UVA CS 4501: Machine Learning

Lecture 10: K-nearest-neighbor Classifier / Bias-Variance Tradeoff

Dr. Yanjun Qi

University of Virginia

Department of
Computer Science

Where are we ? → Five major sections of this course

- ❑ Regression (supervised)
- ❑ Classification (supervised)
- ❑ Unsupervised models
- ❑ Learning theory
- ❑ Graphical models

Three major sections for classification

- We can divide the large variety of classification approaches into roughly three major types

1. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., support vector machine, decision tree, logistic regression,
   - e.g. neural networks (NN), deep NN

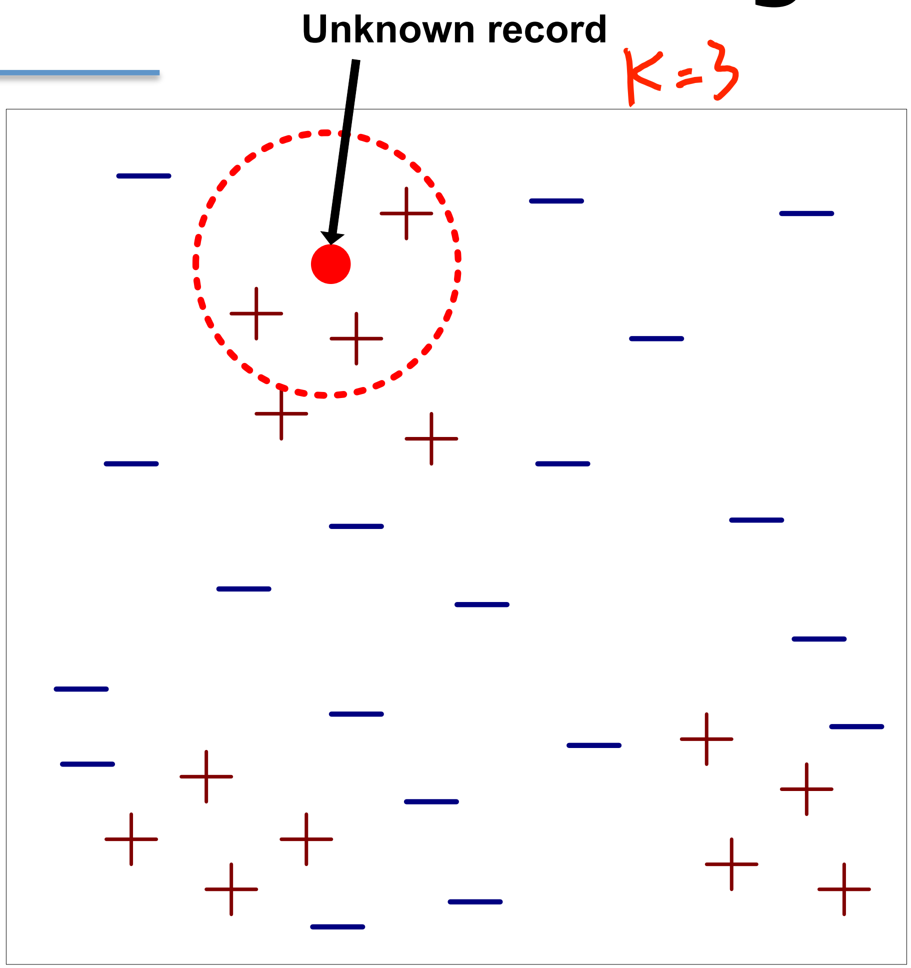
2. Generative:
   - build a generative statistical model
   - e.g., Bayesian networks, Naïve Bayes classifier

3. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

Today :

- ✓ K-nearest neighbor
- ✓ Model Selection / Bias Variance Tradeoff
  - ✓ Bias-Variance tradeoff
  - ✓ High bias ? High variance ? How to respond ?

