Day 9 - Statistical Learning Framework and Bias Variance Tradeoff

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

Statistical Framework for ML (supervised)

Assume:

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

Can estimate the model/predictor by maximum likelihood estimation

Results (usually) in an optimization problem

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

"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 $\{(x_i, y_i)\}_{i=1 \cdots m}$

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

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 $\frac{1}{\sqrt{2\pi}\,\sigma} e^{-z^2/2\sigma^2}$ over $z \in \mathbb{R}$

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

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

by independence of data

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

maximizing data likelihood $\Leftrightarrow$ minimizing square loss

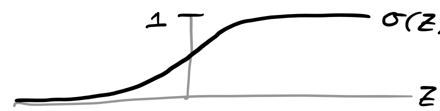
$$\max_a L(a) \iff \min_a \sum_{i=1}^{n} \underbrace{(x_i^t a - y_i)^2}_{\text{square loss } \ell(\hat{y}, y) = |\hat{y} - y|^2}$$

Logistic Regression and Cross Entropy Loss

Model:

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

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



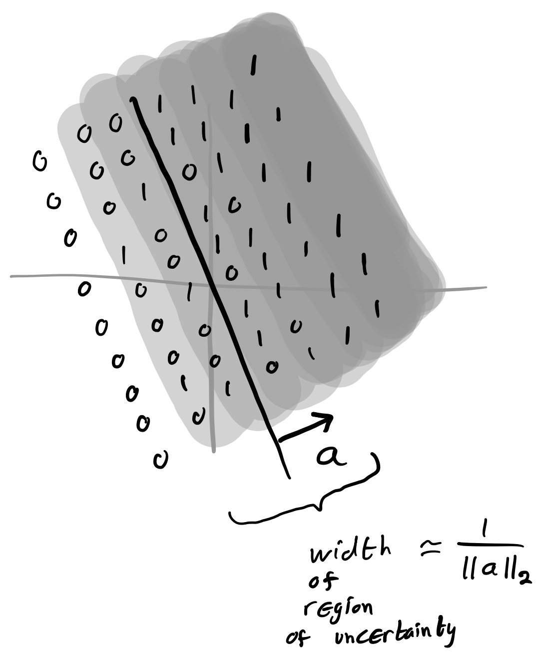
Bernoulli

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

$x^t a$ is a logit

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

Visually:



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

$$\ell_{CE}(p, q) = -\sum_{z \in \mathcal{Z}} p(z) \log q(z) = -\mathbb{E}_p(\log q)$$

discrete r.v.s over $\mathcal{Z}$

$\sum_{z} [\log q(z)] p(z)$

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 $f : \mathcal{X} \to \mathcal{Y}$ 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 $\mathcal{H} = \{$degree d polynomials$\}$

What is the label set for classification with three classes?

$$Y = \{1, 2, 3\} \subset \mathbb{R}$$

or

$$Y = \left\{ \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix} \right\} \subset \mathbb{R}^3$$

one hot encoding

Consider k-dimensional features. When training a binary classifier for logistic regression with no bias term, what is the hypothesis class?

$$P(y=1 \mid x) = \sigma(\theta \cdot x)$$

Let $f_\theta : \mathbb{R}^k \to \mathbb{R}$ or $[0,1]$

$$f_\theta(x) = \sigma(\theta \cdot x)$$

$$\mathcal{H} = \{ f_\theta \mid \theta \in \mathbb{R}^k \} = \bigcup_{\theta \in \mathbb{R}^k} \{ f_\theta \}$$

Is it useful to consider the hypothesis class of ALL functions from X to Y?

When you pick a model (hypothesis class) you are making assumptions on the data. With the set of all functions, one doesn't assume anything about the data?

Could lead to overfitting.

Consider: $f : x_i \mapsto y_i$, all other $x \mapsto 0$ would be in $\mathcal{H}$

Not clear how to optimize over such a class. There's no parameterization of this class.

Want to make regularity assumptions (eg that the relationship is continuous)

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 $D_{y|x}$

In the following example:

(a) Generate training data (x_i, y_i) for $i = 1 \dots 8$ by $x_i \sim \text{Uniform}([0,1])$, and $y_i = f(x_i) + \varepsilon_i$, where $f(x) = 1 + 2x - 2x^2$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\sigma = 0.1$. Plot the training data and the function f.

