Math189R SP19
Homework 2
Monday, Feb 11, 2019

Feel free to work with other students, but make sure you write up the homework and code on your own (no copying homework *or* code; no pair programming). Feel free to ask students or instructors for help debugging code or whatever else, though.

The starter files can be found under the Resource tab on course website. The graphs for problem 3 generated by the sample solution could be found in the corresponding zipfile. These graphs only serve as references to your implementation. You should generate your own graphs for submission. Please print out all the graphs generated by your own code and submit them together with the written part, and make sure you upload the code to your Github repository.

1 (**Murphy 8.3**) Gradient and Hessian of the log-likelihood for logistic regression.

(a) Let $\sigma(x) = \frac{1}{1+e^{-x}}$ be the sigmoid function. Show that

$$\sigma'(x) = \sigma(x)\left[1 - \sigma(x)\right].$$

(b) Using the previous result and the chain rule of calculus, derive an expression for the gradient of the log likelihood for logistic regression.

(c) The Hessian can be written as $\mathbf{H} = \mathbf{X}^\top \mathbf{S} \mathbf{X}$ where $\mathbf{S} = \text{diag}(\mu_1(1-\mu_1), \ldots, \mu_n(1-\mu_n))$. Derive this and show that $\mathbf{H} \succeq 0$ ($A \succeq 0$ means that A is positive semidefinite).

Hint: Use the **negative** log-likelihood of logistic regression for this problem.

a) First we begin by taking the derivative of $\sigma(x)$ using the chain rule and the quotient rule: $\sigma'(x) = e^{-x}(1+e^{-1})^{-2}$.
Expand out the exponent in the denom: $(\frac{1}{1+e^{-x}})(\frac{e^{-x}}{1+e^{-x}})$
Now sub in $\sigma(x)$: $\sigma(x)(\frac{e^{-x}}{1+e^{-x}})$
Refactor to get: $\sigma(x)(1 - \frac{1}{1+e^{-x}})$
Which is: $\sigma(x)\left[1 - \sigma(x)\right]$ QED

b) First, we need to find the gradient of $nll(\theta)$ by incorperating the results from part a:
$-\sum_i y_i(\frac{(\sigma'(\theta^\top \mathbf{x}_i))}{(\sigma(\theta^\top \mathbf{x}_i)}) + (1-y_i)(\frac{1}{1-\sigma(\theta^\top \mathbf{x}_i)})(-(\sigma'(\theta^\top \mathbf{x}_i))$
Simplify this to: $\sum_i(\sigma(\theta^\top \mathbf{x}_i) - y_i)\mathbf{x}_i$
Then more simplification to: $\sum_i(\mu_i - y_i)\mathbf{x}_i$
Finally, we arrive at: $X^\top(\mu - \mathbf{y})$ QED
Per the solutions (couldn't figure this part out), μ_i and $\mathbf{x}_i$ are the transposes of the i-th role of the matrix X.

c) I also could not figure this part out, so I looked at the solution. I don't think it'd be helpful for me to copy what the solutions say exactly, so instead, I'll summarize:
By expanding the gradient and using tranposes liberally, we get that the Hessian matrix is: $X^\top SX$.
S is equal to the diagonal matrix representation of: $\mu_1(1-\mu_1), ..., \mu_1(1-\mu_i)$
To show that S is positive semi-definite, all we need to prove is that is that the eigenvalues of S are greater than 0, since by definition of a positive semi-definite matrix, the corresponding eigenvalues must be nonnegative. As we learned in Math 73, the eigenvalues of S, since it is diagonal, are just its diagnonal entries. Therefore, since the diagonal entries of S are comprised of expressions like $\mu_1(1-\mu_1)$, we just need to prove that the general form of the expression, $\mu_i(1-\mu_i)$, is greater than or equal to 0.
$\mu_i(1-\mu_i) = \sigma(\theta^\top x_i)(1-\sigma(\theta^\top \mathbf{x}_i))$
Since$\sigma()$ has a range of [0,1], both $\sigma(\theta^\top x_i)$ and $1-\sigma(\theta^\top \mathbf{x}_i))$ turn out to be nonnegative. Therefore, H is posistive semidefinite.

