

CS480/680: Introduction to Machine Learning

Lecture 3: Linear Regression

Hongyang Zhang



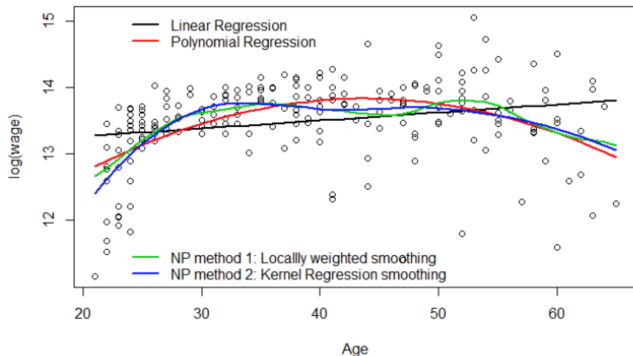
UNIVERSITY OF
WATERLOO

May 17&23, 2023

Regression

- Given training data (\mathbf{x}_i, y_i) , find $f : \mathcal{X} \rightarrow \mathcal{Y}$ such that $f(\mathbf{x}_i) \approx y_i$
 - ▶ $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$: feature vector for the i -th training example
 - ▶ $y_i \in \mathcal{Y} \subseteq \mathbb{R}^t$: t responses, $t = 1$ or even $t = \infty$

Different Types of Regression Lines: Parametric versus nonparametric



The Difficulty

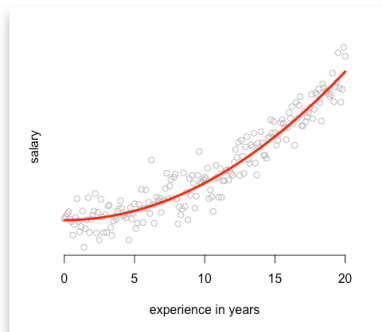
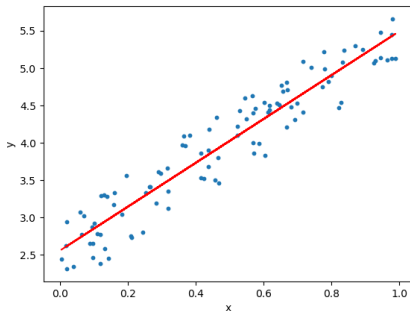
Theorem: Exact interpolation is always possible

For any finite training data $(\mathbf{x}_i, y_i) : i = 1, \dots, n$ such that $\mathbf{x}_i \neq \mathbf{x}_j$ for any i and j , there exist infinitely many functions f such that for all i ,

$$f(\mathbf{x}_i) = y_i.$$

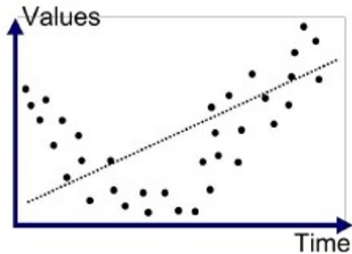
- We cannot decide on a unique f !
- On new data \mathbf{x} , our prediction $\hat{y} = f(\mathbf{x})$ can vary significantly!
- This is where leveraging the prior knowledge of f is important
- “The simplest explanation is usually the correct one”

Prior Knowledge

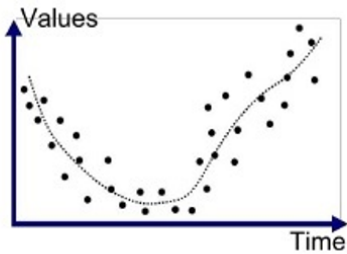


- Prior knowledge on the functional form of f
- Linear vs. nonlinear (e.g., exponential function)