What is the distribution D_X? Uniform([0,1])

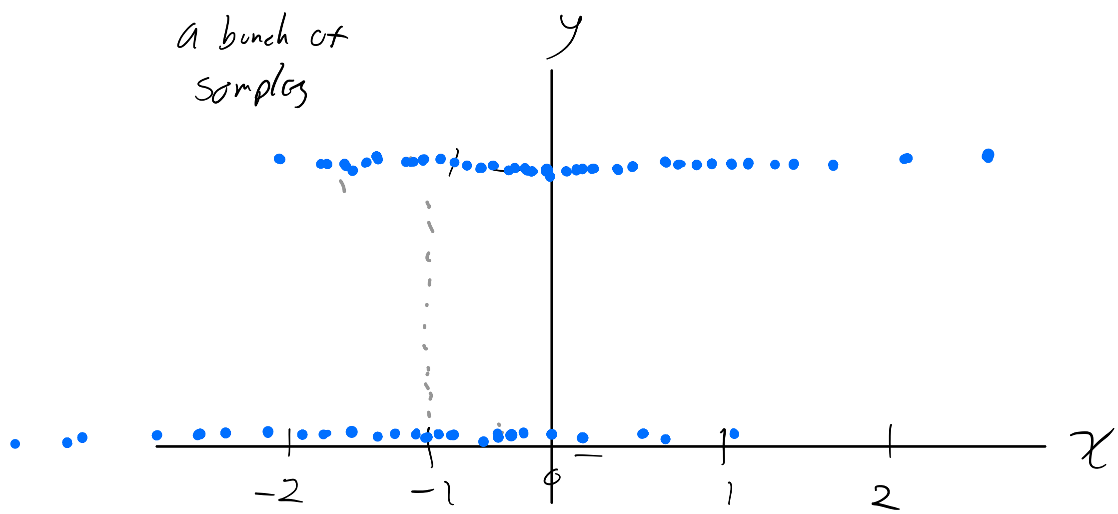
What is the conditional distribution $P_{Y|X}$? $N(1+2x-2x^2, \sigma^2)$

Example: Suppose $X \sim N(0, 1)$

$y|X \sim \text{Bernoulli}[\sigma(X+1)]$

Plot (X, y) according to this distribution

a bunch of samples



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

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

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

Question: Isn't 0-1 loss what I would want to minimize when doing classification?

0-1 loss is not differentiate. Not very useful for training your models. Mostly used for evaluating (particularly for classification)

Risk - expected loss of a predictor for new data samples

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

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

Generalization - ability to perform well on new data

Goal of learning:

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

$$\arg\min_{f \in \mathcal{H}} R_D(f)$$

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

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

empirical mean

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

Test Error — Use a finite test set to assess generalization

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

$$\approx \mathbb{E}_{(x^{test}, y^{test}) \sim \mathcal{D}} \, \ell(y_i^{test}, \hat{f}(x_i^{test}))$$

Model complexity — Cardinality or dimensionality of hypothesis set $\mathcal{H}$

\# unknown parameters

Activity:

Which hypothesis class is more complex?

Let $f_\theta(x) = \theta \cdot x$ for $x \in \mathbb{R}^2$, $\theta \in \mathbb{R}^2$

$$\mathcal{H} = \left\{ f_{\begin{pmatrix}1\\0\end{pmatrix}}, f_{\begin{pmatrix}0\\1\end{pmatrix}} \right\} \quad \text{vs} \quad \mathcal{H} = \left\{ f_{\begin{pmatrix}1\\0\end{pmatrix}}, f_{\begin{pmatrix}0\\1\end{pmatrix}}, f_{\begin{pmatrix}1\\1\end{pmatrix}} \right\}$$

$f(x) = \begin{pmatrix}1\\0\end{pmatrix} \cdot x$

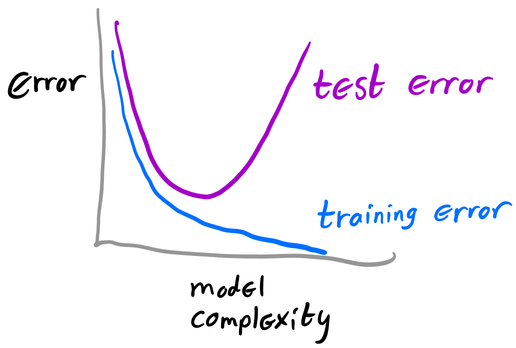
$$\mathcal{H} = \left\{ f_{\begin{pmatrix}1\\0\end{pmatrix}}, f_{\begin{pmatrix}0\\1\end{pmatrix}}, f_{\begin{pmatrix}1\\1\end{pmatrix}} \right\} \quad \text{vs} \quad \left\{ f_\theta \mid \theta \in \mathbb{R}^2 \right\}$$

$\mathcal{H} = \{$ degree 2 polynomials $\}$ vs $\mathcal{H} = \{$ degree 3 polynomials $\}$

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?

