CSE 512: Homework 1 Solutions — Due Monday, Sept. 14

1. *Bayes' rule.*

   (a) I look out of my window and see 10 people in the street. 8 have umbrellas, 2 do not. None of these people know each other; they make decisions independently.
   In general, the area is very windy, so people do not always use an umbrella when it rains. But each day has a 50% chance of getting rain. In fact, even when it rains, people only use an umbrella 75% of the time. When it's not raining, people sometimes use an umbrella as a parasol, 5% of the time.

   i. What is the probability that it is raining, given that I am using an umbrella?
   Ans. (0.5 pts) First, let's try to use Bayes' rule to calculate $\mathbf{Pr}(\text{rain}|\text{umbrella})$:

   $$\mathbf{Pr}(\text{rain}|\text{umbrella}) = \frac{\mathbf{Pr}(\text{umbrella}|\text{rain})\mathbf{Pr}(\text{rain})}{\mathbf{Pr}(\text{umbrella})}$$

   Using Law of Total Probability, we can calculate

   $\mathbf{Pr}(\text{umbrella}) = \mathbf{Pr}(\text{umbrella}|\text{rain})\mathbf{Pr}(\text{rain}) + \mathbf{Pr}(\text{umbrella}|\text{no rain})\mathbf{Pr}(\text{no rain}) = 3/4 \cdot 1/2 + 1/20 \cdot 1/2 = 2/5.$

   Then, using Bayes' rule,

   $$\mathbf{Pr}(\text{rain}|\text{umbrella}) = \frac{\mathbf{Pr}(\text{umbrella}|\text{rain})\mathbf{Pr}(\text{rain})}{\mathbf{Pr}(\text{umbrella})} = \frac{3/4 \cdot 1/2}{2/5} = 15/16 = 93.75\%.$$

   ii. What is the probability that it is raining today, given my observation?
   Ans. (0.5 pts) By observation, we see 8 umbrellas, 2 no umbrellas. Therefore, again using Bayes' rule,

   $$\mathbf{Pr}(\text{rain}|\text{observation}) = \frac{\mathbf{Pr}(\text{observation}|\text{rain})\mathbf{Pr}(\text{rain})}{\mathbf{Pr}(\text{observation})}$$

   Again, we can determine $\mathbf{Pr}(\text{observation})$ using the Law of Total Probability:

   $$\begin{aligned}\mathbf{Pr}(\text{observation}) &= \mathbf{Pr}(\text{observation}|\text{rain})\mathbf{Pr}(\text{rain}) + \mathbf{Pr}(\text{observation}|\text{no rain})\mathbf{Pr}(\text{no rain}) \\ &= (3/4)^8 \cdot 1/4^2 \cdot 1/2 + 1/20^8 \cdot (19/20)^2 \cdot 1/2 \\ &\approx 0.00313.\end{aligned}$$

   Therefore,

   $$\mathbf{Pr}(\text{rain}|\text{observation}) = \frac{(3/4)^8 \cdot 1/4^2 \cdot 1/2}{(3/4)^8 \cdot 1/4^2 \cdot 1/2 + 1/20^8 \cdot (19/20)^2 \cdot 1/2} \approx 0.99999999436$$

   (b) According to the CDC[1], the current percentage of positive COVID tests in the US is 9.1%. Let's use this as a marker for $\mathbf{Pr}(\text{COVID}|\text{took a test})$.
   There are many available tests, each with different performance metrics. For example, if I take a combined IgG/IgM seriology test, then for one company [2], the PPV (the chance of a positive test when COVID is present) is estimated at 82.5%, and the NPV (the chance of a negative test when COVID is not present) is estimated at 99.9%.

   i. If a person goes in for a test and gets a positive result, what are the chances that that person has COVID?
   Ans. (0.5 pts) Following the same procedure as in the previous problem,

   $$\begin{aligned}\mathbf{Pr}(+\text{ test}) &= \underbrace{\mathbf{Pr}(+\text{ test}|\text{COVID})}_{\text{PPV}}\mathbf{Pr}(\text{COVID}) + \underbrace{\mathbf{Pr}(+\text{ test}|\text{not COVID})}_{1-\text{NPV}}\mathbf{Pr}(\text{not COVID}) \\ &= 82.5\% \cdot 9.1\% + .1\% \cdot 90.9\% \approx 7.60\%\end{aligned}$$

   Then

   $$\mathbf{Pr}(\text{COVID}|+\text{ test}) = \frac{\mathbf{Pr}(+\text{ test}|\text{COVID})\mathbf{Pr}(\text{COVID})}{\mathbf{Pr}(+\text{ test})} \approx 98.814\%$$

[1] https://www.cdc.gov/coronavirus/2019-ncov/covid-data/covidview/index.html

[2] https://www.fda.gov/medical-devices/coronavirus-disease-2019-covid-19-emergency-use-authorizations-medical-devices/eua-authorized-serology-test-performance

ii. If a person goes in for a test and gets a negative result, what are the chances that that person does not have COVID?
Ans. (0.5 pts)

$$\begin{aligned}\mathbf{Pr}(-\text{ test}) &= \underbrace{\mathbf{Pr}(-\text{ test}|\text{COVID})}_{\text{1-PPV}}\mathbf{Pr}(\text{COVID}) + \underbrace{\mathbf{Pr}(-\text{ test}|\text{not COVID})}_{\text{NPV}}\mathbf{Pr}(\text{not COVID}) \\ &= 17.5\% \cdot 9.1\% + 99.9\% \cdot 90.9\% \approx 92.40\%\end{aligned}$$

Then

$$\mathbf{Pr}(\text{not COVID}|-\text{ test}) = \frac{\mathbf{Pr}(-\text{ test}|\text{not COVID})\mathbf{Pr}(\text{not COVID})}{\mathbf{Pr}(-\text{ test})} \approx 98.28\%$$

All in all, the tests are not bad!

