UVA CS 6316: Machine Learning

Lecture 10: Bias-Variance Tradeoff

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

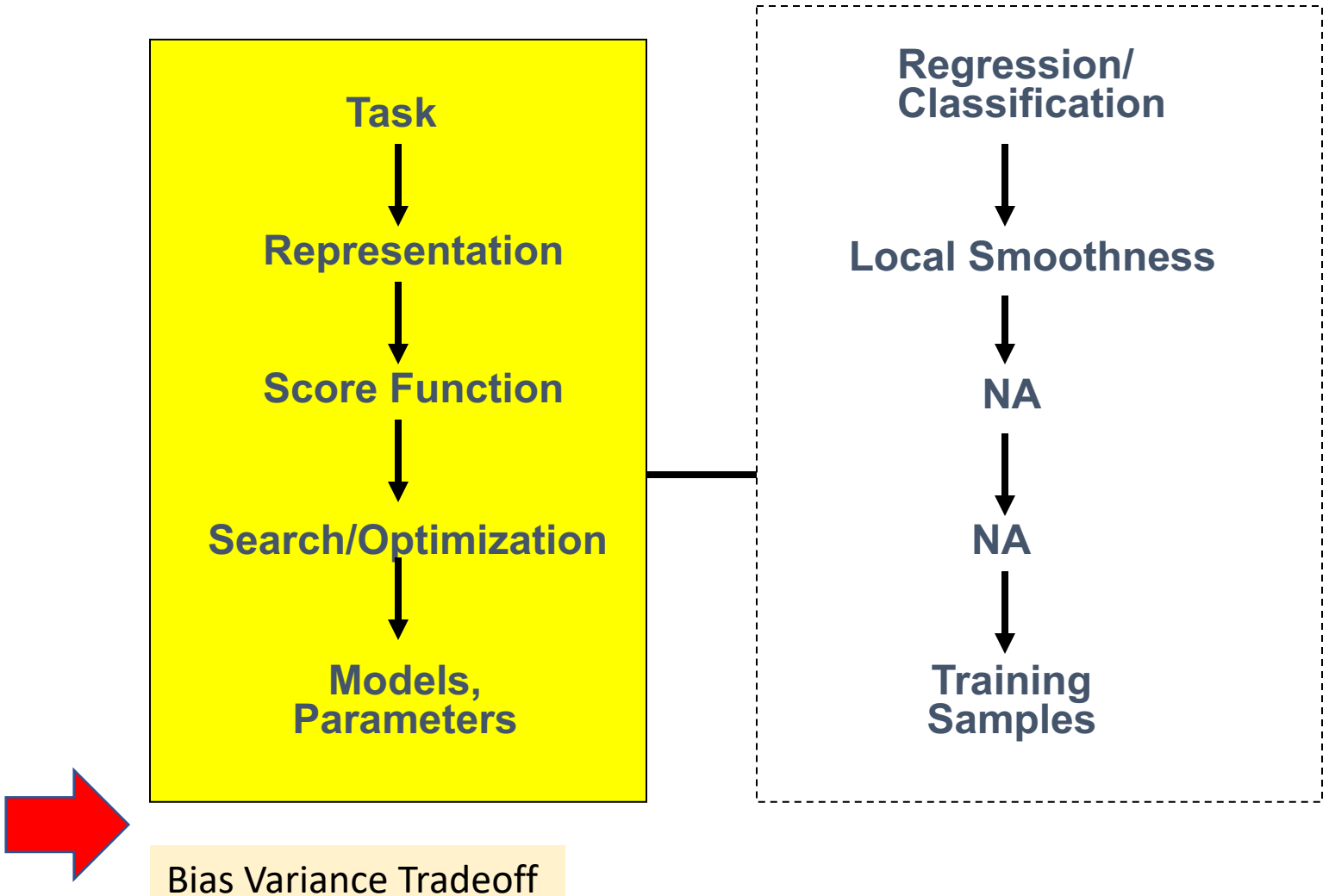
Course Content Plan ➔
Six major sections of this course

- ~~Regression (supervised)~~ — Y is a continuous
- Classification (supervised) — Y is a discrete
- Unsupervised models — NO Y
- Learning theory — About f()

- Graphical models — About interactions among X1,… Xp

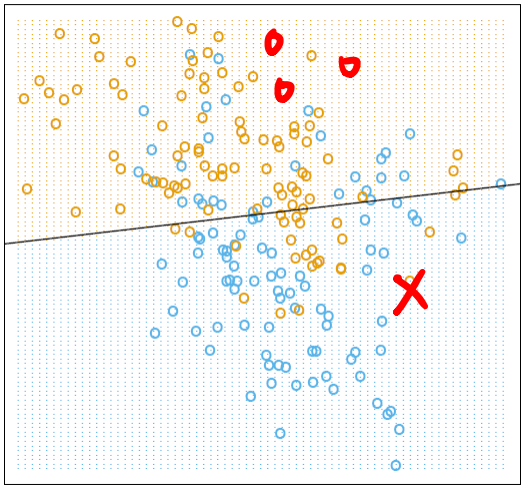
- Reinforcement Learning — Learn program to Interact with its environment

K-Nearest Neighbor



Bias Variance Tradeoff

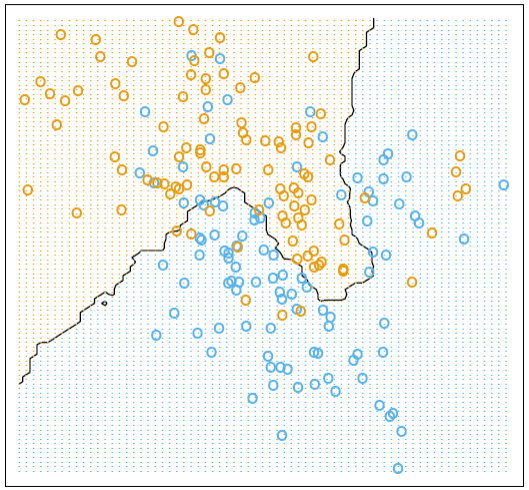
Last : Decision boundaries in global vs. local models



Linear classification

- global
- stable
- can be inaccurate

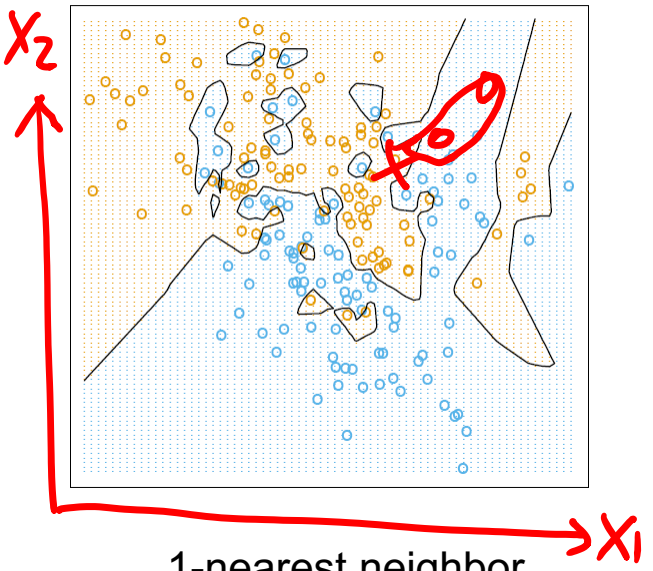
K=15



15-nearest neighbor

- local
- accurate
- unstable

K=1



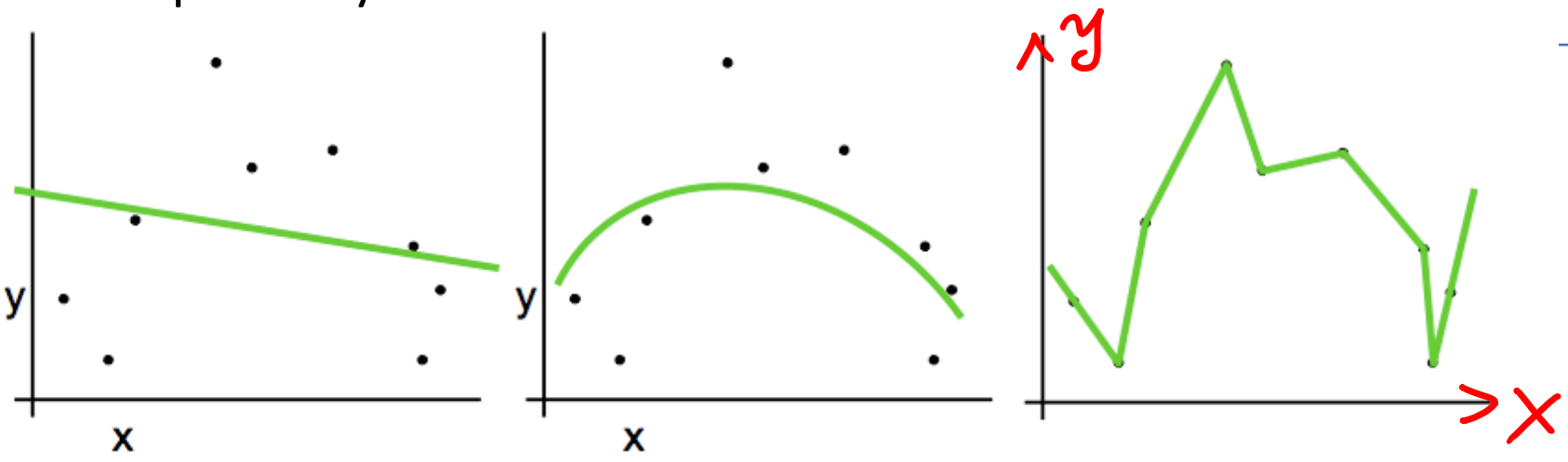
1-nearest neighbor

- K acts as a smoother

What ultimately matters: ***GENERALIZATION***

Previous in Regression:

Complexity versus Goodness of Fit



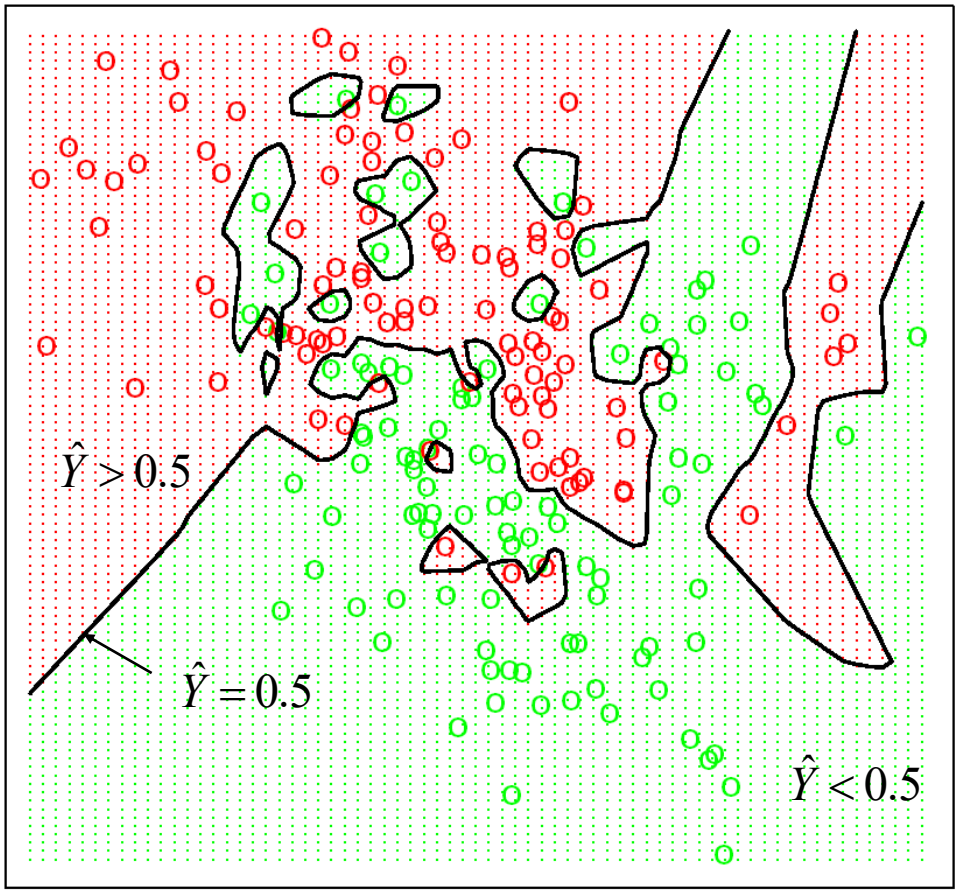
What ultimately matters: ***GENERALIZATION***

