

# EECS 545: Machine Learning

## Lecture 3. Linear Regression (part 2)

Honglak Lee

1/15/2025

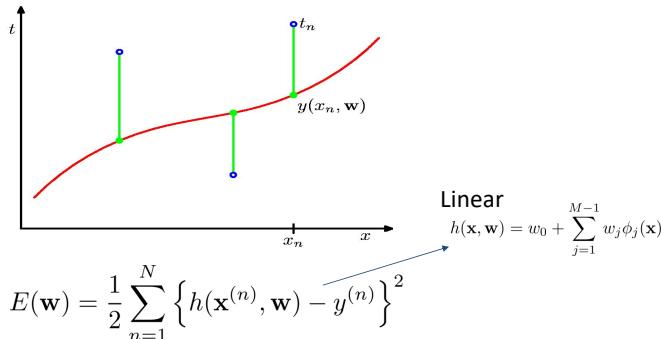


## Outline

- Linear regression review
- Regularized linear regression
- Review on probability
- Maximum likelihood interpretation of linear regression
- Locally-weighted linear regression

2

### Regression, sum of square error



We want to find  $\mathbf{w}$  that minimizes  $E(\mathbf{w})$  over the training data.

3

### Least squares problem

#### • Objective function

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \left( \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}^{(n)}) - y^{(n)} \right)^2$$

M: dimension of feature, N: number of data  
 $\phi_j(\mathbf{x}^{(n)})$ : j-th feature of data

#### • Two ways to find $\mathbf{w}$ that minimizes $E(\mathbf{w})$

1. Gradient descent
2. Closed-form solution

4

### Least squares problem

#### • Objective function

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \left( \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}^{(n)}) - y^{(n)} \right)^2$$

#### 1. Gradient Descent

$$\frac{\partial E(\mathbf{w})}{\partial w_k} = \sum_{n=1}^N \left( \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}^{(n)}) - y^{(n)} \right) \phi_k(\mathbf{x}^{(n)})$$

$$\nabla_{\mathbf{w}} E(\mathbf{w}) = \sum_{n=1}^N \left( \mathbf{w}^T \phi(\mathbf{x}^{(n)}) - y^{(n)} \right) \phi(\mathbf{x}^{(n)})$$

$$\mathbf{w} := \mathbf{w} - \eta \nabla_{\mathbf{w}} E(\mathbf{w})$$

#### 2. Closed-form solution

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \left( \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}^{(n)}) - y^{(n)} \right)^2$$

$$= \frac{1}{2} \mathbf{w}^T \Phi^T \Phi \mathbf{w} - \mathbf{w}^T \Phi^T \mathbf{y} + \frac{1}{2} \mathbf{y}^T \mathbf{y}$$

**Recap**  $\mathbf{w}$ : M by 1     $\mathbf{w} = [w_0, w_1, \dots, w_{M-1}]^T$

$\mathbf{y}$ : N by 1     $\mathbf{y} = [y^{(1)}, y^{(2)}, \dots, y^{(N)}]^T$

$$\Phi: N \text{ by } M$$

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}^{(1)}) & \phi_1(\mathbf{x}^{(1)}) & \dots & \phi_{M-1}(\mathbf{x}^{(1)}) \\ \phi_0(\mathbf{x}^{(2)}) & \phi_1(\mathbf{x}^{(2)}) & \dots & \phi_{M-1}(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}^{(N)}) & \phi_1(\mathbf{x}^{(N)}) & \dots & \phi_{M-1}(\mathbf{x}^{(N)}) \end{pmatrix}$$

5

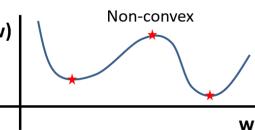
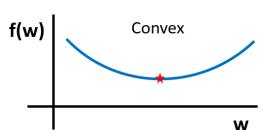
6

### Useful trick: Matrix Calculus

- Compute gradient and set gradient to 0 (condition for optimal solution)