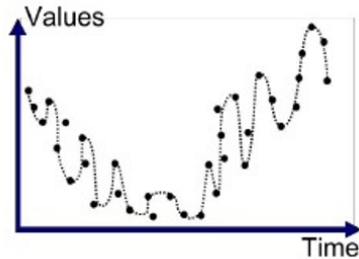
Underfitting, Good Fitting, Overfitting



Underfitted



Good Fit/Robust

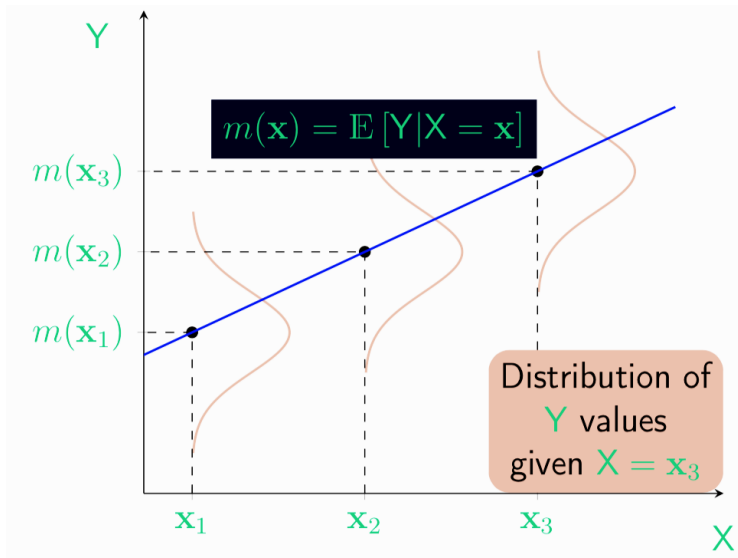


Overfitted

Statistical Learning

- Training and test data are both iid samples from the **same unknown** distribution \mathcal{P}
 - ▶ $(\mathbf{X}_i, Y_i) \sim \mathcal{P}$ and $(\mathbf{X}, Y) \sim \mathcal{P}$
 - ▶ To keep good generalization ability
- Least **squares** regression: $\min_{f: \mathcal{X} \rightarrow \mathcal{Y}} \mathbb{E} \|f(\mathbf{X}) - Y\|_2^2$
 - ▶ Use squared ℓ_2 loss to measure error
 - ▶ Use “square” to make the calculation of the gradient easy
- **Regression function**: $f^*(\mathbf{x}) = m(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$
 - ▶ Regression function is optimal (will show in minutes)
 - ▶ Calculating it needs to know the distribution \mathcal{P} , i.e., all pairs (\mathbf{X}, Y) !
 - ▶ Changing the square loss changes the regression function accordingly

Geometrically



Bias-Variance Decomposition

$$\begin{aligned}\mathbb{E}\|f(\mathbf{X}) - Y\|_2^2 &= \mathbb{E}\|f(\mathbf{X}) - m(\mathbf{X}) + m(\mathbf{X}) - Y\|_2^2 \\ &= \mathbb{E}\|f(\mathbf{X}) - m(\mathbf{X})\|_2^2 + \mathbb{E}\|m(\mathbf{X}) - Y\|_2^2 \\ &\quad + \underbrace{2\mathbb{E}\langle f(\mathbf{X}) - m(\mathbf{X}), m(\mathbf{X}) - Y \rangle}_{=0} \\ &= \mathbb{E}\|f(\mathbf{X}) - m(\mathbf{X})\|_2^2 + \underbrace{\mathbb{E}\|m(\mathbf{X}) - Y\|_2^2}_{\text{noise (variance)}}\end{aligned}$$

- Note that

$$\begin{aligned}\mathbb{E}\langle f(\mathbf{X}) - m(\mathbf{X}), m(\mathbf{X}) - Y \rangle &= \mathbb{E}_{\mathbf{X}}[\mathbb{E}_{Y|\mathbf{X}}\langle f(\mathbf{X}) - m(\mathbf{X}), m(\mathbf{X}) - Y \rangle] \\ &= \mathbb{E}_{\mathbf{X}}\langle f(\mathbf{X}) - m(\mathbf{X}), m(\mathbf{X}) - \mathbb{E}_{Y|\mathbf{X}}[Y] \rangle \\ &= \mathbb{E}_{\mathbf{X}}\langle f(\mathbf{X}) - m(\mathbf{X}), m(\mathbf{X}) - m(\mathbf{X}) \rangle \\ &= 0\end{aligned}$$

Bias-Variance Decomposition — Cont'

$$\mathbb{E}\|f(\mathbf{X}) - \mathbf{Y}\|_2^2 = \mathbb{E}\|f(\mathbf{X}) - m(\mathbf{X})\|_2^2 + \underbrace{\mathbb{E}\|m(\mathbf{X}) - \mathbf{Y}\|_2^2}_{\text{noise (variance)}}$$

- Holds true for any f
- The noise variance is a constant term w.r.t. f !
 - ▶ it is an inherent measure of the difficulty of our problem
- Hence, we aim to choose $f \approx m$ to minimize the squared error
 - ▶ $m(\mathbf{x}) = \mathbb{E}[\mathbf{Y}|\mathbf{X} = \mathbf{x}]$ is our gold rule!
- However, $m(\mathbf{x})$ is unaccessible since we don't know the conditional distribution; learning f from training data D !