Lesson Learned : Training Error from KNN

- When k = 1,
- No misclassifications (on training): Overfit

- Minimizing training error is not always good (e.g., 1-NN)

1-nearest neighbor averaging



Roadmap

- Bias-variance decomposition
- Bias-Variance Tradeoff / Model Selection

Review: Mean and Variance of Random Variable (RV)

X: random variables written in capital letter

$\Rightarrow X = [X_1, X_2, \ldots, X_p]$ random vector

$\Rightarrow Y$ random variable

$\Rightarrow f(X)$ random variable

e.g. $\theta^T \vec{x}$

$\Rightarrow \vec{\theta}$ random variable

Review: Mean and Variance of Random Variable (RV)

Bernoulli $t \in \{-1, 1\}$

$\begin{cases} P(t=-1) = 0.1 \\ P(t=1) = 0.9 \end{cases}$

- Mean (Expectation):
  - Discrete RVs:

$$E(\mathrm{X}) = \sum_{v_i} v_i * \mathrm{P}(\mathrm{X} = v_i)$$

$E(t) = -0.1 + 0.9 = 0.8$

  - Continuous RVs:

$$E(\mathrm{X}) = \int_{-\infty}^{+\infty} x * p(x)\,dx$$

$X \sim N(2, \sigma)$

$E(X) = 2$

Adapt From Carols' prob tutorial

Review: Mean and Variance of Random Variable (RV)

$g(t) = 2t$

- Mean (Expectation):
  - Discrete RVs:

$E(g(t)) = 2 \times (-1) \times 0.1 + 2 \times (1) \times 0.9 = 1.6$

$$E(g(X)) = \sum_{v_i} g(v_i) P(X = v_i)$$

  - Continuous RVs:

$$E(g(X)) = \int_{-\infty}^{+\infty} g(x) * p(x)\, dx$$

Adapt From Carols' prob tutorial

Review: Mean and Variance of RV

- Variance:

$$Var(X) = E((X - \mu)^2) \qquad \mu = E(\mathrm{X})$$

  - Discrete RVs:

$$V(\mathrm{X}) = \sum_{v_i} (v_i - \mu)^2 \, \mathrm{P}(\mathrm{X} = v_i)$$

  - Continuous RVs:

$$V(\mathrm{X}) = \int_{-\infty}^{+\infty} (x - \mu)^2 \, p(x) dx$$

Adapt From Carols' prob tutorial

Statistical Decision Theory (Extra)

- Random input vector: X
- Random output variable: Y
- Joint distribution: Pr(X,Y) $\Rightarrow D = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$
- Loss function L(Y, f(X))

- Expected prediction error (EPE):

$$\mathrm{EPE}(f) = \mathrm{E}(L(Y, f(X))) = \int L(y, f(x)) \Pr(dx, dy)$$

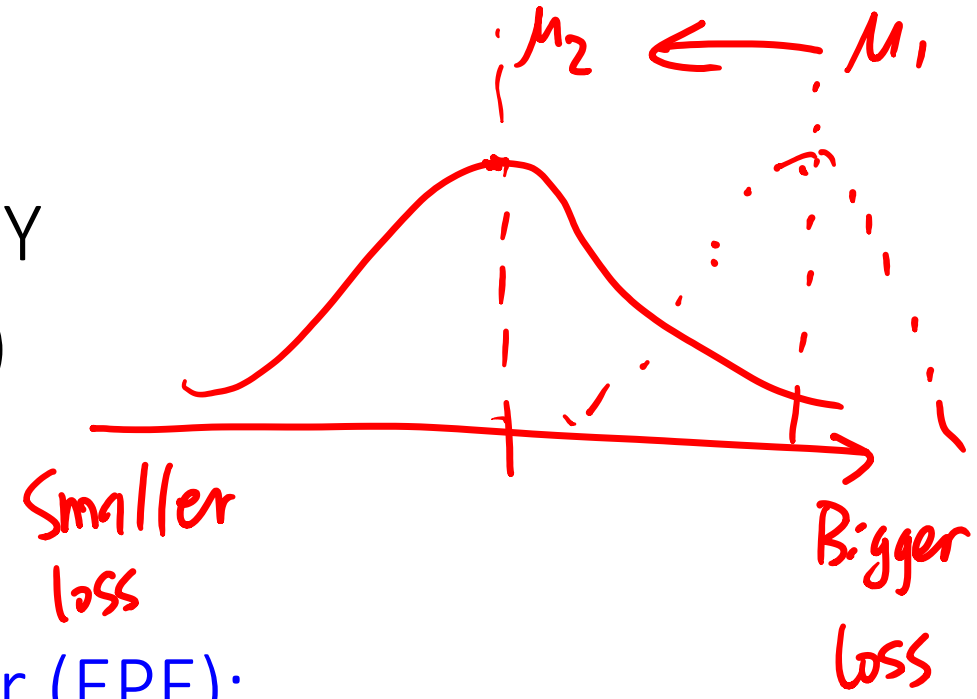
$$\text{e.g.} = \int (y - f(x))^2 \Pr(dx, dy)$$

e.g. Squared error loss (also called L2 loss)

Consider population distribution

Test Error to EPE: (Extra)

- Random input vector: X
- Random output variable:Y
- Joint distribution: Pr(X,Y)
- Loss function L(Y, f(X))



- Expected prediction error (EPE):

$$\mathrm{EPE}(f) = \mathrm{E}(L(Y, f(X))) = \int L(y, f(x)) \Pr(dx, dy)$$

$$\text{e.g.} = \int (y - f(x))^2 \Pr(dx, dy)$$

e.g. Squared error loss (also called L2 loss)

One way to consider generalization: by considering the joint population distribution

$$\begin{cases} Y = f(x) + \varepsilon & \varepsilon \sim N(0, \sigma^2) \Rightarrow f \text{ true} \\ \hat{f}_D(x) & \Rightarrow \hat{f} \text{ estimated} \\ E_D(\hat{f}(x)) = \int_D \hat{f}_D(x)\, p(D) & \Rightarrow \bar{f} \text{ Expected Estimated} \end{cases}$$

$$EPE \approx E_{(x,Y)}\left[(Y - \hat{f}(x))^2\right]$$

$$= E_{(x,Y)}\left[((Y-f) + (f-\hat{f}))^2\right]$$

$$= E\left[(Y-f)^2\right] + E\left[(f-\hat{f})^2\right]$$

$$\text{Bayes error} = \sigma^2 + E\left[(f-\hat{f})^2\right]$$

$$E\left[(f-\hat{f})^2\right]$$

$$= E\left((f-\bar{f})+(\bar{f}-\hat{f})\right)^2$$

$$= E\left[(f-\bar{f})^2\right] + E\left[(\bar{f}-\hat{f})^2\right]$$

Bias^2 Variance

$$E\left[2(f-\bar{f})(\bar{f}-\hat{f})\right]=0$$

$$E(\hat{f})=\bar{f} \Rightarrow$$

$$2E[f\bar{f}] - 2E[\bar{f}\hat{f}] - 2E[\bar{f}\bar{f}] + 2E[\bar{f}\bar{f}]$$

Decomposition of EPE

- When additive error model: $Y = f(X) + \epsilon,\ \epsilon \sim (0, \sigma^2)$
- Notations
  - Output random variable: Y
  - True function: f
  - Prediction estimator: $\hat{f}$

$$
\begin{aligned}
EPE(x_0) &= E[(Y - \hat{f})^2 | X = x_0] \\
&= E[((Y - f) + (f - \hat{f}))^2 | X = x_0] \\
&= E[\underbrace{(Y - f)}_{\epsilon}{}^2 | X = x_0] + \underbrace{E[(f - \hat{f})^2 | X = x_0]} \\
&= \sigma^2 + Var(\hat{f}) + \text{Bias}^2(\hat{f})
\end{aligned}
$$

Irreducible / Bayes error

Bias-Variance Trade-off for EPE:

$$\text{EPE}(x) = \text{noise}^2 + \text{bias}^2 + \text{variance}$$

Unavoidable error

Error due to incorrect assumptions

Error due to variance of training samples

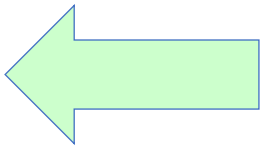
Slide credit: D. Hoiem

- More so than just these intuitive descriptions, the expected test error mathematically decomposes into a sum of three corresponding parts. Begin by writing the model

$$Y = f(X) + \varepsilon,$$

where ε has mean zero, variance σ^2, and is independent of X. Note that the independence condition is the an actual (nontrivial) assumption. Recall that (x_i, y_i), $i = 1, \ldots n$ are independent of each other and of (X, Y), all with the same distribution. We'll look at the expected test error, conditional on $X = x$ for some arbitrary input x. It follows that

$$\mathbb{E}\big[\big(Y - \hat{f}(x)\big)^2 \big| X = x\big] = \sigma^2 + \underbrace{\mathbb{E}\big[\big(f(x) - \hat{f}(x)\big)^2\big]}_{\text{Risk}(\hat{f}(x))}.$$



The first term σ^2 is the *irreducible error*, or sometimes referred to as the *Bayes error*, and the second term is called the risk, or mean squared error (MSE). The risk further decomposes into two parts, so that

$$\mathbb{E}\big[\big(Y - \hat{f}(x)\big)^2 \big| X = x\big] = \sigma^2 + \underbrace{\big(f(x) - \mathbb{E}[\hat{f}(x)]\big)^2}_{\text{Bias}^2(\hat{f}(x))} + \underbrace{\mathbb{E}\big[\big(\hat{f}(x) - \mathbb{E}[\hat{f}(x)]\big)^2\big]}_{\text{Var}(\hat{f}(x))}, \tag{2}$$

the latter terms being the squared *estimation bias* or simply *bias*, and the *estimation variance* or simply *variance*, respectively. The decomposition (2) is called the *bias-variance decomposition* or *bias-variance tradeoff*

http://www.stat.cmu.edu/~ryantibs/statml/review/modelbasics.pdf

$$E\left[\left(Y-\hat{f}(x)\right)^2\right]=E\left[\left(f(X)+\epsilon-\hat{f}(x)\right)^2\right]$$

$$=E\left[\left(f(X)-\hat{f}(x)\right)^2\right]+2E\left[\epsilon\left(f(x)-\hat{f}(x)\right)\right]+E[\epsilon^2]=MSE\left(f,\hat{f}\right)+Var(\epsilon)$$

