Least Squares
Recall: Polynomial Interpolation (Unit 1)

• Given $m$ data points, one can (at best) draw a unique $m - 1$ degree polynomial that goes through all of them
  • As long as they are not degenerate, like 3 points on a line
Recall: Basis Functions (Unit 1)

- Given basis functions $\phi$ and unknowns $c$:  
  $$y = c_1 \phi_1 + c_2 \phi_2 + \cdots + c_n \phi_n$$

- Monomial basis: $\phi_k(x) = x^{k-1}$

- Lagrange basis: $\phi_k(x) = \frac{\prod_{i \neq k} x - x_i}{\prod_{i \neq k} x_k - x_i}$

- Newton basis: $\phi_k(x) = \prod_{i=1}^{k-1} x - x_i$

- Write a (linear) equation for each point, and put into matrix form: $Ac = y$
- Monomial/Lagrange/Newton basis all give the same polynomial, but different matrices
• Given a new input $\hat{x}$, the interpolating polynomial infers/predicts an output $\hat{y}$ that may be far from what one may expect

- Interpolating polynomials are smooth (continuous function and derivatives)
- Thus, they wiggle/overshoot in between data points (so that they can smoothly turn back and hit the next point)
- Overly forcing polynomials to exactly hit every data point is called overfitting (overly fitting to the data)
- It results in inference/predictions that can vary wildly from the training data
Recall: Regularization (Unit 1)

• Using a lower order polynomial that doesn’t (can’t) exactly fit the data points provides some degree of regularization

• A regularized interpolant contains intentional errors in the interpolation, missing some/all of the data points
• However, this hopefully makes the function more predictable/smooth in between the data points

• The data points themselves may contain noise/error, so it is not clear whether they should be interpolated exactly anyways
Recall: Regularization (Unit 1)

• Given $\hat{x}$, the regularized interpolant infers/predicts a more reasonable $\hat{y}$

• There is a trade-off between sacrificing accuracy on fitting the original input data, and obtaining better accuracy on inference/prediction for new inputs
Eliminating Basis Functions

• Consider $Ac = y$:
  
  • Each row of $A$ evaluates all $n$ basis functions $\phi_k$ on a single data point $x_i$
  
  • Each column of $A$ evaluates all $m$ points $x_i$ on a single basis function $\phi_k$

• Regularize by reducing the number of basis functions (and thus the degree of the polynomial)
  
  • Then, write an equation for each point, and put into matrix form $Ac = y$ (as usual)

• When there are more points than basis functions, there are more rows than columns (and the matrix is tall/rectangular)

• This tall matrix has full (column) rank when the basis functions are linearly independent (and the data isn’t degenerate)
Recall: Underfitting (Unit 1)

• Using too low of an order polynomial causes one to miss the data by too much

• A linear function doesn’t capture the essence of this data as well as a quadratic function does
• Choosing too simple of a model function or regularizing too much prevents one from properly representing the data
Tall (Full Rank) Matrices

• Let $A$ be a size $m \times n$ tall (i.e. $m > n$) matrix with full (column) rank (i.e. rank $n$)
• Since there are $n$ entries in each row, the rows span at most an $n$ dimensional space; thus, at least $m - n$ rows are linear combinations of others
• That is, $A$ contains (at least) $m - n$ extra unnecessary equations (that are linear combinations of others)
• Thus, $A$ could be reduced to $n$ equations (and size $n \times n$) without losing any information
• The SVD ($A = U \Sigma V^T$) illustrates this: the last $m - n$ rows of $\Sigma$ are all zeros
• The last $m - n$ columns in $U$ are hit by these zeros, and thus not in the range of $A$
Recall: Example (Unit 3)

\[ A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix} = \]

\[
\begin{pmatrix}
.141 & .825 & -.420 & -.351 \\
.344 & .426 & .298 & .782 \\
.547 & .028 & .644 & -.509 \\
.750 & -.371 & -.542 & .079 \\
\end{pmatrix}
\begin{pmatrix}
25.5 & 0 & 0 \\
0 & 1.29 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
.504 & .574 & .644 \\
-.761 & -.057 & .646 \\
.408 & -.816 & .408 \\
\end{pmatrix}
\]

