ECEN 615
Methods of Electric Power Systems Analysis

Lecture 17: Least Squares, State Estimation

Prof. Tom Overbye

Dept. of Electrical and Computer Engineering

Texas A&M University

overbye@tamu.edu

TEXAS A&M UNIVERSITY

Announcements

- Read Chapter 9 from the book
- Homework 4 is due on Thursday October 31.

Least Squares Solution

- We write $(\mathbf{a}^i)^T$ for the row i of $\mathbf{A}$ and $\mathbf{a}^i$ is a column vector
- Here, $m \geq n$ and the solution we are seeking is that which minimizes $\|\mathbf{Ax} - \mathbf{b}\|_p$, where p denotes some norm
- Since usually an overdetermined system has no exact solution, the best we can do is determine an $\mathbf{x}$ that minimizes the desired norm.

Choice of *p*

- We discuss the choice of *p* in terms of a specific example
- Consider the equation **Ax** = **b** with

$$\mathbf{A} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \qquad \text{with } b_1 \geq b_2 \geq b_3 \geq 0$$

(hence three equations and one unknown)

- We consider three possible choices for *p*:

Choice of *p*

(*i*) $p = 1$

$\|\mathbf{Ax} - \mathbf{b}\|_1$ **is minimized by** $x^* = b_2$

(*ii*) $p = 2$

$\|\mathbf{Ax} - \mathbf{b}\|_2$ **is minimized by** $x^* = \dfrac{b_1 + b_2 + b_3}{3}$

(*iii*) $p = \infty$

$\|\mathbf{Ax} - \mathbf{b}\|_\infty$ **is minimized by** $x^* = \dfrac{b_1 + b_3}{2}$

The Least Squares Problem

- In general, $\| \mathbf{Ax} - \mathbf{b} \|_p$ is not differentiable for $p = 1$ or $p = \infty$
- The choice of $p = 2$ (Euclidean norm) has become well established given its least-squares fit interpretation
- The problem $\min_{\mathbf{x} \in \mathbb{R}^n} \| \mathbf{Ax} - \mathbf{b} \|_2$ is tractable for 2 major reasons
  - First, the function is differentiable

$$\phi(\mathbf{x}) = \frac{1}{2} \| \mathbf{Ax} - \mathbf{b} \|_2^2 = \frac{1}{2} \sum_{i=1}^{m} \left[\left(\mathbf{a}^i \right)^T \mathbf{x} - b_i \right]^2$$

The Least Squares Problem, cont.

– Second, the Euclidean norm is preserved under orthogonal transformations:

$$\left\| \left(\mathbf{Q}^T\mathbf{A}\right)\mathbf{x} - \mathbf{Q}^T\mathbf{b} \right\|_2 = \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|_2$$

with **Q** an arbitrary orthogonal matrix; that is, **Q** satisfies

$$\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I} \qquad \mathbf{Q} \in \mathbb{R}^{n \times n}$$

The Least Squares Problem, cont.

- We introduce next the basic underlying assumption: **A** is full rank, i.e., the columns of **A** constitute a set of linearly independent vectors
- This assumption implies that the rank of **A** is n because $n \leq m$ since we are dealing with an overdetermined system
- Fact: The least squares solution $\mathbf{x}^*$ satisfies

$$\mathbf{A}^T \mathbf{A}\, \mathbf{x}^* = \mathbf{A}^T \mathbf{b}$$

Proof of Fact

- Since by definition the least squares solution $\mathbf{x}^*$ minimizes $\phi(\bullet)$ at the optimum, the derivative of this function zero:

$$\phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{Ax}-\mathbf{b}\|_2^2 = \frac{1}{2}\left(\mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - \mathbf{x}^T\mathbf{A}^T\mathbf{b} - \mathbf{b}^T\mathbf{A}\,\mathbf{x} + \mathbf{b}^T\mathbf{b}\right)$$