Bias-Variance Decomposition — Cont'

Let f_D be the regressor learned on the training dataset D . We have proved:

$$\mathbb{E}_{\mathbf{X}, \mathbf{Y}} \|f_D(\mathbf{X}) - \mathbf{Y}\|_2^2 = \mathbb{E}_{\mathbf{X}} \|f_D(\mathbf{X}) - m(\mathbf{X})\|_2^2 + \underbrace{\mathbb{E}_{\mathbf{X}, \mathbf{Y}} \|m(\mathbf{X}) - \mathbf{Y}\|_2^2}_{\text{noise (variance)}}$$

Define $\bar{f}(\mathbf{X}) = \mathbb{E}_D[f_D(\mathbf{X})]$. Let's continue breaking down the first term in RHS:

$$\begin{aligned}\mathbb{E}_D \mathbb{E}_{\mathbf{X}} \|f_D(\mathbf{X}) - m(\mathbf{X})\|_2^2 &= \mathbb{E}_{D, \mathbf{X}} \|f_D(\mathbf{X}) - \bar{f}(\mathbf{X}) + \bar{f}(\mathbf{X}) - m(\mathbf{X})\|_2^2 \\ &= \mathbb{E}_{\mathbf{X}} \|\bar{f}(\mathbf{X}) - m(\mathbf{X})\|_2^2 + \mathbb{E}_{D, \mathbf{X}} \|f_D(\mathbf{X}) - \bar{f}(\mathbf{X})\|_2^2 \\ &\quad + \underbrace{2\mathbb{E}_{D, \mathbf{X}} \langle \bar{f}(\mathbf{X}) - m(\mathbf{X}), f_D(\mathbf{X}) - \bar{f}(\mathbf{X}) \rangle}_{=0}\end{aligned}$$

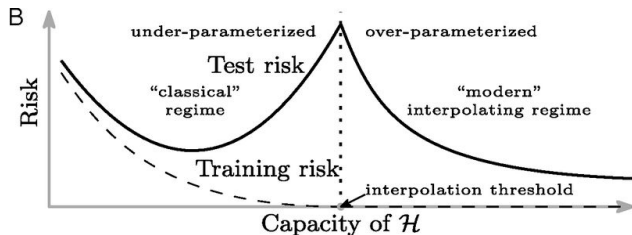
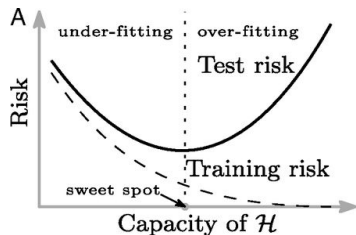
Note that

$$\begin{aligned}\mathbb{E}_{D, \mathbf{X}} \langle \bar{f}(\mathbf{X}) - m(\mathbf{X}), f_D(\mathbf{X}) - \bar{f}(\mathbf{X}) \rangle &= \mathbb{E}_{\mathbf{X}} \mathbb{E}_D \langle \bar{f}(\mathbf{X}) - m(\mathbf{X}), f_D(\mathbf{X}) - \bar{f}(\mathbf{X}) \rangle \\ &= \mathbb{E}_{\mathbf{X}} \langle \bar{f}(\mathbf{X}) - m(\mathbf{X}), \mathbb{E}_D[f_D(\mathbf{X})] - \bar{f}(\mathbf{X}) \rangle \\ &= \mathbb{E}_{\mathbf{X}} \langle \bar{f}(\mathbf{X}) - m(\mathbf{X}), \bar{f}(\mathbf{X}) - \bar{f}(\mathbf{X}) \rangle = 0\end{aligned}$$

