Linear regression

CS 446 / ECE 449

2021-02-03 23:48:16 -0600 (82ce512)
Plan for today

- Linear regression setup revisited.
- Normal equations, SVD, and pseudoinverse.
- Example (if time).
1. Clean/augment data (lecture 10?).
2. Pick model/architecture (anything from lectures 2-13).
3. Pick a loss function measuring model fit to data.
4. Run a gradient descent variant to fit model to data.
5. Tweak 1-4 until training error is small.
6. Tweak 1-5, possibly reducing model complexity, until testing error is small.

Is that all of ML?
No, but these days it’s much of it!
Linear regression — basic setup

1. Start from training data \(((x_i, y_i))_{i=1}^{n}\), with \(x_i \in \mathbb{R}^d\) and \(y_i \in \mathbb{R}\).

2. Model is a linear predictor: pick \(w \in \mathbb{R}^d\) with

   \[x_i \mapsto w^T x_i \approx y_i.\]

3. Loss function is squared loss (standard regression loss):

   \[\ell(w^T x_i, y_i) = \frac{1}{2}(w^T x_i - y_i)^2.\]

   We will minimize the empirical risk (average loss over training examples):

   \[
   \hat{R}(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i, y_i) = \frac{1}{2n} \|Xw - y\|^2 \quad \text{where} \quad X := \begin{bmatrix}
   \leftarrow x_1^T \rightarrow \\
   \vdots \\
   \leftarrow x_n^T \rightarrow
   \end{bmatrix}.
   \]

4. Basic method: gradient descent. Set \(w_0 = 0\), and thereafter

   \[w_{i+1} := w_i - \eta \nabla \hat{R}(w_i) = w_i - \frac{\eta}{n} X^T (Xw_i - y),\]

   where \(\eta\) is a learning rate (step size).
2. **Model** is a linear predictor: pick $\mathbf{w} \in \mathbb{R}^d$ with

$$x_i \mapsto \mathbf{w}^\top x_i \approx y_i.$$ 

- Our model/architecture/function class is $\{x \mapsto \mathbf{w}^\top x : \mathbf{w} \in \mathbb{R}^d\}$.
  
  For each $\mathbf{w} \in \mathbb{R}^d$, we have another predictor.

- This is a simple model; we’ll build off of it to get more powerful ones!

- This model is insufficient for complicated tasks, but often does well, and forms a good baseline.
3. Loss function is squared loss (standard regression loss):

\[ \ell(w^T x_i, y_i) = \frac{1}{2} (w^T x_i - y_i)^2. \]

We will minimize the empirical risk:

\[ \hat{R}(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i, y_i) = \frac{1}{2n} \|Xw - y\|^2. \]

- Regression towards the mean: if \( x_i = 1 \in \mathbb{R}^1 \) for all \( i \), then

\[ \arg \min_{w \in \mathbb{R}^1} \|Xw - y\|^2 = \frac{1}{n} \sum_{i=1}^{n} y_i. \]

Seems a reasonable notion of loss/error.

- There are many choices for \( \ell \). Next lecture we’ll use logistic loss

\[ \ell(\hat{y}, y) = \ln(1 + \exp(-\hat{y}y)). \]

This and squared loss are the most common.
4. Basic method: gradient descent. Set \( w_0 = 0 \), and thereafter

\[
\begin{align*}
    w_{i+1} &:= w_i - \eta \nabla \hat{R}(w_i) = w_i - \frac{\eta}{n} X^T (Xw_i - y),
\end{align*}
\]

where \( \eta \) is a learning rate (step size).

