Day 8 - Statistical Learning Framework

Agenda:

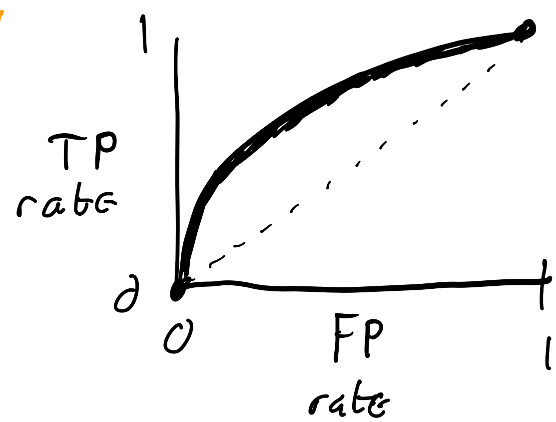
- Statistical learning framework
- Derivation of square loss for regression
- Derivation of log loss / cross-entropy loss for classification
- Terms related to the statistical learning framework
- Bias variance tradeoff

ROC curve

TP rate = recall = sensitivity

= proportion of positive class correctly classified

$$= \frac{TP}{TP + FN}$$



$$\text{FP rate} = \frac{FP}{TN + FP} = 1 - \text{specificity}$$

= proportion of negative class incorrectly classified

$$\text{precision} = \frac{TP}{TP + FP}$$

= proportion of predicted positives that are correct

		Predicted +	Predicted −
True	+	TP	FN
	−	FP	TN

CS 6140: Machine Learning — Fall 2021— Paul Hand

HW 3
Due: Wednesday October 6, 2021 at 2:30 PM Eastern time via Gradescope.

Names: [Put Your Name(s) Here]

You can submit this homework either by yourself or in a group of 2. You may consult any and all resources. Make sure to justify your answers. If you are working alone, you may either write your responses in LaTeX or you may write them by hand and take a photograph of them. If you are working in a group of 2, you must type your responses in LaTeX. You are encouraged to use Overleaf. Create a new project and replace the tex code with the tex file of this document, which you can find on the course website. To share the document with your partner, click Share > Turn on link sharing, and send the link to your partner. When you upload your solutions to Gradescope, make sure to take each problem with the correct page or image.

Question 1. *Linear regression with multivariate responses.*

Consider training data $\{(x^{(i)}, y^{(i)})\}_{i=1...n}$, where $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}^k$. Consider a model $y = Ax$, where $A \in \mathbb{R}^{k\times d}$ is unknown. Estimate A by solving least squares linear regression

$$\min_A \sum_{i=1}^n \|y^{(i)} - Ax^{(i)}\|^2.$$

(a) Find A in the case of training data $\left\{\left(\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix}\right), \left(\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}\right), \left(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\3\\1\end{pmatrix}\right)\right\}$. You may use a computer to perform linear algebra. Hint: the problem can be simplified by observing that each output dimension can be computed separately from the others. If you use this fact, justify it in your response.

Response:

(b) Consider the case of generic training data. Let Y be the $k \times n$ matrix such that $Y_{ji} = y_j^{(i)}$. Let X be the $n \times d$ matrix where $X_{ij} = x_j^{(i)}$. Provide a formula for the least squares estimate of A. Make sure to check that the matrix dimensions match in any matrix products that appear in your answer. Use the same hint as in part (a).

Response:

(c) Show that any prediction under this learned model is a linear combination of the response values $(y^{(1)}, \dots, y^{(n)})$. That is, for the A in part (b), show that $Ax \in \text{span}(y^{(1)}, \dots, y^{(n)})$ for any x. You may assume that X is rank d.

Response:

Question 2. *Logistic Regression*

Consider training data $\{(x_i, y_i)\}_{i=1\ldots n}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. Consider the logistic data model $\hat{y} = \sigma(\theta \cdot x)$, where $x \in \mathbb{R}^d$, $\theta \in \mathbb{R}^d$, and σ is the logistic function $\sigma(z) = e^z/(e^z + 1)$.

(a) Show that $\sigma'(z) = \sigma(z)(1 - \sigma(z))$.

Response:

(b) Let $f(\theta) = \sum_{i=1}^{n} -y_i \log \hat{y}_i - (1 - y_i)\log(1 - \hat{y}_i)$, where $\hat{y}_i = \sigma(\theta \cdot x_i)$. Compute $\nabla f(\theta)$. Use the fact in part (a) to simplify your answer.

Response:

(c) If $M = \sum_{i=1}^{n} x_i x_i^t$, show that $z^t M z \geq 0$ for any $z \in \mathbb{R}^d$.