Note: Least squared is a convex problem.

$$\nabla_{\mathbf{w}} E(\mathbf{w}) = 0$$



- Need to compute the first derivative in matrix form

### Gradient via matrix calculus

- Compute gradient and set to zero

$$\nabla_{\mathbf{w}} E(\mathbf{w}) = \nabla_{\mathbf{w}} \left( \frac{1}{2} \mathbf{w}^T \Phi^T \Phi \mathbf{w} - \mathbf{w}^T \Phi^T \mathbf{y} + \frac{1}{2} \mathbf{y}^T \mathbf{y} \right)$$

$$= \Phi^T \Phi \mathbf{w} - \Phi^T \mathbf{y}$$

$$= \mathbf{0}$$

- Solve the resulting equation (normal equation)

$$\Phi^T \Phi \mathbf{w} = \Phi^T \mathbf{y}$$

$$\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

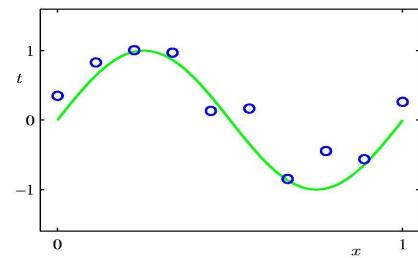
This is the *Moore-Penrose pseudo-inverse*:  $\Phi^\dagger = (\Phi^T \Phi)^{-1} \Phi^T$   
 applied to:  $\Phi \mathbf{w} \approx \mathbf{y}$

7

8

## Back to curve-fitting examples

## Polynomial Curve Fitting

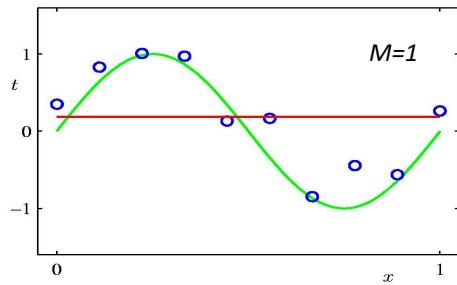


$$h(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_{M-1} x^{M-1} = \sum_{j=0}^{M-1} w_j x^j$$

