Midterm Review

Fall 2019

Comments

- Midterm location on schedule
  - CCIS 1 440
- Covers Chapters 1-6 (up to GLMs, but not including GLMs)
- My office hours are today, from 2-4 p.m.

High-level comments

- In some cases lots of space, but doesn't mean you need to fill the entire page. If the answer can be concisely stated in 2 sentences, that is perfectly reasonable

- I am not looking for one specific "right" answer. I am looking for your thought process, to see if you understood the material

- In the past, the wrong answer has been given (e.g., true, false question), but I gave full marks because the reasoning demonstrated understanding

- **Two common mistakes:** (a) giving multiple answers, in case one is right. You'll lose marks for this (b) answering a different question than the one asked (read the question)

Probability review

- Quantify uncertainty using probability theory
- Discussed sigma-algebras and probability measures
- Discussed random variables as functions of event-space
- Discussed relationships between random variables, including (in)dependence and conditional independence
- Discussed operations, like expected value, marginalization, Bayes rule, chain rule

Exercise: Probability

- Suppose that we have created a machine learning algorithm that predicts whether a link will be clicked. It has a TPR (true positive rate) of 0.99 and a FPR (false positive rate) of 0.01.
  - Let C be binary RV, with C = 1 indicating Predict Click
  - Let Y be binary RV, with Y = 1 indicating Actual Click
  - p(C = 1 | Y = 1) = TPR
  - p(C = 1 | Y = 0) = FPR

- The rate the link is actually clicked is 1/1000 visits to a website. If we predict the link will be clicked on a specific visit, what is the probability it will actually be clicked?

Exercise: MAP and ML

- See a set of random variables X1, …, Xn
- We assumed that these RVs are **independent** and **identically** distributed
- Then we assumed a distribution for each p(Xi l theta)
  - e.g., p(Xi l theta) is a Gaussian
- How would our ML objectives change if we did not assume independence?
  - previously maximized p(D l theta) = prod_{i=1}^n p(Xi l theta)

(we'll do this on the whiteboard)

Exercise: Bias of Sample Average

- See a set of random variables X1, …, Xn (dataset of size n)
- We showed the sample average is unbiased if the Xi are iid
- Is it biased or unbiased if the data is identically distributed, but not independent?
- (Why do we talk about the dataset being random?)

Optimization

- We've discussed first and second-order gradient descent
- We've discussed batch gradient descent, mini-batch gradient descent and stochastic gradient descent
  - one epoch corresponds to going once over the entire dataset (size n)
  - each batch GD update uses one epoch (across all n samples)
  - SGD does n updates for each epoch, noisy gradient
  - mini-batch SGD does n / b updates for each epoch, where b is the batch size and the gradient is slightly less noisy than 1-sample SGD

Optimization

- We've discussed batch gradient descent, mini-batch gradient descent and stochastic gradient descent
  - one epoch corresponds to going once over the entire dataset (size n)
  - each batch GD update uses one epoch (across all n samples)
  - SGD does n updates for each epoch, noisy gradient
  - mini-batch SGD does n / b updates for each epoch, where b is the batch size and the gradient is slightly less noisy than 1-sample SGD

- We've only done first-order SGD. What about second order?
  - Can also consider stochastic Hessian approximations
  - The real goal is to scale the update, to make flat region more curved and really sharp regions more flat
  - Common to use vector of stepsizes (diagonal approximation to Hinv), scales the update to each dimension separately

Practice midterm question

A typical goal behind data normalization is to make all the features of the same scale. For example, the features are rescaled to be zero-mean, with unit variance, by taking the data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and centering and normalizing each column: $\mathbf{X}_{ij} = \frac{\mathbf{X}_{ij} - \mu_j}{\sigma_j}$ where $\mu_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_{ij}$ and $\sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{X}_{ij} - \mu_j)^2$. Why might this be important for gradient descent? Hint: Consider a setting where you have two features, where the first has range $[0, 0.01]$ and the other $[0, 1000]$. Think about the stochastic gradient descent update for linear regression, and what issues might arise.

Exercise: Understand behaviour of algorithms

- **Q1:** Run l1-LinearRegression twice on the same data, using SGD. Should you expect to get about the same error on the testing set?

- **Q2:** Use squared error for binary classification (nonconvex obj). Again, run twice on the same data, using SGD. Should you expect to get about the same error on the testing set?

- **Q3:** Use squared error for binary classification (nonconvex obj). Run, with SGD, on the same training data, in the same order, with the same starting point —> should it produce the same final set of weights?

- **Q4:** When performing batch gradient descent with line search, should you expect loss(wt) to consistently decrease?