Assuming the Bayes error is independent of $\hat{f}(x)$,

$$E\left[\epsilon\left(f(x)-\hat{f}(x)\right)\right]=E[\epsilon]E\left[f(x)-\hat{f}(x)\right]=0$$

$$E[\epsilon^2]=\sigma^2+E[\epsilon]^2=\sigma^2$$

$$E\left[\left(f(X)-\hat{f}(x)\right)^2\right]=E\left[\left(\left(f(X)-E[\hat{f}(x)]\right)+\left(E[\hat{f}(x)]-\hat{f}(x)\right)\right)^2\right]$$

http://alex.smola.org/teaching/10-701-15/homework/hw3_sol.pdf

$$= E\left[\left(f(X) - E[\hat{f}(x)]\right)^2 + 2\left(f(X) - E[\hat{f}(x)]\right)\left(E[\hat{f}(x)] - \hat{f}(x)\right) + \left(E[\hat{f}(x)] - \hat{f}(x)\right)^2\right]$$

$$= E\left[\left(f(X) - E[\hat{f}(x)]\right)^2\right] + 2E\left[\left(f(X) - E[\hat{f}(x)]\right)\left(E[\hat{f}(x)] - \hat{f}(x)\right)\right] + E\left[\left(E[\hat{f}(x)] - \hat{f}(x)\right)^2\right]$$

We can show:

$$2E\left[\left(f(X) - E[\hat{f}(x)]\right)\left(E[\hat{f}(x)] - \hat{f}(x)\right)\right] = 2\left(f(X) - E[\hat{f}(x)]\right)E[E[\hat{f}(x)] - \hat{f}(x)] = 0$$

Finally,

$$E\left[\left(f(X) - \hat{f}(x)\right)^2\right] = E\left[\left(f(X) - E[\hat{f}(x)]\right)^2\right] + E\left[\left(E[\hat{f}(x)] - \hat{f}(x)\right)^2\right]$$

$$= Bias(f(x), \hat{f}(x))^2 + Var(\hat{f}(x))$$

Putting it all together:

$$E\left[\left(Y - \hat{f}(x)\right)^2\right] = Bias(f(x), \hat{f}(x))^2 + Var\left(\hat{f}(x)\right) + \sigma^2$$

http://alex.smola.org/teaching/10-701-15/homework/hw3_sol.pdf

Another View: BIAS AND VARIANCE TRADE-OFF for parameter estimation (Extra)

$$
\begin{aligned}
MSE(\hat{\theta}) &= E[(\hat{\theta} - \theta)^2] \\
&= E[((\hat{\theta} - \bar{\theta}) + (\bar{\theta} - \theta))^2] \\
&= E[(\hat{\theta} - \bar{\theta})^2] + E[(\bar{\theta} - \theta)^2] + 2E[(\hat{\theta} - \bar{\theta})(\bar{\theta} - \theta)] \\
&= Var(\hat{\theta}) + \text{Bias}^2(\hat{\theta}) + 0
\end{aligned}
$$

$E(\hat{\theta}) = \bar{\theta}$

Error due to variance of training samples

Error due to incorrect assumptions

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] = Bias^2(\hat{\theta}) + Var(\hat{\theta})$$

BIAS AND VARIANCE TRADE-OFF for Parameter Estimation (Extra)

- θ: true value (normally unknown)
- $\hat{\theta}$: estimator
- $\bar{\theta} := E[\hat{\theta}]$ (mean, i.e. expectation of the estimator)

- Bias $E[(\bar{\theta} - \theta)^2]$
  - measures accuracy or quality of the estimator
  - low bias implies on average we will accurately estimate true parameter from training data

- Variance $E[(\hat{\theta} - \bar{\theta})^2]$
  - Measures precision or specificity of the estimator
  - Low variance implies the estimator does not change much as the training set varies

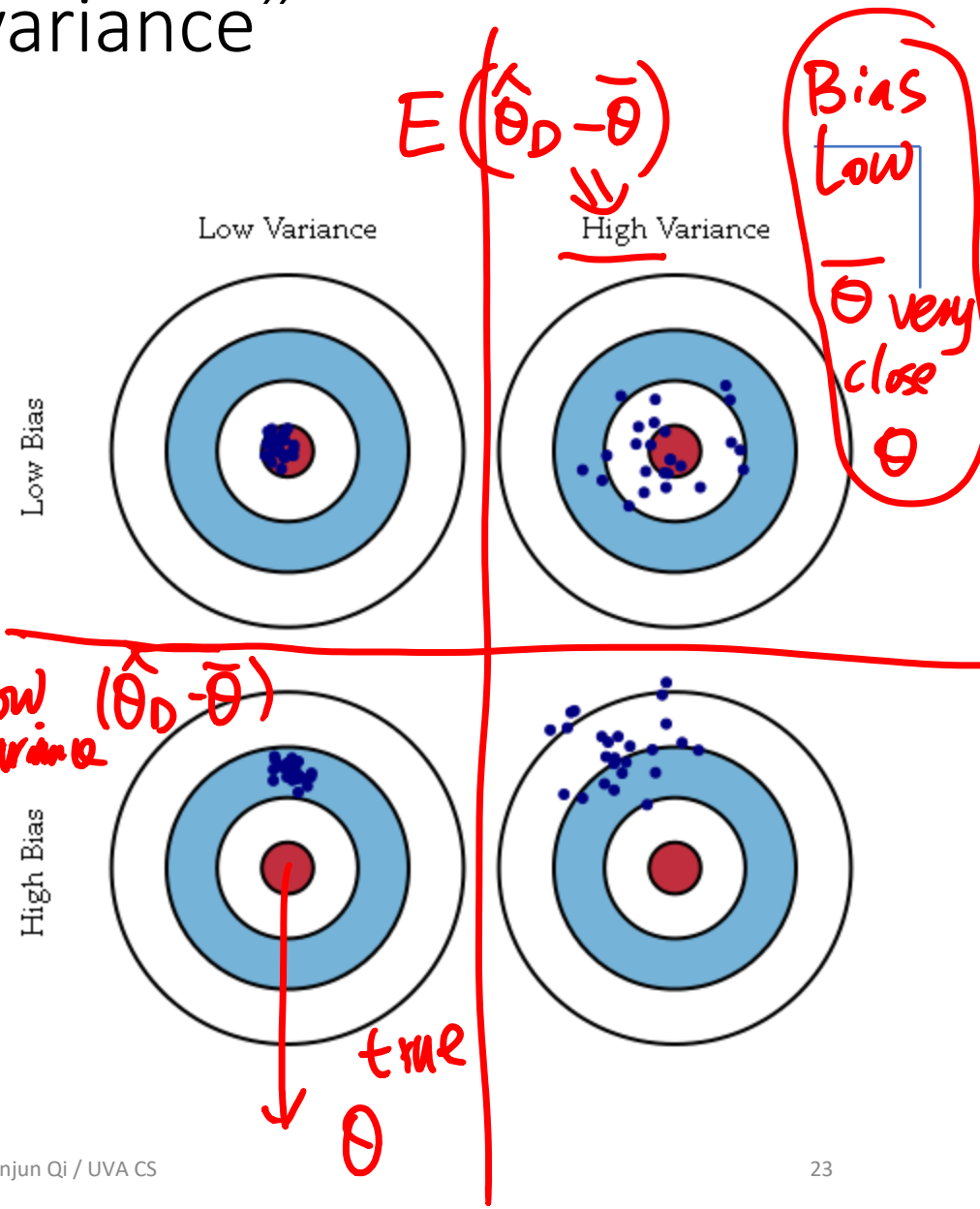
Model "bias" & Model "variance"

- Middle RED:
  - TRUE function
- Error due to bias:
  - How far off in general from the middle red

$$E[(\bar{\theta} - \theta)^2]$$

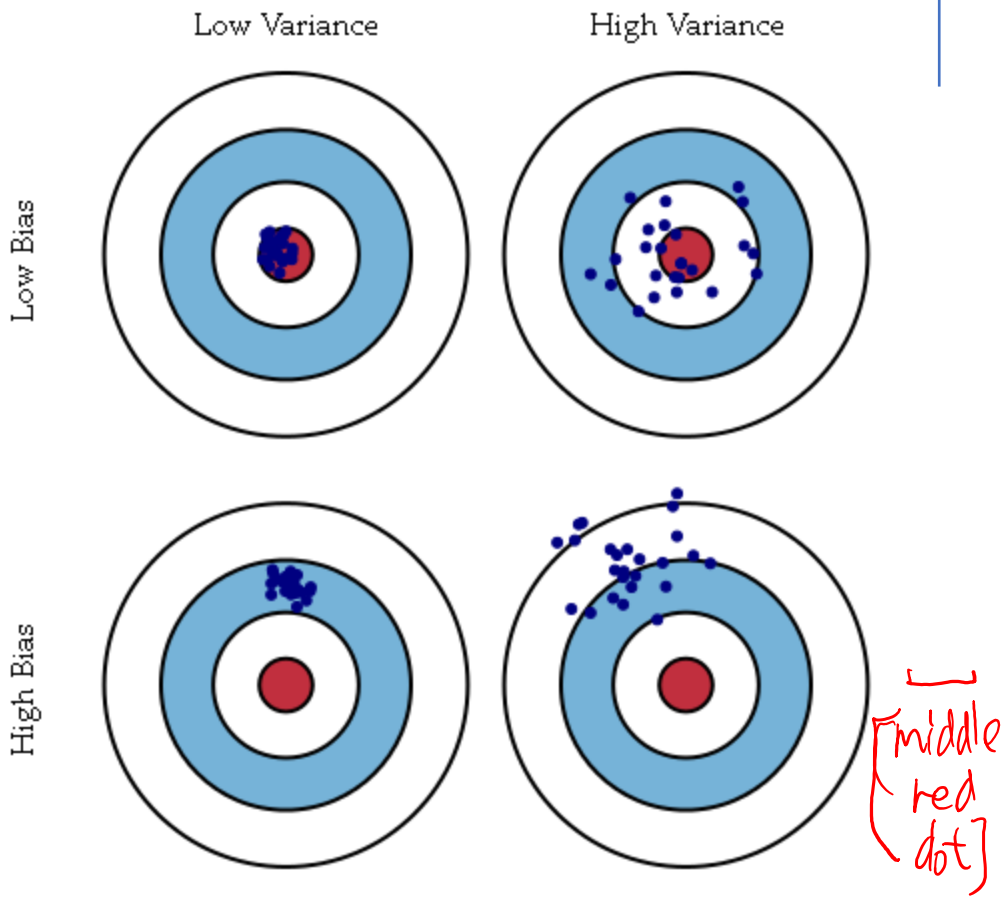
- Error due to variance:
  - How wildly the blue points spread