- In a few lectures, we’ll see that this globally minimizes \( \hat{R} \).
- We’ll spend most of this lecture on other solutions via SVD.
We want to find $\hat{w}$ so that

$$2n\hat{R}(\hat{w}) = \|X\hat{w} - y\|^2 = \min_{w \in \mathbb{R}^d} \|Xw - y\|^2.$$

Idea from calculus: set gradient to zero and solve:

$$0 = \nabla_w \|Xw - y\|^2 = 2X^T(Xw - y),$$

meaning we want $\hat{w}$ so that

$$X^TX\hat{w} = X^Ty.$$
We want to find $\hat{w}$ so that

$$2n\hat{R}(\hat{w}) = \|X\hat{w} - y\|^2 = \min_{w \in \mathbb{R}^d} \|Xw - y\|^2.$$ 

Idea from calculus: set gradient to zero and solve:

$$0 = \nabla_w \|Xw - y\|^2 = 2X^T (Xw - y),$$

meaning we want $\hat{w}$ so that

$$X^TXw = X^Ty.$$ 

These are called the normal equations.
The normal equations are the system of linear equalities

\[ X^T X \hat{w} = X^T y. \]

**Proposition.** \( \hat{w} \) satisfies \( \hat{R}(\hat{w}) = \min_w \hat{R}(w) \) iff \( \hat{w} \) satisfies the normal equations.
The normal equations are the system of linear equalities

\[ X^T X w = X^T y. \]

**Proposition.** \( \hat{w} \) satisfies \( \hat{R}(\hat{w}) = \min_w \hat{R}(w) \) iff \( \hat{w} \) satisfies the normal equations.

**Proof (one direction).** Consider \( w \) with \( X^T X w = X^T y \), and any \( w' \); then

\[ \| Xw' - y \|^2 = \| Xw' - Xw + Xw - y \|^2 \]

\[ = \| Xw' - Xw \|^2 + 2(Xw' - Xw)^T(Xw - y) + \| Xw - y \|^2. \]

Since

\[ (Xw' - Xw)^T(Xw - y) = (w' - w)^T(X^T Xw - X^T y) = 0, \]

then

\[ \| Xw' - y \|^2 = \| Xw' - Xw \|^2 + \| Xw - y \|^2 \geq \| Xw - y \|^2. \]

Later we’ll get a general version by convexity, but it’s nice that we can check this directly so easily!
The normal equations are the system of linear equalities
\[ X^T X \hat{w} = X^T y. \]

**Proposition.** \( \hat{w} \) satisfies \( \hat{R}(\hat{w}) = \min_w \hat{R}(w) \) iff \( \hat{w} \) satisfies the normal equations.
The normal equations are the system of linear equalities

\[ X^T X \hat{w} = X^T y. \]

**Proposition.** \( \hat{w} \) satisfies \( \mathcal{R}(\hat{w}) = \min_w \mathcal{R}(w) \) iff \( \hat{w} \) satisfies the normal equations.

How do we solve for \( \hat{w} \)?

- If \( X^T X \) is invertible, we can use \( (X^T X)^{-1} X^T y \).
- In general, we will use the **SVD**.
The SVD (Singular Value Decomposition).

Let $M \in \mathbb{R}^{n \times d}$ be given. $((s_i, u_i, v_i))_{i=1}^r$ is an SVD of $M$ if:

▶ $M$ has rank $r$;
▶ $s_1 \geq s_2 \cdots \geq s_r > 0$;
▶ $(u_i)_{i=1}^r$ are orthonormal (orthogonal and unit length), and span the column space of $M$;
▶ $(v_i)_{i=1}^r$ are orthonormal, and span the row space of $M$.
▶ $M = \sum_i s_i u_i v_i^T$. 

The SVD always exists, and is real-valued. (When do real eigendecompositions not exist?)

The ordered tuple $(s_1, \ldots, s_r)$ is unique, but the SVD is in general not unique (why not?).

For $k < r$, the low rank approximation $\sum_{i=1}^k s_i u_i v_i^T \approx M$ has many applications (wait for the PCA lecture).
The SVD (Singular Value Decomposition).

Let \( M \in \mathbb{R}^{n \times d} \) be given. \(((s_i, u_i, v_i))_{i=1}^{r} \) is an SVD of \( M \) if:

- \( M \) has rank \( r \);
- \( s_1 \geq s_2 \cdots \geq s_r > 0 \);
- \((u_i)_{i=1}^{r}\) are orthonormal (orthogonal and unit length), and span the column space of \( M \);
- \((v_i)_{i=1}^{r}\) are orthonormal, and span the row space of \( M \).
- \( M = \sum_i s_i u_i v_i^T \).