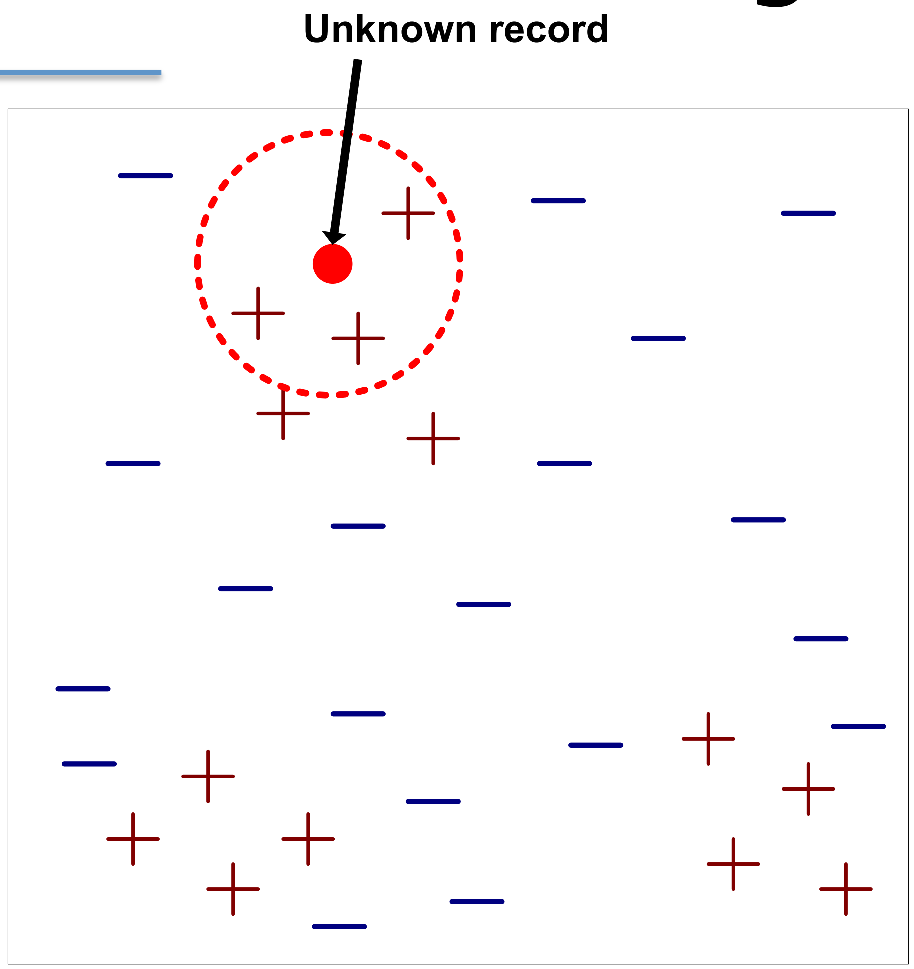
Nearest neighbor classifiers



Requires three inputs:

1. The set of stored training samples
2. Distance metric to compute distance between samples
3. The value of *k*, i.e., the number of nearest neighbors to retrieve

