CPSC 340 Assignment 4

Henry Deng: c1z8

1 Convex Functions

1. $\frac{d}{dx}$ f'(w) $= 2\alpha - \beta \geq 0$
   Since $\alpha \geq 0$, the second derivative is always greater than or equal to zero for the entire domain, therefore, the function is convex.
2. $\frac{d}{dx}$ f'(w) $= \frac{1}{w} \geq 0$
   Since $w > 0$, the second derivative will always be greater or equal to 0 for the entire domain; therefore, the function is convex.
3. $f(w) = \|Xw - y\|^2 + \lambda\|w\|_1$
   The summation of two convex functions is a convex function. Firstly, $\|Xw - y\|^2$ is a convex function because the 2-norm is convex, and so is $\|w\|_1$. Furthermore, lambda is greater than or equal to 0, so $\lambda\|w\|_1$ is convex. All the elements of the function are convex and when added together, the function is still convex
4. $f(w) = \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i))$
   The sum of convex functions is convex, so we need to show that $\log(1 + \exp(-y_i w^T x_i))$ is convex. Let $(-y_i w^T x_i)$ be a constant x. The first differentiation yields $\frac{1}{1+e(-x)}$. Differentiating again, we get $\frac{e(-x)}{(1+e(-x))^2}$. This simplifies to $\frac{1}{1+e(-x)} \cdot \frac{e(-x)}{1+e(-x)}$ This function cannot be negative, therefore, f(w) must be convex.
5. Since $w_0 - w^T x_i$ is a linear function, we know it must be convex. As a result, $\sum_{i=1}^{n} \max[0, w_0 - w^T x_i]$ is also convex because the maximum of a convex function is convex. Since lambda is ≥ 0 and $\|w\|^2$ is convex, the sum of these functions must also be convex.

2 Logistic Regression with Sparse Regularization

2.1 L2-Regularization

Code: Linked in README, `https://github.ugrad.cs.ubc.ca/CPSC340-2017W-T2/c1z8_a4/blob/master/code/linear_model.py`
The updated training error is: 0.02, the validation error is: 0.074, the number of features is 101, and the number of gradient descent iterations is 36.

2.2 L1-Regularization

Code: Linked in README, `https://github.ugrad.cs.ubc.ca/CPSC340-2017W-T2/c1z8_a4/blob/master/code/linear_model.py`
The updated training error is: 0.000, the validation error is: 0.048, the number of features is 72, and the number of gradient descent iterations is 351.

2.3 L0-Regularization

Code: Linked in README, `https://github.ugrad.cs.ubc.ca/CPSC340-2017W-T2/c1z8_a4/blob/master/code/linear_model.py`
The updated training error is: 0.000, the validation error is: 0.040, the number of non-zeros is 25.

2.4 Discussion

From the results, the performance is as follows: $L0 > L1 > L2$. L0 selects more important features since it has the lowest number of non-zeros. As a result, the validation error from L0 is lower than L1 and L2 regularization. However, the run-time for L0 is significantly slower than L1 and L2.

2.5 Comparison with scikit-learn

Code: Linked in README, `https://github.ugrad.cs.ubc.ca/CPSC340-2017W-T2/c1z8_a4/blob/master/code/main.py`
The results from the scikit-learn yielded very similar results to my own. In terms of L2, the training, validation, and number of non-zeros remained the same. For L1, the training error remained the same, but the validation error slightly increased to 0.052 and the non-zeros count decreased from 72 to 71.

3 Multi-Class Logistic

3.1 Softmax Classification, toy example

We want to maximize the inner-product of, $w_c^T \hat{x}$

$w_1^T \hat{x} = (+2)(1) + (-1)(1) = 1$
$w_2^T \hat{x} = (+2)(1) + (+2)(1) = 4$
$w_3^T \hat{x} = (+3)(1) + (-1)(1) = 2$

Under this model, label 2 would maximize the inner-product, and would be chosen for this test example.

3.2 One-vs-all Logistic Regression

Code: Linked in README, `https://github.ugrad.cs.ubc.ca/CPSC340-2017W-T2/c1z8_a4/blob/master/code/linear_model.py`
The validation error is 0.070 and the training error is 0.084.

3.3 Softmax Classifier Implementation

Code: Linked in README, `https://github.ugrad.cs.ubc.ca/CPSC340-2017W-T2/c1z8_a4/blob/master/code/linear_model.py`
The validation error is 0.008 and the training error is 0.

3.4 Comparison with scikit-learn, again

Code: Linked in README, `https://github.ugrad.cs.ubc.ca/CPSC340-2017W-T2/c1z8_a4/blob/master/code/main.py`
The training and validation errors were a bit higher than our implementation for OVA, with training error = 0.100 and validation error = 0.080. For softmax, the results from scikit-learn were roughly the same as our implementation: the training error = 0.008, and the validation error = 0.

3.5 Cost of Multinomial Logistic Regression

1. Computing the derivative for $f(w)$ with respect to one entry in W takes $n(d+dk)$ time. For all entries in W, this will take $dk(nd+ndk)$ time. For t iterations, we get a total cost of $O(tnk^2d^2)$.
2. To predict XW and find the max of all training examples will take $O(tkd)$ time

4 Very-Short Answer Questions

1. Using validation error to choose features tends to overfit. Score BIC makes sure that selecting too many features is penalized in proportion to our sample size, which results in better feature selection.
2. Exhaustively searching takes a very long time (exponential run time) whereas forward selection can be run in polynomial time
3. As lambda decreases, more of our results are determined by the objective function, so train error is small but test error is high. When lambda increases, train error increases but test error decreases.
4. L1 could be preferred when there's a lot of irrelevant features since it does feature selection. L2 could be preferred because it's differentiable and has a unique solution (easier to compute).
5. The penalty for being too right is very significant when using least squares, ie. $w^Tx_i = +100$ and $y_i = +1$
6. SVM will choose a classifier that is farthest from both classes (the largest margin), whereas perceptron finds any classifier with zero error.
7. All of the methods produce a linear classifier.
8. Multi-label: each data point can be assigned multiple labels. Mult-class: a classification task with more than two classes with the assumption that each sample is assigned to only one label
9. Fill in the question marks: for one-vs-all multi-class logistic regression, we are solving one optimization problem(s) of dimension k. On the other hand, for softmax logistic regression, we are solving more than one optimization problem(s) of dimension k.