- The SVD always exists, and is real-valued.
  (When do real eigendecompositions not exist?)
- The ordered tuple \((s_1, \ldots, s_r)\) is unique, but the SVD is in general not unique (why not?).
- For \( k < r \), the low rank approximation \( \sum_{i=1}^{k} s_i u_i v_i^T \approx M \) has many applications (wait for the PCA lecture).
Pseudoinverse.

Given SVD $M = \sum_i s_i u_i v_i^T$, the pseudoinverse is

$$M^+ := \sum_{i=1}^r \frac{1}{s_i} v_i u_i^T.$$
Given SVD \( M = \sum_i s_i u_i v_i^T \), the pseudoinverse is

\[
M^+ := \sum_{i=1}^{r} \frac{1}{s_i} v_i u_i^T.
\]

- The SVD may fail to be unique, but \( M^+ \) is unique.
- \( MM^+ = \sum_{i=1}^{r} u_i u_i^T \) and \( M^+ M = \sum_{i=1}^{r} v_i v_i^T \); in general, neither is an identity matrix. (Consider the case \( M = e_1 e_1^T \).)
- On the other hand,

\[
M M^+ M =
\]

\[
M^+ M M^+ =
\]

- If \( M^{-1} \) exists, then \( M^+ = M^{-1} \).
- If \( M = 0 \), then \( r = 0 \) and \( M^+ = 0 \).
Pseudoinverse.

Given SVD $\mathbf{M} = \sum_i s_i \mathbf{u}_i \mathbf{v}_i^T$, the pseudoinverse is

$$\mathbf{M}^+ := \sum_{i=1}^r \frac{1}{s_i} \mathbf{v}_i \mathbf{u}_i^T.$$  

- The SVD may fail to be unique, but $\mathbf{M}^+$ is unique.
- $\mathbf{M} \mathbf{M}^+ = \sum_{i=1}^r \mathbf{u}_i \mathbf{u}_i^T$ and $\mathbf{M}^+ \mathbf{M} = \sum_{i=1}^r \mathbf{v}_i \mathbf{v}_i^T$; in general, neither is an identity matrix. (Consider the case $\mathbf{M} = \mathbf{e}_1 \mathbf{e}_1^T$.)
- On the other hand,

$$\mathbf{M} \mathbf{M}^+ \mathbf{M} = \left( \sum_{i=1}^r s_i \mathbf{u}_i \mathbf{v}_i^T \right) \left( \sum_{j=1}^r \frac{1}{s_j} \mathbf{v}_j \mathbf{u}_j^T \right) \left( \sum_{k=1}^r s_k \mathbf{u}_k \mathbf{v}_k^T \right) = \mathbf{M},$$

$$\mathbf{M}^+ \mathbf{M} \mathbf{M}^+ = \left( \sum_{i=1}^r \frac{1}{s_i} \mathbf{v}_i \mathbf{u}_i^T \right) \left( \sum_{j=1}^r s_j \mathbf{u}_j \mathbf{v}_j^T \right) \left( \sum_{k=1}^r \frac{1}{s_k} \mathbf{v}_k \mathbf{u}_k^T \right) = \mathbf{M}^+.$$

- If $\mathbf{M}^{-1}$ exists, then $\mathbf{M}^+ = \mathbf{M}^{-1}$.
- If $\mathbf{M} = \mathbf{0}$, then $r = 0$ and $\mathbf{M}^+ = \mathbf{0}$.  

Given a least squares problem $\hat{\mathcal{R}}(w) = \|Xw - y\|^2/(2n)$, the OLS solution

$$\hat{w}_{\text{ols}} = X^+ y$$

satisfies the normal equations (whereby $\hat{\mathcal{R}}(\hat{w}_{\text{ols}}) = \min_w \hat{\mathcal{R}}(w)$).
OLS (Ordinary Least Squares) solution via SVD.

Given a least squares problem \( \hat{R}(w) = \|Xw - y\|^2/(2n) \), the OLS solution

\[
\hat{w}_{\text{ols}} = X^+ y
\]

satisfies the normal equations (whereby \( \hat{R}(\hat{w}_{\text{ols}}) = \min_w \hat{R}(w) \)).

