MS&E 226: Fundamentals of Data Science

Lecture 7: The bias-variance decomposition

Ramesh Johari

Motivation: Evaluating learning algorithms

Evaluation: Fitted models vs. learning algorithms

The validation and test steps of "train-test-validate" show us how to evaluate *fitted models*.

But how do we "evaluate" a *learning algorithm*?

What is a "good" learning algorithm?

Good learning algorithms

In general, when creating learning algorithms,
we are trading off two different goals:

- On one hand, we want fitted models to fit the training data well.
- On the other hand, we want to avoid fitted models that become so finely tuned to the training data that they perform poorly on new data.

In this lecture we develop a systematic vocabulary for talking about "good" learning algorithms, through the notions of *bias* and *variance*.

The frequentist approach: Sampling distributions

An example: NBA rookie points per game (PPG)

Suppose I pick a random NBA (National Basketball Association) rookie from the five seasons 2020-21 to 2024-25.

What would you guess was this player's PPG in their rookie year?

(My son's guess: 10 PPG.)

An example: NBA rookie points per game (PPG)

Now suppose I give you the rookie year PPG for $n = 10$ players (the *sample*), randomly sampled from all rookies in from 2020-25 (the *population*).[1]

The first "learning algorithm" in this course: *sample mean*.

- $Y_1, Y_2, \ldots, Y_{10}$: the PPG of these 10 players
- $\bar{Y} = \frac{1}{10} \sum_{i=1}^{10} Y_i$: the sample mean

[1] Note: throughout the lecture, "randomly sampled" means *each observation* is an independent draw from the entire population.

The population model

We will suspend disbelief: Assume you *don't have access* to the whole population.

In particular, suppose you don't know:

- μ: the true mean PPG of all rookies in the 2020-25 seasons
- σ^2: the true variance of PPG of rookies in the 2020-25 seasons

[A peek at the truth: $\mu = 5.49$ PPG, and $\sigma = 4.31$ PPG.]

The prediction problem

Question: What prediction should we make if we're asked to predict the height of a player we hadn't seen yet, to minimize squared prediction error?

If we knew the heights of all rookies in the last five years (the full population), we would report μ, the population mean. (See Lecture 2 and Problem Set 1.)

The prediction problem

Question: What prediction should we make if we're asked to predict the height of a player we hadn't seen yet, to minimize squared prediction error?

Note that if we could actually predict μ, our expected squared prediction error would be σ^2:

$$\mathbb{E}_Y[(Y - \mu)^2] = \text{Var}(Y) = \sigma^2,$$

where Y is the true PPG of the randomly sampled player.

This error is *irreducible*: We can't do better than this with any other prediction.

But we don't know μ. So what might we do?

Two natural approaches

Two simple approaches ("learning algorithms") we might consider:

1. *Guess a constant*: Predict 10 PPG (my son's guess), or some other constant (what would you guess?).
2. *Use the sample mean*: Predict $\bar{Y}$, the PPG of the 10 players we observed.

Which is "better"?

How does the answer depend on the number of players in our sample?

Evaluating our approaches

A natural question: *What is the MSE of each of these approaches?*

In other words:

- Is "guessing a constant" a "good" idea, i.e., does it generally give low MSE?
- Is "taking the sample mean" a "good" idea, i.e., does it generally give low MSE?

To evaluate the approaches, we need to think more carefully about what we mean by "good."

Evaluating our approaches

Recall that we evaluate alternative *fitted models* by comparing them on a validation set.

But one fitted model might outperform another for two different reasons:

- The learning algorithm that produced it is genuinely "better"
- The particular data sample favored that particular fitted model ("luck")

We want to evaluate the first, not the second. How do we do this?

Thinking like a frequentist

Here's an idea: *rewind time* and imagine resampling 10 players from the full population, again and again.

Each time we resample, we create a new "parallel universe":

- We get a new set of 10 heights: $\mathbf{Y} = (Y_1, \ldots, Y_{10})$.
- We make a prediction based on $\mathbf{Y}$, e.g., the sample mean $\bar{Y}$.

The key insight: *we can evaluate the quality of our approach by imagining its behavior across many possible "parallel universe" samples*.

This is the heart of the *frequentist* viewpoint.

The sampling distribution

Suppose we execute this thought experiment with the number of parallel universes $\to \infty$.

We can summarize the results by the *distribution* of the predictions we obtain, across "parallel universes."

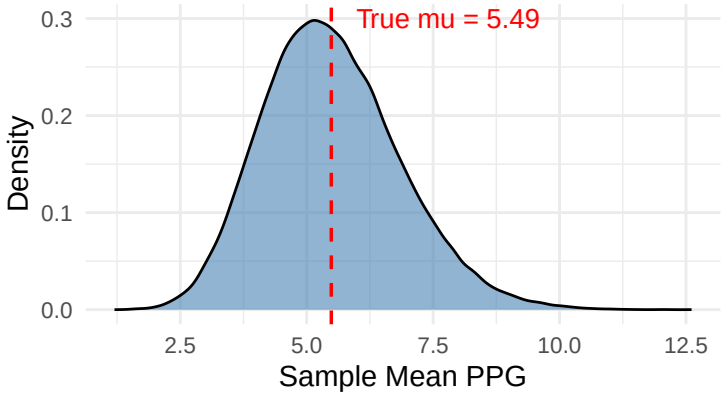
This distribution is called the *sampling distribution* of our approach.

The sampling distribution is the most fundamental concept in frequentist statistics!

The sampling distribution: Example

I actually collected all the rookies' PPG from 2020-25, so I can (approximately) execute this thought experiment for the *sample mean*.

Resulting approximate sampling distribution, over 100,000 parallel universes:



Sampling distributions can't be computed

The sampling distribution is an odd construct:

We can't compute it if we don't know the population model!

So how is it still a useful construct?

1. We will see that we can qualitatively reason about MSE using the sampling distribution, even if we can't exactly compute it.
2. We will see later that we can develop quantitative tools that give us insight into the sampling distribution, even if we can't exactly compute it.