- **Q5:** When using SGD with a small fixed stepsize, should you expect loss(wt) to consistently decrease?

Terminology clarification

- I have used the words Cost, Objective, Loss, and Error
- These are all sometimes used interchangeably in ML
- I try to use these specifically to mean
  - **Cost** — the true cost for errors in our prediction (e.g., for classification, we had a 0-1 cost in Chapter 3).
  - **Objective** — the thing we are trying to optimize, usually labelled as little c in optimizations. It can be more general than an error
  - **Loss** — typically also means the objective, where we minimize Loss. Sometimes we have used it to be the error term in the objective.
  - **Error** — usually for the least-squares error (reflects error in prediction inside objective); I'll try to avoid this extra term

Optimal models versus Estimated models

- Chapter 4: Talked about optimal models
  - For 0-1 Cost, found f*(x) = argmax_y p(y | x)
  - For Squared-error Cost, found f*(x) = E[Y | x]
  - For Absolute-error Cost, found f*(x) = median(Y | x)
  - None of Chapter 3 is about estimation, only about what functions we wish we could get to use for prediction
- Afterwards (Linear regression, GLMs), discussed surrogates to find these f* — we did NOT directly minimize this cost

Exercise: Why don't we directly minimize the Cost?

- In one case we do: linear regression. Otherwise, not often use MLE/MAP objectives as proxy to minimize Expected Cost

- Let's imagine we did directly minimize Expected Cost, for binary classification

$$\mathbb{E}[C] = \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} \text{cost}(f(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x}$$

$$\mathbb{E}[C] = \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} 1(f(\mathbf{x}) \neq y) p(\mathbf{x}, y) d\mathbf{x}$$

- We can obtain an Empirical Cost, to estimate this cost

$$\sum_{i=1}^{n} 1(f(\mathbf{x}_i) \neq y_i)$$

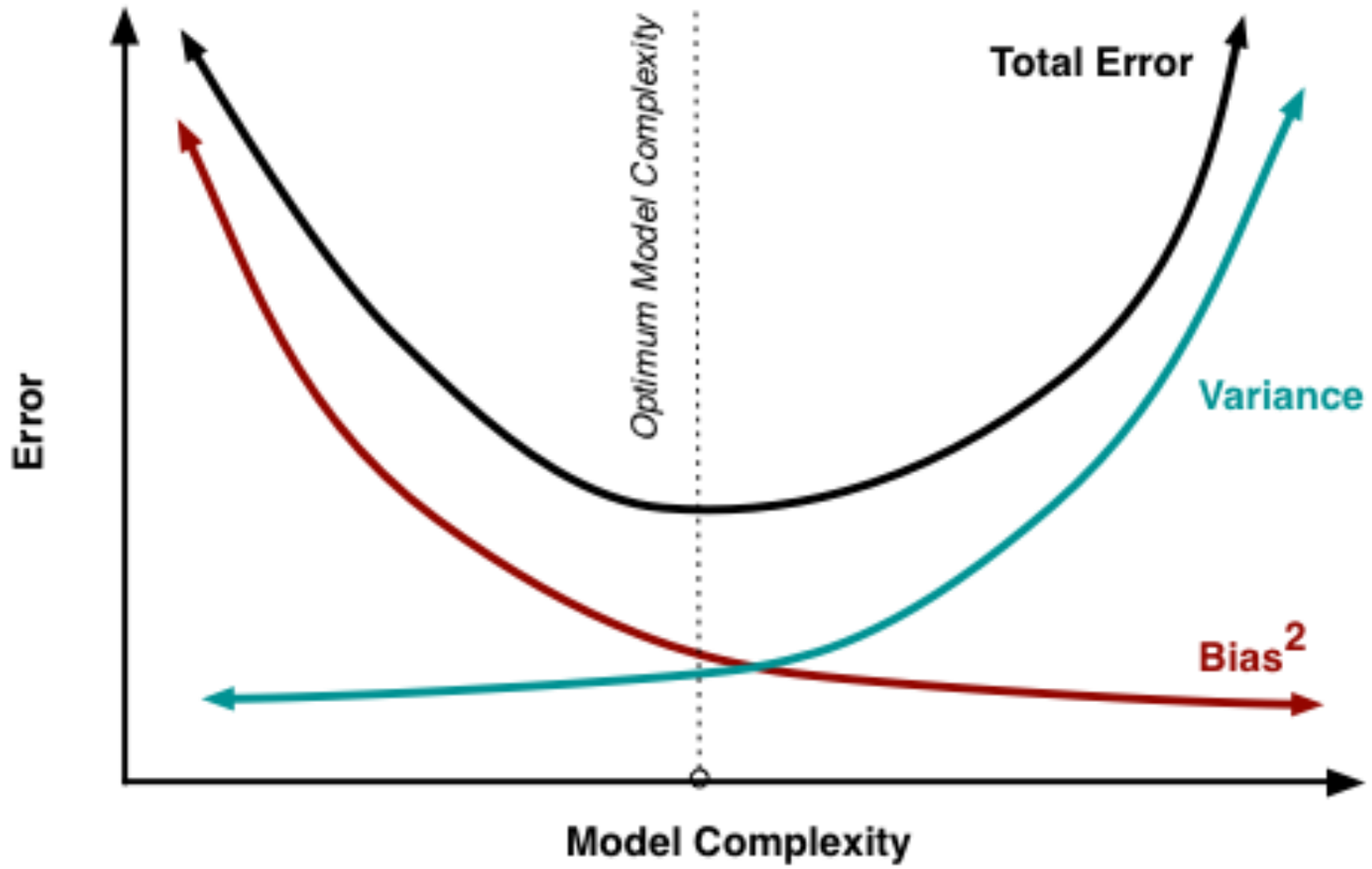
- How might we minimize this?

