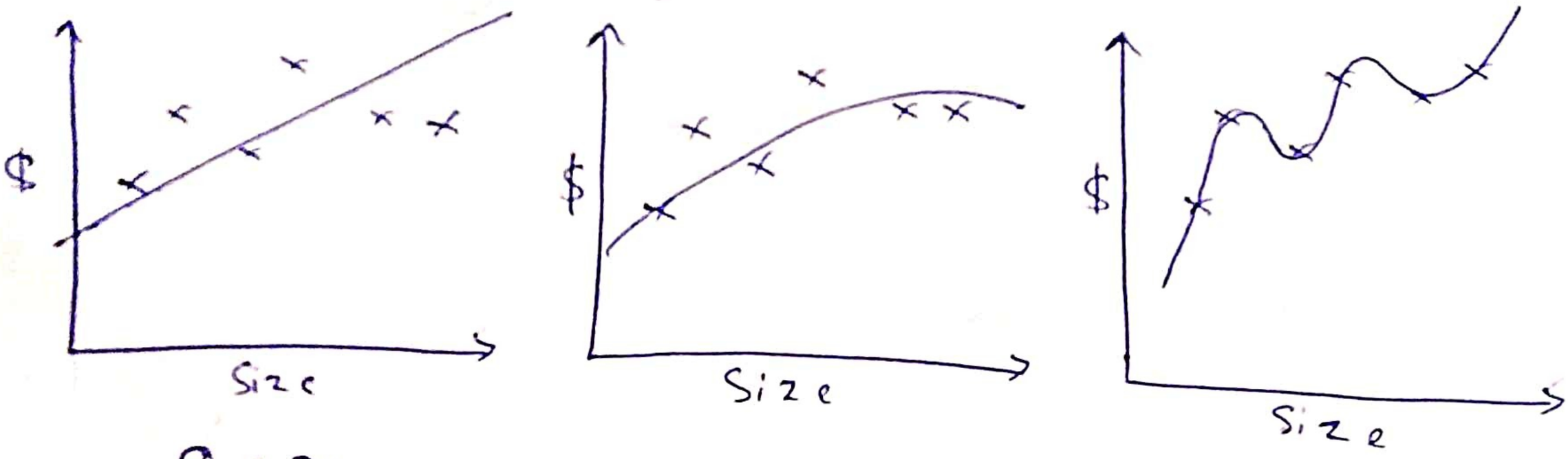
8

Data Split, Models & Cross-Validation



$\theta_0 + \theta_1 x$

Underfit

high bias

{Very Strong Preconception}

$\theta_0 + \theta_1 x + \theta_2 x^2$

Just right

$\theta_0 + \theta_1 x + \cdots + \theta_5 x^5$

Overfit

high variance

{Changing training dataset little bit will drastkly Change the fitted curve}

⇒ One of the most effective ways to prevent overfitting is regularization.

★ Regularization

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{m} \| y^{(i)} - \theta^T x^{(i)} \|^2$$ {Cost function for Linear regression}

↓ After regularization

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{m} \| y^{(i)} - \theta^T x^{(i)} \|^2 + \frac{\lambda}{2} \| \theta \|^2$$

* Bayesian statistics and regularization

Frequentist

→ In the **Frequentist** view of world, θ is not ~~not~~ random, it just happens to be unknown.

↳ and it our job to come up with statistical procedure (example **Maximum Likelihood Estimate**) to try to estimate this parameter.

⇒ In **MLE**, we estimate θ using:

$$\arg\max_{\theta} P(S|\theta)$$

Bayesian

⇒ In **Bayesian** view of world, we think of θ as being a random variable whose value is not known.

↳ In this approach, we would specify a prior distribution $P(\theta)$ on θ that expresses our "prior beliefs" about the parameters.

↳ Given a training set $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ when we are asked to make a prediction on a new value of x, we can then compute the posterior distribution on the parameter

$$\arg\max_{\theta} P(\theta|S)$$

(called **MAP** estimate)