Response:

(d) Using a summation and vector and/or matrix products, write down a formula for the Hessian, H, of f with respect to θ. Show that $z^t H z \geq 0$ for any $z \in \mathbb{R}^d$.

Response:

Statistical Framework for ML (supervised)

Assume:

- (x, y) are sampled from a joint probability distribution
- Training data $D = \{(x_i, y_i)\}_{i=1 \dots n}$ are iid samples
- Test data are also iid samples

Can estimate the model/predictor by maximum likelihood estimation

Results (usually) in an optimization problem

$$\hat{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

"empirical risk minimization"

where

ℓ ~ loss function eg $\ell(\hat{y}, y) = |\hat{y} - y|^2$

$\mathcal{H}$ - hypothesis class eg degree d polynomial

Evaluate performance on test data $\{(x_i, y_i)\}_{i=1 \dots m}$

$$\frac{1}{m} \sum_{i=1}^{m} \ell(y_i, \hat{h}(x_i))$$

Linear Regression and Square Loss

Let $a \in \mathbb{R}^d$, $x \in \mathbb{R}^d$

Model: $y_i = x_i^t a + \varepsilon_i$ w/ $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$

Data: $D = \{(x_i, y_i)\}_{i=1 \cdots n}$

Estimate a by maximum likelihood

pdf of ε_i is $\frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{z^2}{2\sigma^2}}$ over $z \in \mathbb{R}$

likelihood of data (using $\varepsilon_i = y_i - x_i^t a$)

$$L(a) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma} e^{-(y_i - x_i^t a)^2 / 2\sigma^2}$$

$$\log L(a) = -\sum_{i=1}^{n} \frac{(y_i - x_i^t a)^2}{2\sigma^2} + \text{terms constant in } a$$

maximizing data likelihood $\Leftrightarrow$ minimizing square loss

$$\max_a L(a) \quad \Leftrightarrow \quad \min_a \sum_{i=1}^{n} \underbrace{(x_i^t a - y_i)^2}_{\text{square loss } \ell(\hat{y}, y) = |\hat{y} - y|^2}$$

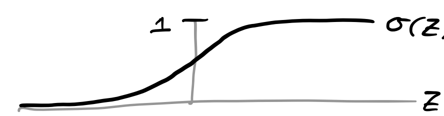
Logistic Regression and Cross Entropy Loss

Model:

Let $a \in \mathbb{R}^d$

$$P(y=1 \mid x) = \sigma(x^t a)$$
$$P(y=0 \mid x) = 1-\sigma(x^t a)$$

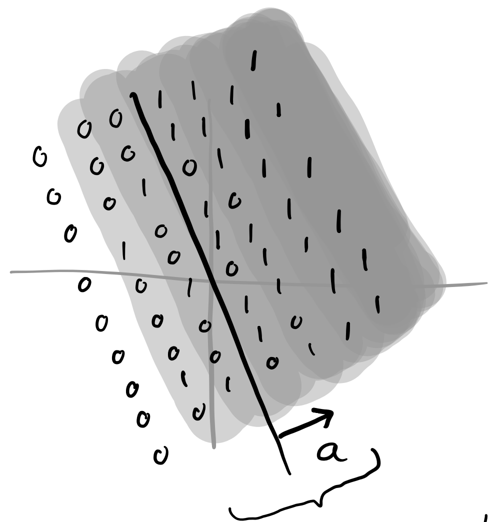
w/ $\sigma(z) = \dfrac{e^z}{e^z+1} = \dfrac{1}{1+e^{-z}}$



$x^t a$ is a <u>logit</u>

Data: $\{(x_i, y_i)\}$

Visually:



width of region of uncertainty $\simeq \dfrac{1}{\|a\|_2}$

Estimate a by maximum likelihood

$$L(a) = \prod_{i=1}^{n} P(y_i=0 \mid x_i)^{1-y_i} \, P(y_i=1 \mid x_i)^{y_i}$$

$$\log L(a) = \sum_{i=1}^{n} (1-y_i) \log P(y_i=0 \mid x_i) + y_i \log P(y_i=1 \mid x_i)$$

Cross entropy loss

$$\underbrace{\ell_{CE}(p, q)}_{\substack{\text{discrete} \\ \text{r.v.s over } \mathcal{Z}}} = -\sum_{z \in \mathcal{Z}} p(z) \log q(z) = -\mathbb{E}_p(\log q)$$

