Assigned: Tuesday 15 November 2011

Due: Monday 21 November 2011, 11:59 pm

Objectives:

- Understand regression-based classification
- Compare different categories of classifiers

Collaboration: This homework assignment will be completed in pairs assigned by the instructor.

Introduction

As we have seen, linear models form a continuous hypothesis space that is an alternative to the discrete hypothesis space of decision trees. Of course, linear models can be used even on problems where features are discrete. In this assignment, we will develop and use two categories of linear models, namely perceptron learning of a threshold function and logistic regression.

For comparison, we will continue to use the mushroom data from the previous assignment. However, we will introduce another application with real-valued features that decision trees are less suited for. Based on measurements of cell nuclei from a fine needle aspirate of a breast mass, our classifier must learn whether a cell is cancerous or benign. There are 683 training examples using just 9 features, which have been downloaded from the UCI Machine Learning repository.[1]

Background

Code

You may obtain the starter code for this assignment on the MathLAN from the directory

```
~weinman/courses/CSC261/code/regression
```

Here is an overview of the files it contains.

`mushroom.scm` Contains routines for loading the mushroom examples and attributes, which reside in `mushrooms.txt` and `mushroom-attribs.txt`. Plain-english explanations of the terse attributes in `mushroom-attribs.txt` are given in `mushroom-info.txt`.

`breast-cancer.scm` Contains the raw Scheme definitions of `breast-cancer-labels` and `breast-cancer-instances`.

`logistic.scm` Contains starter procedures for learning linear models.

`assignment.scm` Contains the documentation for the procedures you will write.

The remaining files are mostly ancillary and loaded by those mentioned above when needed.

A compiled module containing all of the assignment (to aid in analysis if necessary) may be copied from the `compiled` subdirectory of the path given above. Use the statement `(require "linear-models.scm")` to include them.

[1] http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Data Representation

Unlike with decision trees, we must represent data for linear models as a list of numbers. The input vector **x** (that is, a mathematical vector, not a Scheme vector) will be represented as a Scheme list. We convert the categorical mushroom data into a list of numbers by giving each possible value for each attribute an entry in the list, making a number one if the instance has that particular attribute value, and zero in all other cases.

The mushroom data has 21 attributes with a total of 119 possible values, so each instance will be a list of 119 numbers, with 21 of them being one, and the rest being negative one. An instance from the breast cancer data is just 9 numbers. Class labels are benign (0) or malignant (1).

Assignment

Problem 0: From the lab

Include the definitions of the following procedures in your solution: `threshold`, `threshold-output`, and `zero-one-loss`.

Problem 1: Logistic

Part A

Implement the procedure `(logistic z)` (as defined in AIMA 18.6.4, p. 725)

$$Logistic(z) = \frac{1}{1+e^{-z}}.$$

Part B

Implement a procedure `(logistic-output weights instance)` that behaves just like `threshold-output` but using `logistic` rather than `threshold`.

Part C

Whereas `threshold-output` directly produces a class label, zero or one, the `logistic-output` produces something interpretable as the probability of the class being positive (the "one" class, rather than the "zero" class). Implement a procedure `(logistic-classify weights instance)` that produces 1 if the probability (logistic output) is greater than $\frac{1}{2}$, and zero otherwise.

Part D

Wheras `perceptron-update-instance` uses the perceptron-learning rule, we want weight updates for logistic regression to be based on the gradient of quadratic loss. Write the procedure `(logistic-gradient-instance label instance weights)` that calculates the update contribution for an instance as specified in AIMA Equation (18.8), p. 727. You may use `perceptron-update-instance` as a model.

Problem 2: Measuring loss

Part A

Write a procedure `(squared-loss expected produced)` (as defined in AIMA 18.4.2, p. 711) that returns the squared difference of its parameters.

Part B

Write a procedure `(total-loss-functor` *`prediction-fun loss-fun`*`)` that takes `prediction-fun`, a procedure taking weights and an instance (i.e., `threshold-output`), and `loss-fun`, a procedure that takes two numbers (i.e., `zero-one-loss`) and returns a procedure `loss` of the form `(loss` *`labels instances weights`*`)` that calculates the sum of *`loss-fun`* applied to the labels and the result of applying *`prediction-fun`* to the *`instances`* for the given *`weights`*.

Hint: See the in-class lab section E, where you have already constructed the pieces for this.

For example

```
> (define perceptron-0/1-loss
    (total-loss-functor threshold-output zero-one-loss))
> (perceptron-0/1-loss mushroom-labels mushroom-instances
                                       weights-three-updates)
325
```

reveals the total number of errors made with the weights in `weights-three-updates`.

Problem 3: Learning

Write the procedure `learn-weights` as documented in `assignment.scm`. At each iteration, your procedure should display a line containing the iteration number and the current total loss (according to `total-loss`) using the weights at that iteration. For example

```
> (learn-weights mushroom-labels
                 mushroom-instances
                 1 10
                 perceptron-update-instance
                 perceptron-0/1-loss)
0  3916
1  4208
2  831
3  325
4  374
5  289
6  269
7  256
8  239
9  226
10  218
(-100  20  76  232  190  -89  -930  36  251  972  -332  362  -195  -169  -128  342  -1
```

Problem 4: Analysis

You now have everything you need to perform some basic analysis and comparisons of the perceptron and logistic classifiers on a set of data. Experiment with both the perceptron learning rule (which minimizes 0/1 loss) and logistic regression learning (which minimizes squared loss) on the breast cancer data set.

Two variables may need adjusting: the step size (learning rate) and the number of iterations required for convergence. Naturally, smaller steps will require more iterations. However, larger steps may overshoot the point of minimum loss.

Write a brief report of your experiments and make attempts to answer the following questions:

- What is an appropriate step size and iteration limit for the perceptron rule on this data set? How were these determined?

- What is an appropriate step size and iteration limit for logistic regression on this data set? How were these determined?
- Will these values apply to other data sets? Why or why not?
- What is the best performance (number of errors) you are able to get each classifier to achieve on this data set? How do they compare to one another?
- What observations can you make about the learning process for each model? How does the loss function behave during learning?
- Based on these performances and evaluations, what can you conclude about this data?

Nicely formatted graphs and/or tables are welcome as hallmarks of excellent work, but good or satisfactory work can be achieved without them. The audience for your report is your peers in this class.

What to turn in

Your submission should include the following

- Your completed `assignment.scm`
- A short driver program demonstrating each of your procedures in operation
- A single PDF containing (merged)
  - Your Scheme files
  - A transcript of your driver program's output
  - Your analysis of Problem 4

Copyright ©2011 Jerod Weinman. This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.