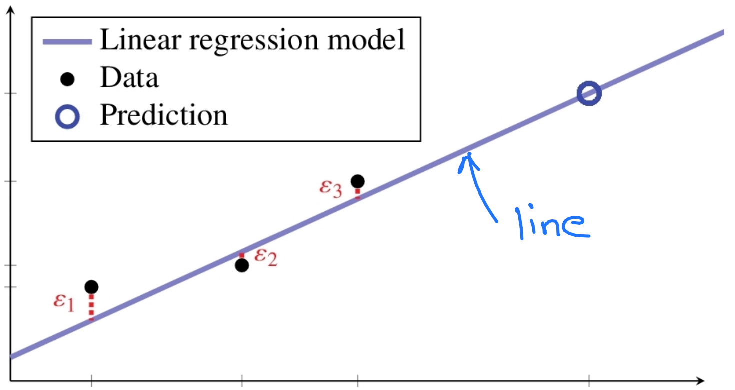
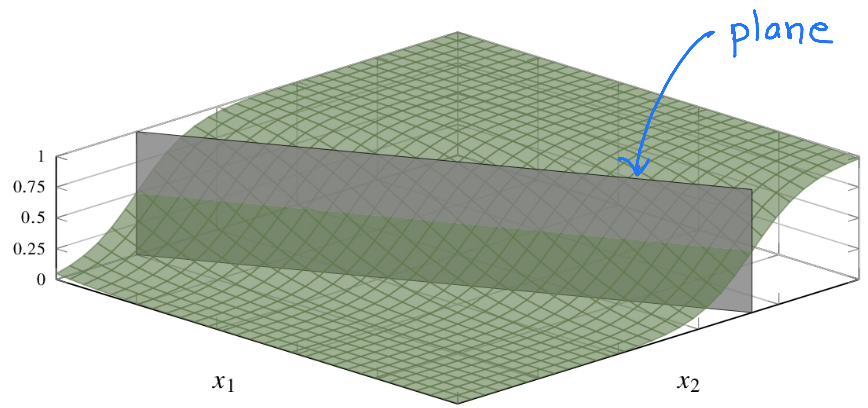
Lecture 7 — Polynomial Regression, Regularization, Generalized linear models

- We looked at two basic parametric models
  - → Linear regression
  - → Logistic regression (linear regression + logistic function)

- Compared to NON-PARAMETRIC models, linear regression and logistic regression appear to be rigid and not very flexible
  - they fit straight lines (or hyperplanes)





- Make linear regression more flexible by increasing the input dimension p

- Question: How to increase input dimension?

- Common Approach: Add non-linear transformation of the input

- A simple nonlinear transformation of one-dimensional input x:

$$\underbrace{y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \cdots + \theta_p x^p + \epsilon}_{\text{Polynomial regression}}$$

— Recall $y = \underline{x}^T \underline{\theta}$ where $\underline{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$, $\underline{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix}$

$$\underbrace{y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \cdots + \theta_p x^p + \epsilon}_{\text{Polynomial regression}}$$

— If $x_1 = x$, $x_2 = x^2$, $x_3 = x^3$, ..., $x_p = x^p$ $\Rightarrow$ $y = \begin{bmatrix} 1 & x & x^2 & x^3 & \cdots & x^p \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_p \end{bmatrix}$

$$= \underbrace{\underline{x}^T \underline{\theta}}$$

Still a linear model

however "lifted" the input from

one-dimension $(p=1)$ to

three-dimension $(p=3)$

— The same polynomial expansion can also be applied to logit z in logistic regression