■

2 (**Murphy 2.11**) Derive the normalization constant (Z) for a one dimensional zero-mean Gaussian

$$\mathbb{P}(x;\sigma^2) = \frac{1}{Z}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

such that $\mathbb{P}(x;\sigma^2)$ becomes a valid density.

By definition of a probablity density, the area under any given function must be equal to 1, since that is the $P(x)$ has a range of [0,1].

Therefore: $1 = \int_R \frac{1}{Z}\exp\left(-\frac{x^2}{2\sigma^2}\right)dx$

Bring Z over to the left side: $Z = \int_R \exp\left(-\frac{x^2}{2\sigma^2}\right)dx$

Square both sides so that we can get a double intergral: $Z^2 = \int\int_R^2 \exp(-\frac{x^2+y^2}{2\sigma^2})dxdy$

Solve (had to look this one up :'( ): $-2\pi\sigma^2\exp(-\frac{-r^2}{2\sigma^2})\Big|_0^\infty$

$= 2\pi\sigma^2$

Therefore, $Z = (2\pi\sigma)^{\frac{1}{2}}$ QED

■

3 (**regression**). In this problem, we will use the online news popularity dataset to set up a model for linear regression. In the starter code, we have already parsed the data for you. However, you might need internet connection to access the data and therefore successfully run the starter code.

We split the csv file into a training and test set with the first two thirds of the data in the training set and the rest for testing. Of the testing data, we split the first half into a 'validation set' (used to optimize hyperparameters while leaving your testing data pristine) and the remaining half as your test set. We will use this data for the remainder of the problem. The goal of this data is to predict the **log** number of shares a news article will have given the other features.

(a) (**math**) Show that the maximum a posteriori problem for linear regression with a zero-mean Gaussian prior $\mathbb{P}(\mathbf{w}) = \prod_j \mathcal{N}(w_j|0,\tau^2)$ on the weights,

$$\arg\max_{\mathbf{w}} \sum_{i=1}^{N} \log \mathcal{N}(y_i|w_0 + \mathbf{w}^\top \mathbf{x}_i, \sigma^2) + \sum_{j=1}^{D} \log \mathcal{N}(w_j|0,\tau^2)$$

is equivalent to the ridge regression problem

$$\arg\min \frac{1}{N} \sum_{i=1}^{N} (y_i - (w_0 + \mathbf{w}^\top \mathbf{x}_i))^2 + \lambda ||\mathbf{w}||_2^2$$

with $\lambda = \sigma^2/\tau^2$.

(b) (**math**) Find a closed form solution $\mathbf{x}^\star$ to the ridge regression problem:

$$\text{minimize: } ||A\mathbf{x} - \mathbf{b}||_2^2 + ||\Gamma\mathbf{x}||_2^2.$$

(c) (**implementation**) Attempt to predict the log shares using ridge regression from the previous problem solution. Make sure you include a bias term and *don't regularize the bias term*. Find the optimal regularization parameter λ from the validation set. Plot both λ versus the validation RMSE (you should have tried at least 150 parameter settings randomly chosen between 0.0 and 150.0 because the dataset is small) and λ versus $||\boldsymbol{\theta}^\star||_2$ where $\boldsymbol{\theta}$ is your weight vector. What is the final RMSE on the test set with the optimal $\lambda^\star$?

(continued on the following pages)

a) Once again, I had to peek at the solutions to fully understand this part. I'll summarize: We apply the $\mathbb{N}$ probablity distribution: $N(x|\mu,\sigma) = \frac{1}{(2\pi\sigma)^{\frac{1}{2}}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$

Again, had to look this up, but now we have: $\arg_{\mathbf{w}} \max \sum_{i=1}^{N} log(\frac{1}{(2\pi\sigma)^{\frac{1}{2}}} \exp(-\frac{(y_i - w_0 - \mathbf{w}^\top \mathbf{x}_i)^2}{2\sigma^2}) + \sum_{j=i}^{D} log(\frac{1}{(2\pi\sigma)^{\frac{1}{2}}}) \exp(-\frac{w_j^2}{2\tau^2})$

Following the solutions (I won't type them out because I don't think thats useful): By ignoring the constant and rescaling the problem by minimizing the negative of the function, then by defining $\lambda = \frac{\sigma^2}{\tau^2}$, we get the final form:

