ECE 6250, Fall 2016

Homework # 11

Due Monday December 5, in class

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

1. Using you class notes, prepare a 1-2 paragraph summary of what we talked about in class in the last week. I do not want just a bulleted list of topics, I want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other things you have learned here or in other classes?). The more insight you give, the better.

2. Please go to http://gatech.smartevals.com and fill out survey for ECE 6250. I would very much appreciate your feedback; I do read and consider carefully the comments for every course that I teach.

3. The file `hw11problem3.mat` contains the input (sequence `u`) and output (sequencey) of an unknown linear time-invariant system whose impulse response is supported on $n = 0, \ldots, 18$.

   (a) Implement the LMS algorithm and use it to estimate the impulse response. Turn in a plot of your estimated filter coefficients. (You can tell how well you did by seeing how well their convolution with `u` agrees with `y`.

   (b) Try different values of the stepsize μ and comment on the effect it has on the speed of convergence (and whether or not it converges). You might gauge this by how many iterations it takes to get a relative error $\|\boldsymbol{h}_n - \boldsymbol{h}_*\|_2/\|\boldsymbol{h}_*\|_2 \leq 0.01$.

   (c) Now add a little bit of noise to the observations, using `yn = y + sigma*randn(length(y),1)`. Comment on the convergence speed when `sigma = 0.05`. You might have to adjust your step size to ensure convergence.

4. Suppose we make a noisy observation of $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, with

$$\boldsymbol{A} = \begin{bmatrix} 2 & 4 & -1 \\ 1 & -2 & 1 \\ 4 & 0 & 1 \\ 5 & 6 & -1 \\ 8 & -4 & 2 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 1 \\ 2 \\ -1 \\ -2 \\ 5 \end{bmatrix}$$

   (a) Find the total-least squares solution to the above linear inverse problem. (Use MATLAB.)

   (b) What is the residual error $\|\boldsymbol{\Delta}\|_F^2$? What are the $\boldsymbol{\Delta A}$ and $\boldsymbol{\Delta y}$ corresponding to your solution?

Last updated 9:14, December 6, 2016

5. Let $\boldsymbol{A}$ be an $M \times N$ matrix with full column rank. Given $\boldsymbol{y} \in \mathbb{R}^M$, we solve

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_\infty.$$

As we have discussed intuitively in class, the ℓ_∞ norm will encourage the error $\hat{\boldsymbol{e}} = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}$ of any solution $\hat{\boldsymbol{x}}$ to be "flat" (many entries of $\hat{\boldsymbol{e}}$ will have the same value). Prove that there exists a solution to the program above such that the corresponding error has at least its $N+1$ largest terms equal in magnitude.

6. I have a secret message of incredible importance that I would like to communicate to you. The message is a 74-byte string of characters, which are mapped to numbers using the standard ASCII encoding ('a'=97, 'b'=98, etc). Given this vectors of numbers $\boldsymbol{x}$, you can view the message in MATLAB using

`char(round(x))'.`

Unfortunately, we have adversaries impeding our ability communicate. These adversaries can change up to one out of every ten numbers I send to you; these changes are arbitrary and the receiver (you) has no knowledge of where they might take place. As I really need you to read all 74 bytes, even having as few as ≈ 5 of them altered is unacceptable.

To protect against this attack, we add redundancy to the message by embedding it into a higher dimensional space. Specifically, instead of sending $\boldsymbol{x}_0 \in \mathbb{R}^{74}$, I will send $\boldsymbol{y} \in \mathbb{R}^{592}$, where $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0$, and $\boldsymbol{A}$ is a 592×74 matrix. You observe

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{e}$$

where $\boldsymbol{e} \in \mathbb{R}^{592}$ can have up to ≈ 60 non-zeros entries. That is, the error between what we have ($\boldsymbol{y}$) and what we want ($\boldsymbol{V}\boldsymbol{x}_0$ — the embedded message) is sparse. We have seen that ℓ_1 norm approximation is likely to produce sparse errors. So you will attempt to recover the message by solving

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1.$$

(a) The file `SecretMessage1.mat` contains a (corrupted) transmission $\boldsymbol{y}$ and the matrix $\boldsymbol{A}$ I used to encode the message. Try to decode the message using least-squares — what do you get?

(b) Decode the message using the file `l1approx.m` that I have provided.

(c) Play around with this encoding/decoding scheme with this particular choice of $\boldsymbol{A}$. Make up some 74-byte messages and encode them; for example:

```
msgstr = 'This is my funny 74 byte message blah blah blah blah blah
blah blah blah..';
x0 = double(msgstr)';
y0 = A*x0;
```

then choose some entries to change and change them to random numbers:

```
q = randperm(592);
S = 60; % number of entries to alter
T = sort(q(1:S)); % indexes of entries to alter
```

```
e = zeros(592,1);
e(T) = 25*randn(S,1); % random, sparse error
y = y0 + e;
```

About how many errors can this encoding $\boldsymbol{A}$ correct on average? (Just do a lot of experiments for different values of `S` to figure this out.)

(d) What are the messages in `SecretMessage2.mat`, `SecretMessage3.mat`, `SecretMessage4.mat`, and `SecretMessage5.mat`? You can get the values of M and N by looking at the dimensions of $\boldsymbol{A}$.

Last updated 9:14, December 6, 2016