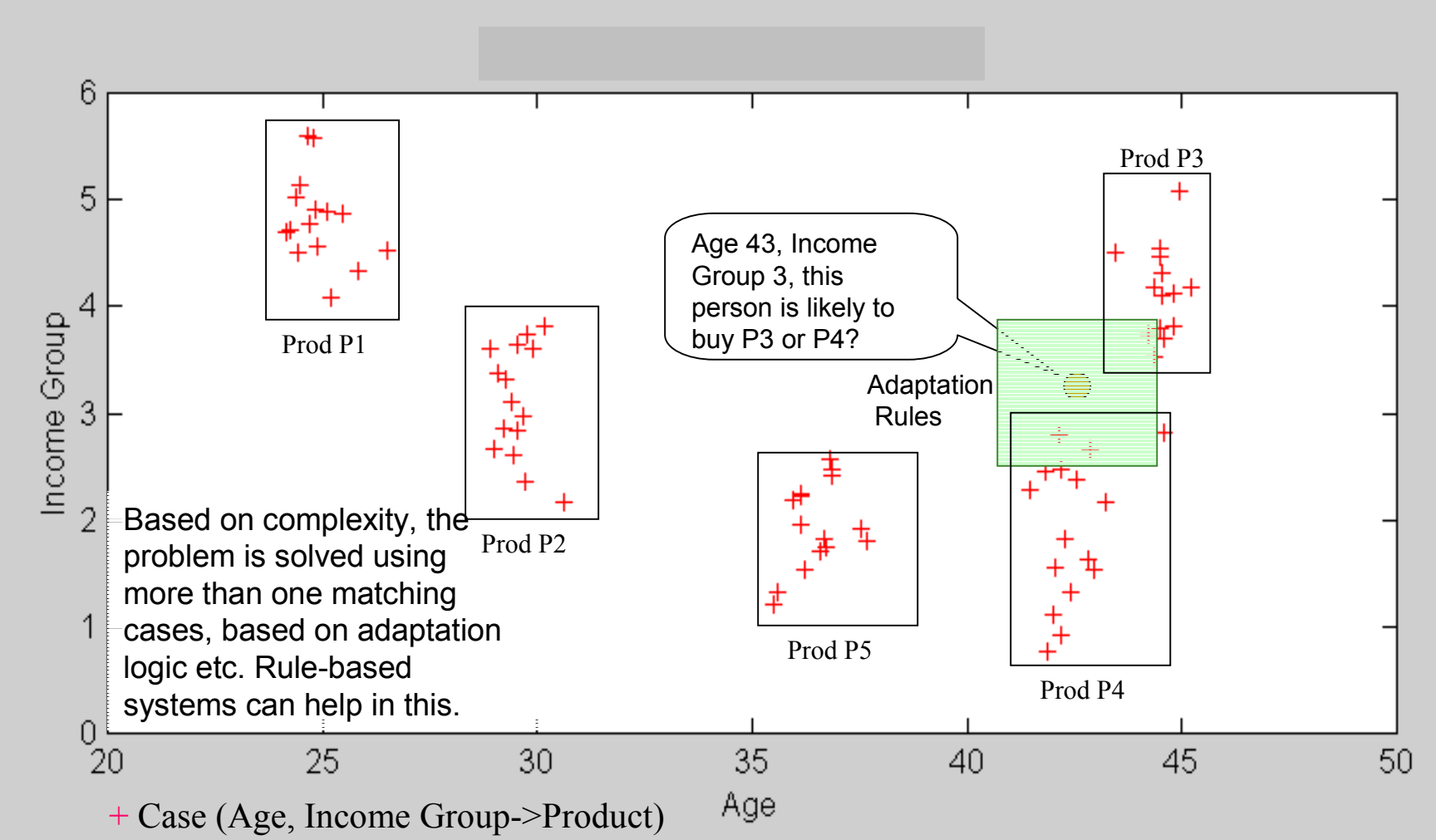
RBR (Clear Cases) + CBR (Specific/Boundary Cases)