$$\mathbf{0} = \left.\frac{\partial\,\phi(\mathbf{x})}{\partial\mathbf{x}}\right|_{\mathbf{x}^*} = \frac{\partial}{\partial\mathbf{x}}\left\{\frac{1}{2}\left(\mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - \mathbf{x}^T\mathbf{A}^T\mathbf{b} - \mathbf{b}^T\mathbf{A}\,\mathbf{x} + \mathbf{b}^T\mathbf{b}\right)\right\}\Bigg|_{\mathbf{x}^*}$$

$$= \frac{\partial}{\partial\mathbf{x}}\left\{\frac{1}{2}\left(\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{x}^T\mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}\right)\right\}\Bigg|_{\mathbf{x}^*}$$

$$= \mathbf{A}^T\mathbf{A}\mathbf{x}^* - \mathbf{A}^T\mathbf{b}$$

Implications

- This underlying assumption implies that

  $\mathbf{A}$ is full rank $\Leftrightarrow \exists\ \mathbf{x} \neq \mathbf{0}\ \ni\ \mathbf{Ax} \neq \mathbf{0}$

- Therefore, the fact that $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is positive definite (*p.d.*) follows from considering any $\mathbf{x} \neq \mathbf{0}$ and evaluating

  $$\mathbf{x}^{T}\mathbf{A}^{T}\mathbf{A}\,\mathbf{x} = \left\| \mathbf{A}\mathbf{x} \right\|_2^2 > 0\,,$$

  which is the definition of a *p.d.* matrix

- We use the shorthand $\mathbf{A}^{\mathrm{T}}\mathbf{A} > \mathbf{0}$ for $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ being a symmetric, positive definite matrix

Implications

- The underlying assumption that **A** is full rank and therefore $\mathbf{A}^T\mathbf{A}$ is *p.d.* implies that there exists a unique least squares solution
- Note: we use the inverse in a conceptual, rather than a computational, sense

$$\mathbf{x}^* = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$$

- The below formulation is known as the normal equations, with the solution conceptually straightforward

$$\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

Example: Curve Fitting

- Say we wish to fit five points to a polynomial curve of the form

$$\mathrm{f}(t,\mathbf{x}) = x_1 + x_2 t + x_3 t^2$$

- This can be written as

$$\mathbf{Ax} = \mathbf{y} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \\ 1 & t_4 & t_4^2 \\ 1 & t_5 & t_5^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

Example: Curve Fitting

- Say the points are **t** =[0,1,2,3,4] and **y** = [0,2,4,5,4]. Then

$$\mathbf{Ax} = \mathbf{y} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 4 \\ 5 \\ 4 \end{bmatrix}$$

$$\mathbf{x}^* = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} = \begin{bmatrix} 0.886 & 0.257 & -0.086 & -0.143 & 0.086 \\ -0.771 & 0.186 & 0.571 & 0.386 & -0.371 \\ 0.143 & -0.071 & -0.143 & -0.071 & 0.143 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 4 \\ 5 \\ 4 \end{bmatrix}$$

$$\mathbf{x}^* = \begin{bmatrix} -0.2 \\ 3.1 \\ -0.5 \end{bmatrix}$$

Implications

- An important implication of positive definiteness is that we can factor $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ since $\mathbf{A}^{\mathrm{T}}\mathbf{A} > \mathbf{0}$

$$\mathbf{A}^{T}\mathbf{A} = \mathbf{U}^{T}\mathbf{D}\,\mathbf{U} = \mathbf{U}^{T}\mathbf{D}^{1/2}\mathbf{D}^{1/2}\mathbf{U} = \mathbf{G}^{T}\mathbf{G}$$

- The expression $\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{G}^{\mathrm{T}}\mathbf{G}$ is called the Cholesky factorization of the symmetric positive definite matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$

A Least Squares Solution Algorithm

Step 1: Compute the lower triangular part of $\mathbf{A}^T\mathbf{A}$

Step 2: Obtain the Cholesky Factorization $\mathbf{A}^T\mathbf{A} = \mathbf{G}^T\mathbf{G}$

Step 3: Compute $\mathbf{A}^T\mathbf{b} = \hat{\mathbf{b}}$

Step 4: Solve for **y** using forward substitution in

$$\mathbf{G}^T\mathbf{y} = \hat{\mathbf{b}}$$

and for **x** using backward substitution in

$$\mathbf{G}\,\mathbf{x} = \mathbf{y}$$

Note, our standard LU factorization approach would work; we can just solve it twice as fast by taking advantage of it being a symmetric matrix

