Case-based Reasoning

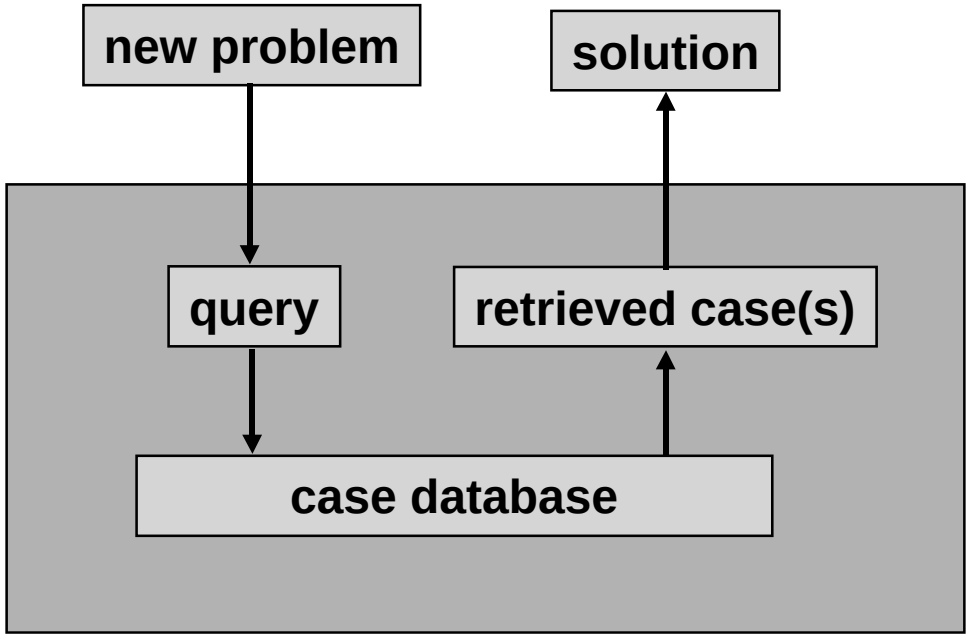
- Idea: experiences themselves are stored. These are called cases.
- Given a new example, the most appropriate case(s) in the knowledge base are found and these are used to predict properties of the new example.

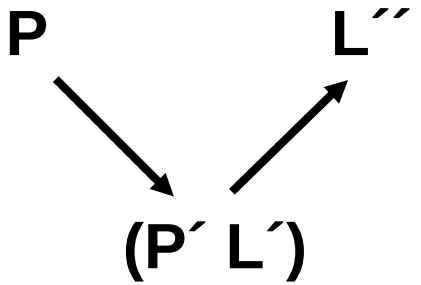
Extremes of Case-based Reasoning

- The cases are simple and for each new example the agent has seen many identical instances. Use the statistics of the cases.
- The cases are simple but there are few exact matches. Use a distance metric to find the closest cases.
- The cases are complex, there are no matches. You need sophisticated reasoning to determine why an old case is like the new case.

  Examples: legal reasoning, case-based planning.

