16831 Statistical Techniques, Fall 2011
Homework 5: Online Learning

You should work in groups of 2 or 3.

Due: Monday December 5th, emailed by the beginning of class

Assignment

This assignment gives you a chance to implement some of the online learning algorithms discussed in class on real data.

Automated interpretation of ladar data is crucial for outdoor vehicle operation. Your goal is to apply algorithms we discussed in class to this problem and in particular to compare and contrast the various methods. You will be building a system to classify ladar points into one of five categories: ground (supporting surface), vegetation, facade, pole, and wire.

You must implement (from the following list) 3 learners we discussed in class. (Talk to us if you're interested in implementing other variants instead.)

- Gaussian Process Regression (choose two classes)
- Bayes Linear Regression (choose two classes)
- Gradient Descent on Square Loss
- Support Vector Machine (choose two, or do multi-class SVM)
- Online Logistic Regression
- NORMA: Kernelized SVM
- NORMA: Kernelized LogReg
- Multi-class Winnow
- Exponentiated Gradient algorithm on another loss (squared loss or log-reg)

At least one must be a feature selecting type algorithm (i.e. winnow or another method using the exponentiated gradient), and at least one must use kernels/covariance functions. We're very flexible about implementation: we're willing to see lots of other options as well as more sophisticated approaches that use factor graphs, etc, but do less algorithms.

Data

The datasets you should use are available in the provided directory, hw5-data. They are 3D point-clouds of Oakland (see `http://www.cs.cmu.edu/~vmr/datasets/oakland_3d/cvpr09/doc/`). Features are provided courtesy of Dan Munoz, but you are welcome to come up with new features to use (please explain if you do). You should not distribute the data without permission.

There are five classes, their labels values are:

- 1004: Veg
- 1100: Wire
- 1103: Pole
- 1200: Ground
- 1400: Facade

Run each algorithm on both datasets and report on the performance.

What to turn in

We're interested in a short performance summary and nice visualizations. You should turn in your *code* and a 2-3 page *report* via email. For each learner you implemented, report on the following:

1. How well did it perform for online learning? Does it perform well on the held-out data?
2. Are there any classes that did not get classified well? Why do you think that is?
3. How easy was the learner to implement?
4. How long does the learner take (in terms of data points, dimensions, classes, etc...) for training and prediction?
5. Show images/movies of the classified data. Note that MATLAB is not very good at displaying thousands of 3D points; use VRML or python.
6. How did you choose (hyper)parameters (priors, kernel width, noise variance, prior variance, learning rate, etc. . .)?
7. How robust is this algorithm to noise? Take the current feature set and:
    - Add a large number of random features
    - Add a large number of features that are noise corrupted versions of the features already in the data-set.

You should also compare the learners' performance to each other. Did kernels help on this data set? Which one would you use on your robot? What would future work include?