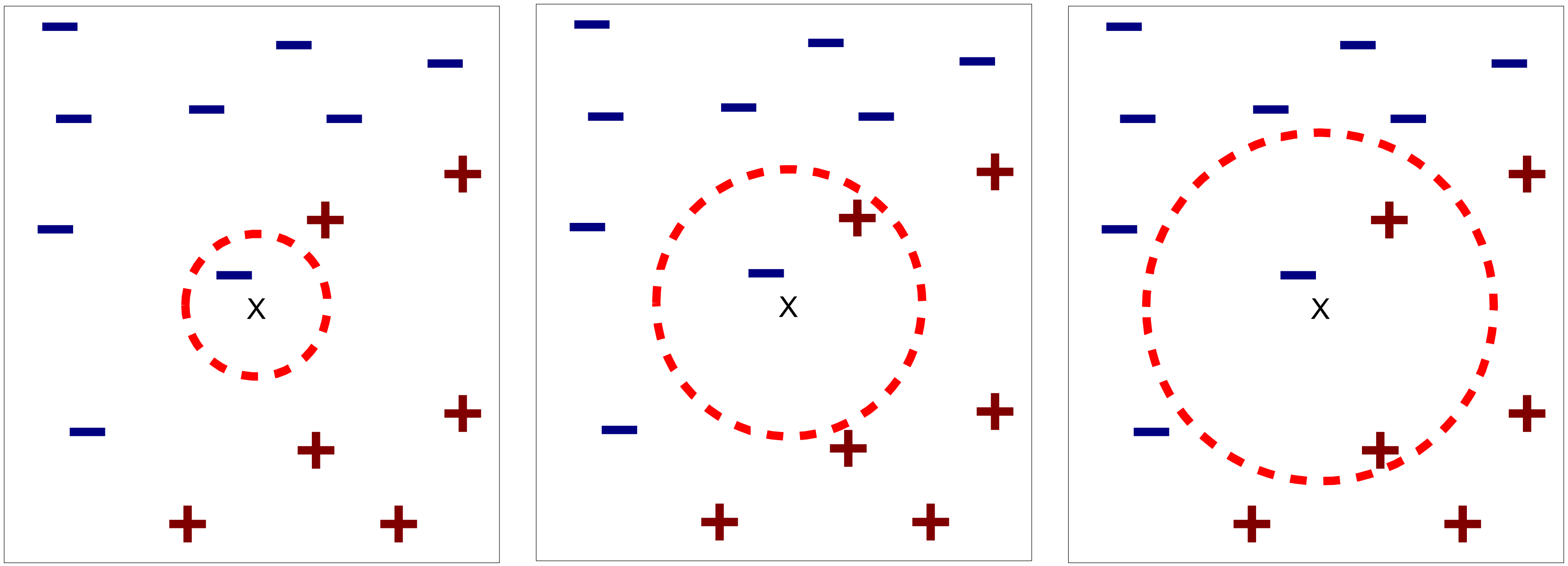
Nearest neighbor classifiers



To classify unknown sample:

1. Compute distance to training records
2. Identify *k* nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of nearest neighbor



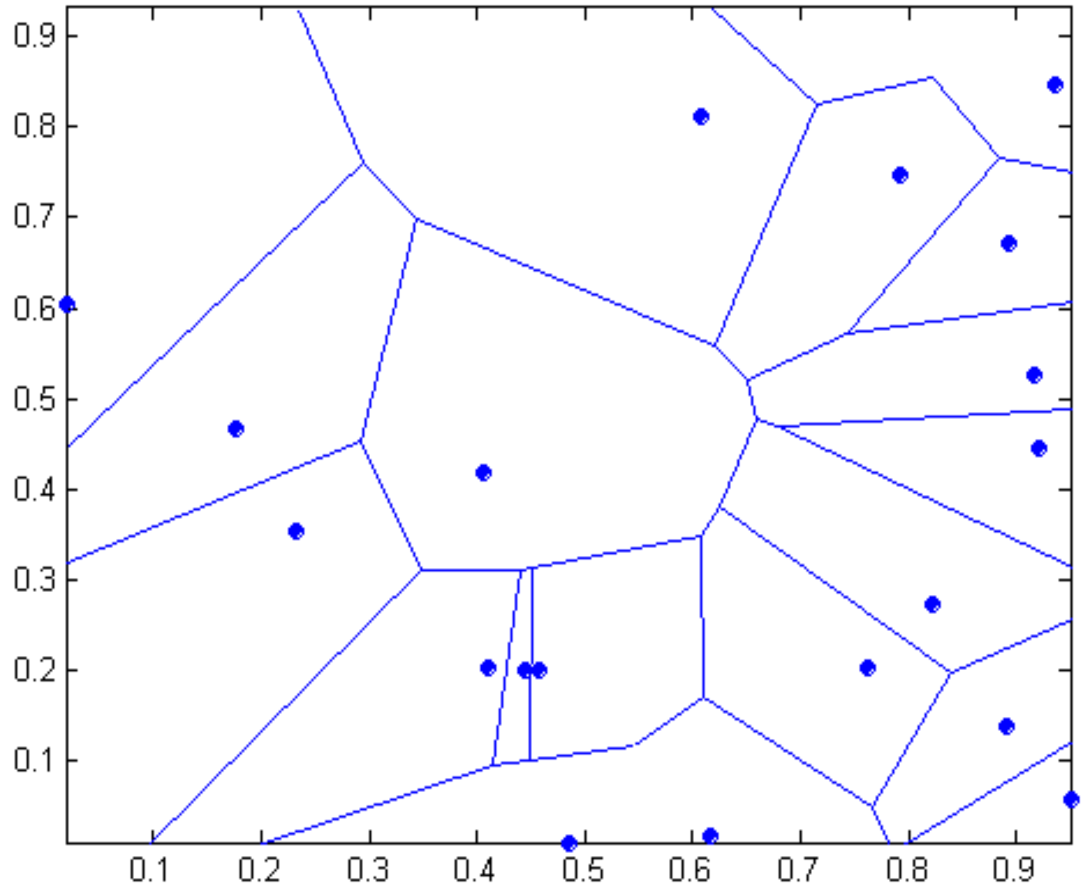
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

