CS2810 Mathematics of Data Models	Gold
Spring 2019	March 11, 2019

Homework 06

Due: March 20, 9PM
Point total: 60

Instructions:

- Submit your PDF and/or .py file to Blackboard by the due date and time. Please do not zip your files together, as this interferes with Blackboard's preview functionality. Always show all your work, and for full credit, you must use the method that the problem instructs you to use (unless none is mentioned). Handwritten or typeset solutions are both acceptable, but unreadable submissions will be penalized. You may discuss problems with other students, but you may not write up solutions together, copy solutions from a common whiteboard, or otherwise share your written work or code. Do not use code or language that is copied from the Internet or other students; attribute the ideas *and* rephrase in your own words.

Problem 1 (9 points)

Find the gradient of each function f at the specified point P, then indicate a vector of movement we should follow to maximally *decrease* f (for gradient descent, for example).

i. $f(x, y) = x^2 + 2xy + y^3$; $P = (1, 1)$

Solution: $\frac{\partial f}{\partial x} = 2x + 2y$
$\frac{\partial f}{\partial y} = 2x + 3y^2$

At (1,1), these evaluate to 4 and 5, respectively. So the gradient is $\begin{bmatrix}4\\5\end{bmatrix}$

and the direction that will minimize f at this point is

$\begin{bmatrix}-4\\-5\end{bmatrix}$

ii. $f(x, y) = 0$ if $x \leq 0$ or $y \leq 0$, else $f(x, y) = xy$; $P = (2, 3)$

Solution: Since we're in a part of the function with both values greater than zero, the other stipulation doesn't apply. $\frac{\partial f}{\partial x} = y$ and $\frac{\partial f}{\partial y} = x$, so the gradient at (2,3) is $\begin{bmatrix}3\\2\end{bmatrix}$ and the direction to minimize it is $\begin{bmatrix}-3\\-2\end{bmatrix}$

iii. $f(w, x, y, z) = 2w + 3x + 4y + 5z$, $P = (6, 7, 8, 9)$ (The gradient works analogously for more dimensions than two.)

Solution: The gradient is $\begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$, regardless of where it is evaluated, and the direction that best minimizes it is $\begin{bmatrix} -2 \\ -3 \\ -4 \\ -5 \end{bmatrix}$.

Problem 2 (15 points, 5 each)

For each function, find the partial derivatives according to x and y, and then find the location(s) where its gradient is zero, and thus, we potentially have a local minimum or maximum. Then find f_{xx} and f_{yy} and determine whether the curvature in each direction is positive (curving up), negative (curving down), or neither.

i. $f(x, y) = 3x^2 + 10y^2 + 3$

Solution: $\frac{\partial f}{\partial x} = 6x$, $\frac{\partial f}{\partial y} = 20y$

These are both 0 at $(0, 0)$.

$\frac{\partial^2 f}{\partial x^2} = 6$, $\frac{\partial^2 f}{\partial y^2} = 20$

These are both positive, so this is a local minimum.

ii. $f(x, y) = -3x^2y - 3x + 27y + 4$

Solution: $\frac{\partial f}{\partial x} = -6xy - 3$ and $\frac{\partial f}{\partial y} = -3x^2 + 27$

The second equation has zeros at $x = \pm 3$, and combined with the constraint that $xy = -1/2$ this gives the two points $(3, -1/6)$ and $(-3, 1/6)$.

$\frac{\partial^2 f}{\partial x^2} = -6y - 3$, $\frac{\partial^2 f}{\partial y^2} = 0$

The curvature in the x direction is negative (down) for $(3, -1/6)$ and positive (up) for $(-3, 1/6)$.

In both cases, the curvature in the y direction is flat.

iii. $f(x, y) = -y^2/(x + 1)^4$ (Recall that this could be interpreted as $-y^2(x + 1)^{-4}$.)

Solution: $\frac{\partial f}{\partial x} = -2y/(x + 1)^4$, $\frac{\partial f}{\partial y} = 4y^2/(x + 1)^5$

This is 0 at all locations where $y = 0$.