$$E[(\hat{\theta} - \bar{\theta})^2]$$



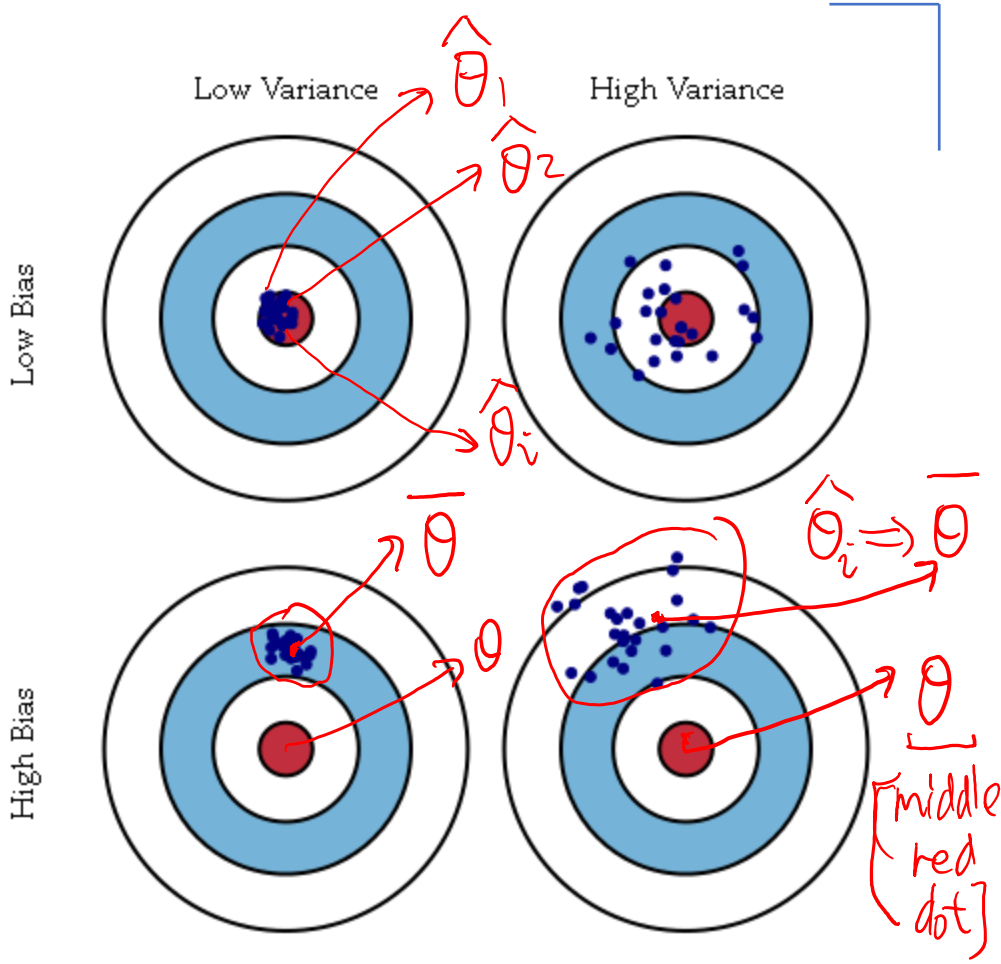
Model "bias" & Model "variance"

- Middle RED:
  - TRUE function
- Error due to bias:
  - How far off in general from the middle red
- Error due to variance:
  - How wildly the blue points spread



Model "bias" & Model "variance"

- Middle RED:
  - TRUE function
- Error due to bias:
  - How far off in general from the middle red
- Error due to variance:
  - How wildly the blue points spread



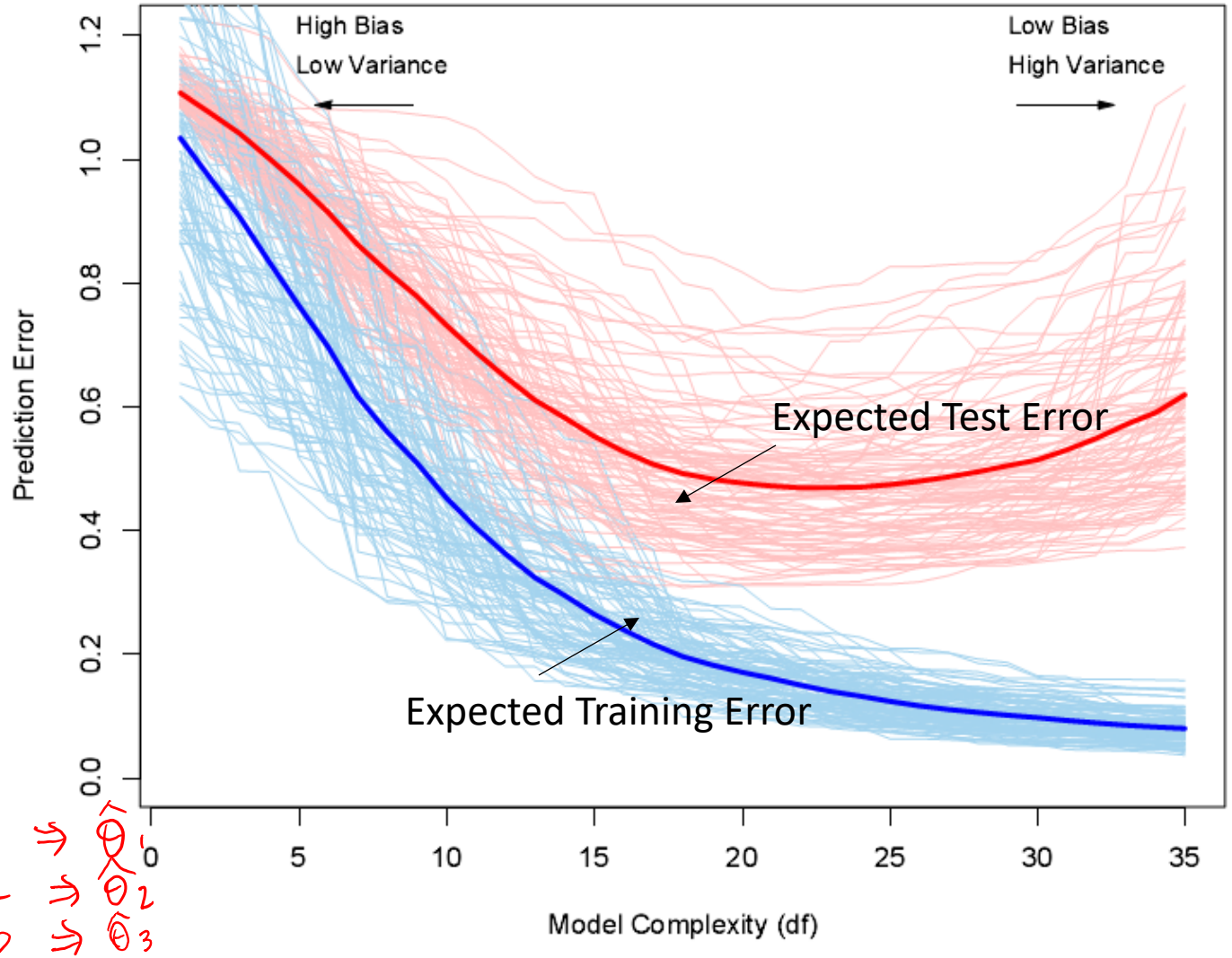
Roadmap

- Bias-variance decomposition
- Bias-Variance Tradeoff / Model Selection

(1) Randomness of Training Sets

$Pr(x, y)$

randomness of train/test set



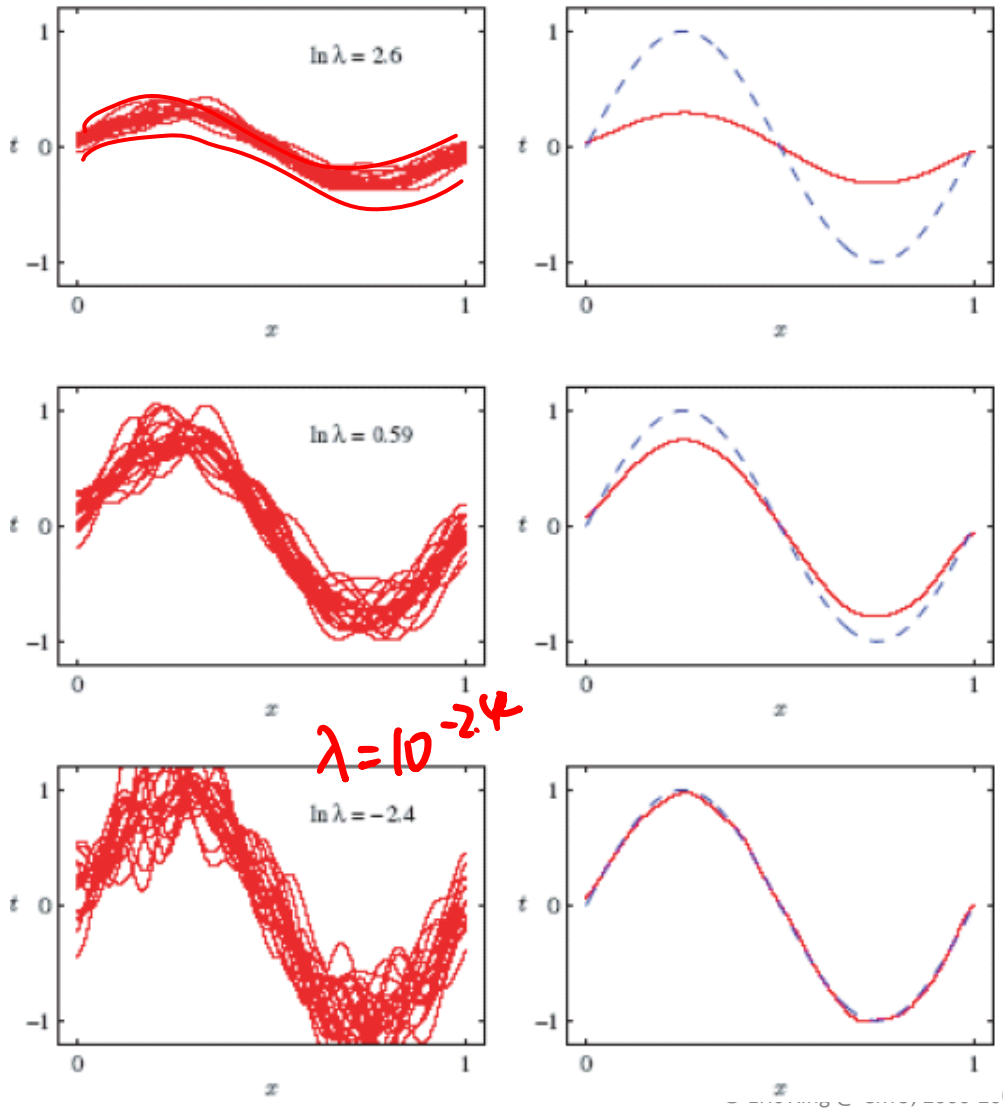
training set_random1 $\Rightarrow \hat{\theta}_1$

training set_random2 $\Rightarrow \hat{\theta}_2$

..... 3 $\Rightarrow \hat{\theta}_3$

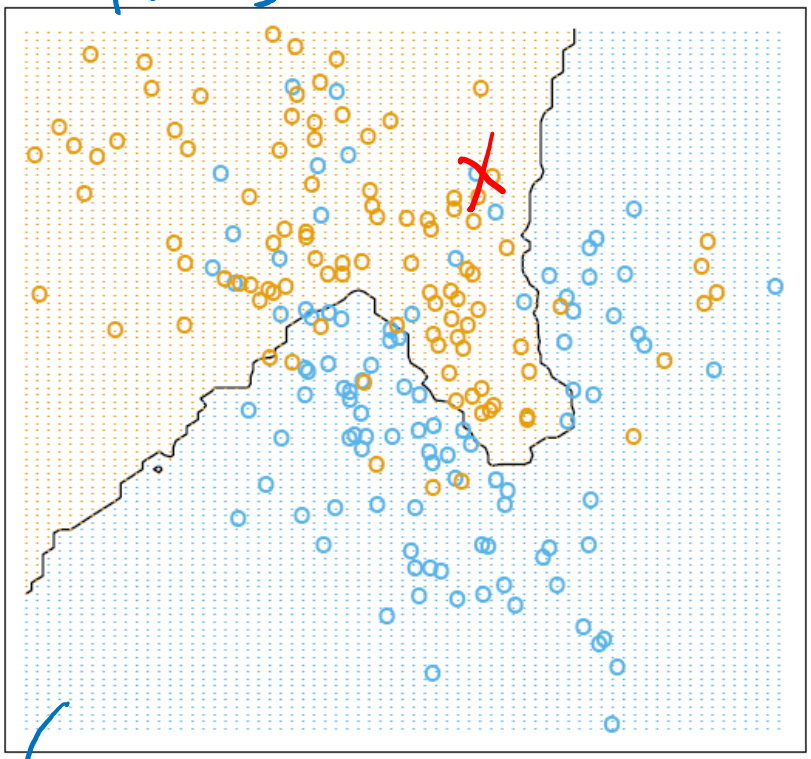
e.g. Regularized LR as an example.