Expected Cost = Reducible and Irreducible Error

$$\begin{aligned}\mathbb{E}[C] &= \mathbb{E}[(f(\boldsymbol{X}) - Y)^2] \\ &= \underbrace{\mathbb{E}[(f(\boldsymbol{X}) - \mathbb{E}[Y|\boldsymbol{X}])^2]}_{\text{reducible error}} + \underbrace{\mathbb{E}[(\mathbb{E}[Y|\boldsymbol{X}] - Y)^2]}_{\text{irreducible error}}.\end{aligned}$$

- We cannot reduce irreducible error
- Our goal is to find f to **minimize reducible error**
  - Note that directly minimizing expected cost will minimize reducible error, since we cannot influence irreducible error
- We cannot directly minimize this true reducible error (expectation over all pairs (x,y)); instead, we use sampled training data to minimize it

Bias-variance (trade-off)

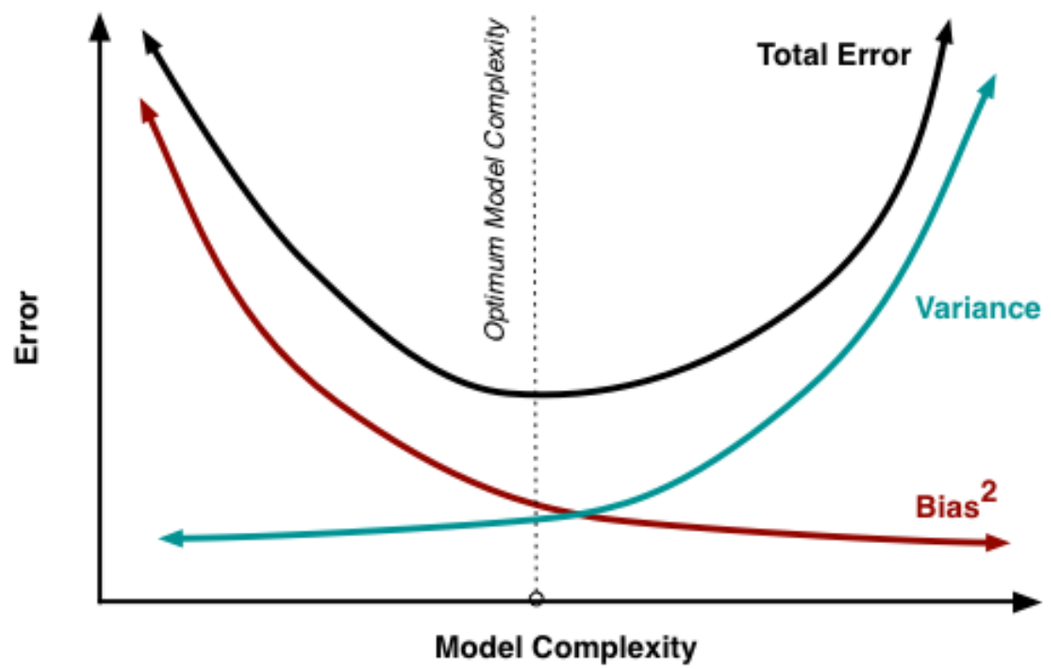


Reducible Error = Bias^2 + Variance

Realizable Setting: $\text{Bias} = \mathbb{E}[\mathbf{w}(\mathcal{D})] - \boldsymbol{\omega}$ $\text{Variance} = \mathbb{V}[\mathbf{w}(\mathcal{D})]$