2. *Logistic regression for Binary MNIST*

(a) For a twice-differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, the *gradient* and *Hessian* are defined as

$$\nabla f(\theta) = \begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_2} \\ \vdots \\ \frac{\partial f}{\partial \theta_d} \end{bmatrix} \in \mathbb{R}^d, \qquad \nabla^2 f(\theta) = \begin{bmatrix} \frac{\partial^2 f}{\partial \theta_1^2} & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_d} \\ \frac{\partial^2 f}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 f}{\partial \theta_2^2} & \cdots & \frac{\partial^2 f}{\partial \theta_2 \partial \theta_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \theta_d \partial \theta_1} & \frac{\partial^2 f}{\partial \theta_d \partial \theta_2} & \cdots & \frac{\partial^2 f}{\partial \theta_d^2} \end{bmatrix} \in \mathbb{R}^{d \times d}.$$

For example, for the function $f(\theta) = \theta_1^2 + 2\theta_1\theta_2 + \theta_3^3$, the gradient and Hessian are

$$\nabla f(\theta) = \begin{bmatrix} 2\theta_1 + 2\theta_2 \\ 2\theta_1 \\ 3\theta_3^2 \end{bmatrix}, \qquad \nabla^2 f(\theta) = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 6\theta_3 \end{bmatrix}.$$

What is the gradient and Hessian of the logistic loss function

$$\mathcal{L}(\theta) = -\frac{1}{m}\sum_{i=1}^{m} \log(\sigma(y_i x_i^T \theta)), \quad \sigma(s) = \frac{1}{1+e^{-s}}$$

where $y_i \in \{-1, 1\}$?
Ans. (1 pts) To reduce notation, we write $z_i = y_i x_i$, and the matrix $Z = [z_1, z_2, ...z_m]^T$ has each vector z_i as a row.
We can work out the derivatives of the sigmoid function, and show that

$$\sigma'(s) = \sigma(s)(1 - \sigma(s)).$$

After that, standard calculus rules gets us to

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = -\frac{1}{m}\sum_{i=1}^{m} \frac{\sigma(z_i^T\theta)(1 - \sigma(z_i^T\theta))}{\sigma(z_i^T\theta)} z_i.$$

This yields

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{m}\sum_{i=1}^{m} (\sigma(z_i^T\theta) - 1) z_i.$$

which we can write succinctly as

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{m} Z^T (d - \mathbf{1})$$

for $d_i = \sigma(z_i^T\theta)$, $i = 1, ..., m$.

The benefit of using the matrix-vector format is that now the Hessian is much easier to write down. Knowing that

$$\frac{\partial \sigma(z_i^T\theta)}{\partial \theta_k} = \sigma(z_i^T\theta)(1-\sigma(z_i^T\theta))z_i[k] \qquad (z_i[k] = k\text{th element of vector } z_i)$$

then

$$\nabla^2 \mathcal{L} = \frac{1}{m} Z^T \mathbf{diag}(d)\mathbf{diag}(\mathbf{1}-d)Z$$

(b) **Coding.**

- Download mnist.mat [**?**]. We will use logistic regression to diffrentiate 4's from 9's, a notoriously tricky problem. If you are using python, you can use `scipy.io.loadmat` to read the matrix. If you are using MATLAB, just `load mnist.mat` should suffice.
- The matrices X and y should contain the vectorized images and corresponding labels. To take a look at how the data is stored, run the following code:
  Python

```
data = sio.loadmat('mnist.mat')
for k in xrange(9):
    plt.subplot(3,3,k+1)
    plt.imshow(np.reshape(data['trainX'][k,:],(28,28)))
    plt.title(data['trainY'][0,k])
plt.tight_layout()
```

  Matlab

```
load mnist.mat
for k = 1:9:
    subplot(3,3,k)
    imshow(reshape(trainX(k,:),28,28))
    title(trainY(k))
end
```

- Select only the data rows corresponding to the labels 4 and 9, and set the remaining labels to be binary.
  Python

```
idx = np.logical_or(np.equal(y,4) ,  np.equal(y,9))
X = X[idx,:]
y = y[idx]
y[np.equal(y,4)] = -1
y[np.equal(y,9)] = 1
```

  Matlab

```
idx = (y == 4) || (y == 9)
X = X(idx,:)
y = y(idx)
y(y==4) = -1
y(y==9)= 1
```

  You should be left with 11791 train images and 1991 test images. Make sure they are stored separately, e.g. X = train images, X_t = test images.
- Normalize the data matrix by first rescaling the pixel values to all be between 0 and 1 (effectively, divide by 255), and then translating all the values so that the sum of all the images in the *train set* is 0. To do this, first compute the average pixel for the training set, e.g.

$$x_{\mathbf{mean}}[k] = \frac{1}{m_{\text{train}}} \sum_{i=1}^{m_{\text{train}}} x_i^{\text{train}}[k], \qquad x_{\mathbf{mean}} \in \mathbb{R}^{784}.$$

