16831 Statistical Techniques, Fall 2009: Homework 4

Due: Tuesday, Dec. 1, beginning of the class

Introduction

You may work in groups of up to 3 people.

1 Online Learning Algs.

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 meth- ods. You will be building a system to classify ladar points into one of three categories: ground (supporting surface), vegetation, and obstacle.

You will implement a few learners we discussed in class. (Talk to us if youre interested in implementing another variant instead.)

Were very flexible about implementation: were willing to see lots of other options as well as more sophisticated approaches that use factor graphs, etc, but do less algorithms. Data set is provided in the files section on the website in hw4 data.zip. README.txt contained within describes the data log formats.

1.1 Question 1: Online Learning and Filtering

- Implement an online gradient descent algorithm on e.g. square loss, (choose two classes), multi-class linear SVM, online logistic regression.
- Implement some variant of a Gauss-Markov (e.g. Kalman) Filter for online learning. (I.e. Bayes Linear Regression, an EKF on a non-linear representation)

Compare performance in the following ways:

Consider the batch performance of the final learner from the methods above with it's online performance. Do you get better performance online then in batch? Why?

Compare the filter version to the online gradient descent version. Both have advantages– what are they?

1.2 Question 2: GPs and Kernel methods

Implement a kernel method (using either the GP equations or NORMA/gradient descent).

For instance, implement one of these:

- Gaussian Processes with squared exponential kernel
- NORMA: Kernelized SVM
- NORMA Kernelized LogReg

Note that implementing GPs will require computational tricks to make efficient enough. Consult the reading material and

Random Features for Large-Scale Kernel Machines, Ali Rahimi, Ben Recht, in Neural Information Processing Systems (NIPS) 2007, pp. 1177-1184.

2 What to turn in

Were interesting in a very short performance summary and nice visualization. A Matlab implementation is fine for this use and a 2-3 page summary. Report on the following:

- 1. What method worked the best for online performance? Does it do so on held-out data?
- 2. Show images/movies of the classified data.
- 3. How did you choose hyper-parameters (kernel width, noise variance, prior variance, learning rate, etc)
- 4. Did kernels help on this data set?
- 5. How would you handle the multi-class problems with a non-multi-class classier?

How do the various methods perform in these situations?

Important Caveat: This data-set may not be distributed or any results published without permission from Drew and the sponsor.