9

10

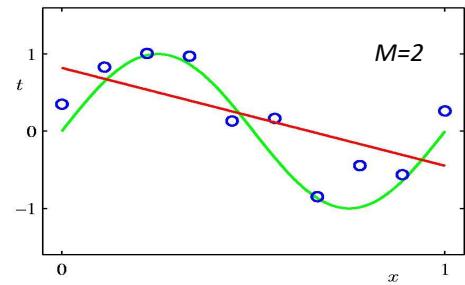
### 0<sup>th</sup> Order Polynomial



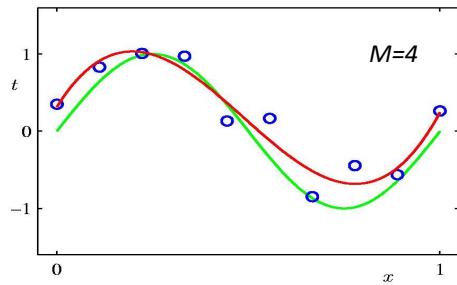
11

12

### 1<sup>st</sup> Order Polynomial



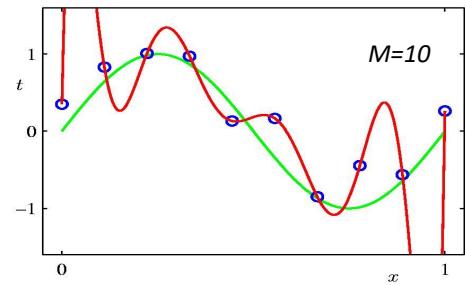
### 3<sup>rd</sup> Order Polynomial



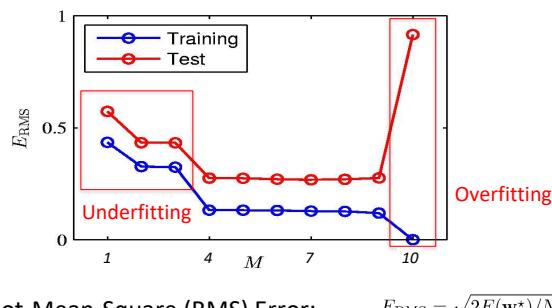
13

14

### 9<sup>th</sup> Order Polynomial



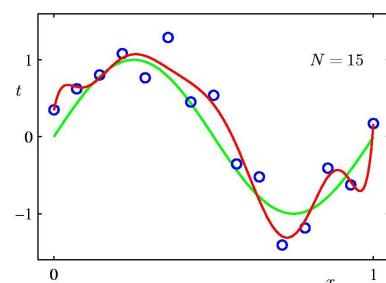
## Over-fitting



Root-Mean-Square (RMS) Error:  $E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$

Q: How do we resolve the over-fitting problem?

### 9<sup>th</sup> Order Polynomial

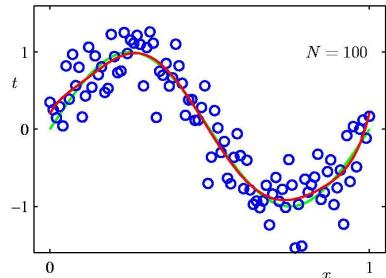


15

16

Data Set Size:  $N = 100$

9<sup>th</sup> Order Polynomial



Increasing data set size can help

Q. How do we choose the degree of polynomial?

17

18

## Rule of thumb

- If you have a small number of data, then use low order polynomial (small number of features).
  - Otherwise, your model will overfit
- As you obtain more data, you can gradually increase the order of the polynomial (more features).
  - However, your model is still limited by the finite amount of the data available (i.e., the optimal model for finite data cannot be infinite dimensional polynomial).
- Controlling model complexity: **regularization**

19

20

## Regularized Linear Regression

### Back to Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^*$	0.19	0.82	0.31	0.35
$w_1^*$		-1.27	7.99	232.37
$w_2^*$			-25.43	-5321.83
$w_3^*$			17.37	48568.31
$w_4^*$				-231639.30
$w_5^*$				640042.26
$w_6^*$				-1061800.52
$w_7^*$				1042400.18
$w_8^*$				-557682.99
$w_9^*$				125201.43

Underfitting      Good      Overfitting; Coefficients are large!

$h(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_{M-1}x^{M-1}$

### Regularized Least Squares (1)

- Consider the error function:
 
$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

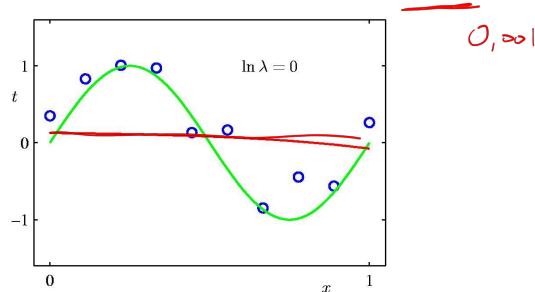
Data term + Regularization term
  - With the sum-of-squares error function and a quadratic regularizer, we get
 
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

New objective function
  - Effect of  $\lambda$
- $\lambda$  is called the regularization coefficient.
- $\sum_j w_j^2 = \frac{1}{2} \|\mathbf{w}\|_2^2$
- Penalizes large coefficient values
- Definition (L2):  $\|\mathbf{w}\|_2^2 = \sum_{j=0}^{M-1} w_j^2$