Maximizing data likelihood $\Longleftrightarrow$ minimizing cross entropy loss

$$\max_a L(a) \Longleftrightarrow \min_a -\sum_{i=1}^n \underbrace{\left(y_i \log(\sigma(x_i^t a)) + (1-y_i)\log(1-\sigma(x_i^t a))\right)}_{\ell_{CE}\left(\begin{pmatrix} y_i \\ 1-y_i \end{pmatrix}, \begin{pmatrix} \sigma(x_i^t a) \\ 1-\sigma(x_i^t a) \end{pmatrix}\right)}$$

Formalism for Statistical Framework for ML (supervised)

Domain Set - $\mathcal{X}$ - arbitrary set of objects/instances that could be labelled
- usually represented as a feature vector in $\mathbb{R}^d$
- could be infinite dimensional

Label Set - $\mathcal{Y}$ - set of possible labels

eg • $\mathbb{R}^d$ for regression
• $\{1,0\}$ for binary classification
• Finite set for multiclass classification

Training data - $S = \{(x_i, y_i)\}_{i=1 \cdots n}$

n points in $\mathcal{X} \times \mathcal{Y}$

Predictor/hypothesis - any function $h : \mathcal{X} \to \mathcal{Y}$ that

$x \mapsto y$

outputs a prediction y for any instance x

Hypothesis Class - $\mathcal{H}$ a set of predictors/hypotheses that are being considered

eg $\mathcal{H} = \{\text{degree } d \text{ polynomials}\}$

Data generation model

Simple version

- Assume $x \sim D$, where D is a probability distribution over $\mathcal{X}$
- Each sample is independent
- $y = f(x)$ for a "correct" function f.

Realistic version

- Assume $(x, y) \sim D$, a joint probability distribution over $\mathcal{X} \times \mathcal{Y}$

There is some marginal distribution of $\mathcal{X}$, $D_{\mathcal{X}}$.

For any x, there is a conditional distribution over y $D_{y|x}$

Loss

– how bad is the prediction of an instance relative to its label

$$\underbrace{\ell(y, \hat{y})}_{} \in \mathbb{R}$$

y: label, $\hat{y}$: prediction

Examples

- Square loss $\ell(y,\hat{y}) = \|y - \hat{y}\|^2$ if $y, \hat{y} \in \mathbb{R}^d$

- log loss $\ell(y,\hat{y}) = \sum_{i=1}^{k} y_i \log \hat{y}_i$ if $y \in \mathbb{R}^k$ are one-hot encodings & $\hat{y} \in \mathbb{R}^k$ is a probability dist over k labels

- 0-1 loss $\ell(y,\hat{y}) = \begin{cases} 1 & \text{if } \hat{y} = y \\ 0 & \text{if o'wise} \end{cases}$

Risk - expected loss of a predictor for new data samples

$$R(h) = \mathbb{E}_{(x,y)\sim D}\, \ell(y, h(x))$$

aka "generalization error"
"error"
"population error"
"test error"

Goal of learning:

To find a h such that $R(h)$ is minimal. Want to solve

$$\min_{h \in \mathcal{H}} R_D(h)$$

Challenge: We don't know D. We only have samples S

Empirical Risk Minimization — approximation of risk based on training data S

$$\hat{h} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{n} \ell(y_i, f(x_i))}_{\text{Empirical risk}}$$

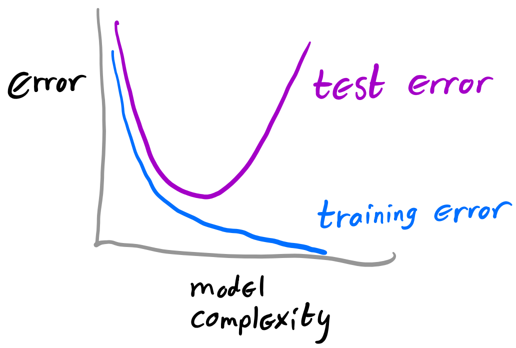
Test Error — Use a finite test set to assess generalization

$$\frac{1}{m} \sum_{i=1}^{m} \ell(y_i^{test}, \hat{h}(x_i^{test}))$$

Bias-Variance Tradeoff

What class of hypotheses should you search over?

Standard Statistical ML story:



higher complexity models have lower bias but higher variance

If complexity is too high, it overfits data, variance term dominates test error

after a certain threshold, "larger models are worse"

Why is training error monotonically decreasing?

Why is test error initially decreasing?

If you have 10^3 data samples, how complex of a data model would you consider?

Why does understanding this tradeoff matter?

