FoDA - L13

Cross - Validation

How will is regression working?

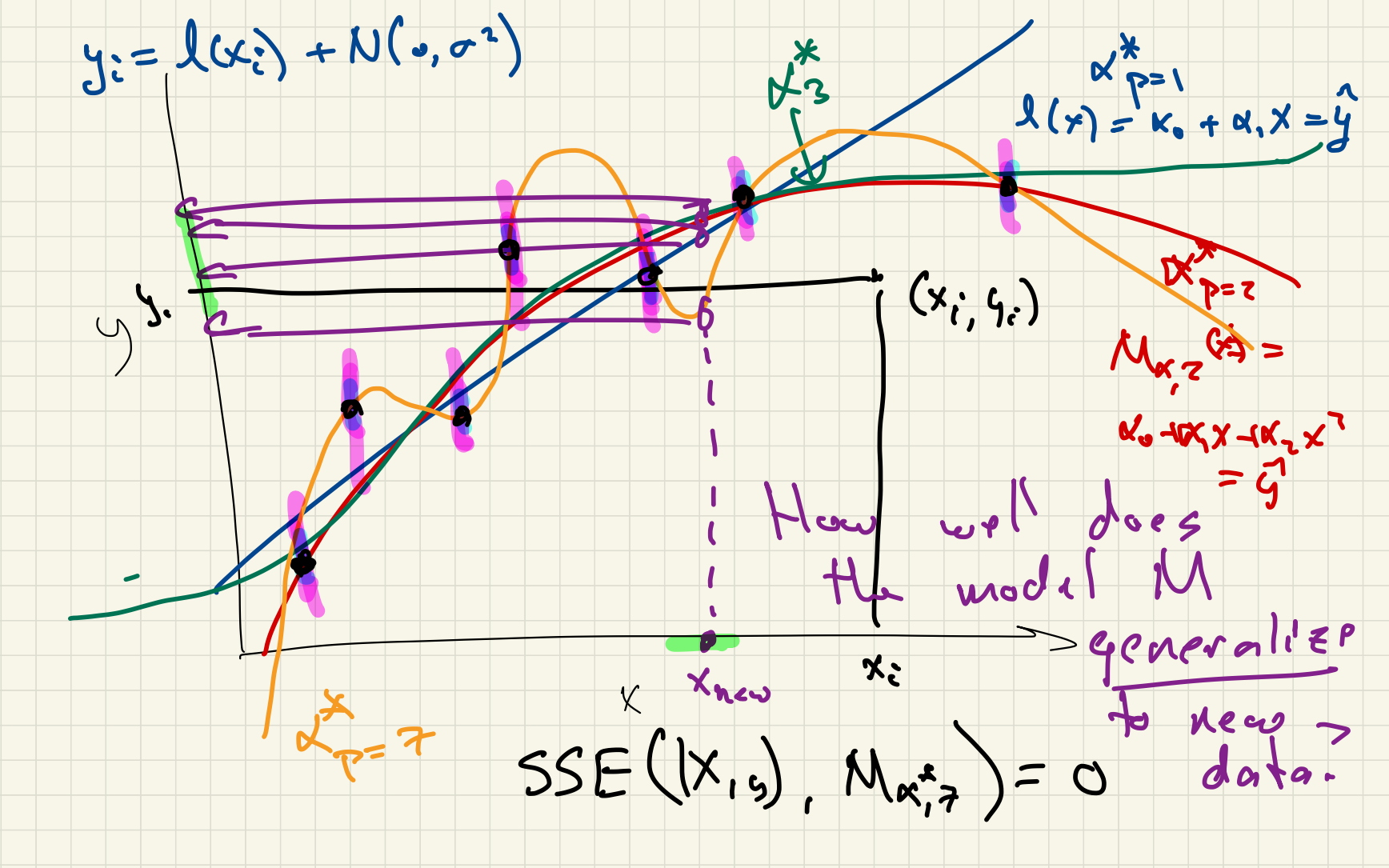
Input (x, y) $\quad x, y \in \mathbb{R}^n$