21

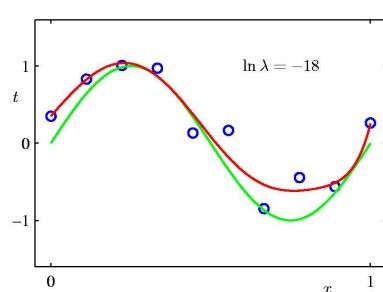
22

L2 Regularization:  $\ln \lambda = 0$



$$M = 9 \quad \tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

L2 Regularization:  $\ln \lambda = -18$

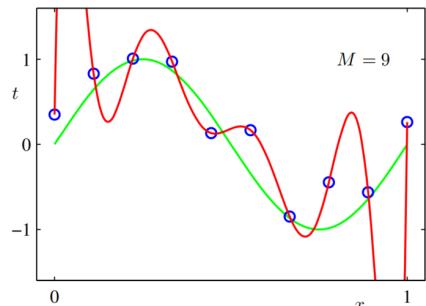


$$M = 9 \quad \tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

23

24

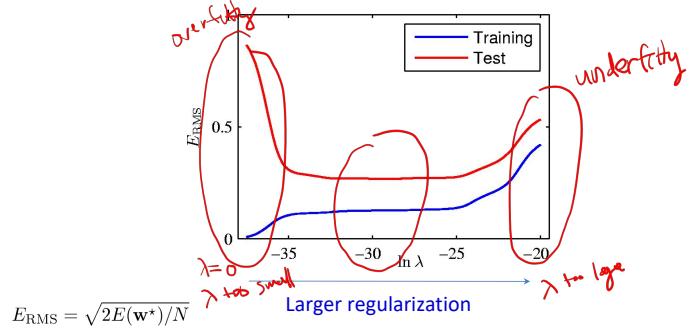
**"No" L2 Regularization:**  $\lambda = 0$   
 (or when L2 regularization is too small) (or  $\ln \lambda \rightarrow -\infty$ )



$$M = 9 \quad \tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

25

**L2 Regularization:**  $E_{\text{RMS}}$  vs.  $\ln \lambda$



NOTE: For simplicity of presentation, we divided the data into training set and test set. However, it's not legitimate to find the optimal hyperparameter based on the test set. We will talk about legitimate ways of doing this when we cover model selection and cross-validation.

26

## Polynomial Coefficients

	(i.e., $\lambda = 0$ )	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^*$	0.35	0.35	0.13	
$w_1^*$	232.37	4.74	-0.05	
$w_2^*$	-5321.83	-0.77	-0.06	
$w_3^*$	48568.31	-31.97	-0.05	
$w_4^*$	-231639.30	-3.89	-0.03	
$w_5^*$	640042.26	55.28	-0.02	
$w_6^*$	-1061800.52	41.32	-0.01	
$w_7^*$	1042400.18	-45.95	-0.00	
$w_8^*$	-557682.99	-91.53	0.00	
$w_9^*$	125201.43	72.68	0.01	
Overfitting; Coefficients are large!	Good	Underfitting		

27

## Regularized Least Squares (1)

- Consider the error function:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

$\lambda$  is called the regularization coefficient.

- With the sum-of-squares error function and a quadratic regularizer, we get

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2 \quad \lambda \cdot \mathbf{w}^\top \mathbf{w}$$

- Closed-form solution:

$$\mathbf{w}_{\text{ML}} = (\lambda \mathbf{I} + \Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}$$

28

## Derivation

Objective function

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

$$= \frac{1}{2} \mathbf{w}^\top \Phi^\top \Phi \mathbf{w} - \mathbf{w}^\top \Phi^\top \mathbf{y} + \frac{1}{2} \mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w}$$

Compute the gradient and set it zero:

$$\nabla_{\mathbf{w}} \tilde{E}(\mathbf{w}) = \nabla_{\mathbf{w}} \left[ \frac{1}{2} \mathbf{w}^\top \Phi^\top \Phi \mathbf{w} - \mathbf{w}^\top \Phi^\top \mathbf{y} + \frac{1}{2} \mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \right]$$

$$= \Phi^\top \Phi \mathbf{w} - \Phi^\top \mathbf{y} + \lambda \mathbf{w}$$

$$= (\Phi^\top \Phi + \lambda \mathbf{I}) \mathbf{w} = \Phi^\top \mathbf{y}$$

$$\mathbf{w}_{\text{ML}} = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}$$