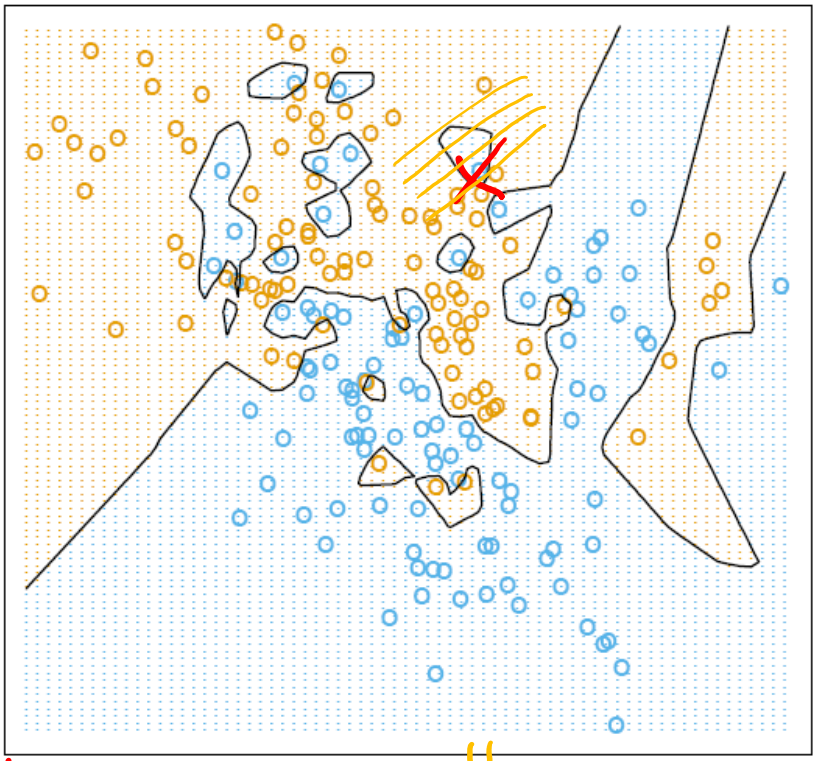
Bias-variance tradeoff



- λ is a "regularization" terms in LR, the smaller the λ, is more complex the model (why?)
  - Simple (highly regularized) models have low variance but high bias.
  - Complex models have low bias but high variance.
- You are inspecting an empirical average over 100 training set.

Randomness of Train Set
=> Variance of Models, e.g.,



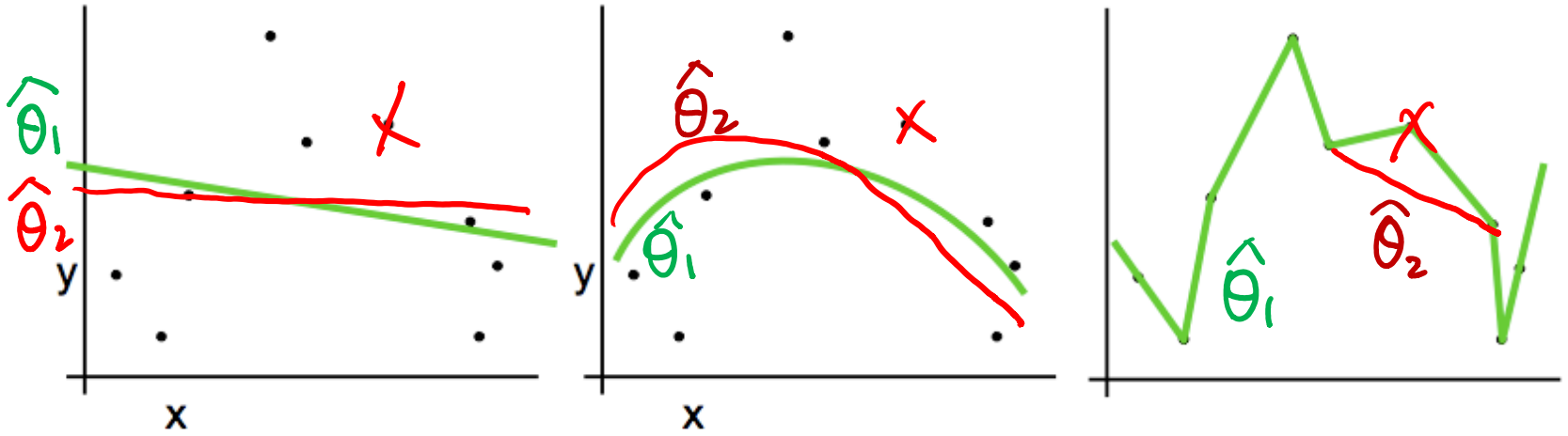


e.g. removing one train sample

[No change of decision boundary]

[decision boundary changed]

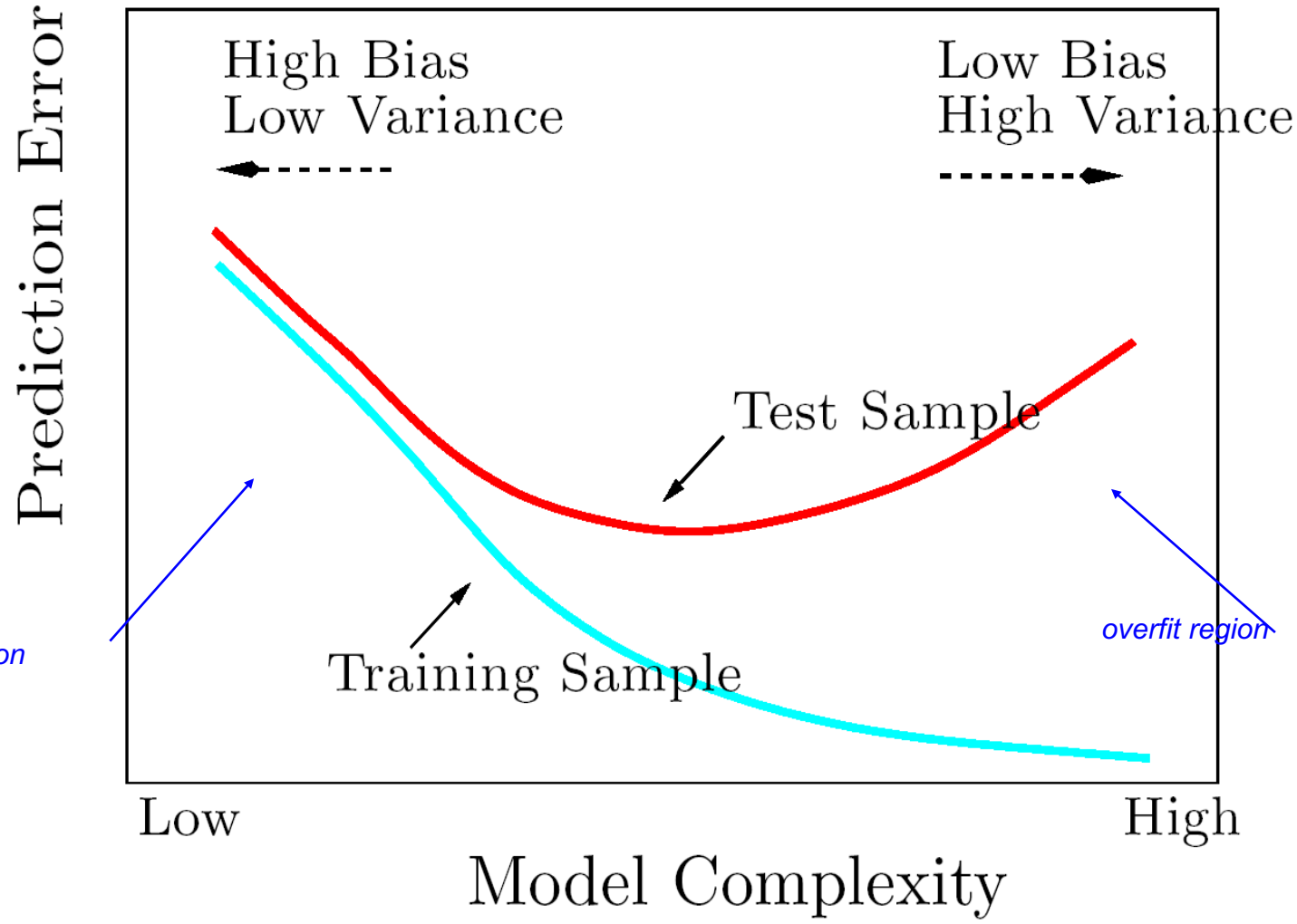
Randomness of Train Set
=> Variance of Models, e.g.,



