Unit 12 – Regularization

Adding an Identity Matrix

- Motivated by the minimal normal solution, one may add new equations of the form $Ic=0$, i.e. $\begin{pmatrix} A \\ I \end{pmatrix} \begin{pmatrix} c \\ b \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$, with the aim of driving to zero whatever could go to zero.
- If the original problem had a least squares (or unique) solution, adding $Ic=0$ creates contradictions modifying the solution incorrectly.
- But in the undetermined case, adding $Ic=0$ aids tremendously by making the resulting columns linearly independent giving the resulting system full rank.
- Thus, one can minimize the residual in the L2 norm (least squares) via $(A^T \ I) \begin{pmatrix} A \\ I \end{pmatrix} \begin{pmatrix} c \\ b \end{pmatrix} = (A^T \ I) \begin{pmatrix} b \\ 0 \end{pmatrix}$ or $(A^T A + I)c = A^T b$ adding an identity matrix to the least squares equations.
- Using $A = U\Sigma V^T$ leads to $(V\Sigma^T V^T + I)c = V\Sigma^T U^T b$ or $(\Sigma^T \Sigma + I)\hat{c} = \Sigma^T \hat{b}$.
- The identity matrix gets added to the diagonal matrix $\Sigma^T \Sigma$ modifying its entries to $\sigma_i^2 + 1$.
- This has a modest effect on large singular values, but a large effect on small singular values.

Full Rank Scenario

- For a full rank $A$, the least squares (or unique) solution is $\hat{c}_i = \frac{1}{\sigma_i} \hat{b}_i$.
- Adding $Ic=0$ gives instead $(\Sigma^T \Sigma + I)\hat{c} = \Sigma^T \hat{b}$ and $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + 1} \hat{b}_i$ perturbing the solution to no longer be the correct least squares (or unique) solution.
- Although this has a modest effect on the parts of the solution corresponding to large singular values, it has a large effect on the parts of the solution corresponding to small singular values.
- Large condition number matrices with very small singular values can be stabilized with such an approach.
- Note: adding a multiple of the identity, i.e. $\epsilon Ic = 0$ for $\epsilon > 0$, leads to $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} \hat{b}_i$ helping to stabilize the $\sigma_i$ that are not significantly larger than $\epsilon$.

Rank Deficient Scenario

- When $A$ is rank deficient, denote $\Sigma = (\Sigma \ \ 0)$ to separate the linearly independent columns from the columns of zeros, so that $\Sigma^T \Sigma \hat{c} = \Sigma^T \hat{b}$ becomes $\begin{pmatrix} \Sigma^T \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{c} \\ 0 \end{pmatrix} = \begin{pmatrix} \Sigma^T \hat{b} \\ 0 \end{pmatrix}$.
- Once again, $\hat{c}_i = \frac{1}{\sigma_i} \hat{b}_i$ for the $\sigma_i \neq 0$ as expected, and a minimum norm approach would choose to set the other (undetermined) $\hat{c}_i$ to zero.
- Meanwhile, $(\Sigma^T \Sigma + I)\hat{c} = \Sigma^T \hat{b}$ becomes $\begin{pmatrix} \Sigma^T \Sigma & 0 \\ 0 & 0 \end{pmatrix} + 1 \begin{pmatrix} \hat{c} \\ 0 \end{pmatrix} = \begin{pmatrix} \Sigma^T \hat{b} \\ 0 \end{pmatrix}$.
- Here, $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + 1} \hat{b}_i$ for the $\sigma_i \neq 0$ just as in the full rank case, but this equation also automatically sets all the other $\hat{c}_i$ to zero (instead of leaving them undetermined).
- Once again, adding $\epsilon Ic = 0$ with $\epsilon > 0$ gives $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} \hat{b}_i$ helping to stabilize $\sigma_i$ that are not significantly larger than $\epsilon$ (while still automatically setting the desired $\hat{c}_i$ to zero).
Initial Guess

- Instead of guessing that $c = 0$ aiming for the minimal norm solution, one might have an initial guess with some of the $c_i \neq 0$
- Adding $Ic = c_{\text{guess}}$ gives $(A^T A + I)c = A^T b + c_{\text{guess}}$
- This leads to $(\Sigma^T \Sigma + I) \hat{c} = \Sigma^T \hat{b} + V^T c_{\text{guess}}$ or $\hat{c} = (\Sigma^T \Sigma + I)^{-1}(\Sigma^T \hat{b} + c_{\text{guess}})$
- This gives $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} \hat{b}_i + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2} (\hat{c}_{\text{guess}})_i$ for the $\sigma_i \neq 0$ , which tends towards $\frac{1}{\sigma_i} \hat{b}_i$ for larger $\sigma_i$ as desired, but tends toward $(\hat{c}_{\text{guess}})_i$ for smaller $\sigma_i$
- In the rank deficient scenario, one automatically obtains $\hat{c}_i = (\hat{c}_{\text{guess}})_i$ for the underdetermined $\hat{c}_i$
- Adding $\epsilon Ic = \epsilon c_{\text{guess}}$ with $\epsilon > 0$ gives $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} \hat{b}_i + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2} \hat{c}_{\text{guess}})_i$ driving the small $\sigma_i$ as compared to $\epsilon^2$ towards $(\hat{c}_{\text{guess}})_i$