  Then translate both the test and train data using this offset:

$$x_i^{\text{train}} \leftarrow x_i^{\text{train}} - x_{\mathbf{mean}}$$

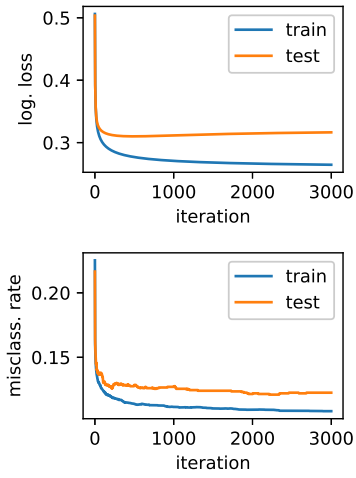


Figure 1: HW 1. MNIST

$$x_i^{\text{test}} \leftarrow x_i^{\text{test}} - x_{\mathbf{mean}}$$

[3]

- Use gradient descent to minimize the logistic loss for this classification problem. Use a step size of 0.001, and run for 5000 iterations. Plot the train / test loss, and train/test misclassification rate, and also report these final values.
- Comment a bit on what you see.

Ans. (2 pts)

Train loss: 0.2645. Train misclassification rate: 10.82%. Test loss: 0.3165. Test misclassification rate: 12.26. See figure 1 for plot.

Some comments: For this problem, using the technique prescribed here, the train/test misclassification rates are around 10%, which is not that impressive. Note that when the task is shifted to 0/1 disambiguation, the rate goes down to 2-3%–in fact, 4/9 is a difficult pair. One thing to notice is that though the train loss keeps decreasing, the test loss saturates quite early. Even more interestingly, the *test misclassification error* does not saturate at the same time, and continues to decrease a bit further! This is an indication of "margin maximizing behavior" which we will cover in a later section.

Note to grader: Give full marks for comments if anything reasonable is written.

3. Using the properties of norms, verify that the following are norms, or prove that they are not norms

   (a) *Direct sum.* $f : \mathbb{R}^d \to \mathbb{R}$, $f(x) = \sum_k x[k]$

   Ans. (0.3 pts) This is not a norm, since it is not nonnegative everywhere.

   (b) *Sum of square roots, squared.* $f : \mathbb{R}^d \to \mathbb{R}$, $f(x) = \left(\sum_{k=1}^d \sqrt{|x[k]|}\right)^2$

   Ans. (0.3 pts) This is not a norm, since it is cannot satisfy triangle inequality. In particular, just take $d = 2$ and

$$f\left(\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}\right) = f\left(\begin{bmatrix}1\\1\end{bmatrix}\right) = (\sqrt{1} + \sqrt{1})^2 = 4$$

$$f\left(\begin{bmatrix}1\\0\end{bmatrix}\right) + f\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = 1 + 1 = 2$$

   and therefore we have shown that $f(x + y) > f(x) + f(y)$ for some choice of x, y.

[3] There are other ways to normalize, but this is a reasonable one I have found works in practice, and in the interest of "normalizing" the assignment, it's what we'll go with.

(c) *Weighted 2-norm.* $f : \mathbb{R}^d \to \mathbb{R}$, $f(x) = \sqrt{\sum_{k=1}^{d} \frac{|x[k]|^2}{k}}$

Ans. (0.4 pts) Yes, this is a norm. To see that, first note that we can write

$$f(x) = \|Wx\|_2, \qquad W = \mathbf{diag}(1, 1/2, 1/3, ..., 1/d).$$

Then we can just go about checking the norm conditions.

- 0 at 0? Yes, $f(0) = \|W\,0\|_2 = \|0\|_2 = 0$
- Positive homogeneity?

  $$f(\alpha x) = \|W(\alpha x)\|_2 = |\alpha|\|Wx\|_2 = |\alpha| f(x)$$

  Check.
- Triangle inequality?

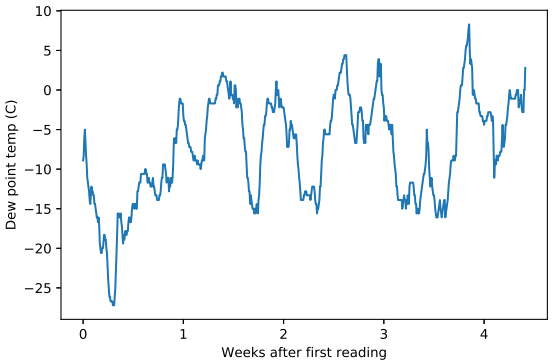
  $$f(x+y) = \|W(x+y)\|_2 = \|Wx + Wy\|_2 \overset{\Delta\text{-ineq of 2-norm}}{\leq} \|Wx\|_2 + \|Wy\|_2 = f(x) + f(y)$$

  Check!

Therefore this is a norm.

4. *Polyfit via linear regression.*

(a) Download weatherDewTmp.mat. Plot the data (`plot(weeks,dew)`). It should look like the following



(b) We want to form a polynomial regression of this data. That is, given w = weeks and d = dew readings, we want to find $\theta_1, ..., \theta_p$ as the solution to

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \tfrac{1}{2} \sum_{i=1}^{m} (\theta_1 + \theta_2 w_i + \theta_3 w_i^2 + \cdots + \theta_p w_i^{p-1} - d_i)^2. \tag{1}$$

Find X and y such that (1) is equivalent to the least squares problem

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \tfrac{1}{2}\|X\theta - y\|_2^2. \tag{2}$$

Ans. (0.25 pts) For w packing the weeks data, we pack y with the dew data, and

$$X = \begin{bmatrix} 1 & w_1 & w_1^2 & w_1^3 & \cdots & w_1^{p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 1 & w_m & w_m^2 & w_m^3 & \cdots & w_m^{p-1} \end{bmatrix}.$$

(c) What are the normal equations for problem (2)? In particular, if θ^* is the minimizer of (2), then θ^* solves a linear system $A\theta^* = b$. What are A and b?