Easy to check: writing \( X = \sum_{i=1}^{r} s_i u_i v_i^\top \),

\[
X^T X \hat{w}_{\text{ols}} = X^T X X^+ y
\]

\[
= \left( \sum_{i=1}^{r} s_i v_i u_i^\top \right) \left( \sum_{j=1}^{r} s_j u_j v_j^\top \right) \left( \sum_{k=1}^{r} \frac{1}{s_k} v_k u_k^\top \right) y
\]

\[
= X^T y.
\]
SVD $M = \sum_i s_i u_i v_i^T$ and orthonormal bases.

We can extend $(u_i)_{i=1}^r$ and $(v_i)_{i=1}^r$ to full orthonormal bases for $\mathbb{R}^n$ and $\mathbb{R}^d$ respectively: write $M \in \mathbb{R}^{n \times d}$ as

$$
\begin{bmatrix}
\uparrow & \uparrow & \uparrow & \uparrow \\
| & \uparrow & | & \uparrow & | & \uparrow \\
\downarrow & \downarrow & \downarrow & \downarrow \\
\end{bmatrix}
\begin{bmatrix}
s_1 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & s_r \\
\end{bmatrix}
\begin{bmatrix}
\uparrow & \uparrow & \uparrow & \uparrow \\
| & \uparrow & | & \uparrow & | & \uparrow \\
\downarrow & \downarrow & \downarrow & \downarrow \\
\end{bmatrix}
\begin{bmatrix}
v_1 & \cdots & v_r & v_{r+1} & \cdots & v_d \\
\end{bmatrix}^T.
$$

The old parts span the column and row spaces of $M$; the new vectors span the left and right nullspaces. Some call this a “full” SVD.
SVD and relationship to eigenvalues.

**Note**

\[
\mathbf{M} \mathbf{M}^T = \sum_{i=1}^{r} s_i \mathbf{u}_i \mathbf{v}_i^T \sum_{j=1}^{r} s_j \mathbf{v}_j \mathbf{u}_j^T = \sum_{i=1}^{r} s_i^2 \mathbf{u}_i \mathbf{u}_i^T,
\]

thus left singular vectors \( \mathbf{u}_i \) for \( i = 1 \) are top eigenvectors of \( \mathbf{M} \mathbf{M}^T \), with eigenvalues \( s_2 \geq \cdots \geq s_r \).

Similarly,

\[
\mathbf{M}^T \mathbf{M} = \sum_{i=1}^{r} s_i \mathbf{v}_i \mathbf{u}_i^T \sum_{j=1}^{r} s_j \mathbf{u}_j \mathbf{v}_j^T = \sum_{i=1}^{r} s_i^2 \mathbf{v}_i \mathbf{v}_i^T,
\]

obtaining right singular vectors from \( \mathbf{M} \mathbf{M}^T \).
SVD and relationship to eigenvalues.

Note

\[ MM^T = \sum_{i=1}^{r} s_i u_i v_i^T \sum_{j=1}^{r} s_j v_j u_j^T = \sum_{i=1}^{r} s_i^2 u_i u_i^T, \]

thus left singular vectors \((u)_{i=1}^{r}\) are top eigenvectors of \(MM^T\), with eigenvalues \(s_1^2 \geq \cdots \geq s_r^2\).

Similarly,

\[ M^T M = \sum_{i=1}^{r} s_i v_i u_i^T \sum_{j=1}^{r} s_j u_j v_j^T = \sum_{i=1}^{r} s_i^2 v_i v_i^T, \]

obtaining right singular vectors from \(M^T M\).
Summary on least squares solutions

We want to approximately solve the empirical risk minimization problem

$$\min_{\mathbf{w} \in \mathbb{R}^d} \hat{\mathcal{R}}(\mathbf{w}) = \min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2n} \| \mathbf{X} \mathbf{w} - \mathbf{y} \|^2.$$ 

Three approaches:

1. Gradient descent: \( \mathbf{w}_0 := 0 \), thereafter \( \mathbf{w}_{i+1} := \mathbf{w} - \eta \nabla \hat{\mathcal{R}}(\mathbf{w}_i) \).