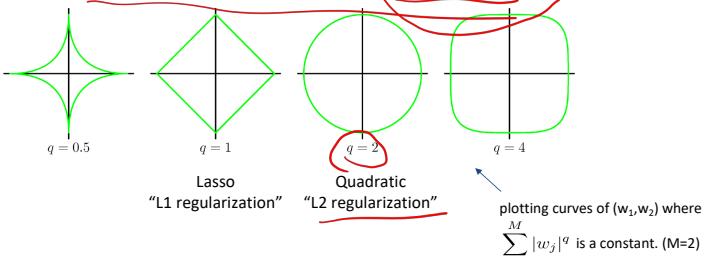
v.s. Ordinary Least Square

Therefore, we get:  $\mathbf{w}_{\text{ML}} = (\lambda \mathbf{I} + \Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}$

## Regularized Least Squares (2)

With a more general regularizer, we have

$$\frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \sum_{j=1}^M |w_j|^q$$



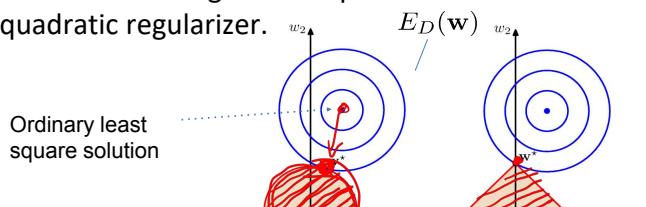
Note: In this lecture, we focus on  $q=2$  (L2 regularization), but other values of  $q>0$  can be used.

plotting curves of  $(w_1, w_2)$  where  $\sum_{j=1}^M |w_j|^q$  is a constant. ( $M=2$ )

30

## Regularized Least Squares (3)

- Lasso tends to generate sparser solutions than a quadratic regularizer.



Assuming a simple scenario of isotropic data covariance, the optimal solution to L2/L1 regularization is closest point to the original solution (center of the concentric circles) that touches the boundary of the L2/L1 constraint.

## Summary: Regularized Linear Regression

- Simple modification of linear regression
- Regularization controls the tradeoff between "fitting error" and "complexity"
  - Small regularization results in complex models (but with risk of overfitting)
  - Large regularization results in simple models (but with risk of underfitting)
- It is important to find an optimal regularization that balances between the two.

31

32

## Maximum Likelihood interpretation of least squares regression

## Review on probability

33

34

## Probability: Terminology

- Experiment:** Procedure that yields an outcome
  - E.g., Tossing a coin three times:
    - Outcome: HHH in one trial, HTH in another trial, etc.
- Sample space:** Set of all possible outcomes in the experiment, denoted as  $\Omega$  (or  $S$ )
  - E.g., for the above example:
    - $\Omega = \{\text{HHH}, \text{HHT}, \text{HTH}, \text{HTT}, \text{THH}, \text{THT}, \text{TTH}, \text{TTT}\}$
- Event:** subset of the sample space  $\Omega$  (i.e., an event is a set consisting of individual outcomes)
  - Event space: Collection of all events, called  $\mathcal{F}$  (aka  $\sigma$ -algebra)
  - E.g., Event that # of heads is an even number.
    - $E = \{\text{HHT}, \text{HTH}, \text{THH}, \text{TTT}\}$
- Probability measure:** function (mapping) from events to probability levels. I.e.,  $P : \mathcal{F} \rightarrow [0, 1]$  (see next slide)
  - Probability that # of heads is an even number:  $4/8 = 1/2$ .
- Probability space:**  $(\Omega, \mathcal{F}, P)$