Using the sampling distribution

How is the sampling distribution specifically helpful in evaluating learning algorithms?

- It reflects the *distribution* of predictions $\hat{Y}$, across parallel universes.

Using the sampling distribution

How is the sampling distribution specifically helpful in evaluating learning algorithms?

- It reflects the *distribution* of predictions $\hat{Y}$, across parallel universes.
- The *average MSE* of these predictions is:

$$\text{Average MSE} = \mathbb{E}_{Y,\mathbf{Y}}[(\hat{Y} - Y)^2].$$

Note that this expectation is over *both* the randomness in the outcome Y, *and* the randomness in the training data sample $\mathbf{Y}$.

Using the sampling distribution

How is the sampling distribution specifically helpful in evaluating learning algorithms?

- It reflects the *distribution* of predictions $\hat{Y}$, across parallel universes.
- The *average MSE* of these predictions is:

$$\text{Average MSE} = \mathbb{E}_{Y,\mathbf{Y}}[(\hat{Y} - Y)^2].$$

  Note that this expectation is over *both* the randomness in the outcome Y, *and* the randomness in the training data sample $\mathbf{Y}$.
- In particular, any "luck" due to a single data sample is averaged out in this Average MSE $\implies$ better learning algorithms have lower Average MSE.

Using the sampling distribution

How is the sampling distribution specifically helpful in evaluating learning algorithms?

- It reflects the *distribution* of predictions $\hat{Y}$, across parallel universes.
- The *average MSE* of these predictions is:

$$\text{Average MSE} = \mathbb{E}_{Y,\mathbf{Y}}[(\hat{Y} - Y)^2].$$

  Note that this expectation is over *both* the randomness in the outcome Y, *and* the randomness in the training data sample $\mathbf{Y}$.
- In particular, any "luck" due to a single data sample is averaged out in this Average MSE $\implies$ better learning algorithms have lower Average MSE.

By decomposing Average MSE, we can identify the sources of our prediction error.

There are two key sources beyond irreducible error: *bias* and *variance*.

The bias-variance decomposition

Bias

Bias is the difference between the *mean of the sampling distribution* and the *actual mean* μ:

$$\text{Bias} = \mathbb{E}_{\mathbf{Y}}[\overline{Y}] - \mu.$$

If our predictions are systematically too high or too low (on average across universes), we have bias.

(Note the expectation is once again over the training data sample $\mathbf{Y}$.)

Variance

The *variance* of the sampling distribution measures how much the sample mean varies from universe to universe:

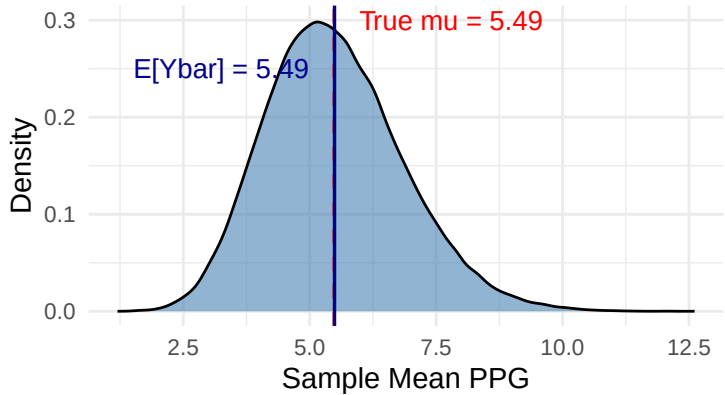
$$\mathrm{Var}_{\mathbf{Y}}(\overline{Y}) = \mathbb{E}_{\mathbf{Y}}\left[\left(\overline{Y} - \mathbb{E}_{\mathbf{Y}}[\overline{Y}]\right)^2\right].$$

If our predictions are highly sensitive to which particular sample we draw across universes, we have high variance.

Note: The standard deviation of the sampling distribution is called the *standard error* (of the sample mean).

Bias and variance: Example

From the sampling distribution of the sample mean shown earlier:



Bias = $\mathbb{E}_{\mathbf{Y}}[\overline{Y}] - \mu = 0$; Sample SD = $\sqrt{\text{sample variance}} = 1.363$.

Bias and variance: Example

In fact, we can exactly compute the bias and variance of the sample mean given n observations:

$$\text{Bias} = \mathbb{E}_{\mathbf{Y}}\left[\frac{1}{n}\sum_{i=1}^{n} Y_i\right] - \mu = \left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathbf{Y}}[Y_i]\right) - \mu = 0;$$

$$\text{Variance} = \text{Var}_{\mathbf{Y}}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) = \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}_{\mathbf{Y}}(Y_i) = \frac{\sigma^2}{n}.$$

So the sample mean is *unbiased*, and the variance goes down *inversely* in the sample size.

(The second result uses the fact that the Y_i are independent.)

Bias and variance: Example

What's the bias and variance of the constant prediction c?

Bias = $c - \mu$: Systematically wrong.

Variance = zero: Prediction *doesn't vary* as data sample changes.

The bias-variance decomposition

Suppose you have a prediction approach ("learning algorithm") that, given the training sample $\mathbf{Y}$, predicts $\hat{Y}$ for the PPG of a player with true PPG Y.

The *bias-variance decomposition* is a fundamental result on the MSE of $\hat{Y}$:

$$\begin{aligned}\text{Average MSE} &= \mathbb{E}_{Y,\mathbf{Y}}[(\hat{Y} - Y)^2] \\ &= \sigma^2 + (\mathbb{E}_{\mathbf{Y}}[\hat{Y}] - \mu)^2 + \text{Var}_{\mathbf{Y}}(\hat{Y}) \\ &= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}.\end{aligned}$$

Key observation: These are the ONLY contributions to MSE!

