Lecture 5: KNN regression and Evaluation

KNN regression:

$$\hat{f}(\underset{\sim}{x}) = \frac{1}{K} \sum_{n \in N_K(\underset{\sim}{x})} y_n$$

= avg. traing y_ns whose corresp. x_ns are one of the K nearest neighbors to $\underset{\sim}{x}$

What happens as I change K?

General rule: K control model flexibility ↓ how complicated my predicter fn is

Small K → very flexible methods

Large K → very inflexible methods

Evaluation

How do I choose among my various models?

e.g. choose K for KNN

e.g. choose design for OLS

We need some way to evaluate which model building process (MBP) is going to perform better.

Naive way: Calculate some performance metric of my model $\hat{f}$ on the training data

e.g. (1) traing RSS

$\hat{y}_n = \hat{f}(x_n)$

$$RSS_{train} = \sum_n (y_n - \hat{y}_n)^2$$

(2) training MSE

$$MSE_{train} = \frac{RSS_{train}}{N}$$

(3) training RMSE

$$RMSE_{train} = \sqrt{MSE_{train}}$$

(4) training R^2

$$R^2_{train} = 1 - \frac{RSS_{train}}{TSS_{train}}$$

$$TSS_{train} = \sum_n (y_n - \bar{y})^2$$

$\approx$ % of var. explained by $\hat{f}$

Why shouldn't we do this?
(why isn't a metric calc. on training good?)

→ I don't actually care about predictions on my train data.

→ I actually care about performance on new/unseen data.

[Generalization Performance]

Focusing too much on train data is misleading i.e. a bad surrogate for generalization performance.

Ex. If want to choose a "best" value for K in KNN what happens if I choose the val. to minimize RSS_{train}?

Always going to choose $K=1$ b/c then $\hat{y}_n = y_n$ (Interpolate) so $RSS_{train} = 0$.

Similarly for OLS if I let

$$P \to N$$

I will eventually Interpolate my traing data.

[If I fit a $N-1^{th}$ degree poly. to N pts, I will interpolate]