Ans. (0.25 pts) The normal equations are characterized by the linear system that emerges from setting the gradient of (2) to 0:

$$X^T X\theta = X^T y$$

or, specifically, $A = X^T X$ and $b = X^T y$.

(d) *Ridge regression* Oftentimes, it is helpful to add a *regularization term* to (2), to improve stability. This also has an interpretation as Bayesian linear regression with a Gaussian 0-mean prior. In other words, we solve

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \tfrac{1}{2}\|X\theta - y\|_2^2 + \tfrac{\alpha}{2}\|\theta\|_2^2. \tag{3}$$

for some $\alpha > 0$. The θ^* that minimizes (3) is the solution to a different linear system $A_{\text{reg}}\theta^* = b_{\text{reg}}$. What are A_{reg} and b_{reg}?

Ans. (0.25 pts) The normal equations again emerge from setting the gradient of (3) to 0:

$$X^T X\theta + \alpha\theta = X^T y$$

or, specifically, $A_{\text{reg}} = X^T X + \alpha I$ and $b_{\text{reg}} = X^T y$.

(e) If A is a positive semidefinite matrix with condition number 5 and largest eigenvalue 1, what is the condition number of $A + \alpha I$ for some $\alpha > 0$?

Ans. (0.25 pts) If A has largest eigenvalue $\lambda_{\max} = 1$ and condition number $\kappa = 5$, then its smallest eigenvalue must be $\lambda_{\min}$ where

$$\kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}} \Rightarrow \lambda_{\min} = 1/5.$$

Recall that the eigenvalues of $A + \alpha I$ are the eigenvalues of A, $+\alpha$. So, the condition number for $A + \alpha I$ is

$$\kappa(A + \alpha I) = \frac{\lambda_{\max} + \alpha}{\lambda_{\min} + \alpha} = \frac{1 + \alpha}{0.2 + \alpha}.$$

(f) In MATLAB or Python, write a function that takes as argument p and returns X and y so that (1) is equivalent to (2). Report the *condition numbers* for A and A_{reg} by filling out this table:

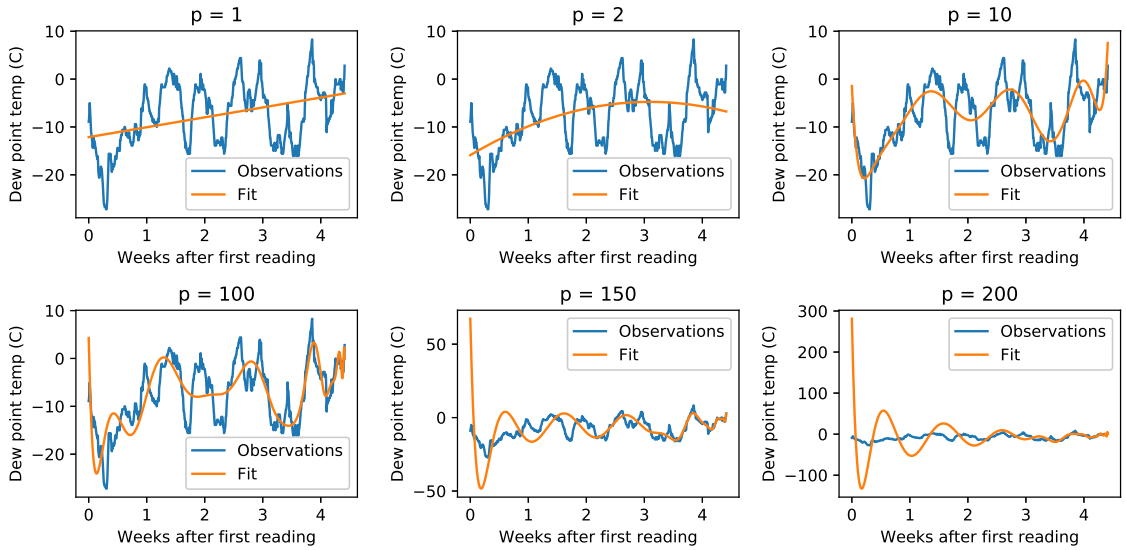
p	A ($\alpha = 0$)	A_{reg}, $\alpha = 0.1 \cdot m$	A_{reg}, $\alpha = m$	A_{reg}, $\alpha = 10 \cdot m$	A_{reg}, $\alpha = 100 \cdot m$
1					
2					
5					
10					

Ans. (.75 pts)

p	A ($\alpha = 0$)	A_{reg}, $\alpha = 0.1 \cdot m$	A_{reg}, $\alpha = m$	A_{reg}, $\alpha = 10 \cdot m$	A_{reg}, $\alpha = 100 \cdot m$
1	3.25e+01	2.28e+01	6.75e+00	1.69e+00	1.07e+00
2	1.48e+03	5.30e+02	7.91e+01	9.20e+00	1.82e+00
5	7.31e+08	2.71e+06	2.72e+05	2.72e+04	2.72e+03
10	7.16e+18	3.97e+12	3.97e+11	3.97e+10	3.97e+09

(g) Compute a polynomial fit by solving (2) for polynomials of order $1, 2, 10, 100, 150$, and 200. Plot all the fits on separate plots (use `subplot`). Comment on your observations.