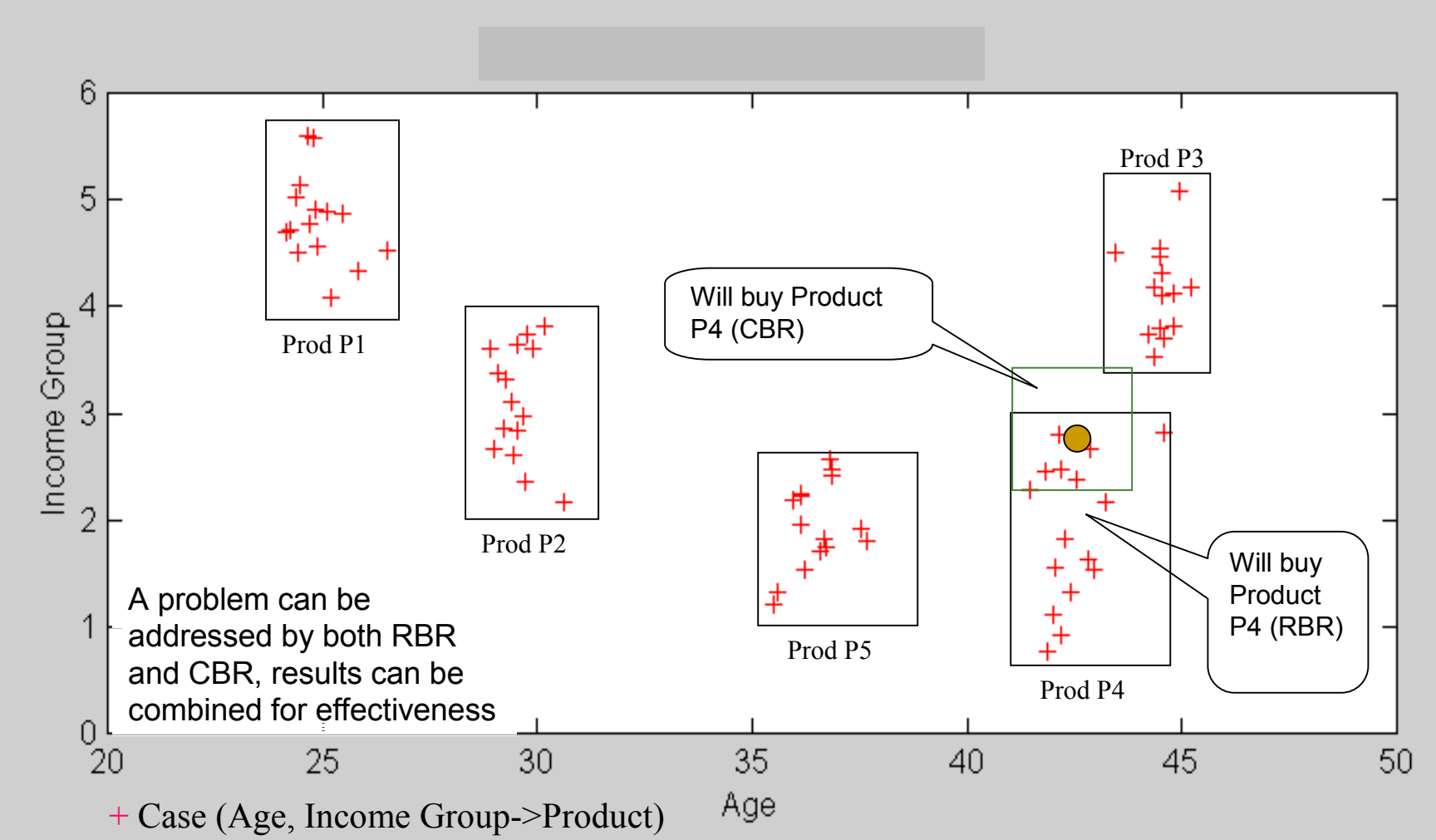
RBR + CBR (Validation)



CBR+RBR (Adaptation)



RBR || CBR


Income Group
Age
Prod P1
Prod P2
Prod P3
Prod P4
Prod P5
Will buy Product P4 (CBR)
Will buy Product P4 (RBR)
A problem can be addressed by both RBR and CBR, results can be combined for effectiveness
+ Case (Age, Income Group->Product)

References

- 1. Larry Medsker, *Hybrid Intelligent Systems.* Kluwer Academic Publishers, Boston, 1995.
- 2. Melanie Hilario,, Christian Pellegrini & Frederic Alexandre. Modular integration of connectionist and symbolic processing in knowledge-based systems. *International Symposium on Integrating Knowledge and Neural Hueristics,* pages 123-132, Pensacola, Florida, 1994.
- 3. Sonar, R.M, "An Enterprise Intelligent System Development and Solution Framework", *International Journal Of Applied Science, Engineering And Technology* Volume 4 Number 1 2007 ISSN 1307-4318
- 4. Sonar, R.M., "A Web-based Hybrid Intelligent System Framework", *Intelligent Systems and Control*, ACTA Press, 2004, pp. 254-259.
- 5. Sonar, R.M. and A. Saha, "An integration framework to develop modular hybrid intelligent systems", *Frontiers in Artificial Intelligence and Applications,* 69, IOS Press, 2001, pp.1499-1506.
- 6. Sonar, R.M. "Integrating intelligent systems using an SQL-database". *Expert Systems with Applications,* Vol.17 (1), July, 1999, 45-49.
- 7. Suran Goonatilake and Phillip Treleaven, editors. *Intelligent Systems for Finance and Business,* John Wiley and Sons, 1995.
- 8. Suran Goonatilake and Sukhdev Khebbal, editors. *Intelligent Hybrid Systems*, John Wiley and Sons, 1995.
- 9. Wermter Stefan and Ron Sun. Overview of Hybrid Neural Systems. In *Hybrid Neural Systems,1:13,* Springer, Heidelberg, New York, January 2000.