MSE decomposition: Intuition

Bias-variance decomposition:

$$\text{MSE} = \sigma^2 + \text{Bias}^2 + \text{Variance}$$

- *Irreducible error* (σ^2): Even if we knew μ exactly, individual players' PPGs vary around μ with variance σ^2. We can't do better than this.
- *Bias*^{2}: If our prediction strategy systematically over- or under-predicts PPG (on average across samples), we pay a price in higher MSE.
- *Variance*: If our predictions vary greatly from sample to sample, we pay a price in higher MSE.

Comparing the two strategies

Bias-variance decomposition:

$$\text{MSE} = \sigma^2 + \text{Bias}^2 + \text{Variance}$$

- *Constant guess (c PPG)*: $\text{MSE} = \sigma^2 + (c - \mu)^2 + 0^2$
- *Sample mean*: $\text{MSE} = \sigma^2 + 0^2 + \sigma^2/n$

So the sample mean is better if $(c - \mu)^2 > \sigma^2/n$,
i.e., if the sample size is large, and/or our constant prediction is highly biased.

The general case

The general case

Now suppose we have training data $\mathbf{X}, \mathbf{Y}$, and we fit a predictive model $\hat{f}$ using a learning algorithm $\mathscr{L}$:

Suppose we use $\hat{f}$ to make a prediction $\hat{f}(\vec{X})$ at a new feature vector $\vec{X}$.

The squared prediction error is $(\hat{f}(\vec{X}) - Y)^2$, where Y is the true outcome associated to $\vec{X}$.

We now discuss the bias-variance decomposition for this setting.

The “best” model: Conditional expectation

Analogous to the previous setting, the *best* (lowest MSE) prediction we can make if we had access to the population model is the *conditional mean* of Y, given the features $\vec{X}$:

$$f^*(\vec{X}) = \mathbb{E}_Y[Y|\vec{X}].$$

The MSE if we use f^* is:

$$\sigma^2(\vec{X}) = \mathbb{E}_Y[(Y - f^*(\vec{X}))^2|\vec{X}] = \text{Var}_Y(Y|\vec{X}).$$

The “best” model: Conditional expectation

The best model we could ever hope for is $\hat{f} = f^*$, i.e.,
we want a fitted model as close to the conditional mean as possible.

We can’t actually make $\hat{f} = f^*$, because *we don’t know the population model!*

The generalization error can’t ever be lower than $\sigma^2(\vec{X})$,
the conditional variance of the outcome Y “left over” even if we know $\vec{X}$;
this is called the *irreducible error*.

Sampling distribution of predictions

Bias and variance are defined via the same "parallel universe" simulation that led to the sampling distribution:

- Rewind time, and resample the training data $\mathbf{X}, \mathbf{Y}$ again and again (parallel universes).

Sampling distribution of predictions

Bias and variance are defined via the same "parallel universe" simulation that led to the sampling distribution:

- Rewind time, and resample the training data $\mathbf{X}, \mathbf{Y}$ again and again (parallel universes).
- In each universe, run $\mathscr{L}$ on the sampled $\mathbf{X}, \mathbf{Y}$ to get a fitted model $\hat{f}$.

Sampling distribution of predictions

Bias and variance are defined via the same "parallel universe" simulation that led to the sampling distribution:

- Rewind time, and resample the training data $\mathbf{X}, \mathbf{Y}$ again and again (parallel universes).
- In each universe, run $\mathscr{L}$ on the sampled $\mathbf{X}, \mathbf{Y}$ to get a fitted model $\hat{f}$.
- In each universe, evaluate $\hat{f}(\vec{X})$.

Sampling distribution of predictions

Bias and variance are defined via the same "parallel universe" simulation that led to the sampling distribution:

- Rewind time, and resample the training data $\mathbf{X}, \mathbf{Y}$ again and again (parallel universes).
- In each universe, run $\mathscr{L}$ on the sampled $\mathbf{X}, \mathbf{Y}$ to get a fitted model $\hat{f}$.
- In each universe, evaluate $\hat{f}(\vec{X})$.

This gives rise to a *sampling distribution* of $\hat{f}(\vec{X})$.

Average MSE (over sampling distribution)

The resulting average MSE (over these universes) is:

$$\text{Average MSE at } \vec{X} = \mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - Y)^2 | \vec{X}].$$

Note in this expectation *both* the true outcome Y *and* the training data $\mathbf{X}, \mathbf{Y}$ are *random*.

The bias-variance decomposition

Definitions:

$$\text{Bias} = \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\hat{f}(\vec{X})|\vec{X}] - f^*(\vec{X});$$
$$\text{Variance} = \text{Var}_{\mathbf{X},\mathbf{Y}}(\hat{f}(\vec{X}|\vec{X}) = \mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[(\hat{f}(\vec{X}) - \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\hat{f}(\vec{X})|\vec{X}])^2\right].$$

The bias-variance decomposition is:

$$\text{Average MSE at } \vec{X} = \sigma^2(\vec{X}) + \text{Bias}^2 + \text{Variance}.$$

As before: **These are the ONLY contributions to MSE!**

Examples: Bias and variance

Suppose you are predicting, e.g., wealth based on a collection of demographic features.

- Suppose we make a constant prediction: $\hat{f}(\vec{X}) = c$.

  Does this have low bias? Does it have low variance?

- Suppose that every time you get your data, you use enough parameters to fit $\mathbf{Y}$ *exactly*: $\hat{f}(\mathbf{X}_i) = Y_i$ for all points in the training data.

  Does this have low bias? Does it have low variance?

Example: k-nearest-neighbors

Generate synthetic data:

We generate 1000 X_1, X_2 as i.i.d. $N(0,1)$ random variables.

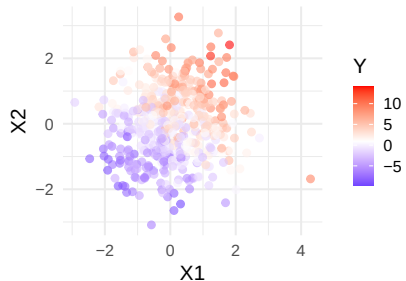
We then generate 1000 Y random variables as:

$$Y_i = 1 + 2X_{i1} + 3X_{i2} + \epsilon_i,$$

where ϵ_i are i.i.d. $N(0,5)$ random variables.