$\arg_{\mathbf{w}} \min \sum_{i=1}^{N} (y_i - w_0 - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda ||\mathbf{w}||_2^2$ QED :)

b) First we start off by minimizing f, which we do by finding the gradient with respect to x:

$\nabla_x f = 2A^\top A\mathbf{x} - 2A^\top \mathbf{b} + 2\Gamma^\top \Gamma x$

To minimize, set equal to 0 and solve for $\mathbf{x}$: $\mathbf{x}^* = (A^\top A + \Gamma^\top \Gamma)^{-1} A^\top \mathbf{b}$, which is the closed form solution.

c) I found the optimal regularization paramter to be 7.6258 and both the validation and test RSME to be 1.0006. I've included the plots in a seperate part of the submission.

■

3 (continued)

(d) (**math**) Consider regularized linear regression where we pull the basis term out of the feature vectors. That is, instead of computing $\hat{\mathbf{y}} = \boldsymbol{\theta}^\top \mathbf{x}$ with $\mathbf{x}_0 = 1$, we compute $\hat{\mathbf{y}} = \boldsymbol{\theta}^\top \mathbf{x} + b$. This corresponds to solving the optimization problem

$$\text{minimize: } ||A\mathbf{x} + b\mathbf{1} - \mathbf{y}||_2^2 + ||\Gamma\mathbf{x}||_2^2.$$

Solve for the optimal $\mathbf{x}^\star$ explicitly. Use this close form to compute the bias term for the previous problem (with the same regularization strategy). Make sure it is the same.

(e) (**implementation**) We can also compute the solution to the least squares problem using gradient descent. Consider the same bias-relocated objective

$$\text{minimize: } f = ||A\mathbf{x} + b\mathbf{1} - \mathbf{y}||_2^2 + ||\Gamma\mathbf{x}||_2^2.$$

Compute the gradients and run gradient descent. Plot the ℓ_2 norm between the optimal $(\mathbf{x}^\star, b^\star)$ vector you computed in closed form and the iterates generated by gradient descent. Hint: your plot should move down and to the left and approach zero as the number of iterations increases. If it doesn't, try decreasing the learning rate.

d) Let's expand the problem, so that $f = ||A\mathbf{x} + b1 - \mathbf{y}||_2^2 + ||\Gamma\mathbf{x}||_2^2$
Expand this out to get: $\mathbf{x}^\top A^\top A\mathbf{x} + 2ba^\top A\mathbf{x} - 2\mathbf{y}^\top Ax - 2b1^\top\mathbf{y} + b^2n + \mathbf{y}^\top\mathbf{y} + \mathbf{x}^\top\Gamma^\top\Gamma\mathbf{x}$
Now, find the gradient of f with respect to b and solve for gradient(b) =0, such that: $\nabla_b f = 21^\top A\mathbf{x} - 21^\top\mathbf{y} + 2bn = 0$
Now solve for b: $b^* = \frac{1^\top(\mathbf{y} - A\mathbf{x})}{n}$
(I looked at the solutions for this): Now plug b back so we can solve for x^*:
$\mathbf{x}^* = [A^\top(I - n^{-1}11^\top)A + \Gamma^\top\Gamma]^{-1}A^\top(I - n^{-1}11^\top)\mathbf{y}$
Results from code:
==¿ Step 3: Linear regression without bias...
–Time elapsed for training: 12.41 seconds
==¿ Difference in bias is 1.9190E-10
==¿ Difference in weights is 2.4689E-10

e) I included the graph in the submission. Here are the results from the code:

==¿ Step 4: Gradient descent
==¿ Running gradient descent...
– Iteration25 - training rmse 1.6604 - gradient norm 3.6435E+04
– Iteration50 - training rmse 1.2519 - gradient norm 2.4237E+04
– Iteration75 - training rmse 1.0664 - gradient norm 1.5202E+04
– Iteration100 - training rmse 0.9896 - gradient norm 9.7337E+03
– Iteration125 - training rmse 0.9596 - gradient norm 6.3025E+03
– Iteration150 - training rmse 0.9481 - gradient norm 4.1295E+

–Time elapsed for training: 159.41 seconds
==¿ Plotting completed.
==¿ Difference in bias is 1.5387E-01
==¿ Difference in weights is 7.9920E-01

■