CS480/680: Introduction to Machine Learning

Lecture 3: Linear Regression

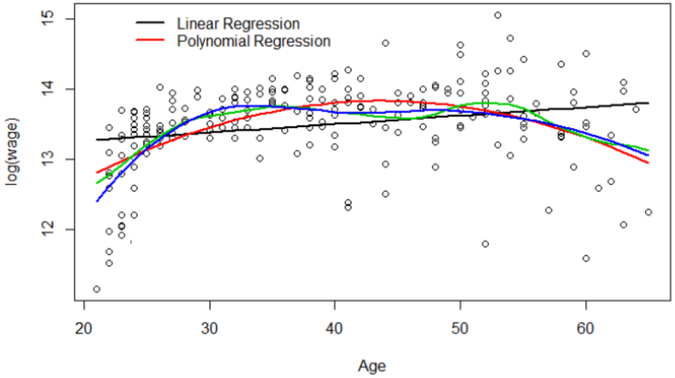
Hongyang Zhang

UNIVERSITY OF WATERLOO

Jan 18&23, 2024

Regression

- Given training data $(\mathbf{x}_i, \mathrm{y}_i)$, find $f : \mathcal{X} \to \mathcal{Y}$ 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
  - $\mathrm{y}_i \in \mathcal{Y} \subseteq \mathbb{R}^t$: t responses, $t = 1$ or even $t = \infty$



The Difficulty

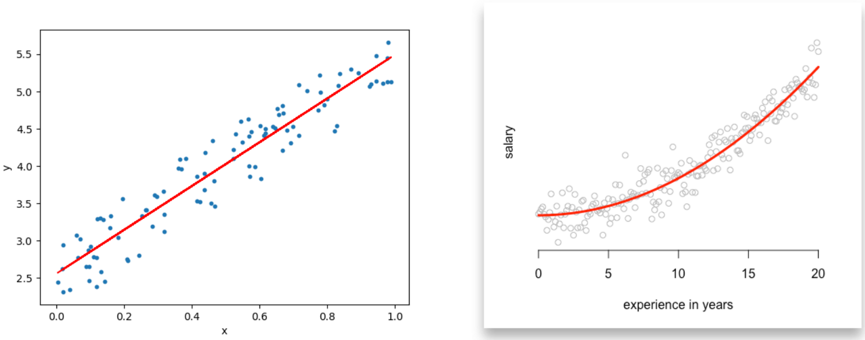
Theorem: Exact interpolation is always possible

For any finite training data $(\mathbf{x}_i, \mathsf{y}_i) : i = 1, \ldots, n$ 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,

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

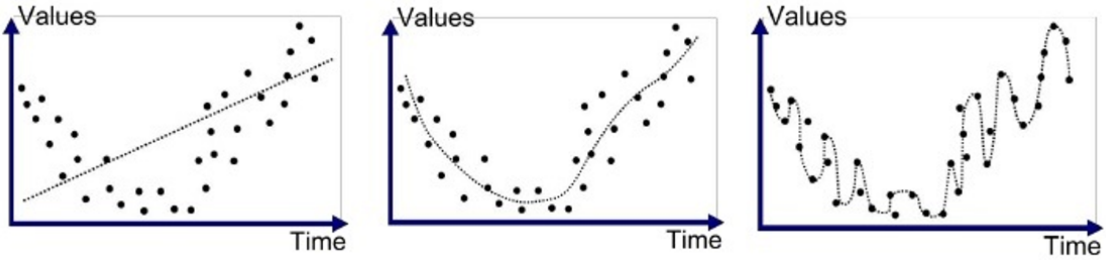
- We cannot decide on a unique f!
- On new data $\mathbf{x}$, our prediction $\hat{\mathsf{y}} = f(\mathbf{x})$ can vary significantly!
- This is where leveraging the prior knowledge of f is important
- "The simplest explanation is usually the correct one"

