CPSC 340 Assignment 4 (due Friday, March 9 at 9:00pm)

1 Convex Functions

1. $f(w) = \alpha w^2 - \beta w + \gamma$ with $w \in \mathbb{R}, \alpha \geq 0, \beta \in \mathbb{R}, \gamma \in \mathbb{R}$ (1D quadratic).

   Find $f''(w)$.

   $$f'(w) = \alpha 2w - \beta$$
   $$f''(w) = \alpha 2$$

   as $\alpha >= 0$, $f''(w) >= 0$ for all 'w', thus, this function is convex.

2. $f(w) = w\log(w)$ with $w > 0$ ("neg-entropy")

   Find $f''(w)$.

   $$f'(w) = log(w) + w(1/w)$$
   $$= log(w) + 1$$
   $$f''(w) = \frac{1}{wln(2)}$$

   as $w > 0$, $f''(w) > 0$, thus, this function is convex.

3. $f(w) = \|Xw - y\|^2 + \lambda\|w\|_1$ with $w \in \mathbb{R}^d, \lambda \geq 0$ (L1-regularized least squares).

   $||w||_1$ is convex as all norms are convex.
   $\lambda||w||$ is convex all convex functions multiplied by a non-negative constant are convex.
   $||Xw - y||^2$ is convex as all squared norms are convex.

   As the sum of convex functions is convex, thus, this function is convex.

4. $f(w) = \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i))$ with $w \in \mathbb{R}^d$ (logistic regression).

   Let $g(z) = \log(1 + \exp(z))$.

   $$g'(z) = \frac{\exp(z)}{1 + \exp(z)} = \frac{1}{1 + \exp(-z)}$$

   $$g''(z) = \frac{\exp(-z)}{(1 + \exp(-z))^2} > 0$$

   for all values of z since $\exp(x) > 0$ for any x, and $(1 + \exp(-z))^2 > 0$ as squaring values makes them positive.
   As the sum of convex functions is convex, thus, this function is convex.

5. $f(w, w_0) = \sum_{i=1}^{N} [\max\{0, w_0 - w^T x_i\} - w_0] + \frac{\lambda}{2}\|w\|_2^2$ with $w \in R^d, w_0 \in \mathbb{R}, \lambda \geq 0$ ("1-class" SVM).

   $\|w\|_2^2$ is convex as all squared norms are convex.
   $\frac{\lambda}{2}\|w\|_2^2$ and a convex function multiplied by non-negative constant is convex.
   Both 0 and $w_0 - w^T x_i$ are linear functions, so they are convex.

The max of two convex functions is convex, so $\max\{0, w_0 - w^T x_i\} - w_0$ is convex, and the subtraction does not affect concavity.
As the sum of convex functions is convex, thus, this function is convex.

2 Logistic Regression with Sparse Regularization

2.1 L2-Regularization

Hand in your updated code. Using this new code with $\lambda = 1$, report how the following quantities change: the training error, the validation error, the number of features used, and the number of gradient descent iterations.

Code included: *code/linear_model.py*

Training error increased to 0.002
Validation error decreased to 0.074
Number of features used did not change
Number of gradient descent iterations decreased to 36

2.2 L1-Regularization

Hand in your updated code. Using this new code with $\lambda = 1$, report how the following quantities change: the training error, the validation error, the number of features used, and the number of gradient descent iterations.

Code included: *code/linear_model.py*

Training error remained the same
Validation error decreased to 0.052
Number of features decreased to 71
Number of gradient descent iterations decreased to 78.

2.3 L0-Regularization

Hand in your updated code. Using this new code with $\lambda = 1$, report the training error, validation error, and number of features selected.

Code included: *code/linear_model.py*

Training error is 0.00.
Validation error is 0.038.
Number of features selected is 24.

2.4 Discussion

In a short paragraph, briefly discuss your results from the above. How do the different forms of regularization compare with each other? Can you provide some intuition for your results? No need to write a long essay, please!

L2-regularization gives us non-zero training error because it doesn't do feature selection (we get incredibly small, specific weights for irrelvant features which allows us to fit exactly.)

L1-regularization allows us to do some feature selection (setting irrelevant features to zero) which helps with the overfitting, thus getting lower validation error.

And L0-regularization does even more which allows us to get even lower validation error. However, the L0-regularization is a fairly slow algorithm as it requires nested loops.

2.5 Comparison with scikit-learn

Compare your results (training error, validation error, number of nonzero weights) for L2 and L1 regularization with scikit-learn's LogisticRegression.