Example: k-nearest-neighbors

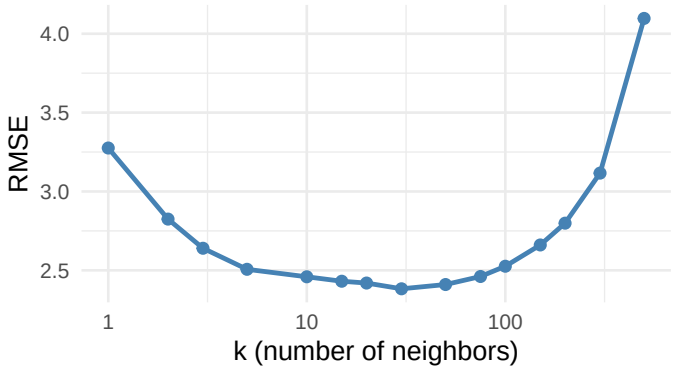
Using the first 500 points, we create a *k-nearest-neighbor* (k-NN) model: For any $\vec{X}$, let $\hat{f}(\vec{X})$ be the average value of Y_i over the k nearest neighbors $\mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(k)}$ to $\vec{X}$ in the training set.



How does this predictive model behave as a function of k?

Example: k-nearest-neighbor fit

The graph shows root mean squared error (RMSE) over the remaining 500 samples (test set) as a function of k:



Example: k-nearest-neighbor fit

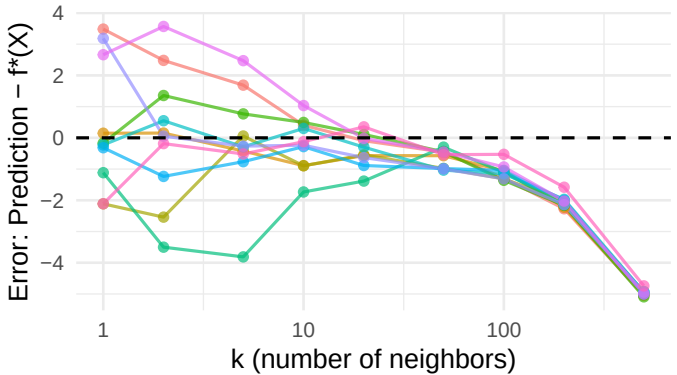
We can get more insight into why RMSE behaves this way if we pick a specific point and look at how our prediction error varies with k.

In particular, repeat the following process 10 times:

- Each time, we start with a new training dataset of 500 samples.
- We use this training dataset to make predictions at the specific point $X_1 = 1$, $X_2 = 1$.
- We plot the error: our prediction minus $f^*(\vec{X})$, where $f^*(\vec{X}) = 1 + 2X_1 + 3X_2 = 6$ is the true conditional mean of Y given $\vec{X}$ in the population model.

Example: k-nearest neighbor fit

Results (each color is one repetition):



In other words, as *k* increases, *variance* goes down while *bias* goes up.

Bias, variance, and linear regression

OLS linear regression

To better understand the role that the set of features plays in bias and variance, let's briefly discuss bias and variance for *OLS linear regression*.

We'll start a classic result on bias and variance under some (unrealistic!) assumptions on the population model, and then use this setting to investigate how bias and variance change as the set of features changes.

The linear population model: Three assumptions

(A1) The population model is *linear*, i.e., there are *parameters* $\beta_1, \ldots, \beta_p$ such that for every observation:

$$Y = \sum_{j=1}^{p} \beta_j X_j + \epsilon,$$

where Y is the outcome, $X_1, \ldots, X_p$ are the associated features, and ϵ is an *error* random variable.

(A2) Observations in the sample are independently drawn from the population model; in particular, errors are independent across observations, and also independent of the features $X_1, \ldots, X_p$.[2]

Crucially: *all features in the population model are also in the sample!*

(A3) All errors have $\mathbb{E}[\epsilon|\vec{X}] = 0$, and $\mathrm{Var}(\epsilon|\vec{X}) = \sigma^2$.

(The assumption that errors have the same variance is called *homoskedasticity*.)

[2]This assumption can actually be relaxed slightly; see the appendix.

The linear population model

This is a very strong set of assumptions!

Example: houses.

(A1) The price of a house is *linear* in a given set of features (e.g., living area, number of bedrooms, number of bathrooms), plus some (random) error.

(A2) The houses in our sample data set are independent draws from this model, *including all the features from (A1)*.

(A3) Errors are independent with zero mean and constant variance.

Hard to believe this is true...

The Gauss-Markov Theorem

The *Gauss-Markov Theorem* (GMT) for OLS (informally) states that if (A1)-(A3) hold, then:

- The irreducible error is σ^2.
- The bias of OLS is **zero**.
- In addition, among *all* linear modeling strategies using the features in $\vec{X}$ that have *zero bias*, OLS has the *lowest variance*.

The Gauss-Markov Theorem

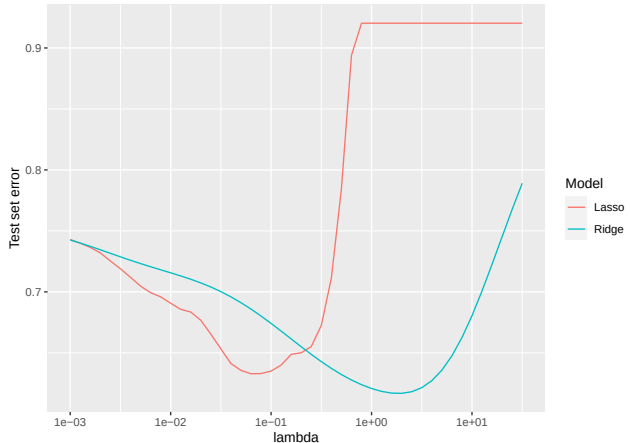
What are the implications?