Why shouldn't you use test data to estimate model parameters? Wouldn't more data lead to a better model?

Bias - Variance Decomposition

Consider regression model

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

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

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

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

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

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

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

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

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

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

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

Evaluating the first term, Conditioning on x,

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

$$= \mathbb{E}_S (\hat{f}_S(x) - \mathbb{E}_S \hat{f}_S(x))^2 + 2\,\underbrace{\mathbb{E}_S \cancel{(\hat{f}_S(x) - \mathbb{E}_S \hat{f}_S(x))}^{\,0}}_{\text{0 in expectation in } S} \underbrace{(\mathbb{E}_S \hat{f}_S(x) - f(x))}_{\text{does not depend on } S} + \mathbb{E}_S (\mathbb{E}_S \hat{f}_S(x) - f(x))^2$$

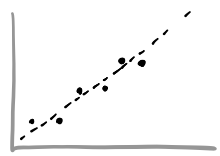
$$= \underbrace{\mathbb{E}_S\left(\hat{f}_S(x) - \mathbb{E}_S \hat{f}_S(x)\right)^2}_{\text{Variance of } \hat{f}_S(x)} + \underbrace{\left(\mathbb{E}_S\left(\hat{f}_S(x) - f(x)\right)\right)^2}_{\text{squared bias}}$$

So,

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

Illustration of bias variance tradeoff

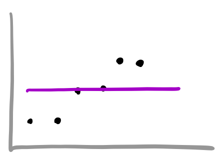
Suppose $y = x + \varepsilon$

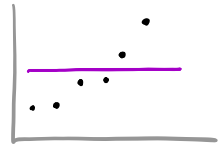


Low complexity model : $y = c$

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

$\mathbb{E}_x \operatorname{Var}_S \hat{f}_S(x)$ is low

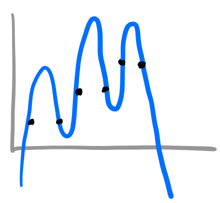


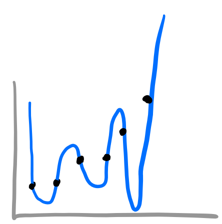


High complexity model : $y = c_0 + c_1 x + c_2 x^2 + \cdots c_6 x^6$

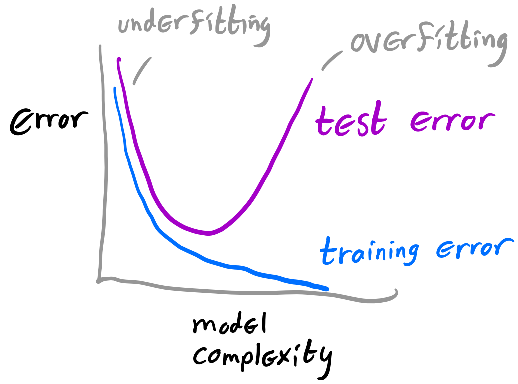
$\mathbb{E}_x (f(x) - \mathbb{E}_S \hat{f}_S)^2$ is low

$\mathbb{E}_x \operatorname{Var}_S \hat{f}_S(x)$ is high





Standard Statistical ML story:

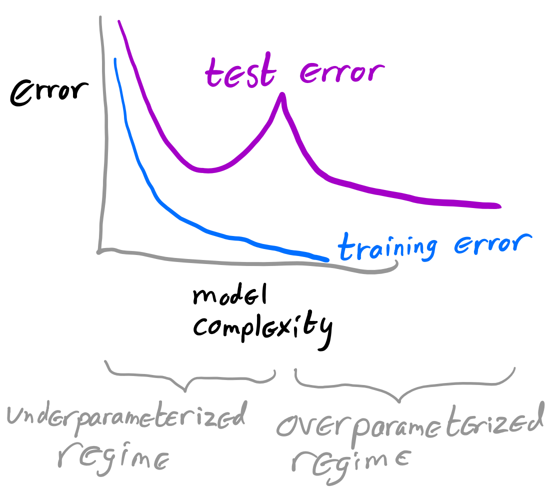


higher complexity models have lower bias but higher variance

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"

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

Why is being critically parametrized bad for generalization?

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

How is good generalization possible in the overparameterized regime?

Why does understanding this tradeoff matter?