Bias-variance (generally)

$$\begin{aligned}
\text{Reducible Error} &= \mathbb{E}[(\hat{f}(X) - f(X))^2] \\
&= \mathbb{E}[\mathbb{E}[(\hat{f}(x) - f(x))^2]|X = x]] \\
\text{Given } x\text{:} &= \mathbb{E}[(\hat{f}(x) - f(x))^2] \\
&= (\mathbb{E}[(\hat{f}(x)] - f(x))^2 + \mathbb{V}[(\hat{f}(x)]
\end{aligned}$$

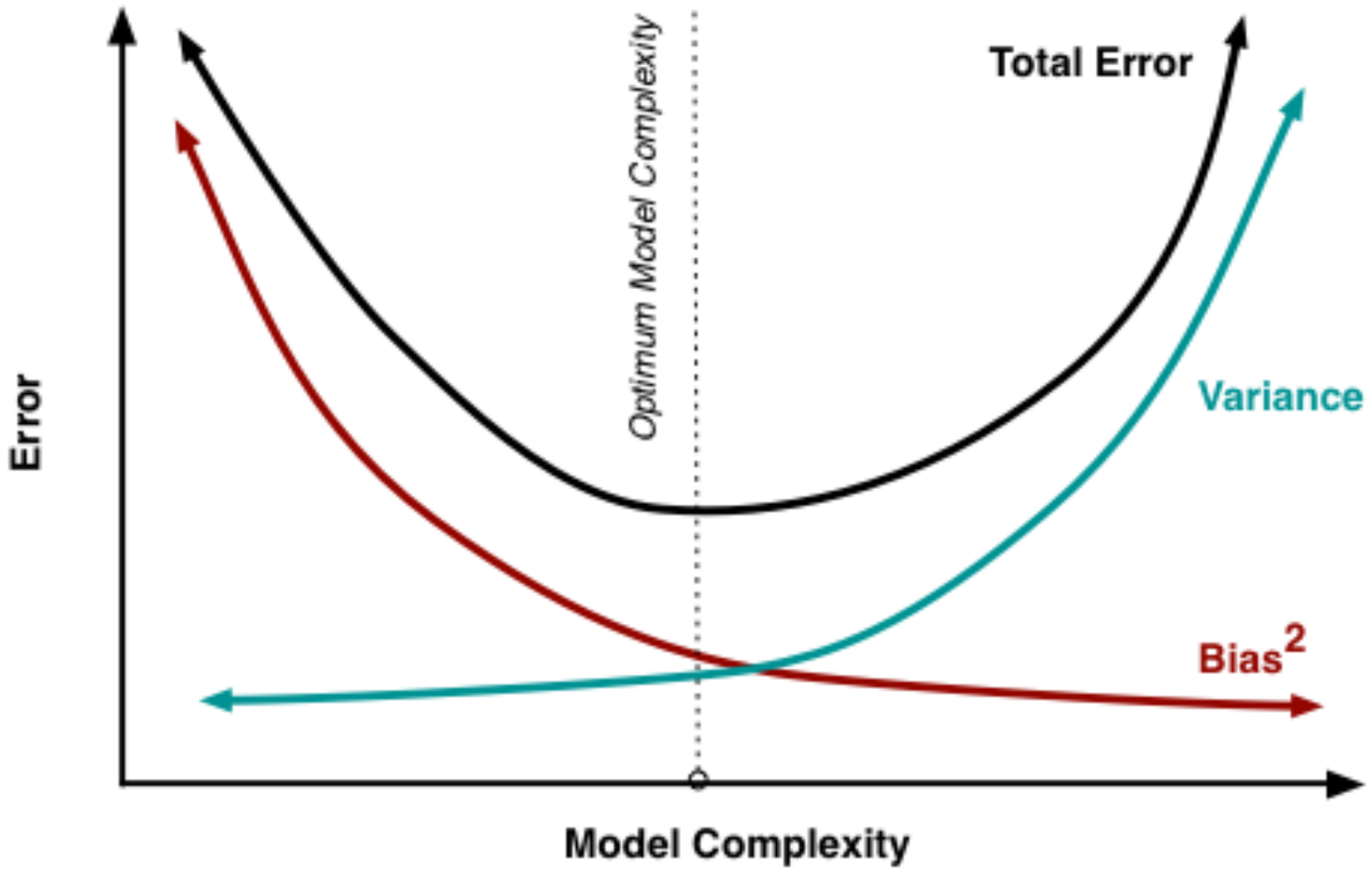


Exercise: Reasoning about Training Error and Expected Error

- Imagine you get 0 training error (say squared error)
- But the irreducible error is sigma^2, the variance of p(y l x)
- This suggests the best error you can get is sigma^2 (i.e., the minimal error, even with zero reducible error)
- What does this tell you about your learned model?