k-nearest neighbors of a sample x are datapoints that have the *k* smallest distances to x

1-nearest neighbor

Voronoi diagram:

partitioning of a plane into regions based on distance to points in a specific subset of the plane.



Nearest neighbor classification

- Compute distance between two points:
  - For instance, Euclidean distance

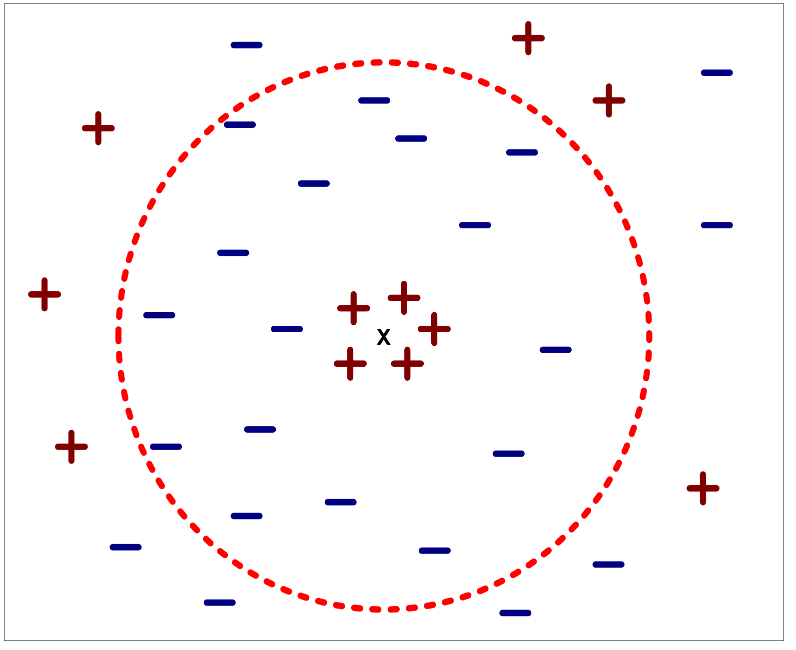
e.g. cosine distance for text

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

- Options for determining the class from nearest neighbor list
  - Take majority vote of class labels among the *k*-nearest neighbors
  - Weight the votes according to distance
    - example: weight factor $w = 1 / d^2$

Nearest neighbor classification

- Choosing the value of *k*:
  - If *k* is too small, sensitive to noise points
  - If *k* is too large, neighborhood may include (many) points from other classes

