CS-534 Machine Learning

Implementation Assignment 1

Author 1 (%) Author 2 (%) Author 3 (%)

Introduction

Given a set a M training pairs $\{\mathbf{x}_i, y_i\}_{i=1}^{M}$, where $\mathbf{x}_i$ is a set of features $\{x_{ij}\}_{j=1}^{N}$ and y_i is the corresponding target variable, we wish to find a set a parameters $\mathbf{w} = \{w_1, w_2, \ldots, w_n\}$ that minimizes the regularized Sum of Squared Error (SSE) objective

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{m}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \lambda||\mathbf{w}||^2.$$

Here, the regularization term $\lambda||\mathbf{w}||^2$ is used to counteract overfitting: a common artifact caused by high dimensional features. Overfitting is often manifest in large parameters, thus the regularization term to encourage small weights. λ is a hyper-parameter that controls the influence of regularization on these weights and subsequent objective value and solution. This value is often not known and difficult to determine. The objective of this assignment is to implement the gradient descent algorithm (GDA) on a high dimensional training set and investigate the effect of the regularization parameter via cross-validation and validation using an independent dataset.

Gradient descent algorithm

To implement GDA, we first derive the gradient of the cost function $J(\mathbf{w})$. The gradient vector is of size M, the number of samples in the training set, and each element is equal to the partial derivative of the cost function $J(\mathbf{w})$ with respect to the parameters $\{w_i\}_{i=1}^{N}$. The chain rule gives

$$\nabla J(\mathbf{w}) = \left[\frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}, \ldots, \frac{\partial J}{\partial w_n}\right]^\top. \tag{1}$$

Thus, the k^{th} element of the gradient vector is

$$\frac{\partial J}{\partial w_k} = \sum_{i=1}^{m}(\mathbf{w}^\top \mathbf{x}_i - y_i)x_{ik} + 2\lambda\mathbf{w}. \tag{2}$$

This gives the following regularized gradient of the cost function $J(\mathbf{w})$:

$$\nabla J(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{w}^\top \mathbf{x}_i - y_i)\mathbf{x}_i + 2\lambda \mathbf{w}. \tag{3}$$

Note that we have scaled the first term by the number of samples in the equation to represent the average gradient per weight. The gradient is now incorporated into the update rule for gradient descent. For a given training pair $\{\mathbf{x}_i, y_i\}$, the batch update rule is

$$w_j =: w_j - \alpha\big((\mathbf{w}^\top \mathbf{x}_i - y_i)x_{ij} + 2\lambda w_j\big). \tag{4}$$

Equations 3 and 4 are used in a batch gradient descent algorithm where the parameters $\mathbf{w}$ are updated simultaneously across features until the norm of $\nabla J(\mathbf{w})$ converges at a small value ϵ. To improve the speed of convergence we will also standardize the features by $s(\mathbf{x}) = \frac{\mathbf{x}-\bar{\mathbf{x}}}{\sigma}$.

Algorithm 1: Batch Gradient Descent

```
$\mathbf{w} = \mathbf{w^0}$;
do
    $\nabla J(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{w}^\top \mathbf{x}_i - y_i)\mathbf{x}_i + 2\lambda \mathbf{w}$;
    $\mathbf{w} =: \mathbf{w} - \alpha \nabla J(\mathbf{w})$;
while $|\nabla J(\mathbf{w})| \geq \epsilon$;
```

In the algorithm, parameters $\mathbf{w}$ are initialized to the zero vector of size M: the number of features in $\mathbf{x}_i$. These parameters are iteratively updated by the regularized gradient of $J(\mathbf{w})$ scaled by the hyper-parameter α, called the learning rate. An inherent trade-off exists in the selection of this parameter. If the learning rate is too small, then the algorithm will be slow to converge or may not converge at all. A large learning rate can result in overstepping the optimum and oscillating out of control to infinity. Prior to tunning the regularization parameter, we will explore different learning rates.

Learning rate selection (10 pts)

Tunning the regularizer (20 pts)

We tested a range of regularization values in order to investigate the effect of λ on the magnitude of the weights and objective value of the cost function. Initially, we found that values less then 10^{-2} had little effect on the weights. Furthermore, values greater than 10^2 caused the gradient to be large enough to overstep the optimum and oscillate out of control. Thus, we restricted our test to the sequence $(\lambda_i)_{i=1}^{n}$ given by $\lambda_i = \frac{10i}{n}, n = 10^3$.