Bias-Variance Trade-off

$$\underbrace{\mathbb{E}_{D, \mathbf{X}, Y} \|f_D(\mathbf{X}) - Y\|_2^2}_{\text{test error}}$$
$$= \underbrace{\mathbb{E}_{\mathbf{X}} \|\mathbb{E}_D[f_D(\mathbf{X})] - m(\mathbf{X})\|_2^2}_{\text{bias}^2} + \underbrace{\mathbb{E}_{D, \mathbf{X}} \|f_D(\mathbf{X}) - \mathbb{E}_D[f_D(\mathbf{X})]\|_2^2}_{\text{variance}} + \underbrace{\mathbb{E}_{\mathbf{X}, Y} \|m(\mathbf{X}) - Y\|_2^2}_{\text{noise (variance)}}$$

- Bias² ↓, as the model capacity ↑ (model is more expressively powerful)
- Var ↑, as the model capacity ↑ (prediction of model is less stable)



Sampling \rightarrow Training

$$\min_{f:\mathcal{X}\rightarrow\mathcal{Y}} \hat{\mathbb{E}}\|f(\mathbf{X}) - \mathbf{Y}\|_2^2 := \frac{1}{n} \sum_{i=1}^n \|f(\mathbf{X}_i) - \mathbf{Y}_i\|_2^2$$

- Replace expectation with sample average: $(\mathbf{X}_i, \mathbf{Y}_i) \sim P$
- (Uniform) law of large numbers: as training data size $n \rightarrow \infty$,

$$\hat{\mathbb{E}} \rightarrow \mathbb{E} \text{ and (hopefully) } \operatorname{argmin} \hat{\mathbb{E}} \rightarrow \operatorname{argmin} \mathbb{E}$$

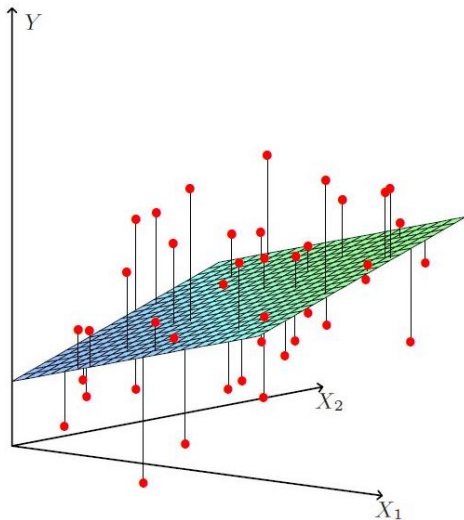
Let's look at the linear function f

Linear Regression

- **Affine function:** $f(\mathbf{x}) = W\mathbf{x} + \mathbf{b}$ with $W \in \mathbb{R}^{t \times d}$ and $\mathbf{b} \in \mathbb{R}^t$
- **Padding:** $\mathbf{x} \leftarrow \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$, $W \leftarrow [W, \mathbf{b}]$, hence $f(\mathbf{x}) = W\mathbf{x}$
- In matrix form: $\frac{1}{n} \sum_i \|f(\mathbf{x}_i) - y_i\|_2^2 = \frac{1}{n} \|WX - Y\|_F^2$
 - ▶ $X = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{(d+1) \times n}$, $Y = [y_1, \dots, y_n] \in \mathbb{R}^{t \times n}$
 - ▶ $\|A\|_F = \sqrt{\sum_{ij} a_{ij}^2}$, where a_{ij} is the element on the i -th row, j -th column of A

$$\min_{W \in \mathbb{R}^{t \times (d+1)}} \frac{1}{n} \|WX - Y\|_F^2$$

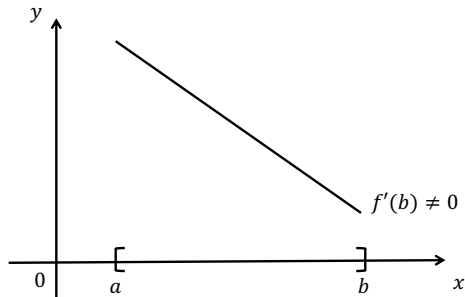
Geometrically



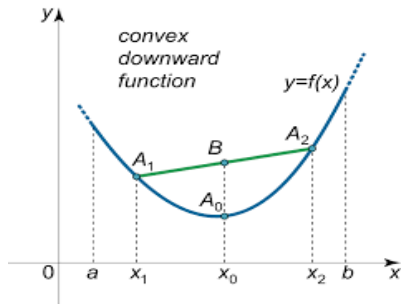
Optimality Condition

Theorem: Fermat's necessary condition for optimality

If \mathbf{w} is a minimizer (or maximizer) of a differentiable function f over an open set, then $f'(\mathbf{w}) = 0$.