convert $\quad x \to \tilde{X}_p = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & \dots & x_2^p \\ & & \vdots & & \end{bmatrix}$

$$\alpha_p^* = (\tilde{X}_p^T \tilde{X}_p)^{-1} \tilde{X}_p y \qquad \alpha_p^* = (\alpha_0, \dots \alpha_p) \in \mathbb{R}^{p+1}$$

$$M_{\alpha_1 p}(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \dots + \alpha_p x^p$$

Which choice of p?


$y_i = f(x_i) + N(0, \sigma^2)$
$\alpha^*_{p=1}$
$f(x) = \alpha_0 + \alpha_1 x = \hat{y}$
α^*_3
$\alpha^*_{p=2}$
$M_{\alpha^*,2}(x) =$
$\alpha_0 + \alpha_1 x + \alpha_2 x^2 = \hat{y}$
(x_i, y_i)
y_i
y
x
x_{new}
x_i
$\alpha^*_{p=7}$
How well does the model M generalize to new data?
$SSE(X_{i,j}, M_{\alpha^*_{i,7}}) = 0$

What makes a model good?

stable to noise (x-coord)

- fit data well (w/ noise tolerance in y)

simple as possible

↳ How well does it generalize to new data?

Cross-Validation

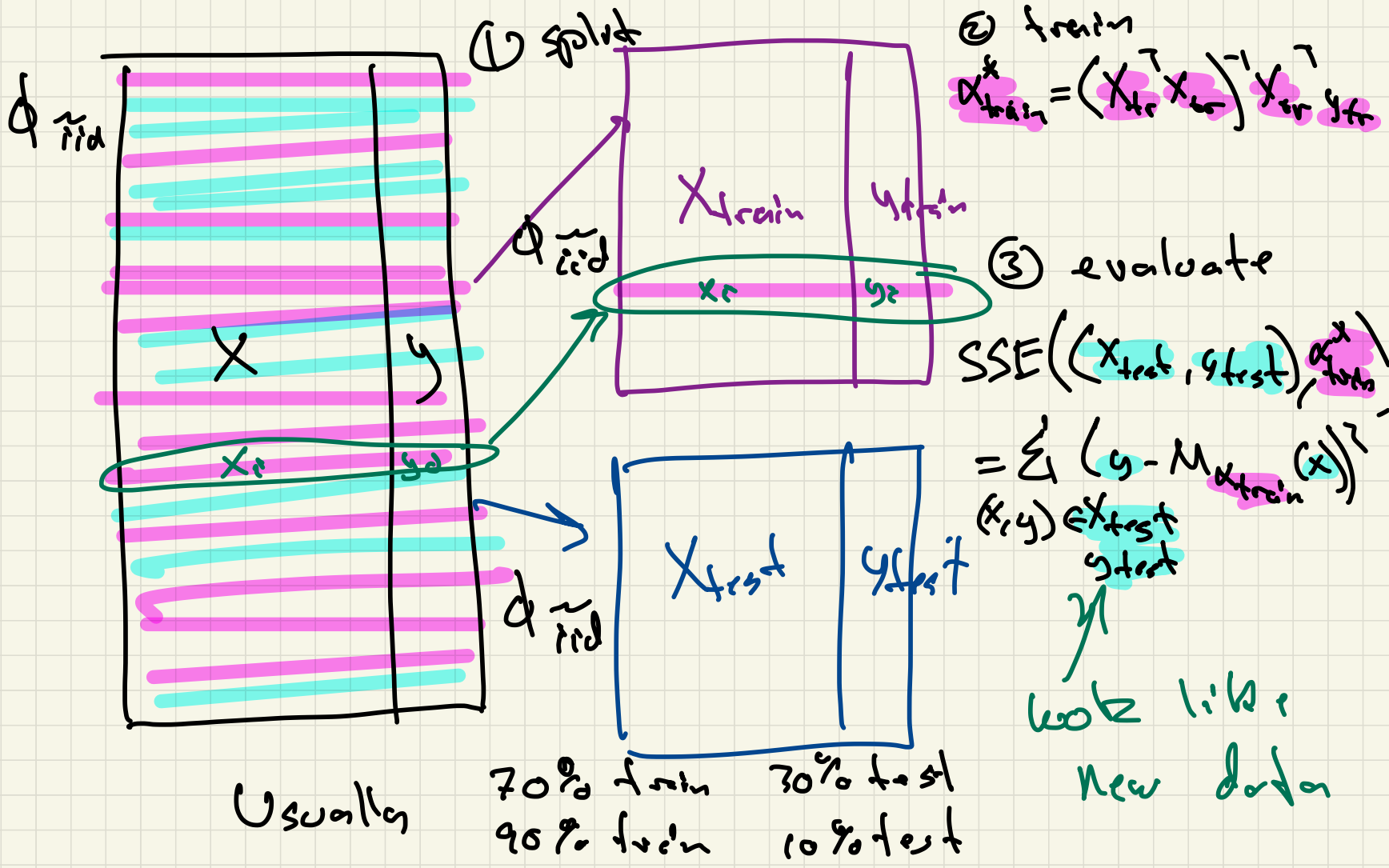
Assume $(X, y) \overset{iid}{\sim} \phi$ ← only have of input