e.g. removing one training sample

model complexity ↑ ⇒ model variance ↑

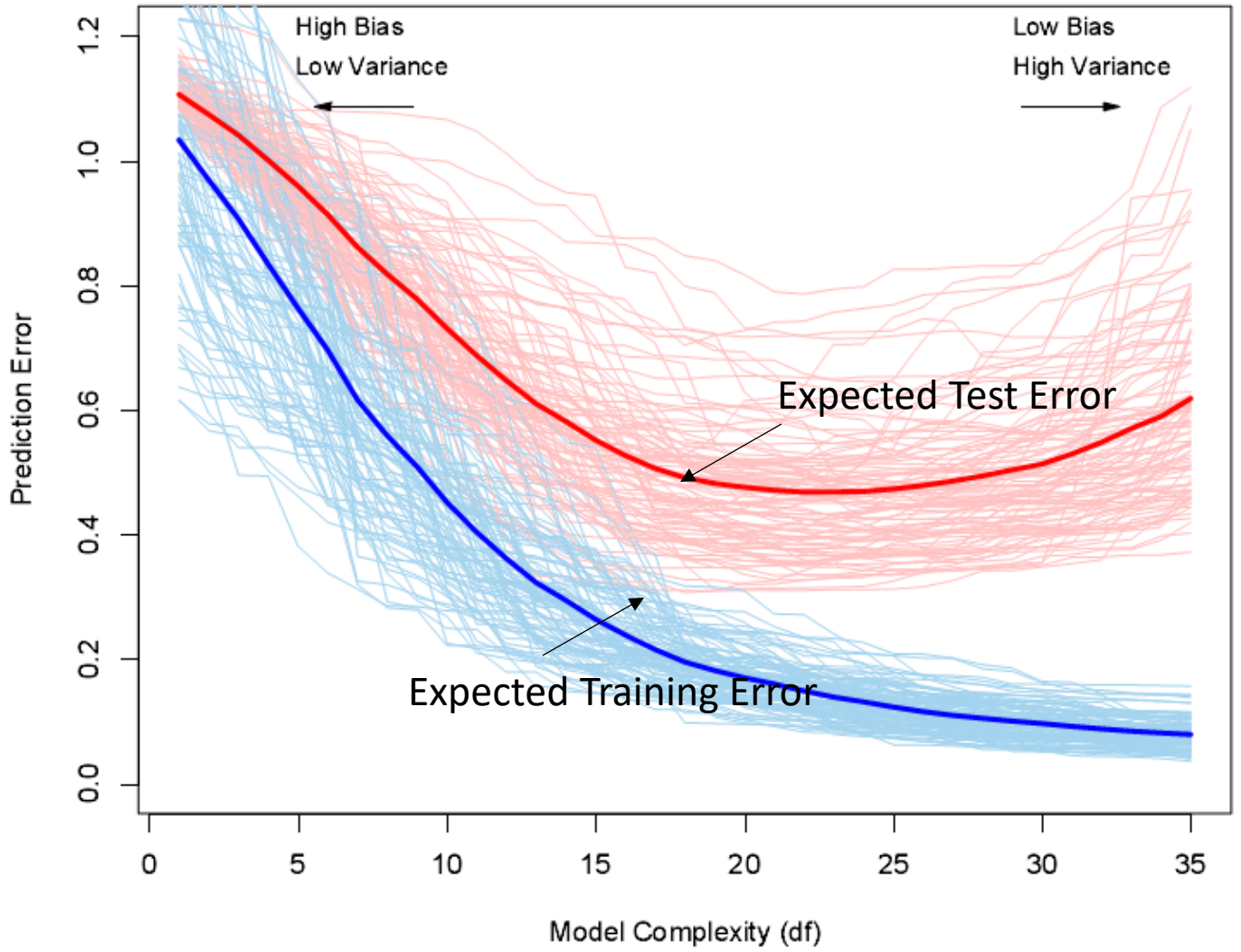
Bias-Variance Tradeoff / Model Selection



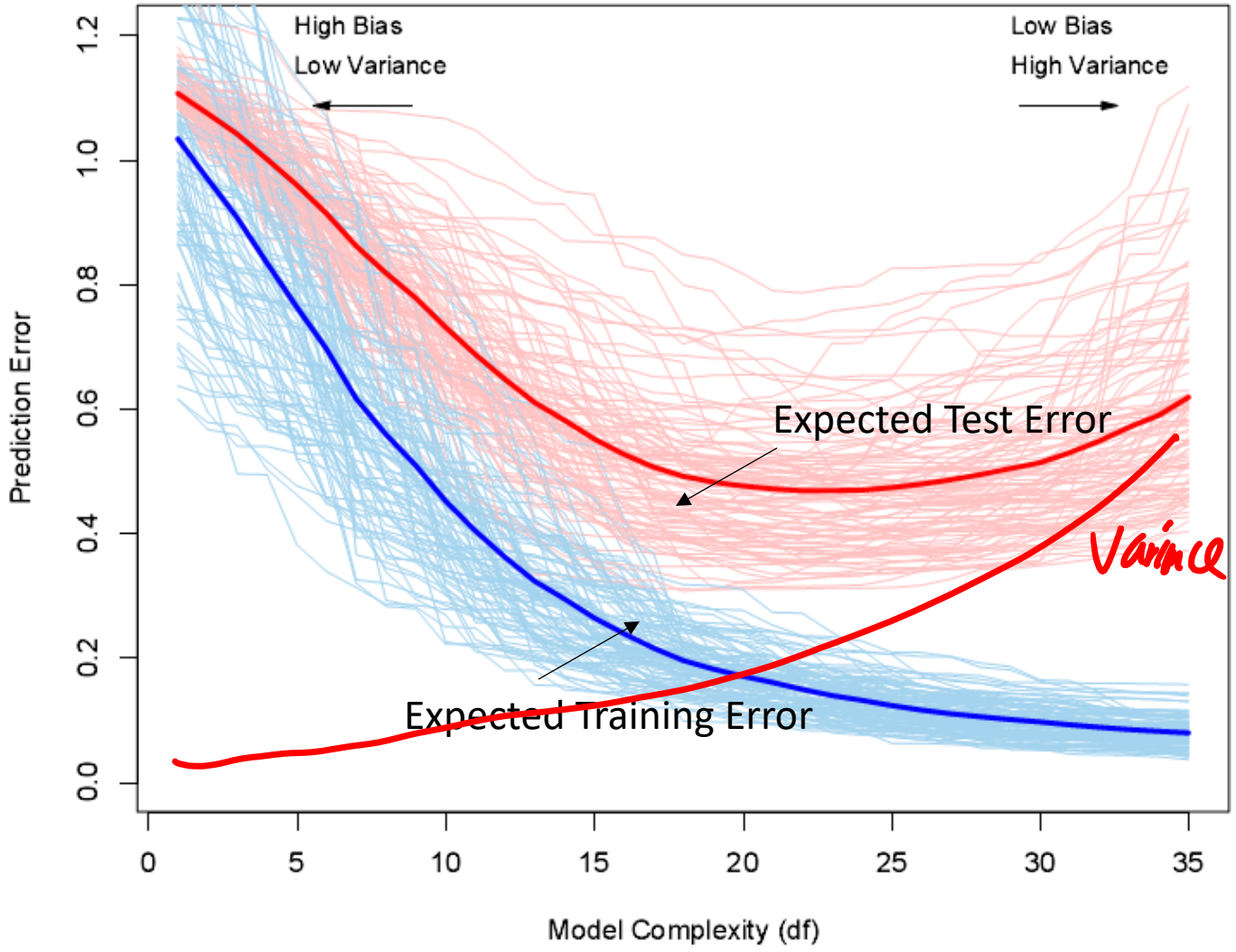
underfit region

overfit region

(2) Training error can always be reduced when increasing model complexity,



Expected Test error and CV error ➔ good approximation of of generalization



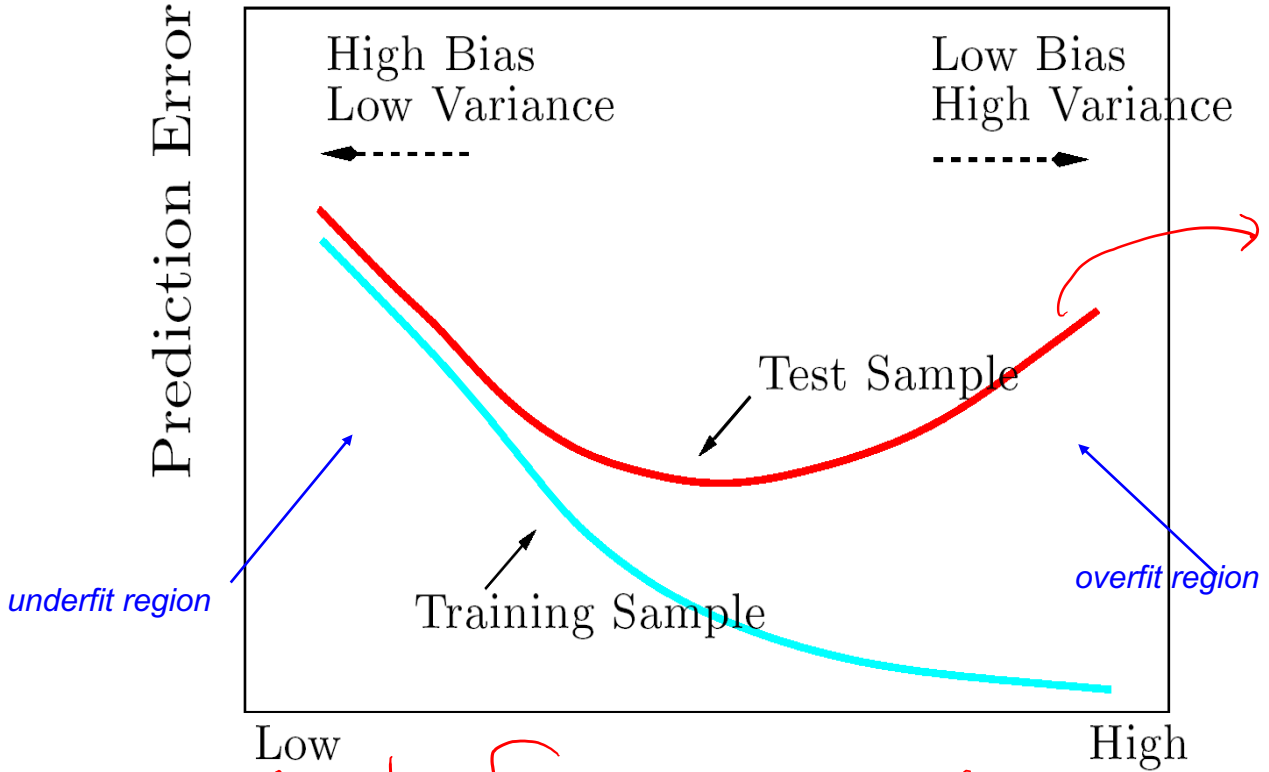
Bias-Variance Trade-off

- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).

- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample randomness).

Slide credit: D. Hoiem

Bias-Variance Tradeoff / Model Selection

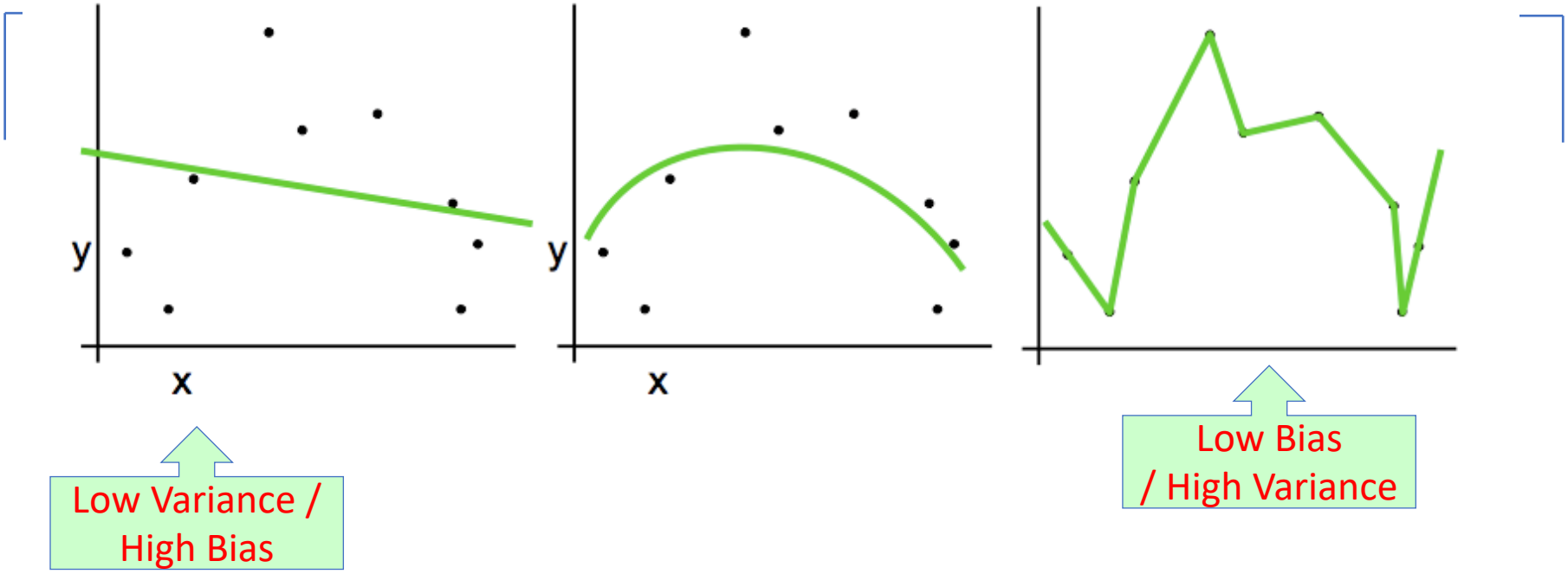


CV / test error
[Good approximation of EPE]