- To lower prediction error, we have to lower bias, variance, or both.
- The GMT says if (A1)-(A3) hold, we can't lower variance if we insist on a linear modeling strategy with zero bias.
- So any linear modeling strategy with lower prediction error than OLS must have **positive bias**, and **lower variance** than OLS.

This observation is one of the important justifications for the value of linear modeling strategies such as lasso and ridge regression: these will generally have positive bias, but might offer sufficiently lower variance in return to lower MSE overall.

Example: Lasso and ridge

Recall the previously discussed example of lasso and ridge predictions with baseball hitters' salaries:



Example: Lasso and ridge

As λ increases, MSE first falls then rises.

Informally, as λ increases, the models put less and less weight on the available features (lasso actually zeroes out coefficients), so:

- The models are systematically wrong $\implies$ higher bias.
- The models also become less sensitive to the training data $\implies$ lower variance.

OLS with fewer features

Let's continue to assume that (A1)-(A3) hold, and explore the role of the number of features in OLS in greater detail.

What happens if we fit our model using only a subset of features $S \subset \{0, \dots, p\}$ from the population model?

In general, the resulting OLS model will be *biased*.

OLS with fewer features

A couple remarks:

- In general, the amount of bias introduced will depend on how correlated the *remaining* features are with the *omitted* features. (This is why it can be possible to make good predictions despite the omission of variables, as you saw on your problem set.)
- When features are omitted, another concern is that the estimates of the coefficients of the included features may be incorrect. This phenomenon often appears as the *omitted variable bias* (OVB) in econometrics; we will return to this in our next unit on inference.

OLS with more features

What happens if we introduce a new feature into our modeling strategy?

As an extreme case, suppose our new feature is *uncorrelated* with the existing features and the outcome. Then:

- The bias remains zero.
- However, the variance will *increase*.

(See the R Shiny dashboard for an example.)

A bias-variance "tradeoff"?

If we measure "complexity" of a modeling strategy in terms of *the number of features we use to fit the model*, then our preceding discussion might lead us to believe that:

- More "complex" modeling strategies (i.e., more features) tend to have higher variance and lower bias.
- Less "complex" modeling strategies (i.e., fewer features) tend to have lower variance and higher bias.

This is usually what's meant by a "bias-variance tradeoff"...

...Is this all there is to the story?

A bias-variance “tradeoff”?

The “tradeoff” view can be misleading!

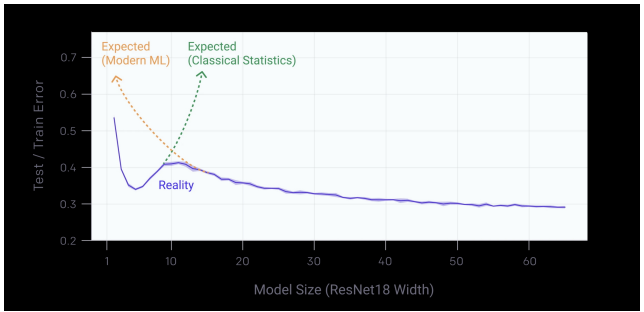
It’s possible to have high bias and high variance at the same time, and bias and variance can move in unusual ways with model “complexity”...
In general, adding features can affect both bias and variance, depending on the correlation between the new features, existing features, and the outcome.

Let’s dig into this a bit more...

Double descent

A puzzle: Double descent

Developments over the last decade in machine learning have challenged the "tradeoff": for some combinations of data context, modeling strategy, and model fitting procedure, the following relationship between train error, test error, and "complexity" is observed:



(From `openai.com/blog/deep-double-descent/`)

“Overparameterization”

The double descent phenomenon highlights the intriguing phenomenon that modeling strategies that appear to be “overparameterized” can in fact generalize very well:

Informally, models so “complex” that they can nearly perfectly fit the training data, are nevertheless not necessarily *overfitting* the data; they appear to have both *low bias* and *low variance*.

How?

Double descent: An example

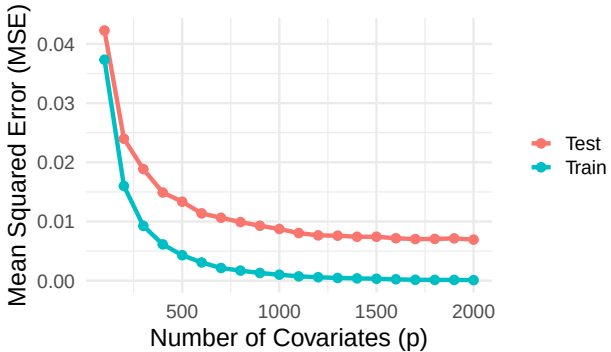
Let's explore double descent with lasso. We'll use synthetic data generated as follows:

1. *First*, generate $Y_1, \ldots, Y_{1000}$ as $N(0,1)$ random variables.
2. *Then*, for each $i = 1, \ldots, 1000$, generate 2000 *features* $X_{i1}, \ldots, X_{i,2000}$ as $X_{ij} = Y_i + s_{ij}$, where s_{ij} is $N(0, \tau^2)$.
3. Finally, we fit lasso models using *only* the first p features, for $p = 100, 200, \ldots, 2000$ (with a fixed value of λ), and test them on a test set with 5000 data points,

What's unusual about this data generating procedure?