Prior Knowledge



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

Underfitting, Good Fitting, Overfitting



Underfitted

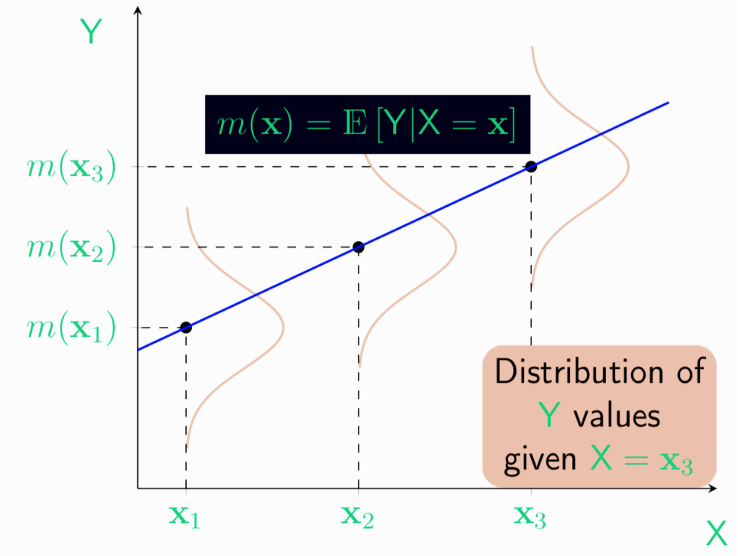
Good Fit/Robust

Overfitted

Statistical Learning

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

Geometrically



Bias-Variance Decomposition

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

- Note that

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

Bias-Variance Decomposition — Cont'

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

- 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
  - $m(\mathbf{x}) = \mathbb{E}[\mathsf{Y}|\mathbf{X} = \mathbf{x}]$ 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:

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

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

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

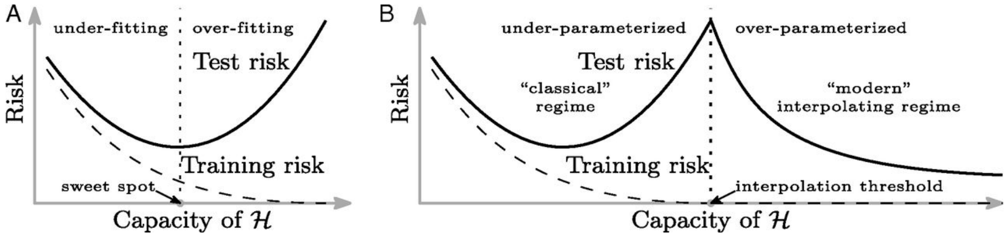
Note that

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

Bias-Variance Trade-off

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

- Bias^2 $\downarrow$, as the model capacity $\uparrow$ (model is more expressively powerful)
- Var $\uparrow$, as the model capacity $\uparrow$ (prediction of model is less stable)



M. Belkin et al. (2019). "Reconciling modern machine-learning practice and the classical bias–variance trade-off". Proceedings of the National Academy of Sciences, vol. 116, no. 32, pp. 15849–15854.

Sampling → Training

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

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

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

Let's look at the linear function f

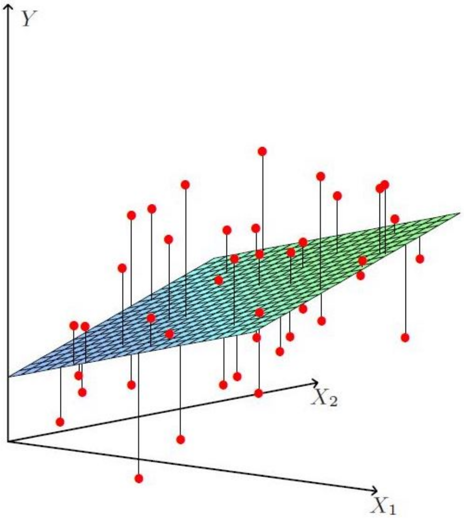
Linear Regression

- 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: $\mathbf{x} \leftarrow \binom{\mathbf{x}}{1}$, $\mathsf{W} \leftarrow [W, \mathbf{b}]$, hence $f(\mathbf{x}) = \mathsf{W}\mathbf{x}$
- In matrix form: $\frac{1}{n}\sum_i \|f(\mathbf{x}_i) - \mathsf{y}_i\|_2^2 \quad = \quad \frac{1}{n}\|\mathsf{WX} - \mathsf{Y}\|_{\mathrm{F}}^2$
  - $\mathsf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{(d+1)\times n}$, $\mathsf{Y} = [\mathsf{y}_1, \ldots, \mathsf{y}_n] \in \mathbb{R}^{t \times n}$
  - $\|A\|_{\mathrm{F}} = \sqrt{\sum_{ij} a_{ij}^2}$, where a_{ij} is the element on the i-th row, j-th column of A