Practical Considerations

- The two key problems that arise in practice with the triangularization procedure are:
  - First, while **A** maybe sparse, $\mathbf{A}^T\mathbf{A}$ is much less sparse and consequently requires more computing resources for the solution
    - In particular, with $\mathbf{A}^T\mathbf{A}$ **s**econd neighbors are now connected! Large networks are still sparse, just not as sparse
  - Second, $\mathbf{A}^T\mathbf{A}$ may actually be numerically less well-conditioned than **A**

Loss of Sparsity Example

- Assume the **B** matrix for a network is

$$\mathbf{B} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

- Then $\mathbf{B}^T\mathbf{B}$ is $\mathbf{B}^T\mathbf{B} = \begin{bmatrix} 2 & -3 & 1 & 0 \\ -3 & 6 & -4 & 1 \\ 1 & -4 & 6 & -3 \\ 0 & 1 & -3 & 2 \end{bmatrix}$

- Second neighbors are now connected!

Numerical Conditioning

- To understand the point on numerical ill-conditioning, we need to introduce terminology
- We define the norm of a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ to be

$$\|\mathbf{B}\| = \max_{\mathbf{x} \neq \mathbf{0}} \left\{ \frac{\|\mathbf{B}\mathbf{x}\|}{\|\mathbf{x}\|} \right\}$$

$$= \textbf{\textit{maximum stretching of the matrix}} \ \mathbf{B}$$

- This is the maximum singular value of **B**

Numerical Conditioning Example

- Say we have the matrix

$$\mathbf{B} = \begin{bmatrix} 10 & 0 \\ 0 & 0.1 \end{bmatrix}$$

- What value of **x** with a norm of 1 that maximizes $\|\mathbf{Bx}\|$?
- What value of **x** with a norm of 1 that minimizes $\|\mathbf{Bx}\|$?

$$\|\mathbf{B}\| = \max_{\mathbf{x} \neq \mathbf{0}} \left\{ \frac{\|\mathbf{B}\,\mathbf{x}\|}{\|\mathbf{x}\|} \right\}$$

$$= \textit{maximum stretching of the matrix } \mathbf{B}$$

Numerical Conditioning

$$= \max_i \left\{ \sqrt{\lambda}_i \,,\ \lambda_i \text{ is an eigenvalue of } \underline{B}^T \underline{B} \right\},$$

i.e., λ_i is a root of the polynomial

$$p(\lambda) = det\left[\mathbf{B}^T\mathbf{B} - \lambda\mathbf{I}\right]$$

Keep in mind the eigenvalues of a p.d. matrix are positive

- In other words, the ℓ_2 norm of **B** is the square root of the largest eigenvalue of $\mathbf{B}^T\mathbf{B}$

Numerical Conditioning

- The conditioning number of a matrix **B** is defined as

$$\kappa(\mathbf{B}) = \|\mathbf{B}\|\,\|\mathbf{B}^{-1}\| = \left.\frac{|\sigma_{max}(\mathbf{B})|}{|\sigma_{min}(\mathbf{B})|}\right\}$$

***the max / min stretching ratio of the matrix* B**

- A well–conditioned matrix has a small value of $\kappa(\mathbf{B})$, close to 1; the larger the value of $\kappa(\mathbf{B})$, the more pronounced is the ill-conditioning

Power System State Estimation

- Overall goal is to come up with a power flow model for the present "state" of the power system based on the actual system measurements
- SE assumes the topology and parameters of the transmission network are mostly known
- Measurements come from SCADA, and increasingly, from PMUs

Power System State Estimation

- Problem can be formulated in a nonlinear, weighted least squares form as

$$\min J(\mathbf{x}) = \sum_{i=1}^{m} \frac{\left[z_i - f_i(\mathbf{x}) \right]^2}{\sigma_i^2}$$

where J(**x**) is the scalar cost function, **x** are the state variables (primarily bus voltage magnitudes and angles), z_i are the m measurements, **f**(**x**) relates the states to the measurements and σ_i is the assumed standard deviation for each measurement