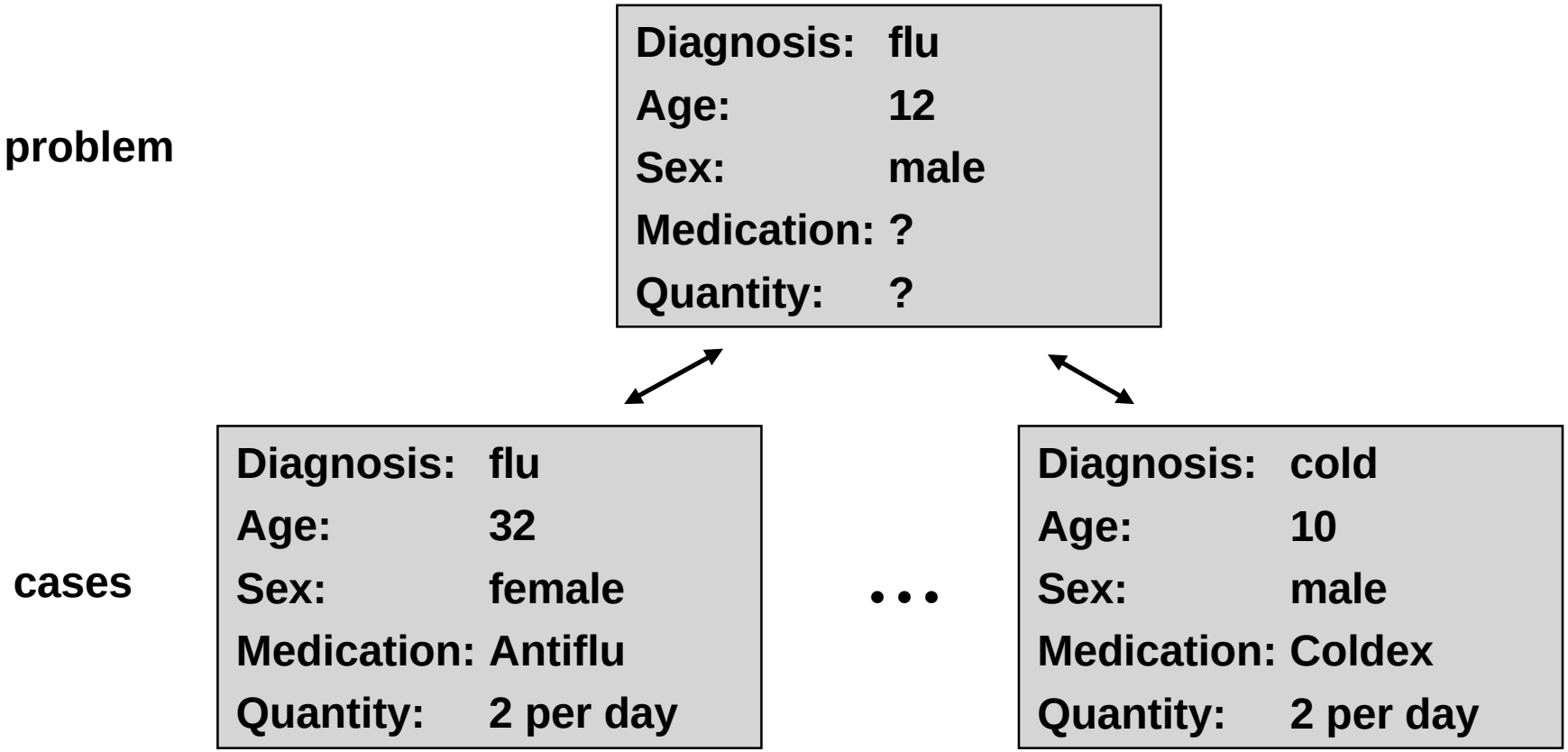
Principle of Case-based Problem Solving





- store problems together with their solutions in case database
- query database with new problems to find matching cases
- adapt solution of retrieved case to new problem

Case-Based Reasoning Example



k-nearest Neighbors

- Need a distance metric between examples.
- Given a new example, find the *k* nearest neighbors of that example.
- Predict the classification by using the mode, median, or interpolating between the neighbors.
- Often want $k > 1$ because there can be errors in the case base.

Euclidean Distance

- Define a metric for each dimension (convert the values to a numerical scale).
- The Euclidean distance between examples x and y is:

$$d(x, y) = \sqrt{\sum_A w_A (x_A - y_A)^2}$$

  - x_A is the numerical value of attribute A for example x
  - w_A is a nonnegative real-valued parameter that specifies the relative weight of attribute A.

kd-tree

- Like a decision tree, but examples are stored at the leaves.
- The aim is to build a balanced tree; so a particular example can be found in $\log n$ time when there are n examples.
- Not all leaves will be an exact match for a new example.
- Any exact match can be found in $d = \log n$ time
- All examples that miss on just one attribute can be found in $O(d^2)$ time.