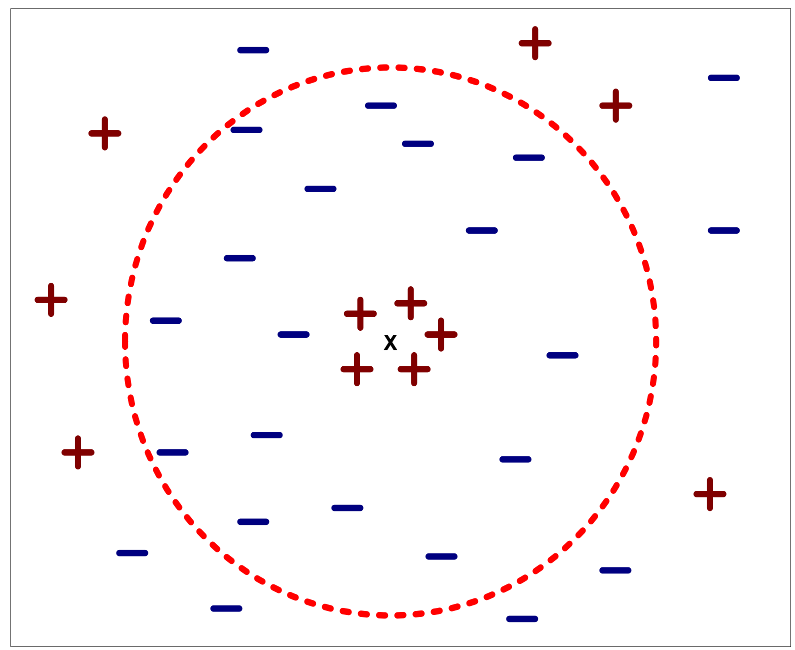


Nearest neighbor classification

- Choosing the value of *k*:
  - If *k* is too small, sensitive to noise points
  - If *k* is too large, neighborhood may include points from other classes

K ↓ flexible / varies a lot

K ↑ Smooth / varies little



Nearest neighbor classification

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5 m to 1.8 m
    - weight of a person may vary from 90 lb to 300 lb
    - income of a person may vary from $10K to $1M

Nearest neighbor classification

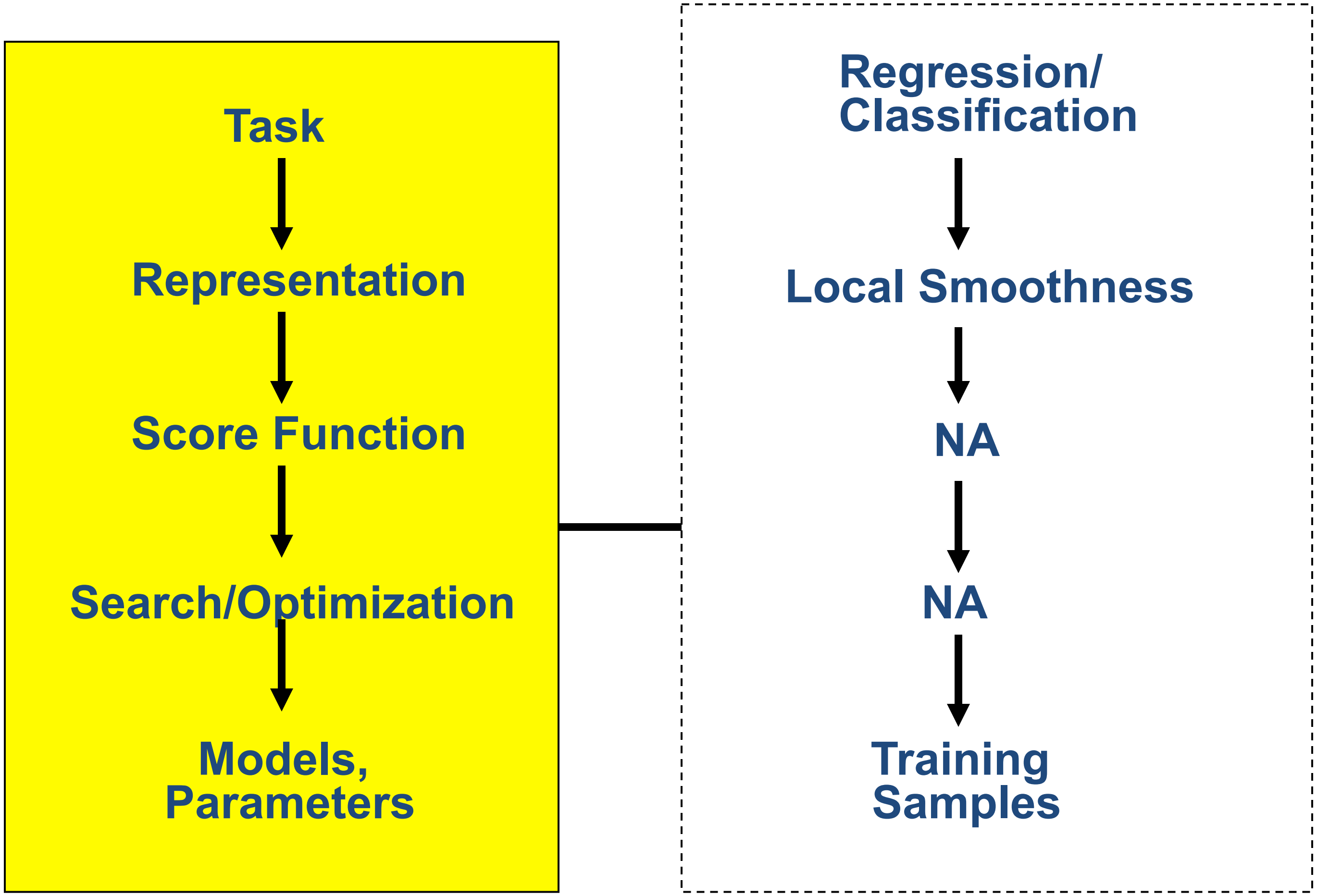
- *k*-Nearest neighbor classifier is a lazy learner
  - Does not build model explicitly.
  - Classifying unknown samples is relatively expensive.