Assumed Error

- Hence the goal is to decrease the error between the measurements and the assumed model states **x**
- The σ_i term weighs the various measurements, recognizing that they can have vastly different assumed errors

$$\min J(\mathbf{x}) = \sum_{i=1}^{m} \frac{\left[z_i - f_i(\mathbf{x})\right]^2}{\sigma_i^2}$$

- Measurement error is assumed Gaussian (whether it is or not is another question); outliers (bad measurements) are often removed

State Estimation for Linear Functions

- First we'll consider the linear problem. That is where

$$\mathbf{z}^{meas} - \mathbf{f}(\mathbf{x}) = \mathbf{z}^{meas} - \mathbf{H}\mathbf{x}$$

- Let **R** be defined as the diagonal matrix of the variances (square of the standard deviations) for each of the measurements

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_m^2 \end{bmatrix}$$

State Estimation for Linear Functions

- We then differentiate J(**x**) w.r.t. **x** to determine the value of **x** that minimizes this function

$$J(\mathbf{x}) = \left[\mathbf{z}^{meas} - \mathbf{H}\mathbf{x}\right]^T \mathbf{R}^{-1} \left[\mathbf{z}^{meas} - \mathbf{H}\mathbf{x}\right]$$

$$\nabla J(\mathbf{x}) = -2\mathbf{H}^T \mathbf{R}^{-1} \mathbf{z}^{meas} + 2\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}\mathbf{x}$$

At the minimum we have $\nabla J(\mathbf{x}) = \mathbf{0}$. So solving for $\mathbf{x}$ gives

$$\mathbf{x} = \left[\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}\right]^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z}^{meas}$$

Simple DC System Example

- Say we have a two bus power system that we are solving using the dc approximation. Say the line's per unit reactance is j0.1. Say we have power measurements at both ends of the line. For simplicity assume **R**=**I**. We would then like to estimate the bus angles. Then

$$z_1 = P_{12} = \frac{\theta_1 - \theta_2}{0.1} = 2.2, \quad z_2 = -2.0 = P_{21} = \frac{\theta_2 - \theta_1}{0.1}$$

$$\mathbf{x} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 10 & -10 \\ -10 & 10 \end{bmatrix}, \mathbf{H}^T\mathbf{H} = \begin{bmatrix} 200 & -200 \\ -200 & 200 \end{bmatrix}$$

We have a problem since $\mathbf{H}^T\mathbf{H}$ is singular. This is because of lack of an angle reference.

Simple DC System Example, cont.

- Say we directly measure θ_1 (with a PMU) to be zero; set this as the third measurement. Then

$$z_1 = P_{12} = \frac{\theta_1 - \theta_2}{0.1} = 2.2, \quad z_2 = -2.0 = P_{21} = \frac{\theta_2 - \theta_1}{0.1}, \quad z_3 = 0$$

$$\mathbf{x} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} 2.2 \\ -2 \\ 0 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 10 & -10 \\ -10 & 10 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{H}^T\mathbf{H} = \begin{bmatrix} 201 & -200 \\ -200 & 200 \end{bmatrix}$$

$$\mathbf{x} = \left[\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}^{meas}$$

$$\mathbf{x} = \begin{bmatrix} 201 & -200 \\ -200 & 200 \end{bmatrix}^{-1} \begin{bmatrix} 10 & -10 & 1 \\ -10 & 10 & 0 \end{bmatrix} \begin{bmatrix} 2.2 \\ -2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.21 \end{bmatrix}$$

Note that the angles are in radians

Nonlinear Formulation

- A regular ac power system is nonlinear, so we need to use an iterative solution approach. This is similar to the Newton power flow. Here assume m measurements and n state variables (usually bus voltage magnitudes and angles) Then the Jacobian is the **H** matrix

$$\mathbf{H}(\mathbf{x}) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{x_1} & \cdots & \frac{\partial f_1}{x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{x_1} & \cdots & \frac{\partial f_m}{x_n} \end{bmatrix}$$