New data $(x', y') \overset{iid}{\sim} \phi$

Assume n (# data points) is large

Randomly split (X, y) into training set (X_{train}, y_{train})

and a test set (X_{test}, y_{test})


① split
② train
③ evaluate
ϕ_{iid}
X
y
X_r
y_r
X_{train}
y_{train}
X_{test}
y_{test}
x_r
y_r
$\alpha^*_{train} = (X_{tr}^T X_{tr})^{-1} X_{tr}^T y_{tr}$
$SSE((X_{test}, y_{test}), \alpha^*_{train}) = \sum_{(x,y) \in X_{test}, y_{test}} (y - M_{\alpha_{train}}(x))^2$
look like new data
Usually
70% train 30% test
90% train 10% test

What is cross-validation good for?

① Predictive: how well does M_α do on new data.

$$\text{RMSE} = \sqrt{\frac{1}{|X_{te}|} \sum_{x,y \in X_{te} \times y_{te}} (y - M_\alpha(x))^2}$$

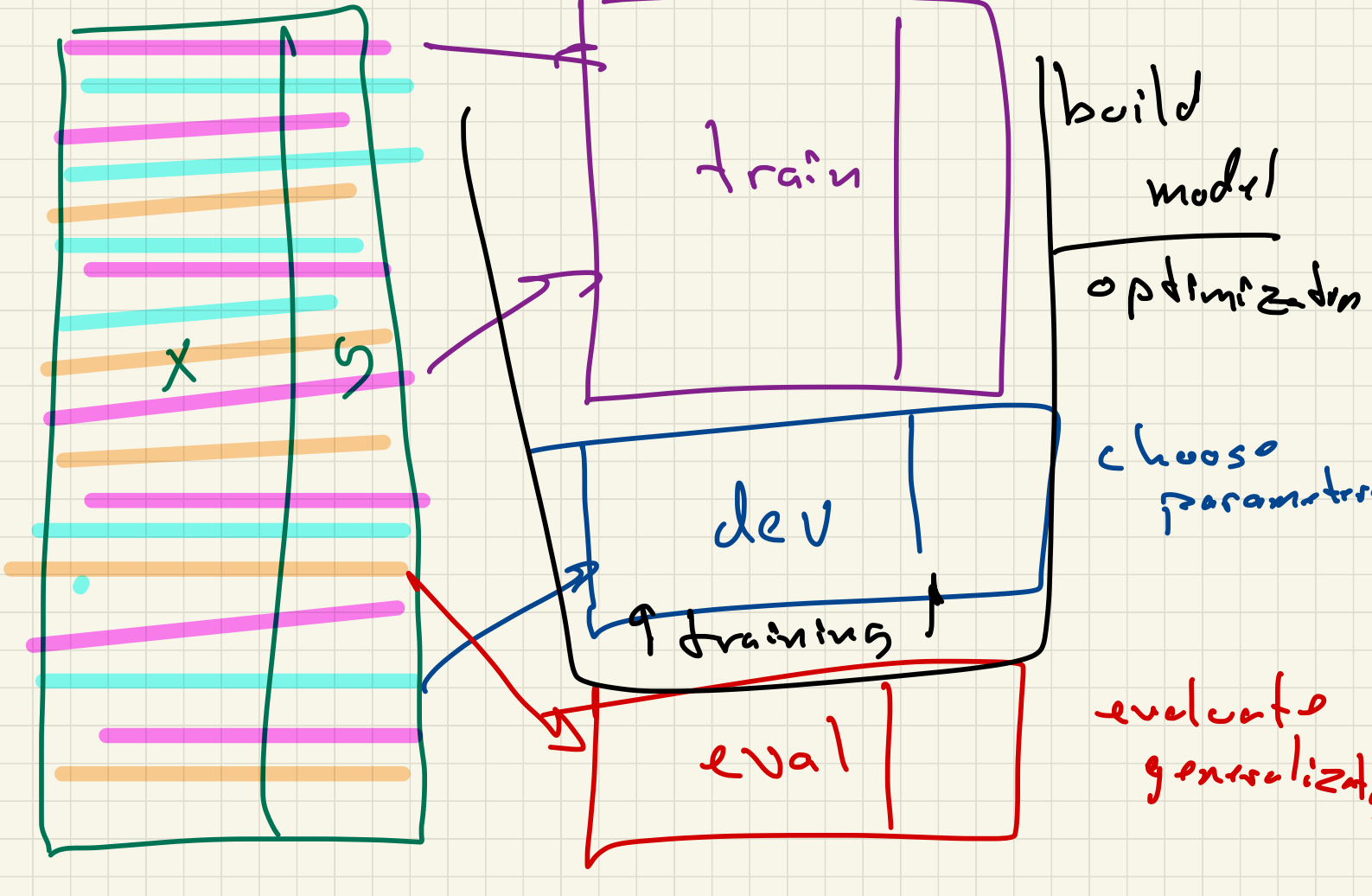
RMSE: root mean sq. err

aux. sq. error

MSE

② Choose the best model (parameter e.g. P)

$\text{MSE}((X_{tr}, y_{tr}), M_{\alpha,2})$ vs. $\text{MSE}((X_{te}, y_{te}), M_{\alpha,7})$


x
y
train
dev
training
eval
build
model
optimization
choose
evaluate

$$p^* = \arg\min_p \; SSE((x_{te}, y_{te}), M_{\alpha_p})$$

$$= \arg\min_p \; RMSE((x_{te}, y_{te}), M_{\alpha_p})$$

$$= \sqrt{\frac{1}{n_{te}} SSE}$$

What to do w/ not enough data to split?

Leave-one-out C-V n data points

Create n test / train splits

$X_{tr,i}$ $i \in [1, \ldots, n]$ $(X_{tr,1}, X_{tr,2}, \ldots X_{tr,n})$

$X_{tr,i} = \{x_1, x_2 \ldots x_{i-1}, x_{i+1}, \ldots x_n\}$

$X_{te,i} = \{x_i\}$

$$Error = \sum_{i=1}^{\hat{i}} \left(y_i - M_{X_{test,i}}(x_i) \right)^2$$

Single test point for model

build model $M_{X_{test,i}}$ for each split.