$\frac{\partial^2 f}{\partial x^2} = 8y/(x + 1)^5$, $\frac{\partial^2 f}{\partial y^2} = 8y/(x + 1)^5$

When $y = 0$, these are also both 0. So this is neither curving up nor curving down.

Problem 3 (15 points [4, 4, 3, 4])

In each case, find the tangent plane and a normal vector of $z = f(x, y)$ at the point $P = (x_0, y_0, z_0)$. Give your tangent plane in the form $c_1x + c_2y + c_3z = c_4$; for example, $-x + 2y + 3z = 10$. (You don't need to normalize the normal vector.)

i. $z = x^2 + y^2$, $P = (3, 4, 5)$

Solution: Partial derivatives are $\frac{\partial f}{\partial x} = 2x$, $\frac{\partial f}{\partial y} = 2y$, which at this particular point evaluate to 6 and 8, respectively. Tangent plane equation is $(z-5) = 6(x-3)+8(y-4)$, which simplifies to $-6x - 8y + z = -45$. Normal vector is $\begin{bmatrix} 6 \\ 8 \\ -1 \end{bmatrix}$.

ii. $z = \sin x + 2\sin y$; $P = (\pi, \pi, 0)$ (Recall if $f(x) = \sin x$, $f'(x) = \cos x$.)

Solution: Partial derivatives are $\frac{\partial f}{\partial x} = \cos x$, $\frac{\partial f}{\partial y} = 2\cos y$, which evaluate at this point to -1 and -2, respectively. Tangent plane equation is $(z - 0) = -1(x - \pi) + -2(y - \pi)$ which simplifies to $x + 2y + z = 3\pi$. A normal vector is $\begin{bmatrix} -1 \\ -2 \\ -1 \end{bmatrix}$

iii. $z = 0, P = (1, 2, 0)$

Solution: This is a plane, so it is its own tangent plane. (The tangent plane formula gives $z = 0$.) A normal vector using our formula is (0, 0, -1), although (0, 0, 1) would do just as well.

iv. Normal vectors are often used in graphics to determine how bright a surface should be as a result of illumination. Lambert's Cosine Law says that the light reflected from a light-scattering surface has a brightness equal to $I\cos\theta$, where I is the original intensity of the light and θ is the angle between the normal vector at the point and a vector from the point to the light source. Find the brightness of illumination at the point $(3, 4, 5)$ (the same as the first part of this problem) if a light source is shining from (1,1,100) with intensity 100. (Flip the direction of the normal so that it's pointing toward the light source.)

Solution: We established earlier that the normal vector was $(6, 8, -1)$, so flipping that becomes $(-6, -8, 1)$. A vector to the light source is $(-2, -3, 95)$, so the dot product is 12 + 24 + 95 = 131. The normal vector has length $\sqrt{36 + 64 + 1} = \sqrt{101} = 10.05$ and the vector to the light source has length $\sqrt{4 + 9 + 9025} = 95.07$, so dividing the dot product by both gives $\cos\theta = 0.137$. So the illumination of this point should be about 13.7 units of brightness.

Problem 4 (9 points, 3 each)

i. What is the directional derivative of a function in a direction orthogonal to the gradient? Explain why your claim must be true. (You can limit your proof to the case of the two-dimensional gradient.)

Solution: It must be 0. If it's orthogonal to the gradient, then its dot product with the gradient is zero. Since the directional derivative is a dot product with the gradient itself, that means that it is zero, too.

ii. Find a function where a finite global minimum exists, but gradient descent may not find it. Give both the function equation and a plot of the function. Your function need not use more than one variable.

Solution: A few strategies here could be high-degree polynomials, such as $x^6 - 4x^4 + 3x^3 - 2x^2 + x - 1$, or trigonometric functions that are being damped over time, such as $0.5^{x^2} \sin x$.

iii. If we don't know a formula for a function that we want to perform gradient descent on, we could simply sample the formula in different directions, and estimate the gradient from that. Explain how we could estimate the gradient from the values $f(x, y)$, $f(x + \Delta x)$, and $f(x, y + \Delta y)$. Is it worse to have a Δx that is too big, or one that is too small?

