Online Learning Techniques

Dan Monga Kilanga, Pushyami Panindrashayi Shandilya

March 13, 2017

1 Introduction

The two main methods used in machine learning for processing of the data are batch learning and online learning. batch learning generates the best predictor by learning on the entire training data set at once. Whereas, the online learning method is used to update our best predictor for future data-data becomes available in a sequential order. Online learning is a common technique used in areas of machine learning where it is computationally infeasible to train over the entire dataset, requiring the need of out-of-core algorithms. It is also used in situations where it is necessary for the algorithm to dynamically adapt to new patterns in the data, or when the data itself is generated as a function of time.

1.1 Gradient Descent on Squared Loss

Gradient descent is a first-order iterative optimization algorithm, which aims to find the solution by taking the gradient and moving in the appropriate direction, ie, $a^{n+1} = a^n$ - $\gamma\nabla\mathrm{F}\big(a^n\big)$, where γ is the step size. When used on an optimization problem that is defined by a mean square loss function, we arrive at a solution iteratively.

1.2 Linear Support Vector Machine

The Support Vector Machine is a two-class classifier which has the form f(x) $= \alpha_0 + \sum\limits_{i=1}^{N} \alpha_i K(x, x_i)$ where the class label y_i takes values -1 or $+1$. SVM can be viewed as a quadratic optimization problem with linear constraints, and requires a quadratic programming algorithm for its solution. The name support vector arises from the fact that typically many of the $\hat{\alpha_i} = 0$.

1.3 Bayesian Linear Regression

The underlying equation for linear regression is $y_i = x_i^T\beta + \epsilon_i$, where ϵ_i is modeled by a normal Gaussian distribution. The addition to the Bayesian version is that the prior and posterior distributions are considered.

2 Parameter Tuning

In practice, parameters are tuned by performing a grid search, where different values of the parameters are tried over a large interval, and subsequently, the search area is narrowed to the region that performed well on the previous iteration until the best values are found. However, a rigorous grid search was not performed in this assignment. The search was done only over one set of interval with a fairly large step, and the values resulting in the seemingly best performance was retained. All the initial weights were taken to be zero for training purposed, however, for testing purposes, the weights resulting from training were used. On Bayesian Linear Regression, the initial prior distribution was arbitrarily chosen with mean 0 and covariance matrix with non-zero element 0.5 on the diagonal. The variance of noise was chosen to be 1.

3 Comparison of Techniques

3.1 Performance

The algorithms were each run with the same pair of classes for ease in comparing them. And the results for the three algorithms and their performance on the respective classes, for a sample pair ((Veg, Facade) for file1 and (Pole, Vegetation) for file2) are as shown below (file1 and file2 will correspond to `oakland_part3_am_rfṅode_features` and `oakland_part3_an_rfṅode_features` respectively):

1. Gradient Descent on Squared Loss
   - file1: Vegetation success ratio (train, test): 0.75292, 0.73273 and Facade success ratio (train, test): 0.77231, 0.73273
   - file2: Pole success ratio (train, test): 0.85333, 0.93058 and Vegetation success ratio (train, test): 0.93200, 0.93058
2. Linear Support Vector Machine
   - file1: Vegetation success ratio (train, test): 0.99253, 0.99297 and Vegetation success ratio (train, test): 0.93805, 0.94313
   - file2: Pole success ratio (train, test): 0.52007, 0.59497 and Vegetation success ratio (train, test): 0.91509, 0.92788
3. Bayesian Linear Regression
   - file1: Vegetation success ratio (train, test): 0.87269, 0.93029 and Facade success ratio (train, test): 0.92090, 0.93029
   - file2: Pole success ratio (train, test): 0.84794, 0.87394 and Vegetation success ratio (train, test): 0.98478, 0.87395

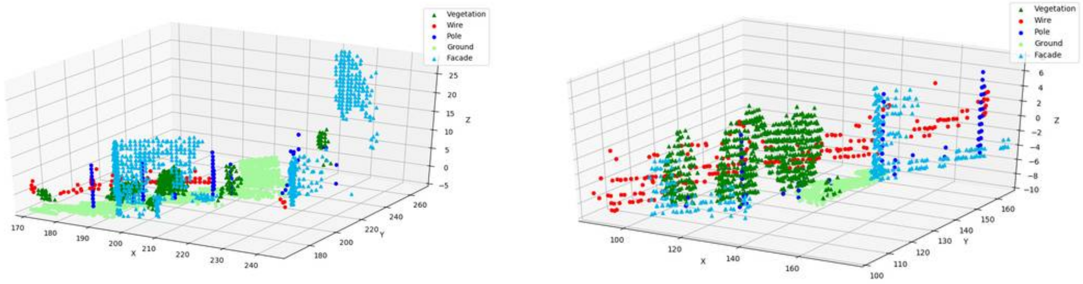
Also, samples of other pairs of classes are included in the zip file.

3.2 Implementation

Gradient Descent needs to go through all the items for each time it updates the model parameters. This is the reason why gradient descent isn't used in practice because there are better algorithms that run much faster. In order to test a data point using an SVM model, you need to compute the dot product of each support vector with the test point. Therefore the computational complexity of the model is linear in the number of support vectors. Fewer support vectors means faster classification of test points. Bayesian Linear Regression uses the statistical analysis within the context of Bayesian inference- makes it more flexible and able to handle complex models.

3.3 Robustness

When 5 additional random features were added, the performance of gradient descent slightly affected with regard to training data, but it was greatly affected when it came to testing data.
Vegetation success ratio (train, test): 0.74508, 0.67405 and Facade success ratio (train, test): 0.76492, 0.69210. The performance of the Linear Support Vector Machine classifier was not very altered.
Vegetation success ratio (train, test): 0.99277, 0.99303 and Facade success ratio (train, test): 0.93997, 0.94361. However, One of the classes of Bayesian Linear Regression, (i.e. vegetation), was affected by the random features added. The change is not enormous, but the difference was more noticeable than it was in the case of Linear Support Vector Machine.
Vegetation success ratio (train, test): 0.87286, 0.89119 and Facade success ration (train, test): 0.92079, 0.93062. When we added noise to existing features, the results were not so much different from when we added random features, and this probably because the original features that are combined to noise don't bring any new information. it's only the noise that has an effect.



(a) File 1 data

(b) File 2 data

Figure 1: Visualization of entire datasets

4 Results

The ease of implementation and robustness of algorithms have been discussed in earlier sections. Another factor that matters while classifying is how skewed the data is to one of the classes. For example, we see in the case of the Bayesian Linear Regression, the density of one of the classes is significantly lesser compared to the other- also coupled with the fact that the datapoints are downsampled for plotting- we can see in the gif for that class that the less dense class is predicted on few instances.
The visualization for all the datapoints in the two files are as shown.