Results

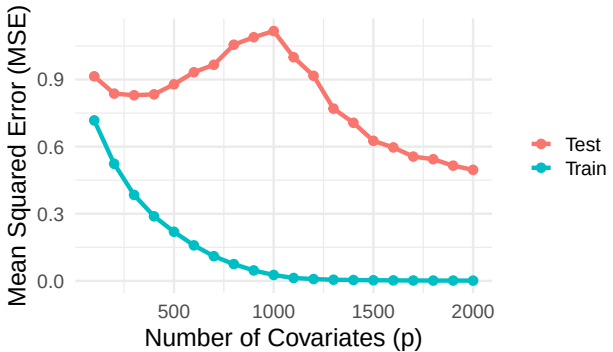
If $\tau = 2$ and $\lambda = 0.001$, our test error continues to drop with increasing p:



Why? Because *every* additional feature brings new information about the outcome.

Results

What happened to "double descent"? Now suppose $\tau = 20$ and $\lambda = 0.001$:



Why? As $p \to 1000$, lasso gets "confused": τ^2 is high, and λ isn't very large, so it fits the training data well ... but then generalizes poorly. But with more features, this "confusion" is overcome.

Moral: "Big data" and highly parameterized models

Usually, we think of "big data" as having more samples in our data (larger n).

But an equally consequential change is the availability (and relevance to the population model!) of *many more features* (larger p).

Moral: "Big data" and highly parameterized models

In many ways, "double descent" is not the central story here: it is a practical artifact of a particular modeling strategy and model fitting procedure passing through "confusion" as p grows, so that variance rises then falls.

Instead a key takeaway is:

With the right data context, right modeling strategy, and right model fitting procedure, *if* those new features all bring novel information about the outcome, bias *and* variance can continue to decrease as you use them. (See R Shiny dashboard for an example.)

A comment on "irreducible" error

In this data generating procedure, *every new feature brings novel information about the outcome*.

Among other things, this means the concept of "irreducible" error $\sigma^2(\vec{X})$ is unusual:

- Irreducible error is defined with respect to the *features* in $\vec{X}$.
- But if we keep adding features, then we are effectively defining a new irreducible error, with that new set of features.
- And if those new features keep bringing additional information, then the "irreducible" error will be reduced!

This is an important point in practice: you shouldn't assume that "irreducible" error is truly irreducible, since new features can actually reduce that error.

Summary

Bias-variance decomposition: Summary

The bias-variance decomposition is a conceptual guide to understanding what influences generalization error:

- Frames the question by asking: *What if we were to repeatedly use the same modeling strategy, but on different training sets?*
- *Generalization error has only three components!* On average (across models built from different training sets):

$$\text{irreducible error} + \text{bias}^2 + \text{variance}.$$

- *Bias*: systematic mistakes in predictions on test data, regardless of the training set
- *Variance*: variation in predictions on test data, as training data changes

Modeling strategies can be "bad" (high test error) because of high bias or high variance (or both).

Coda: "Bias" and "variance" in classification problems

Note that formally, the bias-variance decomposition only applies for *regression problems with squared error loss.*

What about classification problems? Similar qualitative phenomena are observed (e.g., the double descent example above from OpenAI is in the context of image classification.)

There are versions of the B-V decomposition that have been developed for 0-1 loss as well, though one has to be careful in interpreting them.[3]

[3] One example: P. Domingos (2000), "A unified bias-variance decomposition", ICML.

Appendix: Proof of the bias-variance decomposition [$*$]

Bias-variance decomposition theorem [$*$]

In this appendix we prove the bias-variance decomposition for the general case.

Theorem

$$\mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - Y)^2 | \vec{X}] = \sigma^2(\vec{X}) + \mathit{Bias}^2 + \mathit{Variance}, \qquad ((*))$$

where:

$$\begin{aligned}
\sigma^2(\vec{X}) &= \mathbb{E}_Y[(Y - f^*(\vec{X}))^2 | \vec{X}] = \mathrm{Var}_Y(Y | \vec{X}); \\
\mathit{Bias} &= \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\hat{f}(\vec{X}) | \vec{X}] - f^*(\vec{X}); \\
\mathit{Variance} &= \mathrm{Var}_{\mathbf{X},\mathbf{Y}}(\hat{f}(\vec{X} | \vec{X}) = \mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[(\hat{f}(\vec{X}) - \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\hat{f}(\vec{X}) | \vec{X}])^2\right].
\end{aligned}$$

Proof [$*$]

We denote the left hand side of ($*$) by AMSE (for "average MSE").

We first show:

$$\text{AMSE} = \sigma^2(\vec{X}) + \mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[(\hat{f}(\vec{X}) - f^*(\vec{X}))^2 | \vec{X}\right].$$

Proof (continued) [$*$]

Starting with the definition of AMSE, add and subtract $f^*(\vec{X})$:

$$\text{AMSE} = \mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - f^*(\vec{X}) + f^*(\vec{X}) - Y)^2 | \vec{X}].$$

Expand the square:

$$\begin{aligned}\text{AMSE} = \mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}\Big[&(\hat{f}(\vec{X}) - f^*(\vec{X}))^2 \\ &+ 2(\hat{f}(\vec{X}) - f^*(\vec{X}))(f^*(\vec{X}) - Y) \\ &+ (f^*(\vec{X}) - Y)^2 \mid \vec{X}\Big].\end{aligned}$$

Proof (continued) [$*$]

By linearity of expectation, we get three terms:

$$\begin{aligned}\text{AMSE} = &\underbrace{\mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - f^*(\vec{X}))^2 | \vec{X}]}_{\text{Term 1}} \\ &+ \underbrace{2\mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - f^*(\vec{X}))(f^*(\vec{X}) - Y) | \vec{X}]}_{\text{Term 2 (cross term)}} \\ &+ \underbrace{\mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}[(f^*(\vec{X}) - Y)^2 | \vec{X}]}_{\text{Term 3}}.\end{aligned}$$

Term 3: Since $f^*(\vec{X})$ depends only on $\vec{X}$ (which we condition on), and Y is independent of the training data:

$$\mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}[(f^*(\vec{X}) - Y)^2 | \vec{X}] = \mathbb{E}_Y[(Y - f^*(\vec{X}))^2 | \vec{X}] = \sigma^2(\vec{X}).$$

Proof (continued) [$*$]