Bias - Variance Decomposition

Consider regression model

$$y = f(X) + \varepsilon \qquad \text{w/} \quad \mathbb{E}[\varepsilon \mid X] = 0$$

Let $D = \{(X_i, y_i)\}_{i=1\cdots n}$ be iid samples

Estimate f by an algorithm producing $\hat{f}_D$

Evaluate $\hat{f}_D$ by expected loss on a new sample

$$\underbrace{R(\hat{f}_D)}_{\text{risk}} = \underbrace{\mathbb{E}_{x,y}}_{\text{test sample}} \underbrace{(\hat{f}_D(x) - y)^2}_{\text{square loss}}$$

Performance will vary based on D. Take expectation over D.

$$\mathbb{E}_D R(\hat{f}_D) = \mathbb{E}_{x,y,D} (\hat{f}_D(x) - y)^2$$

We will decompose into 3 effects: bias, variance, irreducible error

$$\mathbb{E}_D R(\hat{f}_D) = \mathbb{E}_{x,y,D}\left[(\hat{f}_D(x) - f(x) - \varepsilon)^2\right]$$

$$= \mathbb{E}_{x,y,D}(\hat{f}_D(x) - f(x))^2 - 2\,\mathbb{E}\left[\cancelto{0}{(\hat{f}_D(x) - f(x))\varepsilon}\right] + \underbrace{\mathbb{E}[\varepsilon^2]}_{Var(\varepsilon)}$$

$$= \mathbb{E}_{x,y,D}(\hat{f}_D(x) - f(x))^2 + Var(\varepsilon)$$

Evaluating the first term, Conditioning on x,

$$\mathbb{E}_D(\hat{f}_D(x) - f(x))^2 = \mathbb{E}_D\left[\left((\hat{f}_D(x) - \mathbb{E}_D \hat{f}_D(x)) + (\mathbb{E}_D \hat{f}_D(x) - f(x))\right)^2\right]$$

$$= \mathbb{E}_D\left(\hat{f}_D(x) - \mathbb{E}_D \hat{f}_D(x)\right)^2 + 2\,\mathbb{E}_D \underbrace{\cancelto{0}{\left(\hat{f}_D(x) - \mathbb{E}_D \hat{f}_D(x)\right)}}_{\text{0 in expectation in } D} \underbrace{\left(\mathbb{E}_D \hat{f}_D(x) - f(x)\right)}_{\text{does not depend on } D} + \mathbb{E}_D\left(\mathbb{E}_D \hat{f}_D(x) - f(x)\right)^2$$

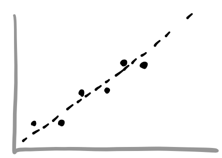
$$= \underbrace{\mathbb{E}_D\left(\hat{f}_D(x) - \mathbb{E}_D \hat{f}_D(x)\right)^2}_{\text{Variance of } \hat{f}_D(x)} + \underbrace{\left(\mathbb{E}_D\left(\hat{f}_D(x) - f(x)\right)\right)^2}_{\text{squared bias}}$$

So,

$$\mathbb{E}_D R(\hat{f}) = \underbrace{\mathbb{E}_x\left(f(x) - \mathbb{E}_D \hat{f}_D(x)\right)^2}_{\text{expected squared bias of estimate}} + \underbrace{\mathbb{E}_x \mathrm{Var}_D \hat{f}_D(x)}_{\text{expected variance of estimate}} + \underbrace{\mathrm{Var}(\varepsilon)}_{\text{irreducible error}}$$

Illustration of bias variance tradeoff

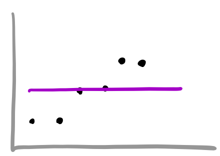
Suppose $y = x + \varepsilon$

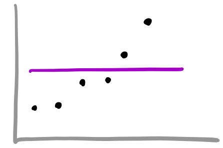


Low complexity model: $y = c$

$\mathbb{E}_x\left(f(x) - \mathbb{E}_D \hat{f}_D\right)^2$ is high

$\mathbb{E}_x \mathrm{Var}_D \hat{f}_D(x)$ is low

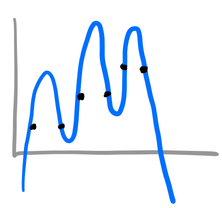


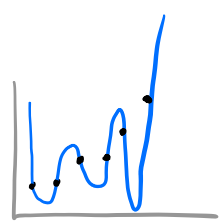


High complexity model: $y = c_0 + c_1 x + c_2 x^2 + \cdots c_6 x^6$

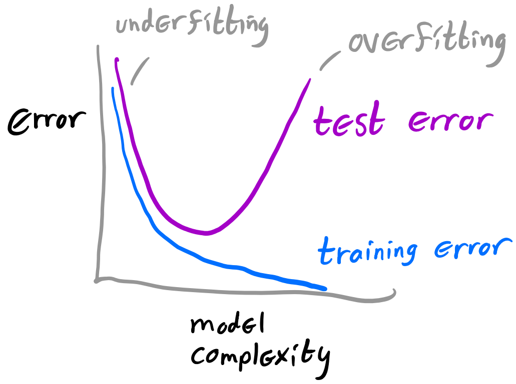
$\mathbb{E}_x\left(f(x) - \mathbb{E}_D \hat{f}_D\right)^2$ is low

