MLC++
Machine Learning Library in C++

Ronny Kohavi

ronnyk@sgi.com

Data Mining and Visualization Group
Silicon Graphics, Inc.

- **http://www.sgi.com/Technology/mlc**

MLC++ : What were the Goals?

MLC++ is a Machine Learning library in C++.
Our original goals were:

- Allow comparisons of algorithms on different datasets (A1, A2, ..A20 on D1, D2, ..D20). This includes good accuracy estimates, learning curves, and statistics.
- Allow visualizations of learned models (e.g., show the decision trees) and target concepts.
- Allow development of variants, hybrid and meta algorithms (voting, stacking).

Comparisons

- Too many papers show an idea, then claim that it is great because performance is better than a previous algorithm on two out of three datasets (two artificial, one real–world).

  We want large scale comparisons!

- The "no–free–lunch" theorems show that no algorithm can dominate all others.

  ~~my algorithm is better~~

  My algorithm is better for domain D7, D8.

  For a given domain, "test drive" different algorithms.

Visualization and Comprehension

- To succeed in learning we must have some bias in the learning algorithms. Many times humans can provide this bias (e.g., this decision tree node doesn't make sense).

- To utilize our background knowledge, we need to understand the results of the learning algorithms.

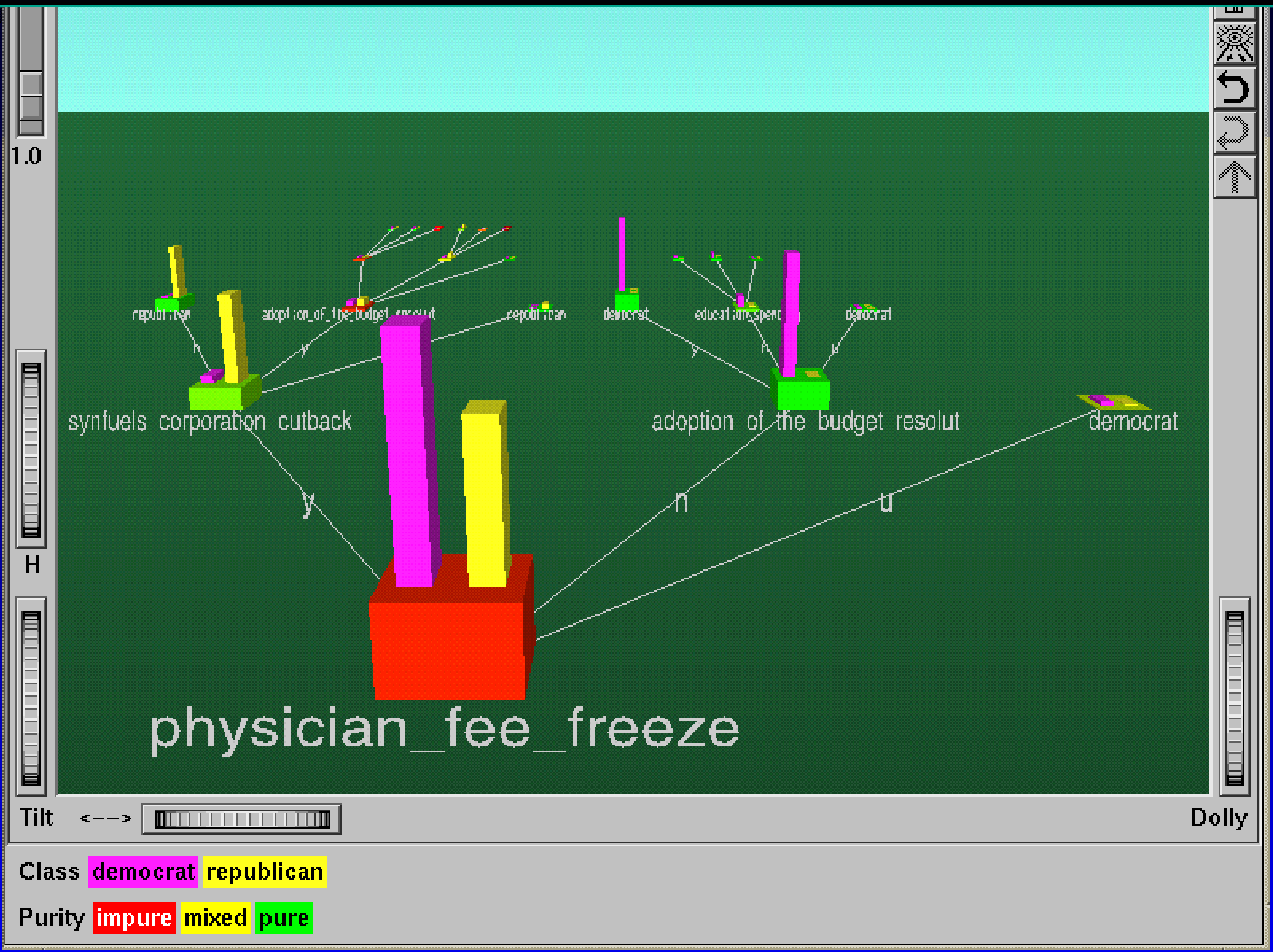
  Visualization is crucial in many cases