- Singular values are 25.5, 1.29, and 0
- Singular value of 0 indicates that the matrix is rank deficient
- The **rank** of a matrix is equal to its number of nonzero singular values
Recall: Extra Dimensions (Unit 3)

\[
A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}
\]

\[
\begin{pmatrix}
.141 & .825 & -.420 & -.51 \\
.344 & .426 & .298 & .782 \\
.547 & .028 & .644 & -.519 \\
.750 & -.371 & -.542 & .09
\end{pmatrix}
\begin{pmatrix}
25.5 & 0 & 0 \\
0 & 1.29 & 0 \\
0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
.504 & .574 & .644 \\
-.761 & -.057 & .646 \\
.408 & -.816 & .408
\end{pmatrix}
\]

- The 3D space of vector inputs can only span a 3D subspace of \( \mathbb{R}^4 \)
- The last (green) column of \( U \) represents the unreachable dimension, orthogonal to the range of \( A \), and is always multiplied by 0
- One can delete this column and the associated portion of \( \Sigma \) (and still obtain a valid factorization)
Solving Tall (Full Rank) Linear Systems

- $Ac = b$ becomes $U \Sigma V^T c = b$ or $\Sigma (V^T c) = (U^T b)$ or $\Sigma \hat{c} = \hat{b}$
- Solve $\Sigma \hat{c} = \hat{b}$ by dividing the entries of $\hat{b}$ by the singular values $\sigma_k$, then $c = V \hat{c}$

- The last $m - n$ equations are identically zero on the left, and need to be identically zero on the right as well in order for a solution to exist
  - E.g. $\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \end{pmatrix} = \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{pmatrix}$ requires $\hat{b}_3 = 0$ in order to have a solution

- The last $m - n$ columns in $U$ are not in the range of $A$, so $b$ must be in the span of the first $n$ columns of $U$ in order for a solution to exist
False Statements

• Reasoning with a false statement leads to infinitely more false statements:

\[ a = b \]
\[ a^2 = ab \]
\[ a^2 - b^2 = ab - b^2 \]
\[ (a + b)(a - b) = b(a - b) \]
\[ a + b = b \]
\[ b + b = b \]
\[ b(1 + 1) = b(1) \]
\[ 2 = 1 \]

• Don’t make false statements!
False Statements

• Reasoning with a false statement leads to infinitely more false statements:

\[ Ac = b \]
\[ A^T Ac = A^T b \]
\[ c = (A^T A)^{-1} (A^T b) \]

• Don’t make false statements!

• A mix of false/true statements makes it difficult to keep track of what is and what is not true
False Statements

• Consider a very simple $Ac = b$ given by: \[
\begin{pmatrix} 1 \\ 1 \end{pmatrix} (c) = \begin{pmatrix} 3 \\ 4 \end{pmatrix}
\]

• This contains the equations $c = 3$ and $c = 4$, and as such is a false statement

• Solve via \[
(1 \quad 1)
\begin{pmatrix} 1 \\ 1 \end{pmatrix} (c) = (1 \quad 1)
\begin{pmatrix} 3 \\ 4 \end{pmatrix},
\] so $2c = 7$ or $c = 3.5$

• Row scale the first equation by 10 to obtain: \[
\begin{pmatrix} 10 \\ 1 \end{pmatrix} (c) = \begin{pmatrix} 30 \\ 4 \end{pmatrix}
\]

• Solve via \[
(10 \quad 1)
\begin{pmatrix} 10 \\ 1 \end{pmatrix} (c) = (10 \quad 1)
\begin{pmatrix} 30 \\ 4 \end{pmatrix},
\] so $101c = 304$ or $c = 3\frac{1}{101}$

• Perfectly valid row scaling leads to a different answer
False Statements

• Again, starting with the same: \( \begin{pmatrix} 1 \\ 1 \end{pmatrix} (c) = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \)

• Subtract 2*(row 1) from row 2 to obtain \( \begin{pmatrix} 1 \\ -1 \end{pmatrix} (c) = \begin{pmatrix} 3 \\ -2 \end{pmatrix} \)

• Solve via \( (1 \quad -1) \begin{pmatrix} 1 \\ -1 \end{pmatrix} (c) = (1 \quad -1) \begin{pmatrix} 3 \\ -2 \end{pmatrix} \), so \( 2c = 5 \) or \( c = 2.5 \)

• A perfectly valid row operation again leads to a different answer

• Note that \( 2.5 \notin [3,4] \) either!