Ans. (.75 pts)

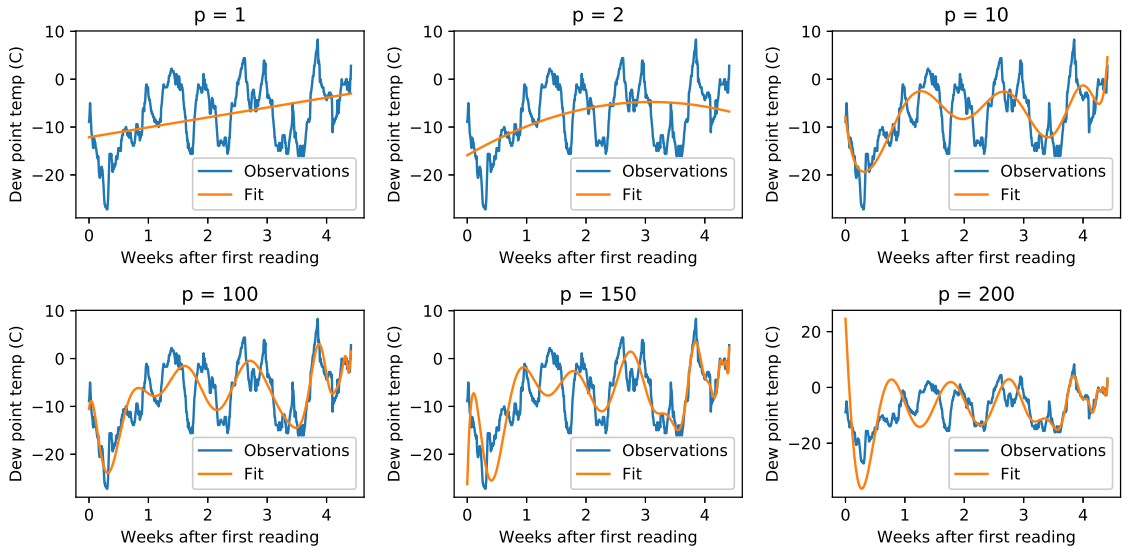


For many values of p, the fit gets better as p gets larger. However, at some point the fit becomes unstable, and is very bad. This can partly be helped by making sure the input values (w) are smaller, but even then, we notice that fits have different qualities for smaller w (more stable, less good fit) and larger ones (less stable, more often good fit).

Note to grader: give full marks for any comment that sounds reasonable.

(h) Now compute a *regularized* polynomial fit by solving (3) for polynomials of order $1, 2, 10, 100, 150$, and 200, for $\alpha = 0.0001$. Plot all the fits on separate plots (use `subplot`). Comment on your observations. How does this compare to the unregularized polynomial fit?

Ans. (.75 pts)

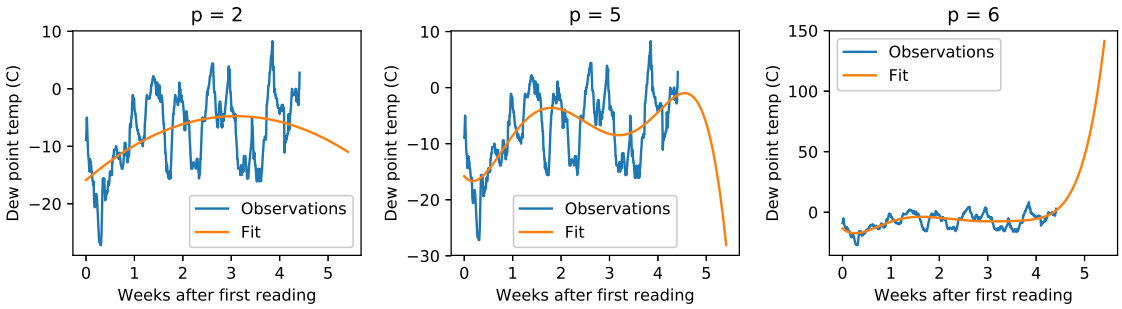


Interestingly, for such a small value of α, already the fits are much more stable, even for large values of p. However, the fit quality doesn't necessarily get that much better than just using a small value of p.

Note to grader: give full marks for any comment that sounds reasonable.

(i) Picking your favorite set of hyperparameters (p, α), forecast the next week's dew point temperature. Plot the forecasted data over the current observations. Do you believe your forecast? Why?

Ans. (.75 pts) There are a number of possible solutions here. Here are some examples, using $\alpha = 0.01$. (In general, for useful values of p, I did not notice much impact of α.)



Overall, I do not believe these fits, as the smaller values of p clearly underfit, and the larger values of p behave wildly in the forecasted region. This suggests that perhaps polynomial fitting is not the best tool for forecasting weather. Still, it makes for a nice "linear" regression exercise!

Challenge!

In this problem we will investigate a *sparse* regularizer, in which we replace the 2-norm regularizer with a 1-norm regularizer. In other words, given $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $\lambda \in \mathbb{R}$, we will solve

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \tfrac{1}{2m}\|Ax-b\|_2^2 + \lambda\|x\|_1 \tag{4}$$

1. This objective function is composed of a smooth (everywhere differentiable) and nonsmooth (not everywhere differentiable) term. Show that $\|x\|_1$ is nonsmooth by describing all the points x where $g(x) = \|x\|_1$ is not differentiable.