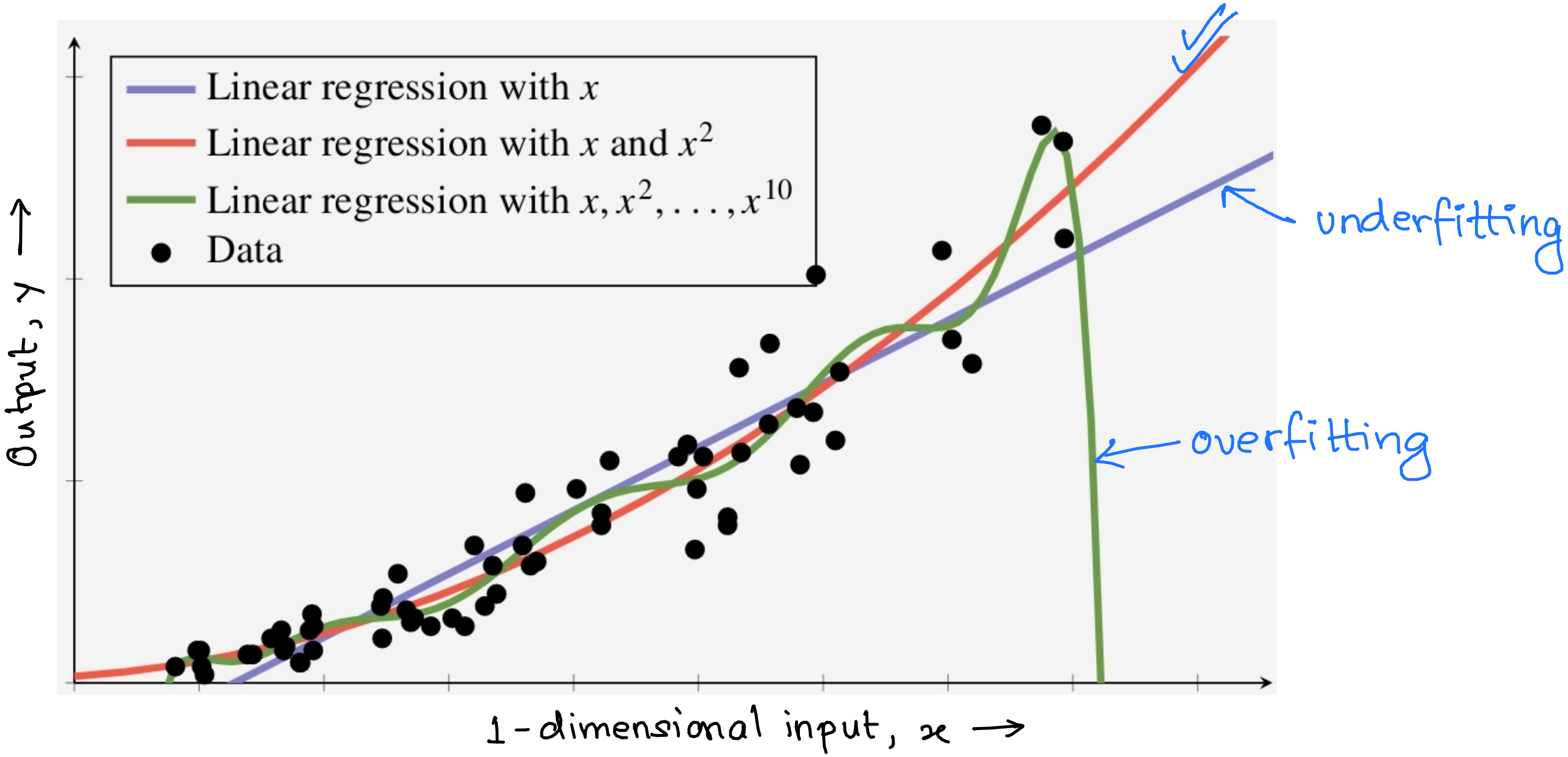
$$z = \begin{bmatrix} 1 & x & x^2 & \cdots & x^p \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix} = \underline{x}^T \underline{\theta}$$

$y = h(z)$ logistic function

— Using nonlinear transformations are quite useful in practice

- effectively increases input dimension p

— Downside : Can lead to overfitting (the model may fit noise in the training data)



— Ways to avoid overfitting

- Carefully select which input transformations to include → add one inputs at a time → removing inputs that are redundant
- Use regularization

Regularization

- Basic idea: Keep the parameters $\underline{\hat{\theta}}$ small unless really required!

- Meaning → if a model with small parameter values $\underline{\hat{\theta}}$ fits the data almost as well as a model with large parameter values, the model with smaller $\underline{\hat{\theta}}$ will be preferred

$$\underline{\hat{\theta}}^{(1)} = \begin{bmatrix} 0.2 \\ 1.5 \\ -0.01 \\ 0.005 \\ 0.01 \end{bmatrix}, \quad \underline{\hat{\theta}}^{(2)} = \begin{bmatrix} 2.3 \\ 10.6 \\ -1.2 \\ 0.1 \\ -1.3 \end{bmatrix}$$

both fit the data well

this set of parameters is more preferrable!