KNN: large k ← [Model Complexity] → small k

Regression: small d → large d

Regression: Complexity versus Goodness of Fit

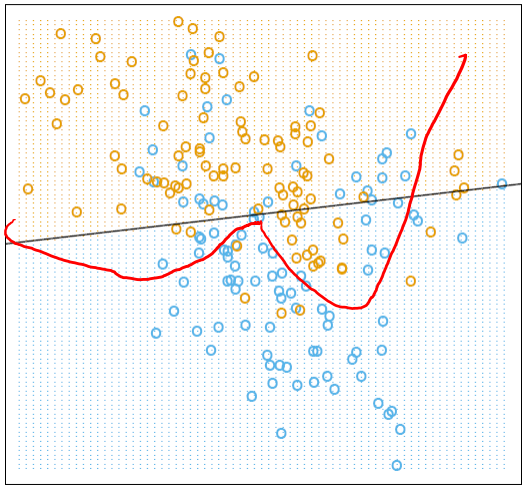


Highest Bias
Lowest variance
Model complexity = low

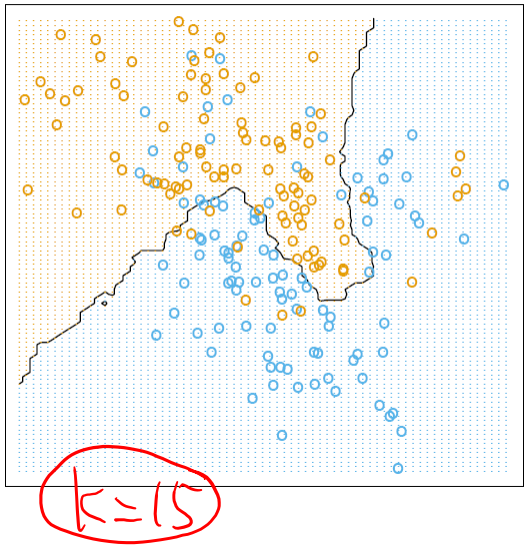
Medium Bias
Medium Variance
Model complexity = medium

Smallest Bias
Highest variance
Model complexity = high

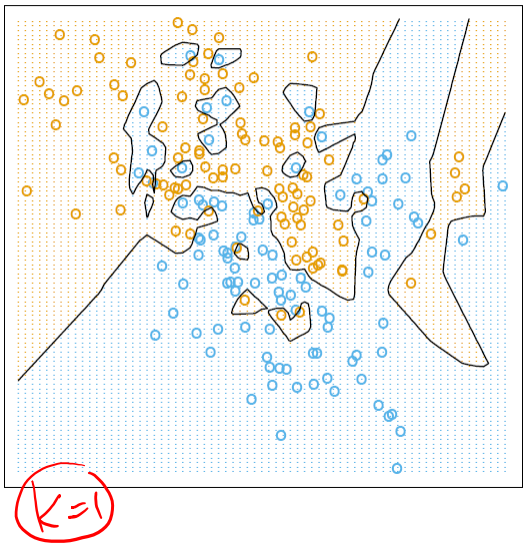
Classification, Decision boundaries in global vs. local models



Low Variance / High Bias



15-nearest neighbor



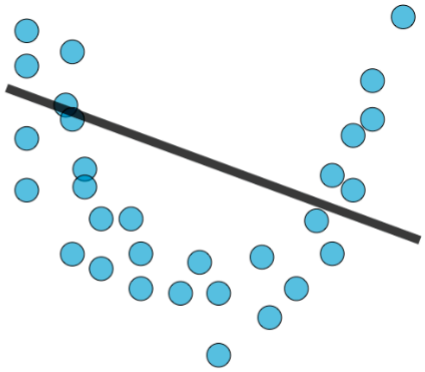
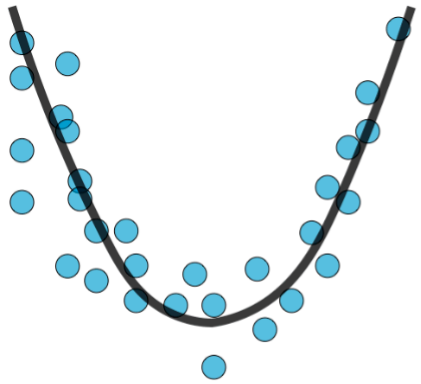
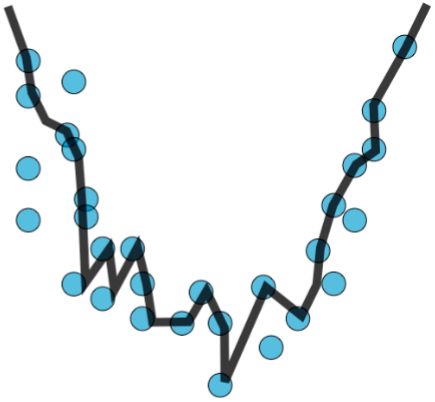
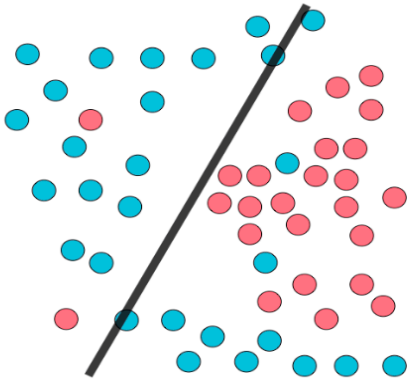
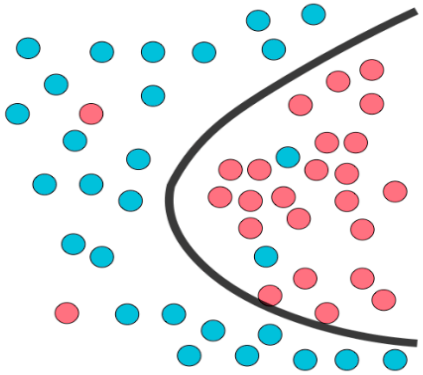
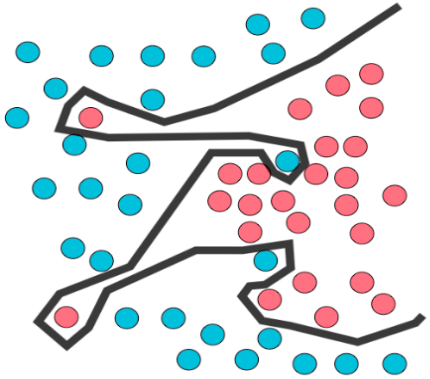
1-nearest neighbor

Low Bias / High Variance

Highest Bias
Lowest variance
Model complexity = low

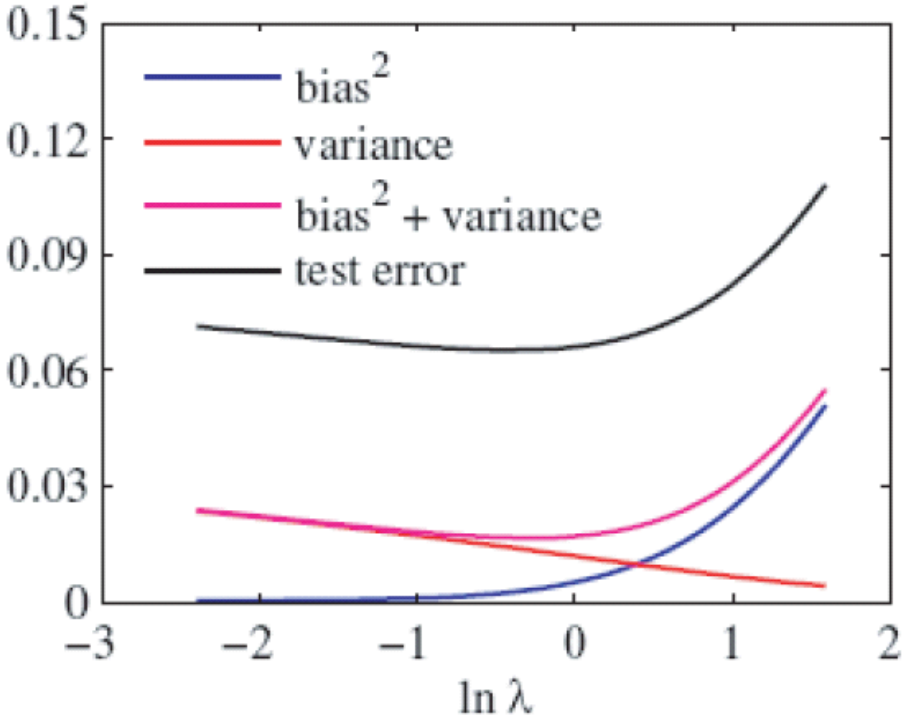
Medium Bias
Medium Variance
Model complexity = medium

Smallest Bias
Highest variance
Model complexity = high

	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			
Remedies	- Complexify model - Add more features - Train longer		- Regularize - Get more data - select feature

Credit: Stanford Machine Learning

Bias^2+variance / Model Selection?



- Bias^2+variance predicts (shape of) test error quite well.
- However, bias and variance cannot be computed since it relies on knowing the true distribution of x and y

References

- Prof. Tan, Steinbach, Kumar's "Introduction to Data Mining" slide
- Prof. Andrew Moore's slides
- Prof. Eric Xing's slides
- Hastie, Trevor, et al. *The elements of statistical learning*. Vol. 2. No. 1. New York: Springer, 2009.

Is the bias-variance trade off dependent on the number of samples? (EXTRA)

https://www.reddit.com/r/statistics/comments/6uajyr/is_the_biasvariance_trade_off_dependent_on_the/

In the usual application of linear regression, your coefficient estimators are unbiased so sample size is irrelevant. But more generally, you can have bias that *is* a function of sample size as in the case of the variance estimator obtained from applying the population variance formula to a sample (sum of squares divided by n)…..

… the bias and variance for an estimator are generally a decreasing function of n. Dealing with this is a core topic in nonparametric statistics. For nonparametric methods with tuning parameters a very standard practice is to theoretically derive rates of convergence (as sample size goes to infinity) of the bias and variance as a function of the tuning parameter, and then you find the optimal (in terms of MSE) rate of convergence of the tuning parameter by balancing the rates of the bias and variance. Then you get asymptotic results of your estimator with the tuning parameter converging at that particular rate. Ideally you also provide a data-based method of choosing the tuning parameter (since simply setting the tuning parameter to some fixed function of sample size could have poor finite sample performance), and then show that the tuning parameter chosen this way attains the optimal rate.