   Ans. $g(x)$ is not differentiable for all points x where any element is 0, e.g. $x[k] = 0$ for some k.

   (That's all I was looking for, but several of you also included proofs, which showed that, fixing all other elements, the limit of $\nabla f(x)[k]$ as $x[k] \to 0$ is 1 if approaching from the positive side, and -1 if approaching from the negative side. Since these two values don't converge to each other, it's a proof that the gradient doesn't exist.)

2. Because the objective has a nonsmooth point, gradient descent will not converge to the global minimum. To see that this is true, consider the case of $m = n = 1$, with $A = b = 1$, $\lambda = 2$. In other words, we consider

   $$\underset{x}{\text{minimize}} \quad \frac{1}{2}(x-1)^2 + 2|x|. \tag{5}$$

   Start with $x^{(0)} = 1$, and with a step size $t = 1$, write out the iterates $x^{(k)}$ for $k = 1, 2, 3, 4$. Is there a limit point $\lim_{k\to+\infty} x^{(k)}$? If so, is this limit point the problem's global minima?

   Ans. For this reduced problem, the iteration scheme can be written simply as

   $$x^{(k+1)} = x^{(k)} - (x^{(k-1)} - 1 + 2\mathbf{sign}(x^{(k)})).$$

   Starting at $x^{(0)} = 1$, we see that

   $$\begin{aligned} x^{(1)} &= 1, \quad \nabla f(x^{(1)}) = 2 \\ x^{(2)} &= -1, \quad \nabla f(x^{(2)}) = -4 \\ x^{(3)} &= 3, \quad \nabla f(x^{(3)}) = 4 \\ x^{(4)} &= -1. \end{aligned}$$

   Once we see this repeated entry, it becomes clear that for all $k \geq 4$, $x^{(k)} \neq 0$, so the limit point is $1/2$.

   However, this is not the global optima. By just using a graphing calculator, we see that the true minimum occurs at $x^* = 0$, and $f(0) = 1/2$.

3. We therefore will introduce a new method called the *proximal gradient descent* method. This method is similar to gradient descent, except we break away the nonsmooth term and deal with it separately. Explicitly, for solving

   $$\underset{x}{\text{minimize}} \quad f(x) + g(x)$$

   where $f(x)$ is smooth and $g(x)$ is nonsmooth, the proximal gradient descent method picks a random point $x^{(0)}$ and iterates

   $$x^{(k+1)} = \mathbf{prox}_{tg}(x^{(k)} - t\nabla f(x^{(t)}))$$

   where the mapping $\mathbf{Pro}x_{tg}$ is the *proximal operator*

   $$\mathbf{prox}_{tg}(z) = \underset{x}{\text{argmin}}\ g(x) + \frac{1}{2t}\|x-z\|_2^2.$$

   We can interpret this as finding the variable x that trades off minimizing the nonsmooth term $g(x)$ and a proximity term (e.g. doesn't want to deviate too far from z).

   Show that the proximal operator of the 1-norm can be computed in closed form, as

   $$\mathbf{prox}_{tg}(z) = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \qquad x_k = \begin{cases} (|z_k| - t)\mathbf{sign}(z_k) & \text{if } |z_k| > t \\ 0 & \text{else.} \end{cases}$$

This operator is called the "shrinkage operator".

Ans. The first step is to show separability. Specifically, the prox operator for $g(x) = \|x\|_1$ can be written the solution to the minimization problem

$$\underset{x}{\text{minimize}} \sum_{i=1}^{n} |x[i]| + \frac{1}{2t}(x[i] - z[i])^2.$$

In general, when minimizing the sum of independent terms, we can separate this into the sum of each minimization, e.g.

$$\min_x \sum_{i=1}^{n} |x[i]| + \frac{1}{2t}(x[i] - z[i])^2 = \sum_{i=1}^{n} \min_{x[i]} \left(|x[i]| + \frac{1}{2t}(x[i] - z[i])^2 \right).$$

So we only need to reduce our attention to the 1-D convex minimization problem of *scalars* $x \in \mathbb{R}$ and $z \in \mathbb{R}$:

$$\min_{x \in \mathbb{R}} \left(|x| + \frac{1}{2t}(x - z)^2 \right).$$

If the solution $x^* = 0$, then for all u,

$$|0| + \frac{1}{2t}(0 - z)^2 = \frac{1}{2t}z^2 \leq |u| + \frac{1}{2t}(u - z)^2 \tag{6}$$

(since x^* is the minimum). Suppose $u > 0$. Then (6) reduces to

$$z \leq t\mathbf{sign}(u) + \frac{1}{2}u = t + \frac{1}{2}u.$$

Taking the limit $u \to 0$, we get $z \leq t$. Now suppose $u < 0$ and take the limit again as $u \to 0$; this gives $z \geq -t$. In other words, if $|z| \leq t$, then $x^* = 0$.

Now suppose the solution $x^* \neq 0$, which requires $|z| > t$. Then this entire objective is differentiable, and has stationary point

$$0 = \mathbf{sign}(x^*) + \frac{1}{t}(x^* - z) \iff x^* = \begin{cases} \max\{z - t, 0\}, & x^* > 0 \\ \min\{z + t, 0\}, & x^* < 0 \end{cases}$$

Finally, we just need to convince ourselves that $\mathbf{sign}(z) = \mathbf{sign}(x^*)$. This is apparent, since the objective

$$|x| + \frac{1}{2t}(x - z)^2 < |-x| + \frac{1}{2t}(-x - z)^2$$

if $\mathbf{sign}(x) = \mathbf{sign}(z)$. Therefore, we know that

$$x^* = \begin{cases} z - t, & z > t \\ z + t, & z < -t. \end{cases}$$

Since the function is convex, then the stationary point is also the minimum. Putting everything together gives us the shrinkage operator.