• Problem: \( \begin{pmatrix} 3 \\ 4 \end{pmatrix} \) is not in the range of \( \begin{pmatrix} 1 \\ 1 \end{pmatrix} \), so \( \begin{pmatrix} 1 \\ 1 \end{pmatrix} (c) \neq \begin{pmatrix} 3 \\ 4 \end{pmatrix} \) for \( \forall c \in \mathbb{R} \)
False Statements

• Consider $y = c_1 \phi_1$ with monomial $\phi_1 = 1$, and data points (1,3) and (2,4)

• This leads to the same $\left( \frac{1}{1} \right) (c_1) = \left( \frac{3}{4} \right)$
• Consider $y = c_1 \phi_1$ with monomial $\phi_1 = 1$, and data points (1,3) and (2,3)

• This leads instead to $\left(\frac{1}{1}\right) (c_1) = \binom{3}{3}$ which is valid and has solution $c_1 = 3$
True Statements

• When $b$ is in the range of $A$, then $Ac = b$ is a true statement
  • There exists at least one $c$ (by definition) constrained by this statement
• When $b$ is in **not** the range of $A$, then $Ac \neq b$ is the true statement
  • In this case, $Ac \neq b$ is true for **all** $c$

• The equation for the **residual** $r = b - Ac$ is **always true** (it’s a definition)
  • When $b$ is in the range of $A$, there exists a $c$ with $Ac = b$ and $r = 0$
  • When $b$ is **not** in the range of $A$, then $Ac \neq b$ and $r \neq 0$ for **all** $c$
• The goal in both cases is to **minimize the residual** $r = b - Ac$
Norm Matters

• Consider \( y = c_1 \phi_1 \) where \( \phi_1 = 1 \) along with data points (1,3), (2,3), and (3,4).

• This leads to \( r = \begin{pmatrix} 3 \\ 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} (c_1) \)

• Setting \( c_1 = 3.5 \) minimizes \( \|r\|_\infty \) with \( r = \begin{pmatrix} -.5 \\ -.5 \\ .5 \end{pmatrix} \), \( \|r\|_\infty = .5 \), \( \|r\|_2 = \frac{\sqrt{3}}{2} \)

• Setting \( c_1 = 3\frac{1}{3} \) minimizes \( \|r\|_2 \) with \( r = \begin{pmatrix} -1/3 \\ -1/3 \\ 2/3 \end{pmatrix} \), \( \|r\|_\infty = \frac{2}{3} \), \( \|r\|_2 = \frac{\sqrt{6}}{3} \)
Row Operations Matter

• Given a set of equations, they can be manipulated in various ways
• These manipulations often change the answer

• Thus, one should carefully choose the residual they want minimized

• Equivalent sets of equations lead to different answers when minimizing the corresponding residuals
Weighted Minimization

• Given $r = b - Ac$, some equations may be deemed more important than others

• Scaling entries in the residual (before taking the norm) changes the relative importance of various equations

• This is accomplished by minimizing $\|Dr\|$ for a diagonal matrix $D$ with non-zero diagonal entries

• This is equivalent to row scaling: $Dr = Db - DAc$

• Column scaling doesn’t effect the residual, e.g. $Dr = Db - DA\hat{D}^{-1}(\hat{D}c)$

• So, it can be used to preserve symmetry: $Dr = Db - (DAD^T)(D^{-T}c)$
  • when $A$ is square and symmetric
Least Squares

- Minimizing $\|r\|_2$ is referred to as **least squares**, and the resulting solution is referred to as the **least squares solution** (it’s really a least squares solution)
  - A least squares solution is the unique solution when $\|r\|_2 = 0$
  - Minimizing $\|Dr\|_2$ is referred to as **weighted least squares**

- $\|r\|_2$ is minimized when $\|r\|_2^2$ is minimized
- And $\|r\|_2^2 = r \cdot r = (b - Ac) \cdot (b - Ac) = c^T A^T Ac - 2b^T Ac + b^T b$ is minimized when $c^T A^T Ac - 2b^T Ac$ is minimized
- Thus, **minimize** $c^T A^T Ac - 2b^T Ac$
- For weighted least squares, **minimize** $c^T A^T D^2 Ac - 2b^T D^2 Ac$