We performed gradient descent on the training set for each λ_i holding all other hyper-parameters constant. For each λ_i we calculated the corresponding

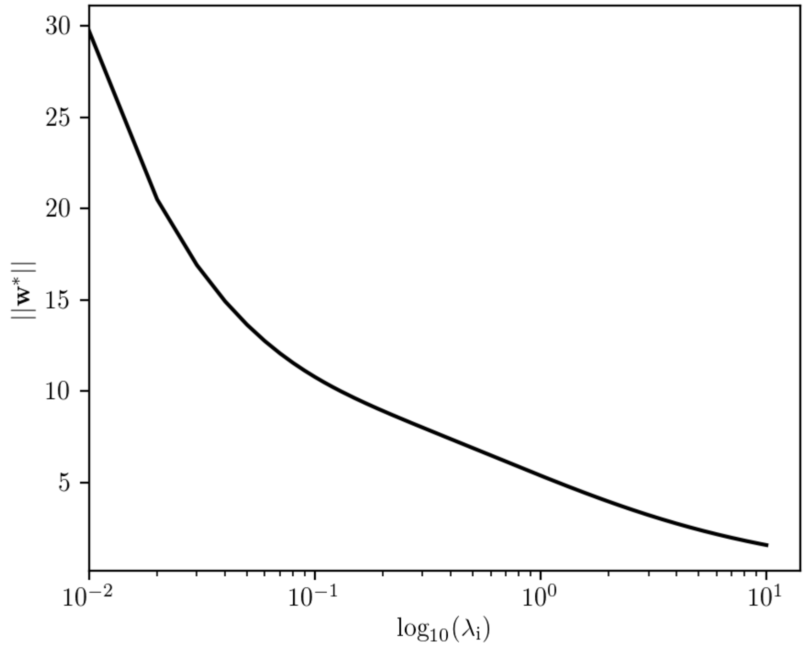

Figure 1: The effect of different λ values on the norm of the optimized weights.

magnitude of the optimized weights vector $\mathbf{w}_i^*$ to confirm that the regularization term was properly driving the weights to be sufficiently small. Figure 1 shows this relationship.

For each λ_i we also calculated SSE for both the training set and test set. This reveals an interesting relationship shown Figure 2. As the regularization parameter increases, the SSE objective function value for the training set increases. This is expected as the absence of regularization results in the best fit regressor for the training set and objective function values depart from the optimum as more control is enforced via regularization.

Conversely, as the regularization parameter increase, the SSE objective function value for the test set decrease, then increases approximately proportional to the training SSE. Regularization is a direct treatment for overfitting, and overfitting is often manifest in the performance of the regressor on a test set. We can see in Figure 2 by enforcing parsimonious weights in the model, performance on the test set improves. However, as expected, this trend will eventually digress when regularization becomes too confining on the magnitude of the weights, discouraging the algorithm to sufficiently learn good parameters for either dataset. This phenomenon is know as underfitting and is often evident in poor performance on both the training and test sets.

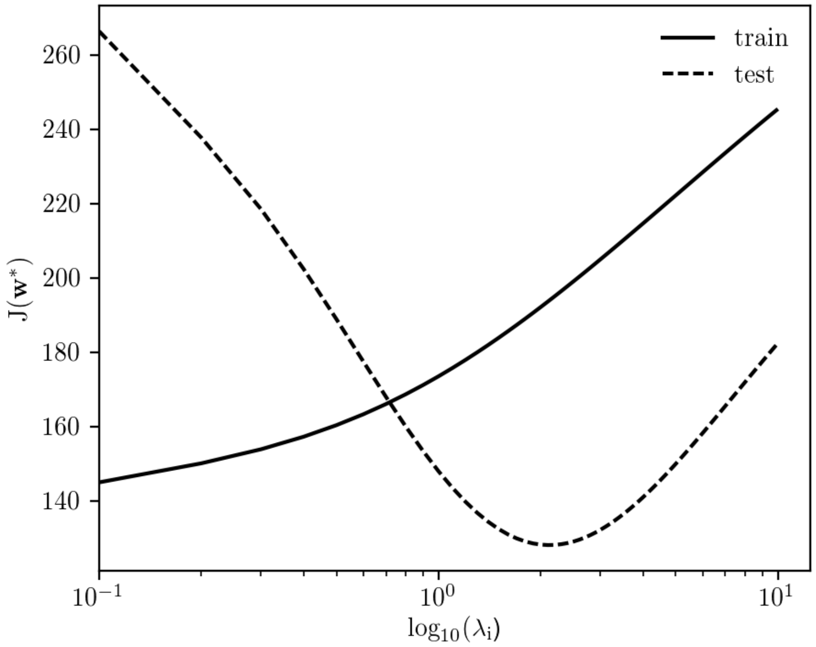

Figure 2: The effect of different λ values on the training SSE (solid) and the test SSE (dashed). **N.B.** This is a subset of the range of λ values test to show intersection of the trends.

According to our tests we recommend values in the range $1 \leq \lambda \leq 2$ as the regularization parameter for this dataset.

10-fold cross-validation (30 pts)