Note: A lot of you tried to solve this by finding the derivative and set to 0. There are 2 things to note: first, you must claim that the function is convex, or finding the 0 gradient point doesn't help you. Second, you must avoid taking derivatives at points which you already declared as non-differentiable. Some of you did this by separating the $x[k] = 0$ case and the $x[k] \neq 0$ case.

Note: Some of you tried to solve this problem using *subdifferentials*, which is admittedly a more elegant solution, but I believe there may have been some confusion over what a subdifferential actually is. I will go over it briefly here, but recognize that this is not core material.

- Consider a convex function $f : \mathbb{R}^n \to \mathbb{R}$, and suppose that f is differentiable at x. Then we know from our first order condition that

$$\nabla f(x)^T(x - y) \geq f(x) - f(y), \quad \forall y.$$

- Now suppose that f is convex but not differentiable at x. Then more than one such vector satisfies this constraint. So, we define a *subgradient* of f at x as any vector g which satisfies

$$g^T(x-y) \geq f(x) - f(y), \quad \forall y.$$

The set of all subgradients is the subdifferential, denoted $\partial f(x)$.

- For example, in the case of the absolute value function, $f(x) = |x|$ and f is not differentiable at $x = 0$. Consider all g where

$$g \cdot (0-y) \geq |0| - |y| \iff |y| \geq -gy \quad \forall y.$$

Well, if $y > 0$ then this can only be true if $g \geq -1$ and if $y < 0$, then $g \leq 1$. The relation is trivially true if $y = 0$. Therefore, the subdifferential $\partial f(0) = [-1, 1]$ (the interval).

- You can also calculate the subdifferential in terms of limits. Assume that f is not differentiable at x, but every point around x is differentiable. Then

$$\partial f(x) = \{\lim_{\alpha \to 0} \partial f(x + \alpha d) : d \in \mathbb{R}^n, \ \|d\|_2 = 1\}.$$

This is sometimes called the *Clarke subdifferential*, and is sometimes used in cases where f is not necessarily convex. (Note that no convex assumptions are used.)

- So, how does this proof work? Well, I alluded that you cannot minimize a function by setting its gradient to 0 if that function is not differentiable where you think the solution is. But, the *subdifferential* always exists! So, suppose f is convex. If f is differentiable at $x = x^*$, and $0 = \nabla f(x^*)$, then $x^* = \operatorname{argmin}_x f(x)$. If, however, f is not differentiable at x^*, then we write

$$0 \in \partial f(x^*) \iff x^* = \operatorname*{argmin}_x f(x).$$

Note that this is a particularly elegant formulation, because even if f *is* differentiable at x^*, then $\partial f(x^*)$ *does* exist as well. (What does it equal, do you think?)

- Ok now let's look at our prox scalar problem. We want to find the argmin of the scalar convex function

$$h(x) = |x| + \frac{1}{2t}(x-z)^2.$$

The subdifferential of h is

$$\partial h(x) = \frac{x-z}{t} + \begin{cases} \{1\} & \text{if } x > 0 \\ \{-1\} & \text{if } x < 0 \\ [-1,1] & \text{if } x = 0. \end{cases}$$

Let's look at each case separately. In the first case,

$$0 \in \frac{x-z}{t} + \{1\} \iff 0 = \frac{x-z}{t} + 1 \iff x = z - t.$$

Ok, so $x^* = z - t$ whenever $x^* = z - t > 0$, e.g. whenever $z > t$.
Similarily, the second case tells us that $x^* = z + t$ whenever $z < -t$.
The third case takes care of 0 for us nicely. It says that

$$0 \in \frac{x-z}{t} + [-1,1] \iff z - x \in [-t, t]$$

whenever $x = 0$; or, to spin the convoluted logic another way, $x = 0$h whenever $z \in [-t, t]$.
And there you have it! all 3 conditions!

4. Again consider the scalar problem (5). Start with $x^{(0)} = 2$, and with a step size $t = 1/2$, write out the iterates $x^{(k)}$ for $k = 1, 2, 3$, but following the proximal gradient scheme. What is the limit point $\lim_{k \to +\infty} x^{(k)}$? Is this limit point the problem's global minima?

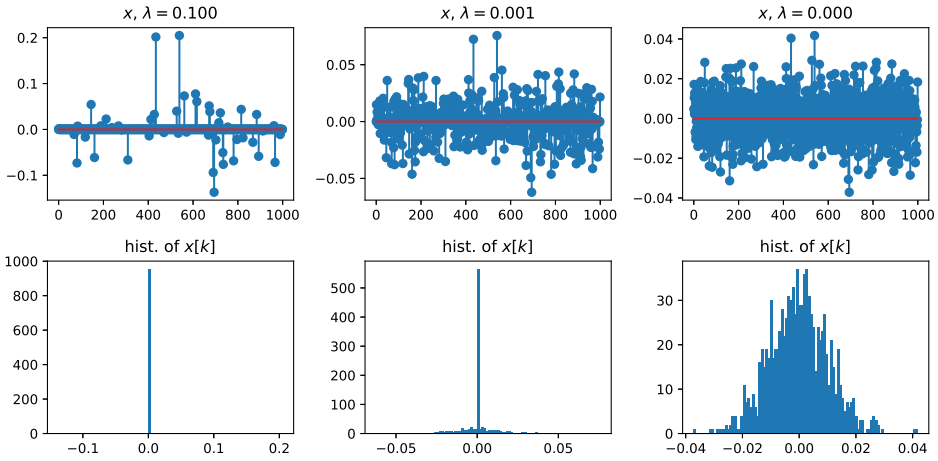
Ans.