Closed Set



Open Set

Training: Solving Linear Regression

$$\text{Loss}(W) = \frac{1}{n} \|WX - Y\|_F^2,$$

- Derivative $\nabla_W \text{Loss}(W) = \frac{2}{n} (WX - Y)X^\top$
 - ▶ Analogous to using chain rule to compute the gradient of $\text{Loss}(w) = \frac{1}{n} (wx - y)^2$
 - ▶ $\nabla_w \text{Loss}(w) = \frac{2}{n} (wx - y)x$
 - ▶ **Not** the focus of this course; check your (matrix) calculus textbook
- Setting derivative to zero:

Normal equation $\boxed{WXX^\top = YX^\top} \implies W = YX^\top (XX^\top)^{-1}$

- $X \in \mathbb{R}^{(d+1) \times n}$ hence $XX^\top \in \mathbb{R}^{(d+1) \times (d+1)}$ (let's assume XX^\top is invertible now)

Prediction

- Once solved W on the training set (X, Y) , can **predict on unseen data** X_{test} :

$$\hat{Y}_{\text{test}} = WX_{\text{test}}$$

- We may evaluate our **test error** if true labels were available:

$$\frac{1}{n_{\text{test}}} \|Y_{\text{test}} - \hat{Y}_{\text{test}}\|_{\text{F}}^2$$

- We may compare to the **training error**:

$$\frac{1}{n} \|Y - \hat{Y}\|_{\text{F}}^2, \hat{Y} := WX$$

- Minimizing the training error as a means to reduce the test error

Ill-conditioning

$$X = \begin{bmatrix} 0 & \epsilon \\ 1 & 1 \end{bmatrix}, \quad y = [1 \quad -1]$$

- Solving linear least squares regression:

$$\mathbf{w} = yX^T(XX^T)^{-1} = [-2/\epsilon \quad 1]$$

- Slight perturbation leads to chaotic behaviour!
- Happens whenever X is ill-conditioned, i.e., (close to) rank-deficient
 - ▶ rank-deficient $X \Rightarrow$ two columns in X are linearly dependent (or simply the same)
 - \Rightarrow but the corresponding y 's might be different
 - \Rightarrow a contradiction and lead to an unstable \mathbf{w}

Ridge Regression

$$\min_W \frac{1}{n} \|WX - Y\|_F^2 + \boxed{\lambda \|W\|_F^2}$$

- Normal equation: $W(\mathbf{X}\mathbf{X}^\top + n\lambda I) = \mathbf{Y}\mathbf{X}^\top$
- $\mathbf{X}\mathbf{X}^\top + n\lambda I$ is far from rank-deficient matrices for large λ . **Proof (optional):**
 - ▶ Consider SVD of $\mathbf{X} = U\Sigma V^\top \Rightarrow \mathbf{X}\mathbf{X}^\top = U\Sigma^2 U^\top \Rightarrow \mathbf{X}\mathbf{X}^\top + n\lambda I = U(\Sigma^2 + n\lambda I)U^\top$
 - ▶ $\Sigma^2 + n\lambda I$ is a diagonal matrix with **strictly positive** diagonal elements
 - ▶ Thus, $\mathbf{X}\mathbf{X}^\top + n\lambda I$ has **no** zero singular value, i.e., it is of full-rank
 - ▶ If you are not familiar with SVD, check your linear algebra textbook
- Regularization parameter. λ controls trade-off
 - ▶ $\lambda = 0$ reduces to ordinary linear regression
 - ▶ $\lambda = \infty$ reduces to $W \equiv \mathbf{0}$

Data Augmentation

$$\frac{1}{n} \|WX - Y\|_F^2 + \boxed{\lambda \|W\|_F^2} = \frac{1}{n} \|W \underbrace{\begin{bmatrix} X & \sqrt{n\lambda}I \end{bmatrix}}_{\tilde{X}} - \underbrace{\begin{bmatrix} Y & \mathbf{0} \end{bmatrix}}_{\tilde{Y}}\|_F^2$$

- Augment X with $\sqrt{n\lambda}I$, i.e., p data points $\mathbf{x}_j = \sqrt{n\lambda}\mathbf{e}_j, j = 1, \dots, p$
- Augment Y with zero

regularization = data augmentation

Questions

?

?

Answers

?