2. Pick any \( \hat{\mathbf{w}} \) satsifying the normal equations

\[
\mathbf{X}^\top \mathbf{X} \mathbf{w} = \mathbf{X}^\top \mathbf{y}.
\]

3. Use the ordinary least squares (OLS) solution \( \hat{\mathbf{w}}_{\text{ols}} = \mathbf{X}^+ \mathbf{y} \).
Summary on least squares solutions

We want to approximately solve the **empirical risk minimization problem**

\[
\min_{w \in \mathbb{R}^d} \hat{R}(w) = \min_{w \in \mathbb{R}^d} \frac{1}{2n} \|Xw - y\|^2.
\]

Three approaches:

1. **Gradient descent**: \( w_0 := 0 \), thereafter \( w_{i+1} := w - \eta \nabla \hat{R}(w_i) \).
2. Pick any \( \hat{w} \) satisfying the **normal equations**
   \[
   X^T X w = X^T y.
   \]
3. Use the **ordinary least squares (OLS)** solution \( \hat{w}_{\text{ols}} = X^+ y \).

(Side note: are these different?...)

Example: Old Faithful geyser (Yellowstone)
**Task:** Predict time of next eruption.
# Time between eruptions

**Source data:** start and end times \((a_i, b_i)\) of \(n = 136\) eruptions.

\[
\begin{array}{ccccc}
  a_0 & b_0 & a_1 & b_1 & a_2 & b_2 & a_3 & b_3 & \ldots \\
\end{array}
\]

Let's pre-process: form time between eruptions \(y_i = a_i + 1 - b_i\).

Reformulated task: to estimate next eruption, find last end time \(t\), compute \(\hat{y}\), and output \(t + \hat{y}\).

Let's use linear regression.

- Set \(x_i = 1\) and the OLS solution is the mean: \(\hat{y} = \frac{1}{136} \sum_{i=1}^{136} y_i = 70.7941\).
- Can we do better with another \(x_i\)?
Time between eruptions

Source data: start and end times \((a_i, b_i)\) of \(n = 136\) eruptions.

\[
\begin{array}{cccccccc}
\hspace{1cm} & a_0 & b_0 & a_1 & b_1 & a_2 & b_2 & a_3 & b_3 & \ldots \\
\leftarrow & - & - & - & - & - & - & - & - & - & \rightarrow \\
& Y_1 & & Y_2 & & Y_3 & & & &
\end{array}
\]

Let’s pre-process: form time between eruptions \(y_i := a_{i+1} - b_i\).
Time between eruptions

Source data: start and end times \((a_i, b_i)\) of \(n = 136\) eruptions.

\[
\ldots \ a_{n-1} \ b_{n-1} \ a_n \ b_n \ \ldots \quad \text{data} \quad \ldots \ t
\]

Let’s pre-process: form time between eruptions \(y_i := a_{i+1} - b_i\).

Reformulated task:

to estimate next eruption, find last end time \(t\), compute \(\hat{y}\), and output \(t + \hat{y}\).
Time between eruptions

Source data: start and end times \((a_i, b_i)\) of \(n = 136\) eruptions.

Let’s pre-process: form time between eruptions \(y_i := a_{i+1} - b_i\).

Reformulated task:

to estimate next eruption, find last end time \(t\), compute \(\hat{y}\), and output \(t + \hat{y}\).

Let’s use linear regression.

- Set \(x_i = 1\) and the OLS solution is the mean:

\[
\hat{y} = \frac{1}{136} \sum_{i=1}^{136} y_i = 70.7941.
\]

- Can we do better with another \(x_i\)?
Eruption length and time to eruption are correlated.

Let choose $x_i := [b_i - a_i]$. 
Eruption length and time to eruption are correlated.

Let choose \( x_i := \left[ b_i - a_i \right] \).

**(Side note:** the extra “1” will be discussed extensively later.)
1. Form pairs \( x_i := [b_i - a_i] \), and matrix 

\[
X := \begin{bmatrix}
\leftarrow & x_1^T & \rightarrow \\
\vdots & \vdots & \vdots \\
\leftarrow & x_n^T & \rightarrow \\
\end{bmatrix} = \begin{bmatrix}
b_1 - a_1 & 1 \\
\vdots & \vdots \\
b_n - a_n & 1 \\
\end{bmatrix} \in \mathbb{R}^{n \times 2}.
\]

Form labels \( y \in \mathbb{R}^n \), \( y_i := a_{i+1} - b_i \).