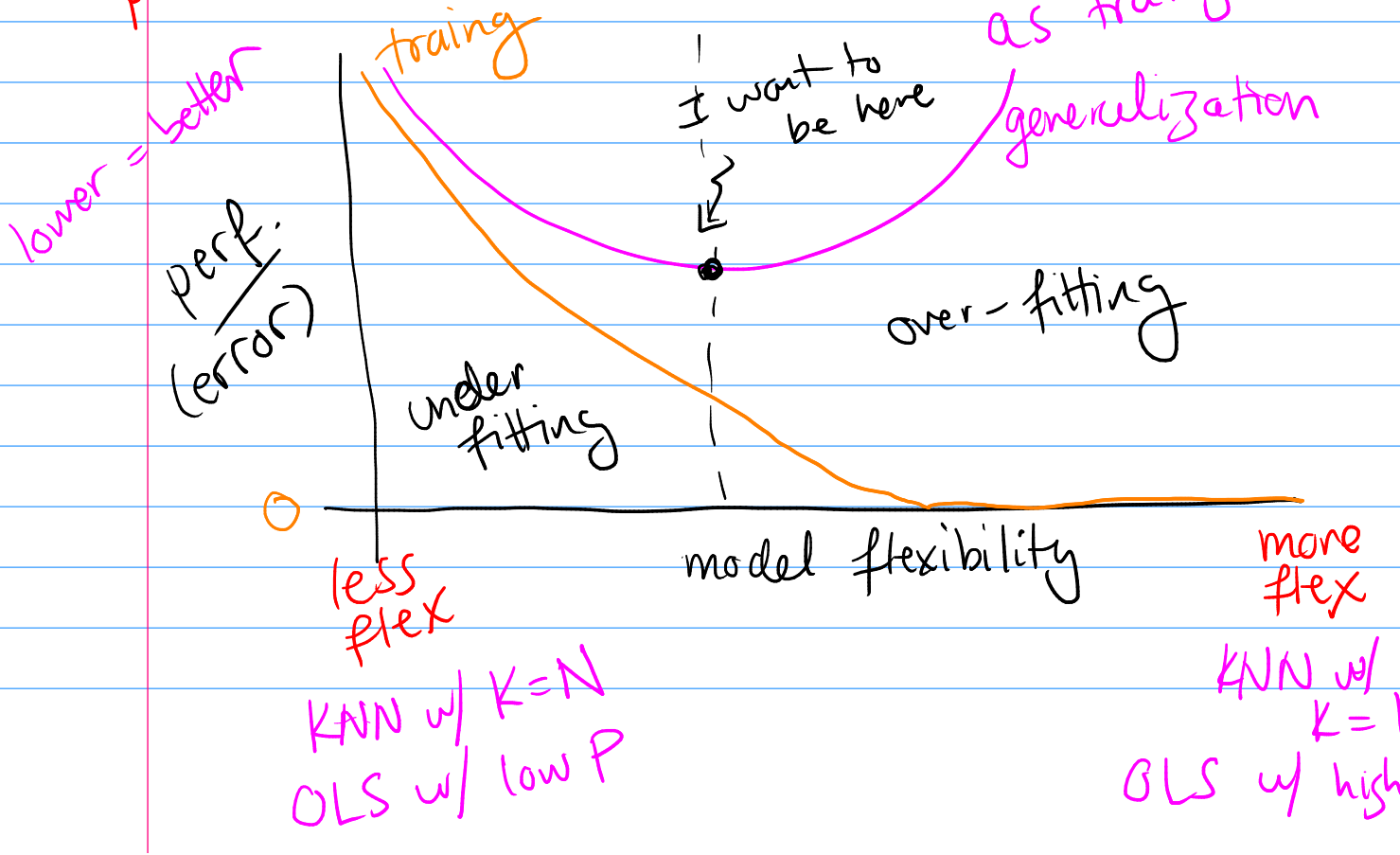
What we really want to calculate is

$$\mathbb{E}[M(X,Y)]$$

gen perf.

metric

X, Y pts draw from the same dist as traing data are new data



How do we do this?

I need some data my approach (MBP) hasn't seen before

We'll call this test data: $\{(x_{n,test}, y_{n,test})\}_{n=1,\dots,N_{test}}$

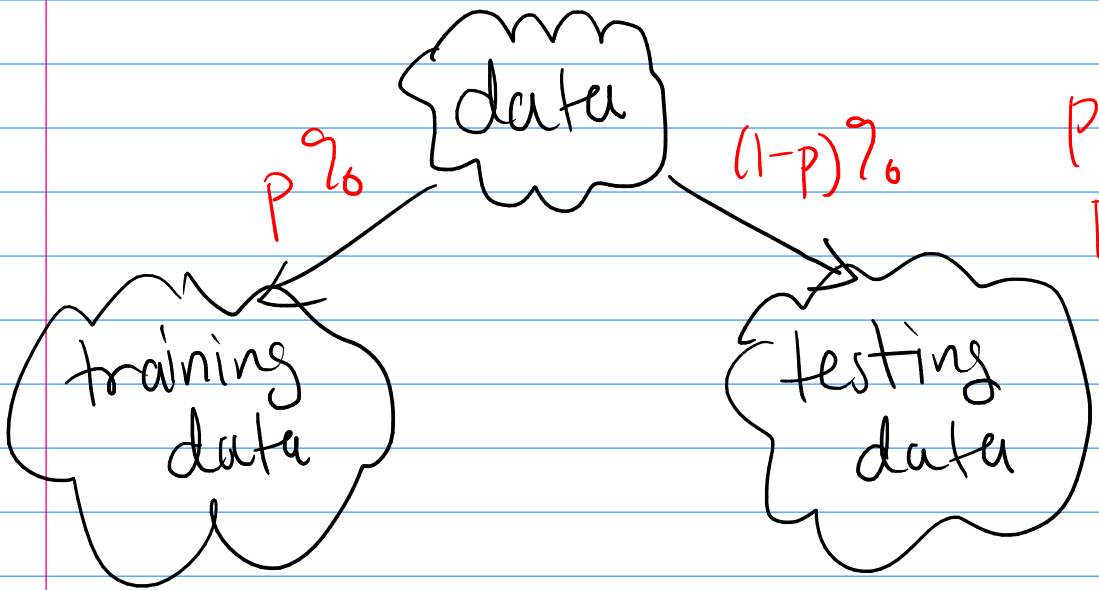
$$E[M(X,Y)] \approx M(\text{test data})$$

Procedure: (1) train $\hat{f}$ on traing data

(2) evaluate $\hat{f}$ on test data

How do I get test data?

Do a train/test split



$p \approx .9$

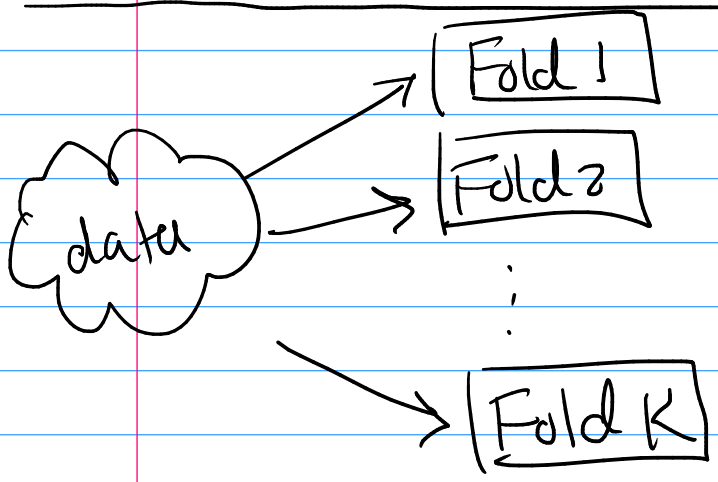
$p \approx .8$

Can also do this multiple times.

Cross-Validation

K-Fold X-Validation

e.g. $K=10$



For $k = 1, \ldots, K$

(1) train model on all but k^{th} fold

(2) evaluate on the k^{th} fold

Let m_k = perf. metric for this iter

At the end I have

$$m_1, m_2, m_3, \ldots, m_K$$

Can combine together to get an overall metric e.g.

$$m = \text{mean}(m_1, \ldots, m_K).$$

OK, so now I have K diffent models, which do I use?

None. I should probably train using all my data.