- Several ways to implement the idea of "small parameter values"
  - L_0 - regularization
  - L_1 - regularization

  } maybe covered later

  - L_2 - regularization (will look into this here)

L_2 - REGULARIZATION

- Purpose is to prevent overfitting

- To keep $\hat{\underline{\Theta}}$ small, an extra penalty term $\lambda \|\hat{\underline{\Theta}}\|_2^2$ is added to the cost function

regularization parameter
(which is a hyper-parameter)
chosen by user

- Regularization parameter, $\lambda \geq 0$, controls the strength of regularization effect
  - Larger the λ value, smaller will be the values of $\hat{\underline{\Theta}}$
  - $\lambda = 0$ has no effect of regularization
  - $\lambda \to \infty$ will force all parameters $\hat{\underline{\Theta}}$ to 0
  - Use cross-validation to select λ or use L-curve method

- Regularization parameter, $\lambda \geq 0$, controls the strength of regularization effect
  - Larger the λ value, smaller will be the values of $\hat{\underline{\theta}}$
  - $\lambda = 0$ has no effect of regularization
  - $\lambda \to \infty$ will force all parameters $\hat{\underline{\theta}}$ to 0
  - Use cross-validation to select λ or use L-curve method

Cross-validation

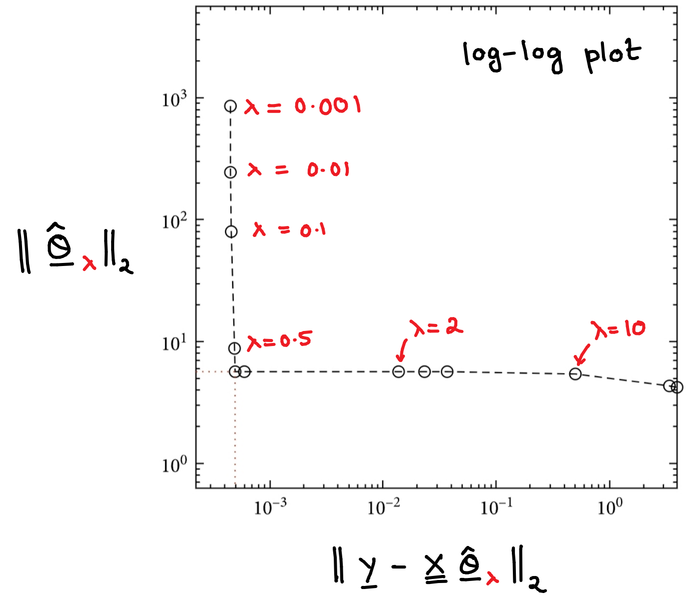
Training set	Validation set	Test set

Training w/ $\lambda = 0.01$ → err = 5 ✗

Training w/ $\lambda = 1$ → err = 1.3 ✓ → test err = 1.4

Training w/ $\lambda = 3$ → err = 7 ✗

L-curve method



- Previously studied loss function for (non-regularized) linear regression:

$$\hat{\underline{\theta}} = \underset{\underline{\theta}}{\operatorname{argmin}} \frac{1}{N} \underbrace{\| \underline{y} - \underline{\underline{X}}\, \underline{\theta} \|_2^2}_{\text{squared loss}} \longrightarrow (\underline{\underline{X}}^T \underline{\underline{X}}) \hat{\underline{\theta}} = \underline{\underline{X}}^T \underline{y}$$

- With L_2-regularization, add a penalty over $\underline{\theta}$ to the loss

$$\hat{\underline{\theta}} = \underset{\underline{\theta}}{\operatorname{argmin}} \left(\underbrace{\frac{1}{N} \| \underline{y} - \underline{\underline{X}}\, \underline{\theta} \|_2^2}_{\substack{\text{tries to fit} \\ \text{the data}}} + \underbrace{\lambda \| \underline{\theta} \|_2^2}_{\substack{\text{tries to} \\ \text{keep parameters} \\ \text{small}}} \right)$$

* Usually, the intercept parameter θ_0 is kept out of regularization

- Just like the non-regularized linear regression, the regularized problem also has a closed-form solution

$$(\underline{\underline{X}}^T \underline{\underline{X}} + N \lambda \underline{\underline{I}}) \hat{\underline{\theta}} = \underline{\underline{X}}^T \underline{y}$$

$\underline{\underline{I}} \leftarrow$ identity matrix

- This particular application of L_2-regularization is called RIDGE REGRESSION

— L_2-regularization is not just restricted to linear regression

- The $\|\underline{\hat{\theta}}\|_2^2$ penalty can be applied to any method that involves optimization