- *k*-Nearest neighbor classifier is a local model, vs. global model like linear classifiers.

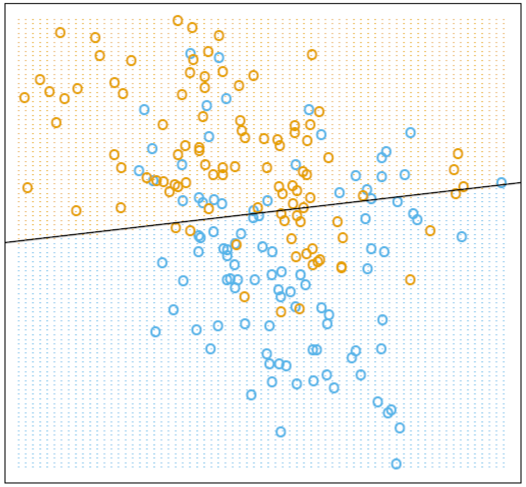
Computational Time Cost

	Train (n)	Test (m=1)
Linear Regression	$O(np^2+p^3)$	$O(p)$
KNN	$O(1)$	$O(np)+$ $O(\text{sort } n\text{-}K)$???

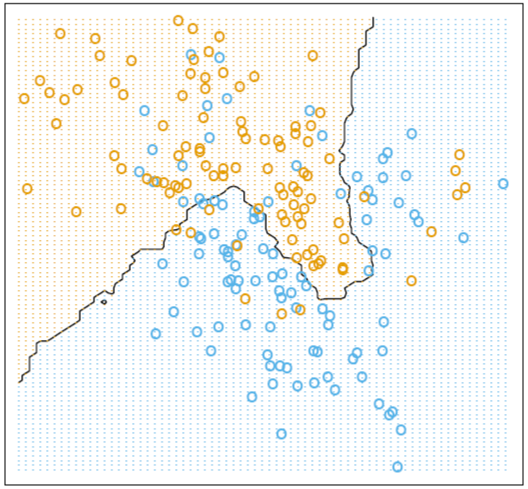
K-Nearest Neighbor



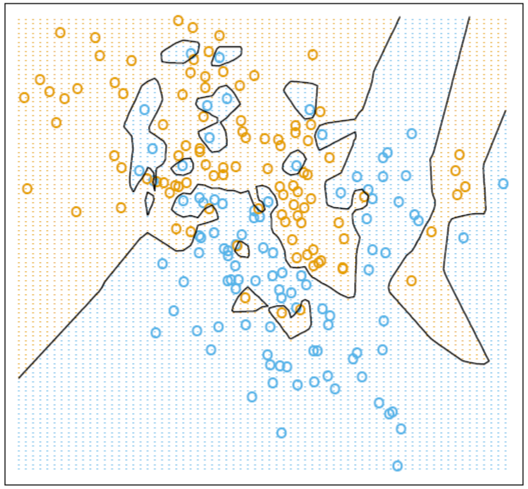
Decision boundaries in global vs. local models