2. Choose OLS solution \( \hat{w}_{\text{ols}} := X^+ y \).

3. Given a new eruption \((a, b)\), estimate next eruption time \( b + w^T [b-a] \).
“pytorch meta-algorithm” on Old Faithful data

1. Clean/augment data.
   From \((a_i, b_i)\), form \(x'_i = (1, )\) or \(x_i = (b_i - a_i, 1)\), and \(y_i = a_i - b_{i-1}\).

2. Pick model/architecture (anything from lectures 2-13).
   Linear predictor.

3. Pick a loss function measuring model fit to data.
   Squared loss.

4. Run a gradient descent variant to fit model to data.

5. Tweak 1-4 until training error is small.
   \(x'_i\) was bad, so we added a feature and got \(x_i\).

6. Tweak 1-5, possibly reducing model complexity, until testing error is small.
   We didn’t try this!
Summary for today

- Linear regression setup revisited.
- Normal equations, SVD, and pseudoinverse.
- Example (if time).
(Appendix.)
The normal equations are the system of linear equalities

\[ X^T X w = X^T y. \]

**Proposition.** \( \hat{w} \) satisfies \( \hat{R}(\hat{w}) = \min_w \hat{R}(w) \) iff \( \hat{w} \) satisfies the normal equations.
The normal equations are the system of linear equalities

\[ X^T X w = X^T y. \]

**Proposition.** \( \hat{w} \) satisfies \( \hat{R}(\hat{w}) = \min_{w} \hat{R}(w) \) iff \( \hat{w} \) satisfies the normal equations.

**Proof (other direction).**
Suppose \( w \) is optimal; since \( \hat{w}_{ols} \) satisfies the normal equations, then expanding the square as in the proof of the other direction gives

\[ \| Xw - y \|^2 = \| Xw - X\hat{w}_{ols} \|^2 + \| X\hat{w}_{ols} - y \|^2. \]

Since \( w \) and \( \hat{w}_{ols} \) are optimal, then \( \hat{R}(w) = \hat{R}(\hat{w}_{ols}) \), so the preceding implies
The normal equations are the system of linear equalities

\[ X^T X w = X^T y. \]

**Proposition.** \( \hat{w} \) satisfies \( \hat{R}(\hat{w}) = \min_w \hat{R}(w) \) iff \( \hat{w} \) satisfies the normal equations.

**Proof (other direction).**
Suppose \( w \) is optimal; since \( \hat{w}_{\text{ols}} \) satisfies the normal equations, then expanding the square as in the proof of the other direction gives

\[ \| Xw - y \|^2 = \| Xw - X\hat{w}_{\text{ols}} \|^2 + \| X\hat{w}_{\text{ols}} - y \|^2. \]

Since \( w \) and \( \hat{w}_{\text{ols}} \) are optimal, then \( \hat{R}(w) = \hat{R}(\hat{w}_{\text{ols}}) \), so the preceding implies

\[ 0 = \| Xw - X\hat{w}_{\text{ols}} \|^2 = \| X(w - \hat{w}_{\text{ols}}) \|^2, \]

therefore \( X(w - \hat{w}_{\text{ols}}) = 0 \) and \( Xw = X\hat{w}_{\text{ols}} \), which by the normal equations for \( \hat{w}_{\text{ols}} \) means

\[ X^T y = X^T X\hat{w}_{\text{ols}} = X^T X w, \]

thus \( w \) satisfies the normal equations. \( \square \)
Geometric interpretation of least squares ERM

Let \( a_j \in \mathbb{R}^n \) be the \( j \)-th column (not row!) of matrix \( X \in \mathbb{R}^{n \times d} \), so

\[
X = \begin{bmatrix}
\leftarrow x_1^\top & \rightarrow \\
\vdots & \\
\leftarrow x_n^\top & \rightarrow \\
\end{bmatrix}
= \begin{bmatrix}
\uparrow & \cdots & \uparrow \\
\downarrow & \cdots & \downarrow \\
\end{bmatrix}.
\]

Minimizing \( \|Aw - b\|_2^2 \) means finding \( \hat{b} \in \text{span}(a_1, \ldots, a_d) \) closest to \( b \).