Ex: Regularized logistic regression

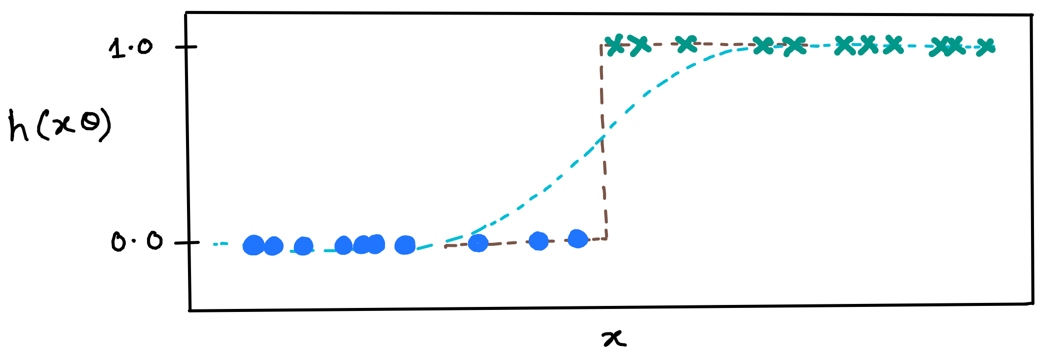
Logistic regression with L_2-regularization (very commonly used)

$$\underline{\hat{\theta}} = \underset{\underline{\theta}}{\operatorname{argmin}} \; \frac{1}{N} \sum_{i=1}^{N} \underbrace{\ln\left(1 + \exp\left(- y_i \cdot \underline{x}_i^T \underline{\theta}\right)\right)}_{\text{logistic loss}} + \lambda \|\underline{\theta}\|_2^2$$

- Reasons to use L_2-regularization in logistic regression

(a) to prevent overfitting

(b) to prevent unstable (or infinite) values of $\underline{\hat{\theta}}$



Linearly separable data causes a Heaviside step function

Generalized Linear Models

- We saw two basic parametric models:
  - linear regression (used for regression)
  - logistic regression (used for classification)

- In logistic regression, we adapted linear regression by passing the output through a nonlinear (in this case, a logistic) function
  - the output of the nonlinear logistic function was interpreted as class probability

- The same principle can be generalized to adapt linear regression model to different other properties of output as well. Such models are called Generalized linear models

- Different properties of output y
  - Output y corresponds to count of some quantity

    ex. number of cars crossing a bridge, number of earthquakes in a region

  - In such cases, y is a natural number taking values $0, 1, 2, \ldots$

  - Such count data, despite being numerical variables, cannot be well described by linear regression

    Reason: output from linear regression are not restricted to discrete or non-negative values

- To address this issue, we need to change the conditional probability model $p(y|\underline{x};\underline{\theta})$

- First step: Choose a suitable form of $p(y|\underline{x};\underline{\theta})$
    - This step is guided by properties of output data (such as natural numbers only)
    - Compute $z = \underline{x}^T\underline{\theta}$
    - Then, let $p(y|\underline{x};\underline{\theta})$ depend upon z in an appropriate way

      e.g. logistic function in logistic regression

Example: Poisson Regression

The Poisson distribution models natural numbers (including 0)

$$\mathrm{Pois}(y;\mu) = \frac{\lambda\, e^{-\mu}}{y!} \qquad y = 0, 1, 2, \ldots$$

$\mu \leftarrow$ rate-parameter, $\mu \geq 0$

$\mu = \mathbb{E}[y]$

To use the Poisson distribution for generalized linear models:

- we can let $\mu = \exp(\underline{x}^T\underline{\theta})$ to ensure $\mu \geq 0$
- $p(y|\underline{x};\underline{\theta}) = \mathrm{Pois}\left(y;\ \exp(\underline{x}^T\underline{\theta})\right)$

- Poisson regression model
  - y has a conditional Poisson distribution $p(y \mid \underline{x}; \underline{\theta})$
  - We can calculate the conditional mean, variance, etc.
    - → Conditional mean of output y

$$\mu = \mathbb{E}[y \mid \underline{x}; \underline{\theta}] = \phi^{-1}(z), \quad z = \underline{x}^T \underline{\theta}$$

$$\phi(\mu) \triangleq \log(\mu)$$

- Generalized linear models consist of:

(a) A choice of output conditional distribution $p(y \mid \underline{x}; \underline{\theta})$

[commonly from exponential family of distributions]

$y \in \{0,1\}$ → Bernoulli distribution

$y \in \mathbb{R}^+$ → Exponential distribution

$y \in \mathbb{N}$ → Poisson distribution

(b) A linear regression term $z = \underline{x}^T \underline{\theta}$

(c) A strictly increasing link function ϕ, s.t. $\mathbb{E}[y \mid \underline{x}; \underline{\theta}] = \phi^{-1}(z)$

(If μ denotes the mean of $p(y \mid \underline{x}; \underline{\theta})$, we can express $\phi(\mu) = \underline{x}^T \underline{\theta}$)