Assignment #2

Date Assigned: Monday, Feb. 23, 1998
Date Due: Tuesday, Mar. 17, 1998

---

1. Give examples (one each) of data mining problems where

    1. HOODG performs better than ID3
    2. PROGOL performs better than ID3
    3. ID3 performs better than PROGOL
    4. A feed forward neural network is best

    "Performs better" can be taken to refer to any performance criteria such as simplicity, interpretability, accuracy etc., but please state which one(s) you are referring to.

2. Consider the IRIS data set. Select 3-4 algorithms from MLC++ (other than ID3) to classify this data set and perform first order stacking with them, this time using ID3 to classify the newly formed instances. Visualize the tree produced by ID3 and determine if you could make qualitative statements about the relative efficacies of the algorithms that you selected in the first place (Stacking is described in the handout "Decision Trees and Instance Based Classifiers", Quinlan, page 533). A copy of this dataset suitable for use with MLC++ is available in the **`/u/u61/cse/MLC++/mlc/db/`** directory.

3. *No programming/code is required for this question, though you might submit them for extra credit*

    Goldbach's conjecture states that every even number greater than two can be expressed as the sum of two primes. There is a well known program called AM (Automated Mathematician) that discovered this, but its internal design is beyond the scope of this course.

    Instead, suggest a scheme/some combination of schemes that could lead us to this conclusion. i.e., you want to data-mine that

    ```
    For all even x, there exist primes y and z such that x=y+z, if x>2.
    ```

    Assume that all you know is the notion of numbers, the notion of ordinal operators (<,>,<=,>= etc.), and concepts such as "even" and "prime". In particular, you can assume the existence of predicates such as **`prime(n)`** that returns true if the argument is a prime and false otherwise. Likewise, you can assume the availability of **`even(n)`**. (In reality, there are more predicates available to you, like **`odd(n)`**, **`composite(n)`**, and so on; part of the task is to figure out how exactly you narrow down on **`even`** and **`prime`**, as potential interesting concepts.)

    You can use anything from ILP, decision trees, nearest neighbor methods, neural networks, genetic algorithms or a combination of these. You have to make sure that your design is (i) powerful enough to express the above concept, (ii) can be expected to converge to the same.