The battle against overfitting (Extra) :

- Cross validation
- Regularization
- Feature selection
- Model selection --- Occam's razor
- Model averaging
  - The Bayesian-frequentist debate
  - Bayesian learning (weight models by their posterior probabilities)

For instance, if trying to solve "spam detection" using (Extra)

L2 - logistic regression, implemented with gradient descent.

Fixes to try: If performance is not as desired

– Try getting more training examples.	Fixes high variance.
– Try a smaller set of features.	Fixes high variance.
– Try a larger set of features.	Fixes high bias.
– Try email header features.	Fixes high bias.
– Run gradient descent for more iterations.	Fixes optimization algorithm.
– Try Newton's method.	Fixes optimization algorithm.
– Use a different value for λ.	Fixes optimization objective.
– Try using an SVM.	Fixes optimization objective.

Slide credit: A. Ng

Expected prediction error (EPE)

Consider joint distribution

$$\mathrm{EPE}(f) = \mathrm{E}(L(Y, f(X))) = \int L(y, f(x)) \Pr(dx, dy)$$

- For L2 loss:

$$\text{e.g.} = \int (y - f(x))^2 \Pr(dx, dy)$$

under L2 loss, best estimator for EPE (Theoretically) is :

Conditional mean

$$\hat{f}(x) = \mathrm{E}(Y \mid X = x)$$

e.g. KNN

NN methods are the direct implementation (approximation)

kNN for minimizing EPE

- We know under L2 loss, best estimator for minimize EPE (theoretically) is :

Conditional mean $f(x) = \mathrm{E}(Y \mid X = x)$

- Nearest neighbours assumes that f(x) is well approximated by a locally constant function.

Minimize EPE using L2

- Expected prediction error (EPE) for L2 Loss:

$$\mathrm{EPE}(f) = \mathrm{E}(Y - f(X))^2 = \int (y - f(x))^2 \Pr(dx, dy)$$

- Since Pr(X,Y)=Pr(Y |X)Pr(X), EPE can also be written as

$$\mathrm{EPE}(f) = \mathrm{E}_X \mathrm{E}_{Y|X}([Y - f(X)]^2 \mid X)$$

- Thus it suffices to minimize EPE pointwise

Best estimator under L2 loss: conditional expectation

$$f(x) = \arg\min_c \mathrm{E}_{Y|X}([Y - c]^2 \mid X = x)$$

Conditional mean

Solution for Regression:

Solution for kNN:

$$f(x) = \mathrm{E}(Y \mid X = x)$$

Minimize EPE using L2 (another proof)

- Let *t* be the true (target) output and $y(x)$ be our estimate. The expected squared loss is

$$E(L) = \iint L(t, y(x))\, p(x,t)\, dxdt$$

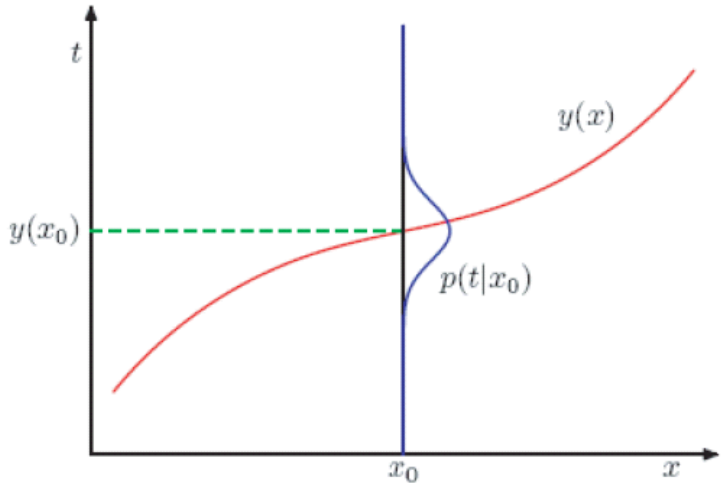
$$= \iint (t - y(x))^2\, p(x,t)\, dxdt$$

- Out goal is to choose $y(x)$ that minimize $E(L)$:
  - Calculus of variations:

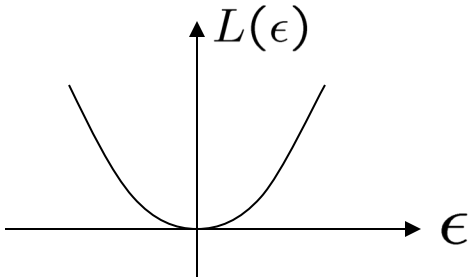
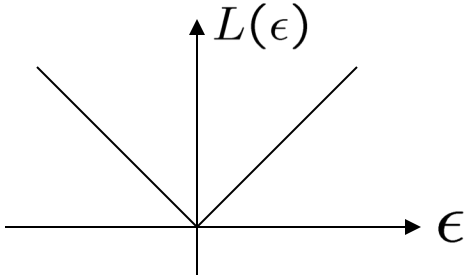
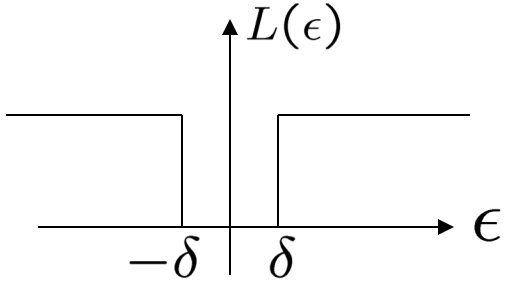
$$\frac{\partial E(L)}{\partial y(x)} = 2\int (t - y(x))\, p(x,t)\, dt = 0$$

$$\int y(x)\, p(x,t)\, dt = \int t\, p(x,t)\, dt$$

$$y^*(x) = \int \frac{t\, p(x,t)}{p(x)}\, dt = \int t\, p(t \mid x)\, dt = E_{t|x}[t] = E[t \mid x]$$



Review : EPE with different loss

Loss Function	Estimator $\hat{f}(x)$
L_2 	$\hat{f}(x) = E[Y\|X = x]$
L_1 	$\hat{f}(x) = \text{median}(Y\|X = x)$
0-1 	$\hat{f}(x) = \arg\max_Y P(Y\|X = x)$ (Bayes classifier / MAP)

Expected prediction error (EPE)

Consider joint distribution

$$\mathrm{EPE}(f) = \mathrm{E}(L(Y, f(X))) = \int L(y, f(x)) \Pr(dx, dy)$$

For 0-1 loss: $L(k, \ell) = 1 - d_{kl}$

Bayes Classifier

$$\hat{f}(X) = C_k \text{ if}$$
$$\Pr(C_k \mid X = x) = \max_{g \in C} \Pr(g \mid X = x)$$

Bayesian and Frequentist (Extra)

- Frequentist interpretation of probability
  - Probabilities are objective properties of the real world, and refer to limiting relative frequencies (e.g., number of times I have observed heads). Hence one cannot write $P(\text{Katrina could have been prevented}|D)$, since the event will never repeat.
  - Parameters of models are *fixed, unknown constants*. Hence one cannot write $P(\vartheta|D)$ since ϑ does not have a probability distribution. Instead one can only write $P(D|\vartheta)$.
  - One computes point estimates of parameters using various *estimators*, $\vartheta^* = f(D)$, which are designed to have various desirable qualities when *averaged over future data D* (assumed to be drawn from the "true" distribution).
- Bayesian interpretation of probability
  - Probability describes degrees of belief, not limiting frequencies.
  - Parameters of models are *hidden variables*, so one can compute $P(\vartheta|D)$ or $P(f(\vartheta)|D)$ for some function f.
  - One estimates parameters by computing $P(\vartheta|D)$ using Bayes rule:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

Cross Validation and Variance Estimation

- Cross-validation (CV) is quite a general tool for estimating the expected test error (1), that makes minimal assumptions—i.e., it doesn't assume that $Y = f(X) + \varepsilon$ with ε independent of X, it doesn't assume that the training inputs $x_1, \ldots x_n$ are fixed, all it really assumes is that the training samples $(x_1, y_1), \ldots (x_n, y_n)$ are i.i.d.

  We split up our training set into K divisions or folds, for some number K; usually this is done randomly. Write these as $F_1, \ldots F_K$, so $F_1 \cup \ldots \cup F_K = \{1, \ldots n\}$. Now for each $k = 1, \ldots K$, we fit our prediction function on all points but those in the kth fold, denoted $\hat{f}^{-(k)}$, and evaluate squared errors on the points in the kth fold,

$$\mathrm{CV}_k(\hat{f}^{-(k)}) = \frac{1}{n_k} \sum_{i \in F_k} \left(y_i - \hat{f}^{-(k)}(x_i)\right)^2.$$

http://www.stat.cmu.edu/~ryantibs/statml/review/modelbasics.pdf

http://www.stat.cmu.edu/~ryantibs/statml/review/modelbasics.pdf

- What is the difference between choosing say $K = 5$ (a common choice) versus $K = n$?
  - When $K = 5$, the function $\hat{f}^{-(k)}$ in each fold k is fit on about $4/5 \cdot n$ samples, and so we are looking at the errors incurred by a procedure that is trained on less data than the full $\hat{f}$ in (1). Therefore the mean of the CV estimate (7) could be off. When $K = n$, this is not really an issue, since each $f^{-(k)}$ is trained on $n - 1$ samples
  - When $K = n$, the CV estimate (7) is an average of n extremely correlated quantities; this is because each $\hat{f}^{-(k)}$ and $\hat{f}^{-(\ell)}$ are fit on $n - 2$ common training points. Hence the CV estimate will likely have very high variance. When $K = 5$, the CV estimate will have lower variance, since it the average of quantities that are less correlated, as the fits $\hat{f}^{-(k)}$, $k = 1, \ldots 5$ do not share as much overlapping training data

  This is tradeoff (the bias-variance tradeoff, in fact!). Usually, a choice like $K = 5$ or $K = 10$ is more common in practice than $K = n$, but this is probably an issue of debate

- For K-fold CV, it's can be helpful to assign a notion of variability to the CV error estimate. We argue that

$$\mathrm{Var}\big(\mathrm{CV}(\hat{f})\big) = \mathrm{Var}\Big(\frac{1}{K}\sum_{k=1}^{K}\mathrm{CV}_k(\hat{f}^{-(k)})\Big) \approx \frac{1}{K}\mathrm{Var}(\mathrm{CV}_1(\hat{f}^{-(1)})). \tag{8}$$

  Why is this an approximation? This would hold exactly if $\mathrm{CV}_1(\hat{f}^{-(1)}), \ldots \mathrm{CV}_K(\hat{f}^{-(K)})$ were i.i.d., but they're not. This approximation is valid for small K (e.g., $K = 5$ or 10) but not really for big K (e.g., $K = n$), because then the quantities $\mathrm{CV}_1(\hat{f}^{-(1)}), \ldots \mathrm{CV}_K(\hat{f}^{-(K)})$ are highly correlated

Practical issues for CV

- How to decide the values for K *in* KCV and α =1/K
  - Commonly used K = 10 and α = 0.1.
  - when data sets are small relative to the number of models that are being evaluated, we need to decrease α and increase K
  - K needs to be large for the variance to be small enough, but this makes it time-consuming.
- Bias-variance trade-off
  - Small α usually lead to low bias. In principle, *LOOCV* provides an almost unbiased estimate of the generalization ability of a classifier, especially when the number of the available training samples is severely limited; but it can also have high variance.
  - Large α can reduce variance, but will lead to under-use of data, and causing high-bias.
- One important point is that the test data D_{test} is never used in CV, because doing so would result in overly (indeed dishonest) optimistic accuracy rates during the testing phase.