35

36

## Law of Total Probability

$$P(A) \geq 0, \forall A \in \mathcal{F}$$

$$P(\Omega) \geq 1$$

- Law of total probability**

$$\underline{P(A)} = \underline{P(A \cap B)} + \underline{P(A \cap B^C)}$$

$$P(A) = \sum_i P(A \cap B_i) \quad \text{Discrete } B_i$$

$$\underline{P(A)} = \int P(A \cap B_i) dB_i \quad \text{Continuous } B_i$$



## Conditional Probability

For events  $A, B \in \mathcal{F}$  write the conditional probability of  $B$  given  $A$ , we may

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$\begin{aligned} P(A|B_1) &= 1 \\ P(A|B_2) &= 0.12/(0.12 + 0.04) = 0.75 \\ P(A|B_3) &= 0 \quad (\text{disjoint}) \\ P(A) &= 0.30 + 0.10 + 0.12 = 0.52 \\ &\quad (\text{the unconditional probability}) \end{aligned}$$

From Wikipedia

35

## Bayes' Rule

Using the chain rule we may see:

$$P(A|B)P(B) = P(A \cap B) = P(B|A)P(A)$$

Rearranging this yields **Bayes' rule**:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Often this is written as:

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_i P(A|B_i)P(B_i)}$$

Where  $B_i$  are a partition of  $\Omega$  (note the bottom is just the law of total probability).

36

38

## Likelihood Functions

Why is Bayes' so useful in learning? Allows us to compute the posterior of  $w$  given data  $D$

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)} \quad \text{Prior}$$

Bayes' rule in words:

$$p(w|D) \propto p(D|w)p(w)$$

The likelihood function,  $p(w | D)$ , is evaluated for observed data  $D$  as a function of  $w$ . It expresses how parameter settings  $w$ .

## Maximum Likelihood Estimation (MLE)

- Maximum likelihood:**

- choose parameter setting  $w$  that maximizes likelihood function  $p(D | w)$ .
- choose the value of  $w$  that maximizes the probability of observed data.

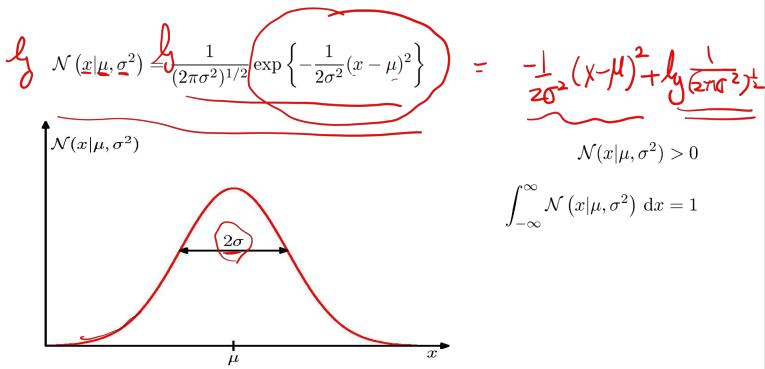
- Cf. MAP (Maximum a posteriori) estimation

- Equivalent to maximizing  $p(w|D) \propto p(D|w)p(w)$
- Can compute this using Bayes rule!
- This will be covered in later lectures

39

40

## The Gaussian Distribution | *PDcase*



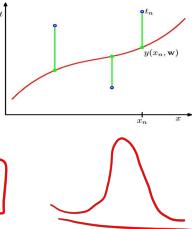
41

Maximum Likelihood interpretation of least squares regression

42

## MLE for Linear Regression

- Assume a stochastic model:  
 $y^{(n)} = \mathbf{w}^\top \phi(\mathbf{x}^{(n)}) + \epsilon$  where  $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- This gives a likelihood function:  
 $p(y^{(n)} | \phi(\mathbf{x}^{(n)}), \mathbf{w}, \beta) = \mathcal{N}(y^{(n)} | \mathbf{w}^\top \phi(\mathbf{x}^{(n)}), \beta^{-1})$
- With input matrix  $\Phi$  and output matrix  $\mathbf{y}$ , the data likelihood is:  
 $P(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(N)} | \Phi, \mathbf{w}, \beta) = P(\mathbf{y}^{(1)} | \Phi, \mathbf{w}, \beta) \dots P(\mathbf{y}^{(N)} | \Phi, \mathbf{w}, \beta)$   
 $p(\mathbf{y} | \Phi, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(y^{(n)} | \mathbf{w}^\top \phi(\mathbf{x}^{(n)}), \beta^{-1})$