$$\begin{aligned} x^{(0)} &= 2 \\ x^{(1)} &= \mathbf{prox}_{1/2\,\|\cdot\|_1}(2-1) = 1/2 \\ x^{(2)} &= \mathbf{prox}_{1/2\,\|\cdot\|_1}(1/2-1/2) = 0 \\ x^{(3)} &= \mathbf{prox}_{1/2\,\|\cdot\|_1}(0+1/2) = 0 \end{aligned}$$

In fact, we reach the limit point and global minimum $x^* = 0$ in two steps.

5. **Coding.** In MATLAB or Python, generate a sample problem with `A = randn(m,n)` and `b = randn(m,1)`. Pick $m = 100$, $n = 1000$. Solve (4). For $\lambda = 0, 0.001, 0.1$, show histograms of the elements of x^*. Comment on the sparsifying property of the 1-norm.

   Ans.



As λ grows, the distribution of the elements becomes "spiky", e.g. a lot of elements are exactly 0.

Comparison with 2-norm regularization. We can also consider a 2-norm regularized version as well, where we solve

$$\underset{x}{\text{minimize}} \quad \frac{1}{2m}\|Ax-b\|_2^2 + \lambda\|x\|_2 \tag{7}$$

6. Show that this regularization term $\|x\|_2$ is also nonsmooth.

   Ans. The gradient of $\|x\|_2$ exists only if $x \neq 0$. In particular, for nonzero x,

$$\nabla g(x) = \frac{1}{\|x\|_2}x$$

   which is undefined if $\|x\|_2 = 0$.

7. Derive the proximal operator $\mathbf{prox}_{tg}$ for $g(x) = \|x\|_2$.

   Ans. We are now after the solution to the minimization problem

$$\min_x \ \|x\|_2 + \frac{1}{2t}\|x-z\|_2^2.$$

   First, suppose that we bound the norm of x. Then, the "direction" of x should face the direction of z; that is, $x = \eta z$ for some η. More explicitly,

$$\underset{x:\|x\|_2=1}{\text{argmin}} \ \underbrace{\|x\|_2}_{\text{doesn't matter}} + \frac{1}{2t}\|x-z\|_2^2 = \underset{x:\|x\|_2=1}{\text{argmin}} \ (-2x^Tz)$$

which is minimized for $x = z/\|z\|_2$. (Think cosine inequality.)

So, more generally, we just fix this parametrization $x = \eta z$. Then the minimization is only over a 1-D scalar:

$$\min_x \ \|x\|_2 + \frac{1}{2t}\|x - z\|_2^2 \quad \rightarrow \quad \min_{\eta \geq 0} \ \underbrace{\eta\|z\|_2 + \frac{(\eta - 1)^2}{2t}\|z\|_2^2}_{h(\eta)}$$

and this function is smooth (though the domain is constrained). Setting the derivative $h'(\eta) = 0$ yields $\eta = 1 - \frac{t}{\|z\|_2}$, but this solution is only feasible if $t \leq \|z\|_2$. In the case that $t > \|z\|_2$, then

$$h'(\eta) = 1 + \underbrace{\frac{\|z\|_2}{t}}_{<1}(\eta - 1) > 0 \ \forall \eta \geq 0$$

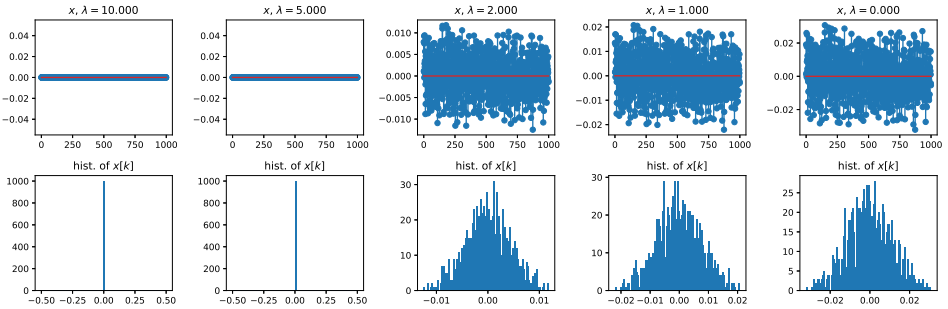
so we know that h is a scalar function that is always increasing over the domain $\eta \geq 0$. Therefore, the minimum must be achieved at $\eta = 0$.

Putting it all together, we get

$$x = \max\left\{0, 1 - \frac{t}{\|z\|_2}\right\} \cdot z.$$

8. Use proximal gradient descent to solve (7), using the same choices of A and b as in the previous section. Histogram the final solutions x^* for $\lambda = 0, 1, 2, 5, 10$. Comment on the sparsifying properties of the 2-norm vs the 1-norm.

   Ans.



Basically, compared to the 1-norm case, there are no sparsifying properties of the 2-norm. It has 2 modes: elements of x distributed agnostically to λ, or $x = 0$.