$$\boxed{\min_{\mathsf{W} \in \mathbb{R}^{t \times (d+1)}} \quad \frac{1}{n}\|\mathsf{WX} - \mathsf{Y}\|_{\mathrm{F}}^2}$$

S. M. Stigler (1981). "Gauss and the Invention of Least Squares". The Annals of Statistics, vol. 9, no. 3, pp. 465–474.

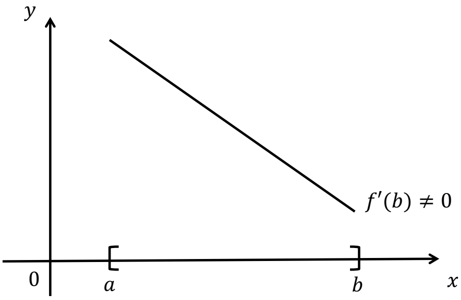
Geometrically



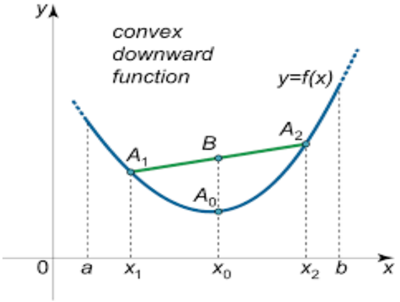
Optimality Condition

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 $f'(\mathbf{w}) = \mathbf{0}$.



Closed Set



Open Set

Training: Solving Linear Regression

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

- Derivative $\nabla_{\text{W}}\text{Loss}(\text{W}) = \frac{2}{n}(\text{WX} - \text{Y})\text{X}^\top$
  - 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:

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

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

Prediction

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

$$\hat{\mathsf{Y}}_{\text{test}} = \mathsf{W}\mathsf{X}_{\text{test}}$$

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

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

- We may compare to the training error:

$$\frac{1}{n}\|\mathsf{Y} - \hat{\mathsf{Y}}\|_{\mathrm{F}}^2, \hat{\mathsf{Y}} := \mathsf{W}\mathsf{X}$$

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

Ill-conditioning

$$\mathsf{X} = \begin{bmatrix} 0 & \epsilon \\ 1 & 1 \end{bmatrix}, \qquad \mathsf{y} = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

- Solving linear least squares regression:

$$\mathbf{w} = \mathsf{y}\mathsf{X}^\top(\mathsf{X}\mathsf{X}^\top)^{-1} = \begin{bmatrix} -2/\epsilon & 1 \end{bmatrix}$$

- 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

$$\min_{\mathsf{W}} \frac{1}{n}\|\mathsf{WX}-\mathsf{Y}\|_{\mathrm{F}}^2 + \boxed{\lambda\|\mathsf{W}\|_{\mathrm{F}}^2}$$

- Normal equation: $\mathsf{W}(\mathsf{XX}^\top + n\lambda I) = \mathsf{YX}^\top$
- $\mathsf{XX}^\top + n\lambda I$ is far from rank-deficient matrices for large λ. Proof (optional):
  - Consider SVD of $\mathsf{X} = U\Sigma V^\top \Rightarrow \mathsf{XX}^\top = U\Sigma^2 U^\top \Rightarrow \mathsf{XX}^\top + n\lambda I = U(\Sigma^2 + n\lambda I)U^\top$
  - $\Sigma^2 + n\lambda I$ is a diagonal matrix with strictly positive diagonal elements
  - Thus, $\mathsf{XX}^\top + n\lambda I$ 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 $\mathsf{W} \equiv \mathbf{0}$

Data Augmentation

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

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

$$\boxed{\text{regularization} = \text{data augmentation}}$$


Questions
Answers