K=15



K=1



Linear classification

- global
- stable
- can be inaccurate

15-nearest neighbor

1-nearest neighbor

- local
- accurate
- unstable

- K acts as a smoother

What ultimately matters: ***GENERALIZATION***

Today :

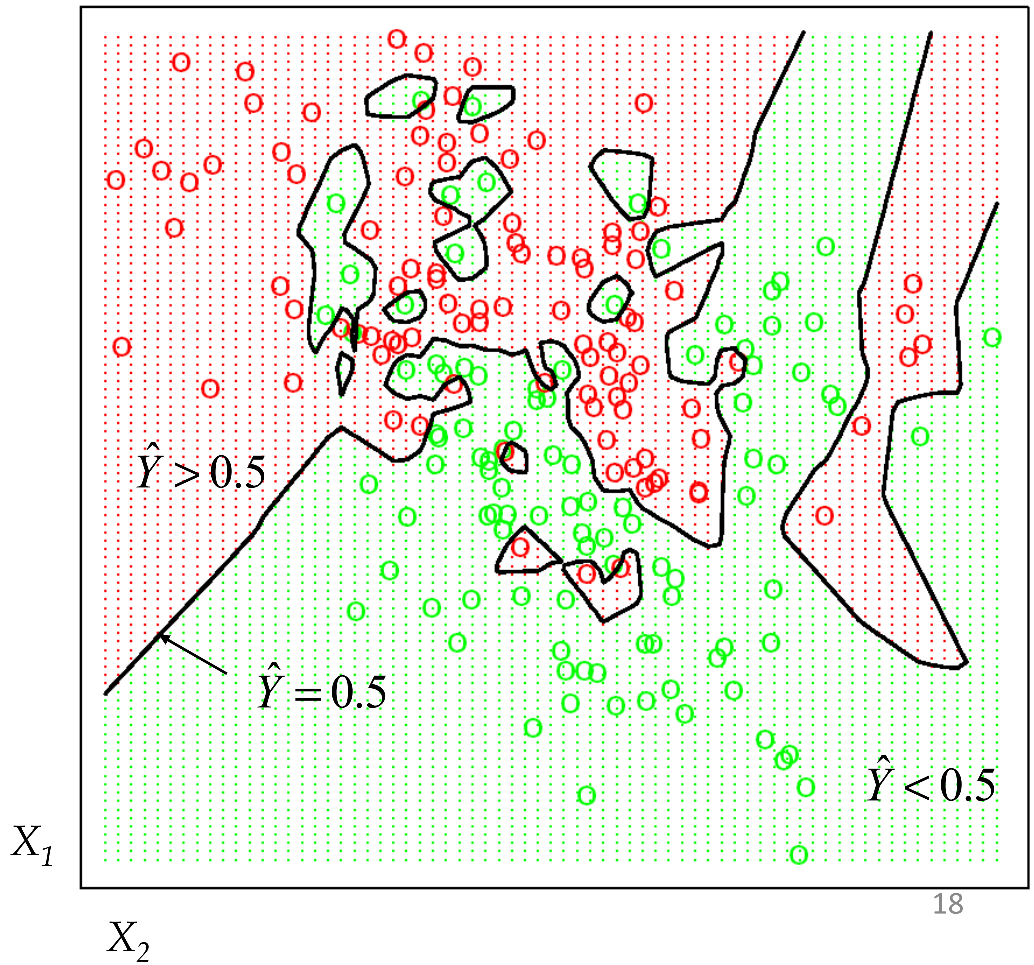
- ✓ K-nearest neighbor
- ✓ Model Selection / Bias Variance Tradeoff
  - ✓ Bias-Variance tradeoff
  - ✓ High bias ? High variance ? How to respond ?

e.g. Training Error from KNN, Lesson Learned

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

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

1-nearest neighbor averaging



Review: Mean and Variance of Random Variable (RV)

- Mean (Expectation): $\mu = E(\mathrm{X})$
  - Discrete RVs:

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

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

Adapt From Carols' prob tutorial

Review: Mean and Variance of Random Variable (RV)

- Mean (Expectation): $\mu = E(\mathrm{X})$
  - Discrete RVs: $E(\mathrm{X}) = \sum_{v_i} v_i \mathrm{P}(\mathrm{X} = v_i)$