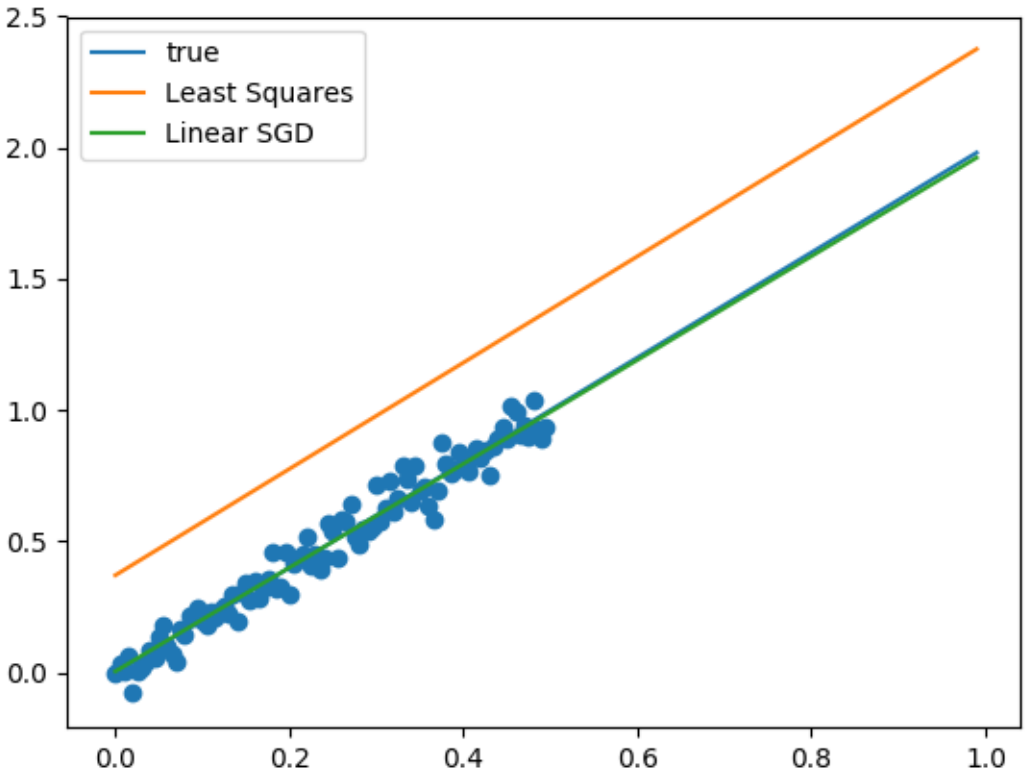
Solution: We can estimate the gradient to just be the vector $\begin{bmatrix} f(x + \Delta x, y) - f(x, y) \\ f(x, y + \Delta y) - f(x, y) \end{bmatrix}$ since these are how much f changes in response to the changes in x and y, and thus approximate the partial derivatives. It's worse to have a Δ that is too big, since we could then think the gradient is more like the gradient of a point that is farther away; having a small step size just gets us closer to the actual definition of the gradient, which uses an infinitesimal step size.

Problem 5 (12 points [4, 2, 4, 2])

For this problem, you're going to gain some experience using the machine learning tools in `scikit-learn`, a popular Python package for ML. You can find starter code in `sgd.py`.

Install the `scikit-learn` module before performing these exercises.

i. Refer to the documentation for `sklearn.linear_model.SGDRegressor` to train a stochastic gradient descent model on 100 noisy datapoints created with `create_data()`. The parameters of the model should be: use squared error as loss; iterate for 10^6 iterations (set tol to "None" and ignore the warning), use L1 regularization, and set the regularization parameter alpha to 0.001. Use `plot_fits()` and `solve_least_squares()` to compare the fit at $y = 0, z = 0$ and at $y = 3, z = 3$ to a least squares fit, and include the latter plot in your PDF submission.