$$\begin{aligned}\mathbb{E}[C] &= \mathbb{E}[(f(\boldsymbol{X}) - Y)^2] \\ &= \underbrace{\mathbb{E}[(f(\boldsymbol{X}) - \mathbb{E}[Y|\boldsymbol{X}])^2]}_{\text{reducible error}} + \underbrace{\mathbb{E}[(\mathbb{E}[Y|\boldsymbol{X}] - Y)^2]}_{\text{irreducible error}}.\end{aligned}$$

Bias-variance (trade-off)



Q1: Why might variance increase with increasing model complexity?
Q2: Is this really a trade-off? Does decreasing variance necessarily imply we have increased bias?

Practice Midterm Question

- Imagine you transform the input observations to higher-order polynomials, before using linear regression. You consider all polynomials of orders k = 1,2,...,100. You are given a training set, with a separate test set. How might you determine which models are overfitting or underfitting?

Practice Midterm Question

Let us assume a setting where the true model is linear, i.e., $Y = w_0 + \sum_{j=1}^{a} w_i X_i + \epsilon$ for weights $w_j \in \mathbb{R}$, random variables X_j and $\epsilon \sim \mathcal{N}(0, 1)$. Imagine you get a dataset with $n = 100$ samples, and train one model with linear regression and one model with cubic regression—linear regression, where you first expand the features into all the polynomial terms in a cubic polynomial. Which model do you think will obtain lower training error—or will they perform the same—and why?

Generalized linear models

- Can pick any natural exponential family distribution for p(y | x)
- If p(y | x) is Gaussian, then we get linear regression with <x, w> approximating E[y | x]
- If p(y | x) is Bernoulli, then we get logistic regression with sigmoid(<x, w>) approximating E[y | x]
- If p(y | x) is Poisson, then we get Poisson regression with exp(<x, w>) approximating E[y | x]
- If p(y | x) is a Multinomial (multiclass), then we get multinomial logistic regression with softmax(<x, w>) approximating E[y | x]

Note: GLMs are not on the midterm, but this summary is useful for thinking about distributions

Exercise: Selecting distributions

- What distributions might you use if
  - we have binary features and targets?
  - binary targets and continuous features?
  - positive targets?
  - categorical features with a large number of categories?
  - multi-class targets, with continuous features?

Exercise: Selecting algorithms

- When might logistic regression do better than linear regression?

- When might Poisson regression do better than linear regression?

- When might Ridge Regression (l2-regularized linear regression) do better than linear regression?

Practice Midterm Question

- Discuss any one form of regularization that is used to train linear regression models. Why is such regularization used?

- Example answer that would get full marks:

- One form of regularization that is used with linear regression is the l2 regularizer, where $l2(\mathbf{w}) = \sum_j w_j^2 = \|\mathbf{w}\|_2^2$. This function of $\mathbf{w}$ is added to the least-squares loss $\|\mathbf{Xw} - \mathbf{y}\|_2^2$, to get the modified minimization
$\min_w \|\mathbf{Xw} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$.
This regularizer helps prevent overfitting, by encoding a preference for weights that are closer to zero—the l2 corresponds to using a zero-mean Gaussian prior on the weights. From an algebraic perspective, it add a non-negligble positive constant λ to the eigenvalues of the matrix $\mathbf{X}^\top\mathbf{X}$, making this matrix better conditioned and so the solution that uses the inverse of this matrix more stable.

Another answer that would get full marks

- Discuss any one form of regularization that is used to train linear regression models. Why is such regularization used?

One form of regularization that is used with linear regression is the l2 regularizer, where $l2(\mathbf{w}) = \sum_j w_j^2 = \|\mathbf{w}\|_2^2$. This regularizer helps prevent overfitting, by encoding a preference for weights that are closer to zero—the l2 corresponds to using a zero-mean Gaussian prior on the weights.

Challenge Exercise

- Imagine you believed p(ylx) is a mixture of two Gaussians
- Recall we can take the convex combination of two Gaussians to represent a more complex bimodal distribution

$$c_1\mathcal{N}(\mu_1, \sigma_1^2) + (1 - c_2)\mathcal{N}(\mu_2, \sigma_2^2)$$

- If you want to estimate p(ylx), how might you parameterize it?
- How might you estimate those parameters?