Term 2 (cross term): Using the law of iterated expectations:

$$\begin{aligned} &2\mathbb{E}_{Y,\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - f^*(\vec{X}))(f^*(\vec{X}) - Y)|\vec{X}] \\ &= 2\mathbb{E}_{\mathbf{X},\mathbf{Y}}\left[(\hat{f}(\vec{X}) - f^*(\vec{X})) \cdot \mathbb{E}_Y[(f^*(\vec{X}) - Y)|\vec{X}] \mid \vec{X}\right]. \end{aligned}$$

Since $f^*(\vec{X}) = \mathbb{E}_Y[Y|\vec{X}]$ by definition:

$$\mathbb{E}_Y[(f^*(\vec{X}) - Y)|\vec{X}] = f^*(\vec{X}) - \mathbb{E}_Y[Y|\vec{X}] = f^*(\vec{X}) - f^*(\vec{X}) = 0.$$

Therefore, **Term 2 = 0**.

Combining all terms completes the first part:

$$\text{AMSE} = \sigma^2(\vec{X}) + \mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - f^*(\vec{X}))^2|\vec{X}].$$

Proof (continued) [$*$]

Now we complete the proof by decomposing the second term into Bias^2 + Variance.

Let $\bar{f}(\vec{X}) = \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\hat{f}(\vec{X})|\vec{X}]$ denote the average prediction across parallel universes.

Parallel to the first step: Add and subtract $\bar{f}(\vec{X})$:

$$\hat{f}(\vec{X}) - f^*(\vec{X}) = [\hat{f}(\vec{X}) - \bar{f}(\vec{X})] + [\bar{f}(\vec{X}) - f^*(\vec{X})].$$

Square both sides:

$$\begin{aligned}(\hat{f}(\vec{X}) - f^*(\vec{X}))^2 =& (\hat{f}(\vec{X}) - \bar{f}(\vec{X}))^2 \\ &+ 2(\hat{f}(\vec{X}) - \bar{f}(\vec{X}))(\bar{f}(\vec{X}) - f^*(\vec{X})) \\ &+ (\bar{f}(\vec{X}) - f^*(\vec{X}))^2.\end{aligned}$$

Proof (continued) [$*$]

Take expectations to get three terms:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - f^*(\vec{X}))^2 | \vec{X}] \\
&= \underbrace{\mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - \bar{f}(\vec{X}))^2 | \vec{X}]}_{\text{Term A}} \\
&\quad + \underbrace{2\mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - \bar{f}(\vec{X}))(\bar{f}(\vec{X}) - f^*(\vec{X}))|\vec{X}]}_{\text{Term B (cross term)}} \\
&\quad + \underbrace{\mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\bar{f}(\vec{X}) - f^*(\vec{X}))^2 | \vec{X}]}_{\text{Term C}}.
\end{aligned}
$$

Term A: By definition, this is exactly the Variance:

$$\mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - \bar{f}(\vec{X}))^2 | \vec{X}] = \text{Variance}.$$

Proof (continued) [$*$]

Term C: Since both $\bar{f}(\vec{X})$ and $f^*(\vec{X})$ are deterministic given $\vec{X}$:

$$\mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\bar{f}(\vec{X}) - f^*(\vec{X}))^2 | \vec{X}] = (\bar{f}(\vec{X}) - f^*(\vec{X}))^2 = \text{Bias}^2.$$

Term B (cross term): Factor out the deterministic part:

$$\begin{aligned} &2\mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - \bar{f}(\vec{X}))(\bar{f}(\vec{X}) - f^*(\vec{X}))|\vec{X}] \\ &= 2(\bar{f}(\vec{X}) - f^*(\vec{X})) \cdot \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\hat{f}(\vec{X}) - \bar{f}(\vec{X})|\vec{X}]. \end{aligned}$$

But $\mathbb{E}_{\mathbf{X},\mathbf{Y}}[\hat{f}(\vec{X}) - \bar{f}(\vec{X})|\vec{X}] = \bar{f}(\vec{X}) - \bar{f}(\vec{X}) = 0.$

Therefore, **Term B = 0**.

Proof (conclusion) [$*$]

Combining all terms:

$$\mathbb{E}_{\mathbf{X},\mathbf{Y}}[(\hat{f}(\vec{X}) - f^*(\vec{X}))^2 | \vec{X}] = \text{Bias}^2 + \text{Variance}.$$

Together with the first part of the proof, we have:

$$\boxed{\text{AMSE} = \sigma^2(\vec{X}) + \text{Bias}^2 + \text{Variance}.}$$

Key insight: Both parts used the same technique: add and subtract an expectation, then show the cross term vanishes because we're adding/subtracting exactly that expectation.

Appendix: Proof of the Gauss-Markov Theorem [$*$]

Technical details for linear regression [$*$]

Throughout this appendix, we assume that (A1)-(A3) as stated in the lecture hold.

We begin by noting that we can slightly relax assumption (A2) as stated in the text:

Rather than independence of the errors, it suffices that they are uncorrelated with the covariates and conditionally homoskedastic given the covariates:

$$\mathbb{E}[\epsilon_i|\mathbf{X}] = 0; \quad \mathrm{Var}(\epsilon_i|\mathbf{X}) = \sigma^2 \text{ for all } i.$$

Linear regression: bias [$*$]

This derivation shows bias of OLS is zero:

$$\begin{aligned}
\mathbb{E}_{\mathbf{Y}}[\hat{f}(\vec{X})|\vec{X},\mathbf{X}] &= \mathbb{E}_{\mathbf{Y}}[\vec{X}\hat{\boldsymbol{\beta}}|\vec{X},\mathbf{X}] \\
&= \mathbb{E}_{\mathbf{Y}}[\vec{X}(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{Y})|\vec{X},\mathbf{X}] \\
&= \vec{X}(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top\left(\mathbb{E}_{\boldsymbol{\epsilon}}[\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}|\vec{X},\mathbf{X}]\right) \\
&= \vec{X}(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{X})\boldsymbol{\beta} = \vec{X}\boldsymbol{\beta} = f(\vec{X}).
\end{aligned}$$

The Gauss-Markov theorem: Precise statement [$*$]

The Gauss-Markov theorem shows that among all unbiased linear modeling strategies, OLS has minimum variance.