Solution \( \hat{b} \) is orthogonal projection of \( b \) onto \( \text{range}(A) = \{Aw : w \in \mathbb{R}^d\} \).

\( \hat{b} \) is uniquely determined; indeed, \( \hat{b} = AA^+b = \sum_{i=1}^{r} u_i u_i^\top b \).

If \( r = \text{rank}(A) < d \), then one way to write \( \hat{b} \) as linear combination of \( a_1, \ldots, a_d \).

If \( \text{rank}(A) < d \), then ERM solution is not unique.

To get \( w \) from \( \hat{b} \): solve system of linear equations \( Aw = \hat{b} \).
Geometric interpretation of least squares ERM

Let $a_j \in \mathbb{R}^n$ be the $j$-th column (not row!) of matrix $X \in \mathbb{R}^{n \times d}$, so

$$X = \begin{bmatrix}
\leftarrow & x_1^T & \rightarrow \\
\vdots & & \\
\leftarrow & x_n^T & \rightarrow 
\end{bmatrix} = \begin{bmatrix}
\uparrow & \cdots & \uparrow \\
\downarrow & \cdots & \downarrow \\
\end{bmatrix}
= \begin{bmatrix}
\uparrow & \cdots & \uparrow \\
\downarrow & \cdots & \downarrow \\
\end{bmatrix} = \begin{bmatrix}
\uparrow & \cdots & \uparrow \\
\downarrow & \cdots & \downarrow \\
\end{bmatrix}.
$$

Minimizing $\|A\hat{w} - b\|_2^2$ means finding $\hat{b} \in \text{span}(a_1, \ldots, a_d)$ closest to $b$. 
Geometric interpretation of least squares ERM

Let \( a_j \in \mathbb{R}^n \) be the \( j \)-th column (not row!) of matrix \( X \in \mathbb{R}^{n \times d} \), so

\[
X = \begin{bmatrix}
\leftarrow x_1^\top \rightarrow \\
\vdots \\
\leftarrow x_n^\top \rightarrow
\end{bmatrix} = \begin{bmatrix}
\uparrow \cdots \uparrow \\
\downarrow a_1 \cdots \downarrow a_d
\end{bmatrix}.
\]

Minimizing \( \| Aw - b \|_2^2 \) means finding \( \hat{b} \in \text{span}(a_1, \ldots, a_d) \) closest to \( b \).

Solution \( \hat{b} \) is **orthogonal projection** of \( b \) onto \( \text{range}(A) = \{ Aw : w \in \mathbb{R}^d \} \).
Geometric interpretation of least squares ERM

Let \( a_j \in \mathbb{R}^n \) be the \( j \)-th column (not row!) of matrix \( X \in \mathbb{R}^{n \times d} \), so

\[
X = \begin{bmatrix}
\leftarrow \ x_1^T \rightarrow \\
\vdots \\
\leftarrow \ x_n^T \rightarrow 
\end{bmatrix} = \begin{bmatrix}
\uparrow & \cdots & \uparrow \\
\downarrow \ a_1 & \cdots & \downarrow a_d 
\end{bmatrix}.
\]

Minimizing \( \| Aw - b \|_2^2 \) means finding \( \hat{b} \in \text{span}(a_1, \ldots, a_d) \) closest to \( b \).

Solution \( \hat{b} \) is **orthogonal projection** of \( b \) onto \( \text{range}(A) = \{ Aw : w \in \mathbb{R}^d \} \).

\[ \Rightarrow \hat{b} \text{ is uniquely determined; indeed, } \hat{b} = AA^+ b = \sum_{i=1}^{r} u_i u_i^T b. \]
Geometric interpretation of least squares ERM

Let $a_j \in \mathbb{R}^n$ be the $j$-th column (not row!) of matrix $X \in \mathbb{R}^{n \times d}$, so

$$X = \begin{bmatrix} \leftarrow x_1^\top \rightarrow \\ \vdots \\ \leftarrow x_n^\top \rightarrow \end{bmatrix} = \begin{bmatrix} \uparrow & \cdots & \uparrow \\ a_1 & \cdots & a_d \downarrow & \cdots & \downarrow \end{bmatrix}.$$ 

Minimizing $\|Aw - b\|^2_2$ means finding $\hat{b} \in \text{span}(a_1, \ldots, a_d)$ closest to $b$.