- Black–box approaches fail too many times.

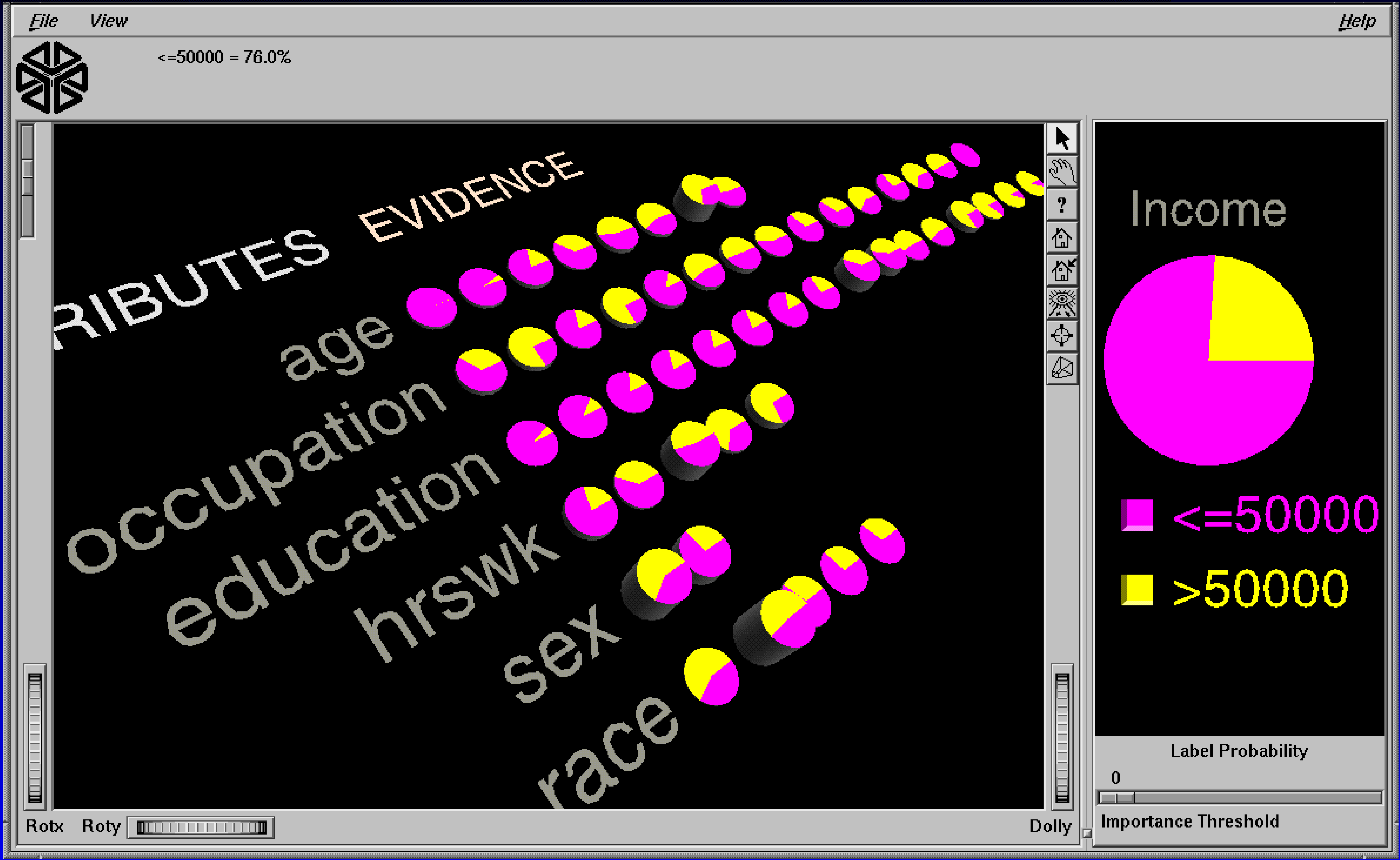
We Want to Avoid

```
physician fee freeze = n:
|   adoption of the budget resolution = y: democrat (151.0)
|   adoption of the budget resolution = u: democrat (1.0)
|   adoption of the budget resolution = n:
|   |   education spending = n: democrat (6.0)
|   |   education spending = y: democrat (9.0)
|   |   education spending = u: republican (1.0)
physician fee freeze = y:
|   synfuels corporation cutback = n: republican (97.0/3.0)
|   synfuels corporation cutback = u: republican (4.0)
|   synfuels corporation cutback = y:
|   |   duty free exports = y: democrat (2.0)
|   |   duty free exports = u: republican (1.0)
|   |   duty free exports = n:
|   |   |   education spending = n: democrat (5.0/2.0)
|   |   |   education spending = y: republican (13.0/2.0)
|   |   |   education spending = u: democrat (1.0)
physician fee freeze = u:
|   water project cost sharing = n: democrat (0.0)
|   water project cost sharing = y: democrat (4.0)
|   water project cost sharing = u:
|   |   mx missile = n: republican (0.0)
|   |   mx missile = y: democrat (3.0/1.0)
```