Iterative Approach

- First solve with $Ic=0$, and call the result $c_{\text{guess}}$ so that $(\hat{c}_{\text{guess}})_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} \hat{b}_i$ when $\sigma_i \neq 0$ and $(\hat{c}_{\text{guess}})_i = 0$ otherwise
- Then preform as second solve with $Ic = c_{\text{guess}}$ instead of $Ic=0$
- This gives $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} \hat{b}_i + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2} (\hat{c}_{\text{guess}})_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} (1 + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}) \hat{b}_i$ when $\sigma_i \neq 0$, but still gives $\hat{c}_i = (\hat{c}_{\text{guess}})_i$ otherwise
- Repeated solves do not alter the $\hat{c}_i = (\hat{c}_{\text{guess}})_i$ case where there is no corresponding $\sigma_i \neq 0$
- The 3rd solve gives $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} \hat{b}_i + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2} \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} (1 + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}) \hat{b}_i$
- This is equivalent to $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} (1 + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2} + \left(\frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}\right)^2) \hat{b}_i$
- The 4th solve gives $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} (1 + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2} + \left(\frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}\right)^2 + \left(\frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}\right)^3) \hat{b}_i$
- Leading to $\hat{c}_i = \frac{\sigma_i}{\sigma_i^2 + \epsilon^2} (1 + \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2} + \left(\frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}\right)^2 + \left(\frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}\right)^3 + \cdots) \hat{b}_i$ where the term in parenthesis is a geometric series with $r = \frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}$ that converges to $\frac{1}{1-r}$ or $\frac{\sigma_i^2 + \epsilon^2}{\sigma_i^2 + \epsilon^2}$
- Thus, $\hat{c}_i$ properly converges to $\frac{1}{\sigma_i} \hat{b}_i$ for the $\sigma_i \neq 0$
- After n terms $\frac{1-r^n}{1-r} = 1 - \left(\frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}\right)^n \frac{\sigma_i^2 + \epsilon^2}{\sigma_i^2}$, and thus $\hat{c}_i = \frac{1}{\sigma_i} \hat{b}_i \left(1 - \left(\frac{\epsilon^2}{\sigma_i^2 + \epsilon^2}\right)^n\right)$ so the convergence is monotonic increasing from $\hat{c}_i = 0$ towards $\hat{c}_i = \frac{1}{\sigma_i} \hat{b}_i$
- The convergence is quick for small r (i.e. $\sigma_i$ large compared to $\epsilon$), which is good news
- The convergence is slow for r is closer to 1 (i.e. $\sigma_i$ small compared to $\epsilon$), which is fine in the sense that being closer to the initial guess of $\hat{c}_i = 0$ is regularized via minimum norm and also somewhat discarded as it would be in PCA as well
Adding a Diagonal Matrix

- One may feel more strongly about weighting some variables towards zero than others, in which case one might add a diagonal matrix times the variables, i.e. $Dc=0$, or $(A D) c = (b_0)$, in which case the full rank system has normal equations of $(A^T A + D^2)c = A^T b$ leading to $(\Sigma^T \Sigma + \nu^T D^2 \nu)\hat{c} = \Sigma^T \hat{b}$
- Manipulating the original system to use these variables $(\Sigma D V) \hat{c} = \begin{pmatrix} \hat{b} \\ 0 \end{pmatrix}$ results in the same normal equations
- Here, $DV$ shears the coordinate system $V$ is written in creating issues
- A better approach would be to column scale before proceeding $(AD^{-1}) Dc = (b_0)$ or $(\tilde{A} I) \tilde{c} = (b_0)$ where $\tilde{A} = AD^{-1}$ and $\tilde{c} = Dc$
- Then the discussion proceeds as above adding $I\tilde{c} = 0$ to $\tilde{A}\tilde{c} = b$

Column space search method

- $Ac=b$ indicates that a linear combination of the columns of $A$ give the right hand side $b$
- Geometrically, the columns of $A$ are vectors, and adding multiples of those vectors together gives a new vector $b$
- Solving $Ac=b$, or approximating it, involves specifying a weight $c_i$ on each column of $A$
- Thinking geometrically, one doesn’t care about the rank of $A$ with this approach
- Other concerns, may be more important:
  - Using as few columns as possible
    - Each column scales with a $c_i$ that indicates the activation of a degree of freedom
    - Setting as many $c_i$ to zero as possible gives a sparse solution that is easier to glean semantic information from: “what’s happening?”
  - Correlation
    - Columns of $A$ that are more parallel to $b$ probably are more relevant than those that are more perpendicular
    - A normalized dot product can be used to check this
  - Gains
    - Columns of $A$ that have a large dot product with the direction of $b$ make more progress towards $b$ with smaller $c_i$ values
    - It’s easier or more natural for these columns to be used

PowerPoint Presentation on Mike’s stuff….