43

## Log-likelihood

- Given data likelihood (prev. slide)

$$p(\mathbf{y} | \Phi, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(y^{(n)} | \mathbf{w}^\top \phi(\mathbf{x}^{(n)}), \beta^{-1})$$

- Log likelihood:  $= \sum_{n=1}^N \log \mathcal{N}(\cdot)$

$$\log p(\mathbf{y} | \Phi, \mathbf{w}, \beta) = \frac{N}{2} \log \beta - \frac{N}{2} \log 2\pi - \frac{1}{2} E_D(\mathbf{w})$$

$$\text{where } E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2$$

- Derivation?

44

## Derivation of log-likelihood of p

$$\text{From } p(y^{(n)} | \phi(\mathbf{x}^{(n)}), \mathbf{w}, \beta) = \mathcal{N}(y^{(n)} | \mathbf{w}^\top \phi(\mathbf{x}^{(n)}), \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2\right)$$

Derive:  $\log p(y^{(1)}, y^{(2)}, \dots, y^{(N)} | \Phi, \mathbf{w}, \beta)$

$$\begin{aligned} &= \log \prod_{n=1}^N \mathcal{N}(y^{(n)} | \mathbf{w}^\top \phi(\mathbf{x}^{(n)}), \beta^{-1}) \\ &= \sum_{n=1}^N \log \left( \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2\right) \right) \\ &= \sum_{n=1}^N \left( \frac{1}{2} \log \beta - \frac{1}{2} \log 2\pi - \frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2 \right) \\ &= \frac{N}{2} \log \beta - \frac{N}{2} \log 2\pi - \sum_{n=1}^N \frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2 \end{aligned}$$

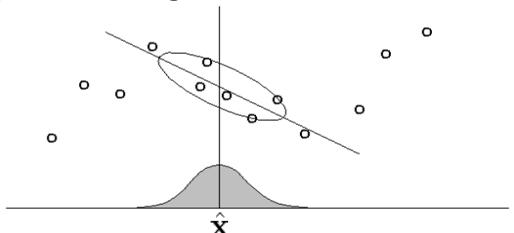
## Maximum likelihood estimation (MLE)

- Let's maximize the log-likelihood!
  - Set the gradient of log-likelihood = 0 (Why?)
- $$\begin{aligned} \nabla_{\mathbf{w}} \log p(\mathbf{y} | \Phi, \mathbf{w}, \beta) &= \nabla_{\mathbf{w}} \left( \frac{N}{2} \log \beta - \frac{N}{2} \log 2\pi - \sum_{n=1}^N \frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2 \right) \\ f(\mathbf{w}) &\quad \text{Concave} \\ &= \beta \sum_{n=1}^N (y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)}) \phi(\mathbf{x}^{(n)})) \\ &= \beta \left( \sum_{n=1}^N y^{(n)} \phi(\mathbf{x}^{(n)}) - \phi(\mathbf{x}^{(n)}) \phi(\mathbf{x}^{(n)})^\top \mathbf{w} \right) = 0 \end{aligned}$$
- In matrix form,  $\beta(\Phi^\top \mathbf{y} - \Phi^\top \Phi \mathbf{w}) = 0$   
 $\mathbf{w}_{\text{ML}} = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}$
  - MLE solution is equivalent to OLS solution!

46

## Locally weighted linear regression

- Main idea: When predicting  $f(\hat{\mathbf{x}})$ , give high weights for "neighbors" of  $\hat{\mathbf{x}}$ .