$\mathbb{E}_x \mathrm{Var}_D \hat{f}_D(x)$ is high





Standard Statistical ML story:

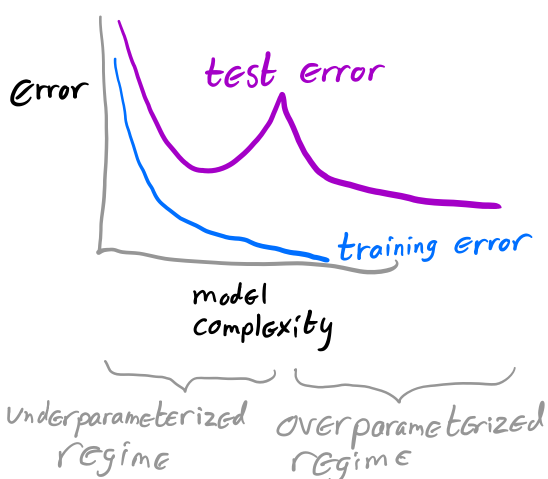


higher complexity models have lower bias but higher variance

If complexity is too high, it overfits data, variance term dominates test error

after a certain threshold, "larger models are worse"

Modern Story based on Neural Nets:



Test error can decrease as model complexity continues increasing.

And it can be lower than in underparameterized regime

Phenomenon: double descent

"larger models are better"

Q: Are larger models better b/c we have so much data that it captures the entire problem domain and is actually overfitting?

If you have 10^3 data samples, how complex of a data model would you consider?

Choose a neural network with 10000 or 100000 parameters

Why is being critically parametrized bad for generalization?

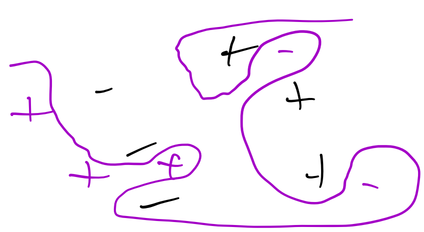
Critically parameterized: #parameters = # data points

How many values of parameters would fit data exactly? 1. Neural net must contort itself to fit the exact data. No expectation for generalization.

In the overparameterized regime, do all models with 0 training error generalize well?

there is an infinity of model parameters that fit data exactly. Gradient descent will find one of them. Would all solutions generalize well?

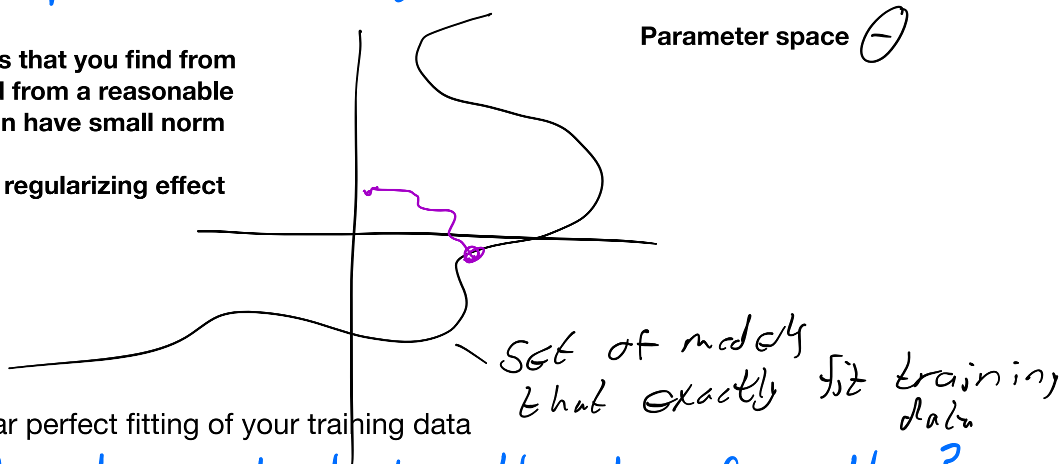
There are solutions that don't generalize well.
Build them by adding poison training data



How is good generalization possible in the overparameterized regime?

Parameters that you find from running Gd from a reasonable initialization have small norm

That has a regularizing effect



Expect near perfect fitting of your training data

Why does understanding this tradeoff matter?