Training error:
L2-Regularization: 0.002
L1-Regularization: 0.000

scikit-learn's + L2-Regularization: 0.002
scikit-learn's + L1-Regularization: 0.000

Validation error:
L2-Regularization: 0.074
L1-Regularization: 0.052

scikit-learn's + L2-Regularization: 0.074
scikit-learn's + L1-Regularization: 0.052

Nonzero weights:
L2-Regularization: 101
L1-Regularization: 71

scikit-learn's + L2-Regularization: 101
scikit-learn's + L1-Regularization: 71

3 Multi-Class Logistic

3.1 Softmax Classification, toy example

Under this model, what class label would we assign to the test example? (Show your work.)

We would want $max(w_c^T \hat{x})$ - need to get all cases of [1,1] in x and make a prediction.

$$w = \begin{bmatrix} +2 & -1 \\ +2 & +2 \\ +3 & -1 \end{bmatrix}$$

$$w^T = \begin{bmatrix} -1 & +2 & -1 \\ +3 & +2 & +2 \end{bmatrix}$$

$$c1 = (-1)(1) + (3)(1) = 2$$
$$c2 = (2)(1) + (2)(1) = 4$$
$$c3 = (-1)(1) + (2)(1) = 1$$

c2 produces the max, so we choose c2 as the class label.

3.2 One-vs-all Logistic Regression

Hand in the code and report the validation error.

Code included: *code/linear_model.py*
Validation error: 0.070

3.3 Softmax Classifier Implementation

Hand in the code and report the validation error.

Code included: *code/linear_model.py*
Validation error: 0.008

3.4 Comparison with scikit-learn, again

Compare your results (training error and validation error for both one-vs-all and softmax) with scikit-learn's `LogisticRegression`, which can also handle multi-class problems.

One-vs-all:
Training error: 0.084
Validation error:0.070

Softmax function:
Training error: 0.000
Validation error: 0.008

Sklearn one-vs-all function:
Training error: 0.084
Validation error:0.070

Sklearn softmax function:
Training error: 0.000
Validation error: 0.012

Our one-vs-all function produces the exact same result as the Sklearn one-vs-all function, in both training and test error.

For the 2 softmax functions, for a high C(1000) - theyre nearly exactly the same, except that our validation error is slightly lower.

3.5 Cost of Multinomial Logistic Regression

1. In $O()$ notation, what is the cost of training the softmax classifier?

   O(ndkT) - T iterations over n examples - which goes through each class (c) and feature (d).

2. What is the cost of classifying the test examples?

   O(tdk) - The dimensions of the XW matrix - have to go through the whole thing to classify the examples.

4 Very-Short Answer Questions

1. Why would you use a score BIC instead of a validation error for feature selection?

   BIC is better than validation error for feature selection because it penalises model complexity - thus avoiding over-fitting. Penalisies additional features

2. Why do we use forward selection instead of exhaustively search all subsets in search and score methods?

   exhaustive search and score finds the best for every single combination of features so it takes a lot longer than forward selection.

3. In L2-regularization, how does λ relate to the two parts of the fundamental trade-off?

   L1 > L2: L1 is robust and L2 is not very robust. L2 > L1: L2 is stable/ always produces one solution, while L1 is unstable and possibly returns multiple solutions.

4. Give one reason why one might chose to use L1 regularization over L2 and give one reason for the reverse case.

   They dont work the same way. For classification, the distance to the classifier doesnt correlate to the squared error - the error should be binary, distance to the answer is arbitrary.

5. What is the main problem with using least squares to fit a linear model for binary classification?

   A classifier always exists for perceptron, but not always for SVM

6. For a linearly separable binary classification problem, hit ow does a linear SVM differ from a classifier found using the perceptron algorithm?

   b, c, and d

7. Which of the following methods produce linear classifiers? (a) binary least squares as in Question 3, (b) the perceptron algorithm, (c) SVMs, and (d) logistic regression.

   In multi-class classification there is one true label, whereas in multi-label several of the labels (or none) can be correct.

8. What is the difference between multi-label and multi-class classification?

   In multi-class there is one label vs. multi-label = many labels where many can be right (or none can be right).

9. Fill in the question marks: for one-vs-all multi-class logistic regression, we are solving ?? optimization problem(s) of dimension ??. On the other hand, for softmax logistic regression, we are solving ?? optimization problem(s) of dimension ??.

   (a) 1

   (b) 1

   (c) n

   (d) d