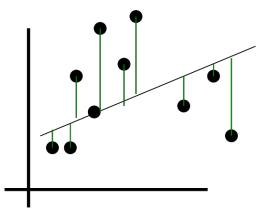
In locally weighted regression, points are weighted by proximity to the current  $\hat{\mathbf{x}}$  in question using a kernel. A regression is then computed using the weighted points.

## Locally-weighted Linear Regression

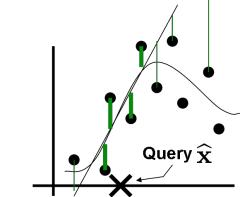
47

Slide credit: William Cohen 48

## Regular linear regression vs. locally weighted linear regression



Regular linear regression



Locally weighted linear regression

$$\sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2$$

$$\sum_{n=1}^N r^{(n)}(\hat{\mathbf{x}}) (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2$$

49

## Linear regression vs. Locally-weighted Linear Regression

- A query point  $\hat{\mathbf{x}}$ , training set  $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$

- Linear regression

- Fit  $\mathbf{w}$  to minimize  $\sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2$
- Predict  $\mathbf{w}^\top \phi(\hat{\mathbf{x}})$

- Locally-weighted linear regression

- Fit  $\mathbf{w}$  to minimize  $\sum_{n=1}^N r^{(n)}(\hat{\mathbf{x}}) (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2$
- Predict  $\mathbf{w}^\top \phi(\hat{\mathbf{x}})$

weights are dependent on the query  $\hat{\mathbf{x}}$   
(i.e., need to solve the optimization for each query value)

50

## Linear regression vs. Locally-weighted Linear Regression

- Locally-weighted linear regression

- Fit  $\mathbf{w}$  to minimize  $\sum_{n=1}^N r^{(n)}(\hat{\mathbf{x}}) (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2$
- Predict  $\mathbf{w}^\top \phi(\hat{\mathbf{x}})$

- Remarks:

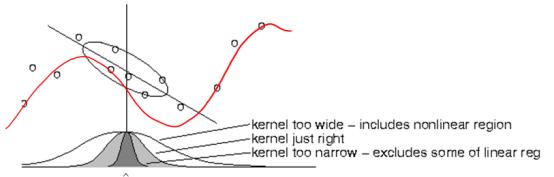
- Standard choice:  $r^{(n)}(\hat{\mathbf{x}}) = \exp\left(-\frac{\|\phi(\mathbf{x}^{(n)}) - \phi(\hat{\mathbf{x}})\|^2}{2\tau^2}\right)$
- Note that  $r^{(n)}(\hat{\mathbf{x}})$  depends on  $\hat{\mathbf{x}}$  (query point), and you solve linear regression for each query point  $\hat{\mathbf{x}}$
- The problem can be formulated as a modified version of least squares problem (HW#1)

Gaussian kernel with kernel width  $\tau$

51

## Locally weighted linear regression

- Choice of kernel width  $\tau$  matters
  - Requires hyper-parameter tuning



The estimator is minimized when kernel includes as many training points as can be accommodated by the model. Too large a kernel includes points that degrade the fit; too small a kernel neglects points that increase confidence in the fit.

Slide credit: William Cohen 52

## Summary

- $L_2$  Regularized linear regression  $\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$ 
  - Adding  $L_2$  regularizer
  - Can be solved via closed form (simple modification of the original linear regression)
  - penalizes complex solutions (with high weights)
- Maximum likelihood interpretation of linear regression
  - Linear regression can be interpreted as performing MLE assuming the Gaussian noise distribution for targets
- Locally-weighted linear regression

Any feedback (about lecture, slide, homework, project, etc.)?

(via anonymous google form: <https://forms.gle/fpYmiBtG9Me5qbP37>)



Change Log of lecture slides: <https://docs.google.com/document/d/e/2PACX-1vSSiHjklypK7rKFSR1-5GYxyBCEW8UPtpSfCR9AR6Ml7K9ZQEmxfFwaWaW7kLDxusthsF8WICyZJ-/pub>

53

54