Theorem (Gauss-Markov)

Assume a linear population model with uncorrelated errors. Fix a (row) covariate vector $\vec{X}$*, and let* $\gamma = \vec{X}\boldsymbol{\beta} = \sum_j \beta_j X_j$.

Given data $\mathbf{X}, \mathbf{Y}$*, let* $\hat{\boldsymbol{\beta}}$ *be the OLS solution. Let* $\hat{\gamma} = \vec{X}\hat{\boldsymbol{\beta}} = \vec{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$.

Let $\hat{\delta} = \mathbf{g}(\mathbf{X}, \vec{X})\mathbf{Y}$ *be any other estimator for* γ *that is linear in* $\mathbf{Y}$ *and* unbiased *for all* $\mathbf{X}$*:* $\mathbb{E}_{\mathbf{Y}}[\hat{\delta}|\mathbf{X}, \vec{X}] = \gamma$.

Then $\mathrm{Var}(\hat{\delta}|\mathbf{X}, \vec{X}) \geq \mathrm{Var}(\hat{\gamma}|\mathbf{X}, \vec{X})$*, with equality if and only if* $\hat{\delta} = \hat{\gamma}$.

The Gauss-Markov theorem: Proof [$*$]

Proof:[4] We compute the variance of $\hat{\delta}$.

$$\begin{aligned}
&\mathbb{E}[(\hat{\delta} - \mathbb{E}[\hat{\delta}|\vec{X}, \mathbf{X}])^2|\vec{X}, \mathbf{X}] \\
&\quad = \mathbb{E}[(\hat{\delta} - \vec{X}\boldsymbol{\beta})^2|\vec{X}, \mathbf{X}] \\
&\quad = \mathbb{E}[(\hat{\delta} - \hat{\gamma} + \hat{\gamma} - \vec{X}\boldsymbol{\beta})^2|\vec{X}, \mathbf{X}] \\
&\quad = \mathbb{E}[(\hat{\delta} - \hat{\gamma})^2|\vec{X}, \mathbf{X}] \\
&\qquad + \mathbb{E}[(\hat{\gamma} - \vec{X}\boldsymbol{\beta})^2|\vec{X}, \mathbf{X}] \\
&\qquad + 2\mathbb{E}[(\hat{\delta} - \hat{\gamma})(\hat{\gamma} - \vec{X}\boldsymbol{\beta})|\vec{X}, \mathbf{X}].
\end{aligned}$$

Look at the last equality: If we can show the last term is zero, then we would be done, because the first two terms are uniquely minimized if $\hat{\delta} = \hat{\gamma}$.

[4] Throughout this proof we suppress subscripts on the expectations for ease of exposition.

The Gauss-Markov theorem: Proof [$*$]

Proof continued: For notational simplicity let $\mathbf{c} = \mathbf{g}(\mathbf{X}, \vec{X})$. We have:

$$\begin{aligned}&\mathbb{E}[(\hat{\delta} - \hat{\gamma})(\hat{\gamma} - \vec{X}\boldsymbol{\beta})|\vec{X}, \mathbf{X}] \\ &\quad = \mathbb{E}[(\mathbf{c}\mathbf{Y} - \vec{X}\hat{\boldsymbol{\beta}})^2(\vec{X}\hat{\boldsymbol{\beta}} - \vec{X}\boldsymbol{\beta})|\vec{X}, \mathbf{X}].\end{aligned}$$

Now using the fact that $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$, that $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, and the fact that $\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\vec{X}, \mathbf{X}] = \sigma^2\mathbf{I}$, the last quantity reduces (after some tedious algebra) to:

$$\sigma^2\vec{X}(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{c}^\top - \vec{X}^\top).$$

The Gauss-Markov theorem: Proof [$*$]

Proof continued: To finish the proof, notice that from unbiasedness we have:

$$\mathbb{E}[\mathbf{c}\mathbf{Y}|\mathbf{X}, \vec{X}] = \vec{X}\boldsymbol{\beta}.$$

But since $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbb{E}[\boldsymbol{\epsilon}|\mathbf{X}, \vec{X}] = 0$, we have:

$$\mathbf{c}\mathbf{X}\boldsymbol{\beta} = \vec{X}\boldsymbol{\beta}.$$

Since this has to hold true for every $\mathbf{X}$, we must have $\mathbf{c}\mathbf{X} = \vec{X}$, i.e., that:

$$\mathbf{X}^\top\mathbf{c}^\top - \vec{X}^\top = 0.$$

This concludes the proof.

Variance of OLS [$*$]

We can explicitly work out the variance of OLS in the linear population model, *if* we assume that the covariates $\mathbf{X}$ stay fixed across parallel universes, and we *only* resample the outcomes $\mathbf{Y}$.

$$\begin{aligned}\mathrm{Var}(\hat{f}(\vec{X})|\mathbf{X}, \vec{X}) &= \mathrm{Var}(\vec{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}|\mathbf{X}, \vec{X}) \\ &= \vec{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \mathrm{Var}(\mathbf{Y}|\mathbf{X})\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\vec{X}^\top.\end{aligned}$$

Now note that $\mathrm{Var}(\mathbf{Y}|\mathbf{X}(= \mathrm{Var}(\boldsymbol{\epsilon}|\mathbf{X}) = \sigma^2\mathbf{I}$ where $\mathbf{I}$ is the $n \times n$ identity matrix.

Therefore:

$$\mathrm{Var}(\hat{f}(\vec{X})|\mathbf{X}, \vec{X}) = \sigma^2\vec{X}(\mathbf{X}^\top\mathbf{X})^{-1}\vec{X}^\top.$$