Solution $\hat{b}$ is orthogonal projection of $b$ onto $\text{range}(A) = \{Aw : w \in \mathbb{R}^d\}$.

- $\hat{b}$ is uniquely determined; indeed, $\hat{b} = AA^+ b = \sum_{i=1}^r u_i u_i^\top b$.

- If $r = \text{rank}(A) < d$, then $>1$ way to write $\hat{b}$ as linear combination of $a_1, \ldots, a_d$. 

$\implies$ $\hat{b}$ is orthogonal projection of $b$ onto $\text{span}(a_1, a_2)$. 

$\implies$ $\hat{b} = \frac{1}{2} a_1 + \frac{1}{2} a_2$.
Geometric interpretation of least squares ERM

Let $a_j \in \mathbb{R}^n$ be the $j$-th column (not row!) of matrix $X \in \mathbb{R}^{n \times d}$, so

$$X = \begin{bmatrix} \leftarrow & x_1^\top & \rightarrow \\ \vdots \\ \leftarrow & x_n^\top & \rightarrow \end{bmatrix} = \begin{bmatrix} \uparrow & \cdots & \uparrow \\ \downarrow & \cdots & \downarrow \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_d \end{bmatrix}.$$

Minimizing $\|Ax - b\|_2^2$ means finding $\hat{b} \in \text{span}(a_1, \ldots, a_d)$ closest to $b$.

Solution $\hat{b}$ is orthogonal projection of $b$ onto $\text{range}(A) = \{Aw : w \in \mathbb{R}^d\}$.

- $\hat{b}$ is uniquely determined; indeed, $\hat{b} = AA^+b = \sum_{i=1}^r u_i u_i^\top b$.
- If $r = \text{rank}(A) < d$, then $>1$ way to write $\hat{b}$ as linear combination of $a_1, \ldots, a_d$.

If $\text{rank}(A) < d$, then ERM solution is not unique.
Geometric interpretation of least squares ERM

Let $a_j \in \mathbb{R}^n$ be the $j$-th column (not row!) of matrix $X \in \mathbb{R}^{n \times d}$, so

$$X = \begin{bmatrix}
\leftarrow & x_1^\top & \rightarrow \\
\vdots & \hspace{1cm} & \\
\leftarrow & x_n^\top & \rightarrow \\
\end{bmatrix} = \begin{bmatrix}
\uparrow & \hspace{1cm} & \uparrow \\
\downarrow & \hspace{1cm} & \downarrow \\
a_1 & \cdots & a_d \\
\end{bmatrix}.$$ 

Minimizing $\|Aw - b\|_2^2$ means finding $\hat{b} \in \text{span}(a_1, \ldots, a_d)$ closest to $b$.

Solution $\hat{b}$ is **orthogonal projection** of $b$ onto $\text{range}(A) = \{Aw : w \in \mathbb{R}^d\}$.

- $\hat{b}$ is uniquely determined; indeed, $\hat{b} = AA^+ b = \sum_{i=1}^r u_i u_i^\top b$.
- If $r = \text{rank}(A) < d$, then $>1$ way to write $\hat{b}$ as linear combination of $a_1, \ldots, a_d$.

If $\text{rank}(A) < d$, then **ERM solution is not unique**.

To get $w$ from $\hat{b}$:
solve system of linear equations $Aw = \hat{b}$. 

---

23 / 25
Computing the SVD

Typical solver is an iterative, greedy method. For more information, see the excellent data science book by Blum, Hopcroft, Kannan.
Why include GD, since pseudoinverse seems sufficient?

- GD is easy to implement, pseudoinverse more painful.
- Pseudoinverse after all implemented as an iterative solver.
- GD generalizes to other cases of squared loss (e.g., deep network training with squared loss).