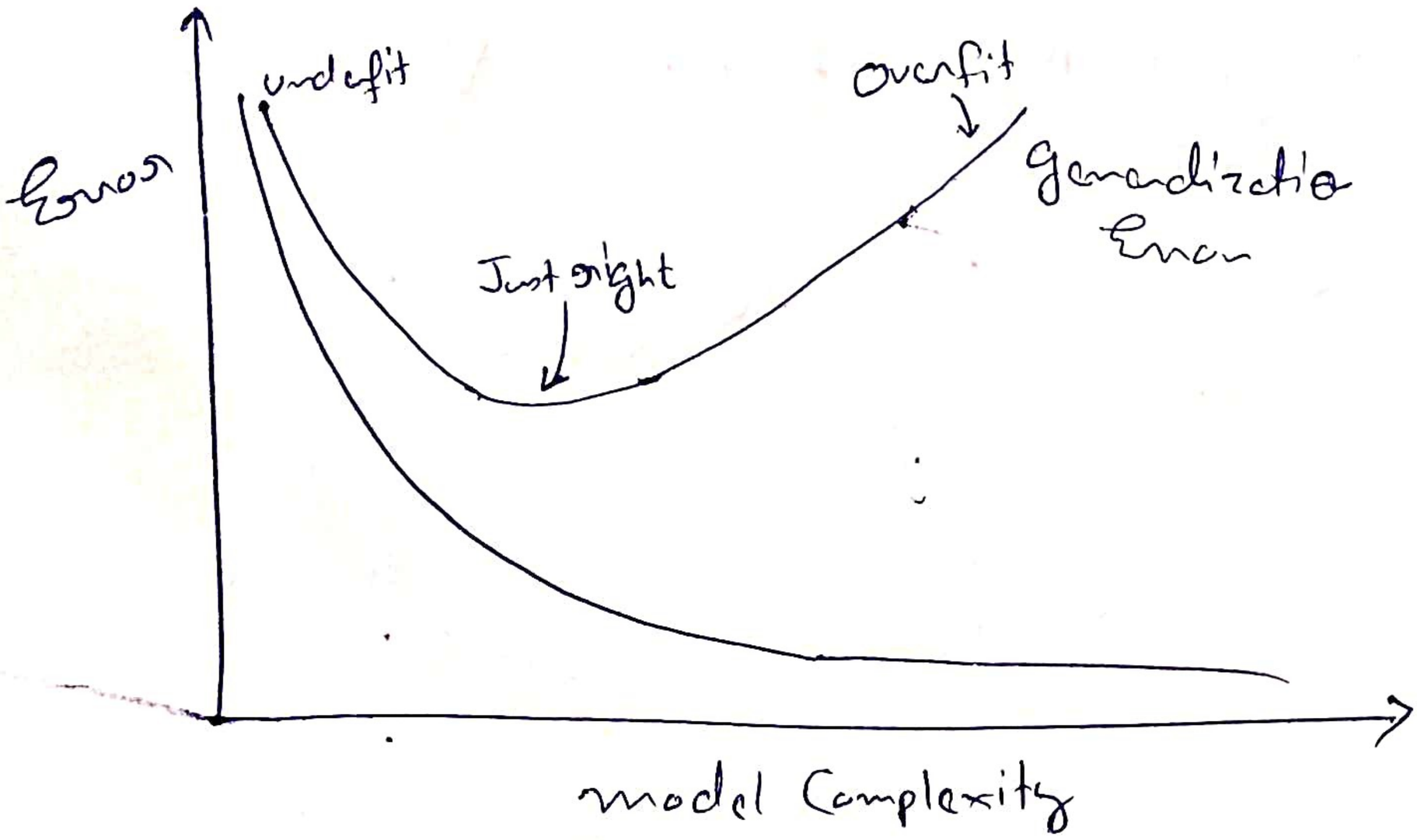
↓

Maximum a posteriori

⇒ In practical applications, a common choice for the prior $p(\theta)$ is to assume that $\theta \sim N(0, \tau^2 I)$.

⇒ Using this choice of prior, the fitted parameters θ_{MAP} will have smaller norm than that selected by Maximum Likelihood (ML).

⇒ In practice, this causes the Bayesian MAP estimate to be less susceptible to overfitting than the ML estimate of the parameters.



★ Simple Holdout Cross validation

(Train Set) + (~~Dev Set~~ Dev Set) + (Test Set) = (Data Set)

S

S_{train} S_{dev} S_{test}

60% 20% 20%

{Also called Cross-validation Set}

⇒ Train each model on S_{train}

↳ Get some hypothesis's h_i

⇒ Measure the error on S_{dev}. Pick the one with lowest error on S_{dev}.

⇒ Evaluate the algorithm on Seperate S_{test} and report that error.

* K-fold CV {Use this only for small dataset}

$$\begin{bmatrix} x^{(1)} & y^{(1)} \\ \vdots & \\ x^{(100)} & y^{(100} \end{bmatrix}$$

⇒ Divide dataset into K subsets.
(i.e. $K=5$)

→ For $i=1 \dots K$

→ Train on K-1 pieces

→ Test on the remaining 1 piece

→ Take Average.

→ Refit the best model found on 100% of the data

★ Feature Selection

Start with $F=\phi$

Forward Search

Repeat

1) Try adding each feature to F, and See which feature addition, most improves the dev set performance.

2) Add that feature to F.