We Want to See



And This (for Naive–Bayes)



Development of New Algorithms

Newton said he saw farther because he stood no the shoulders of giants. Computer programmers stand on each other's toes
–– James Coggins

- As we understand learning algorithms better, we can tailor them to specific scenarios and conditions.
- MLC++ gives you the ability to write new algorithms faster and in a reliable manner.
- Hybrids and meta–algorithms are easy to write and test.

MLC++ Utilities

- The utilities are programs that use the library and provide a useful high–level function. Precompiled and geared at most "end users."

Examples:

– Inducer utility allows you to train and test.

– PerfEst allows you to estimate performance using cross validation or bootstrap.

– LearnCurve generates a learning curve (performance versus number of instances).

– BiasVar shows the bias–variance

MLC++ Utilities (cont)

- Each utility can run on any learning algorithm (inducer), including: decision trees, nearest neighbors, naive–Bayes, OODG, 1R, Perceptron, Winnow.
- Wrapper algorithms allow meta–learning and hybrid algorithms.
  Examples: feature selection, discretization, "auto"–tuning of parameters, decision trees with naive–Bayes at the leafs (NBTree).

MLC++ for Developers

- Original source code developed at Stanford is public domain.
- Enhancements at SGI (10 man years) are research domain (free for research purposes, cannot be commercialized).
- Over 100,000 lines of very tight code, 40,000 lines of regression tests.
  Utilities are 5,000 lines of code using the library.

Silicon Graphics' MineSet™

- MineSet is Silicon Graphics' data mining product. The analytic components are all based on MLC++.
- The MLC++/MineSet interface is simple and is about 4,000 lines of code. The file reader is replaced with "datamove," which connects to DBs and MineSet flat files.

Summary

MLC++ serves three different purposes:

- **Provides analytical engine for MineSet and could provide similar engine for other products (e.g., databases, vertical apps).**
- **Compiled utilities allow fast comparisons and evaluations for research at academia.**
- **The library itself allows R&D of new algorithms.**