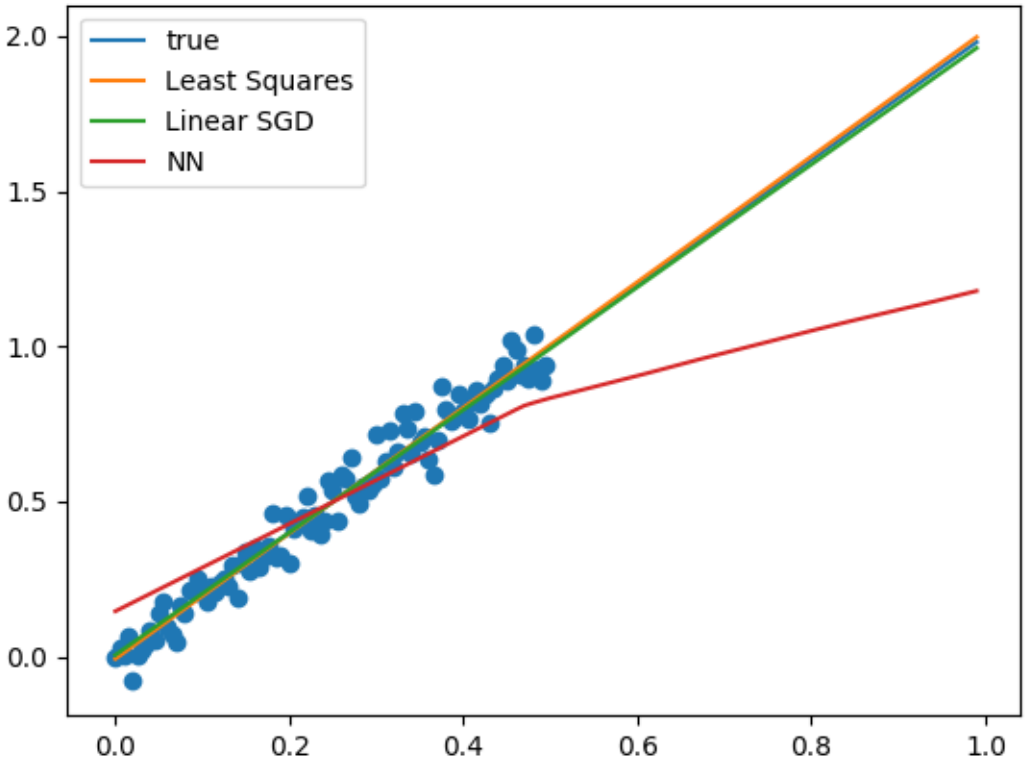


Solution:

ii. Why do you think SGD captures the true function better than least squares, which we thought was optimal? (Hint: Notice that small nonzero values for y and z coefficients will have a big effect at $y = 3, z = 3$, which are larger than any actually observed inputs. How did SGD avoid this effect?)

Solution: The stochastic gradient descent method has a regularization term, which tells it to prefer explanations with fewer nonzero parameters. Least squares, on the other hand, is just trying to fit the noisy data, and could introduce nonzero parameters to "explain" the noise.

iii. Now train a neural network on the same data using `sklearn.neural_network.MLPRegressor` using the following parameters: 50 hidden units, L2 regularization with $\alpha = 0.001$, a million iterations, stochastic gradient descent, learning rate 0.01, tolerance 0.001, and allow 1000 iterations without improving more than the tolerance. Plot the performance versus your other two models for $y = 0, z = 0$, and include this plot in your PDF submission.



Solution:

iv. Suggest one change to the neural network that might help it match the underlying function better. (You can suggest a direction and a parameter, without needing an exact value to set it to; for example, "increase the number of hidden units." There is more than one possible answer.)

Solution: Some reasonable possible responses, roughly in the order of most likely to succeed to less likely: more regularization to force a simpler, straight function; fewer hidden units so that it is less able to make a too-complicated function; more training time. Decreasing the learning rate or the tolerance could also keep the network from stopping before it has really nailed down the best solution.