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

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

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

Adapt From Carols' prob tutorial

Review: Mean and Variance of RV

- Variance: $Var(X) = E((X-\mu)^2)$
  - 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

BIAS AND VARIANCE TRADE-OFF

- 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

BIAS AND VARIANCE TRADE-OFF for Mean Squared Error of parameter estimation

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

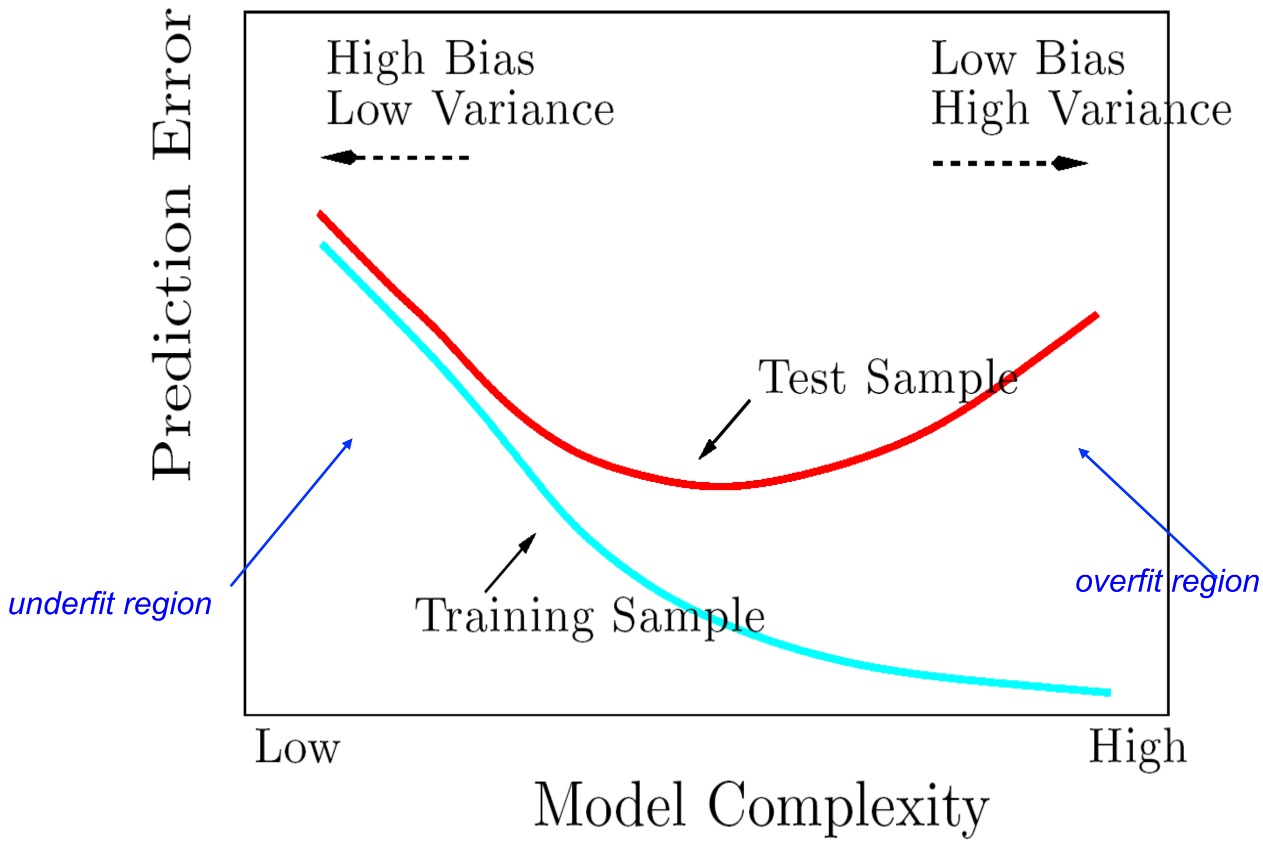
E($\hat{\theta}$)= $\bar{\theta}$

Error due to variance of training samples

Error due to incorrect assumptions

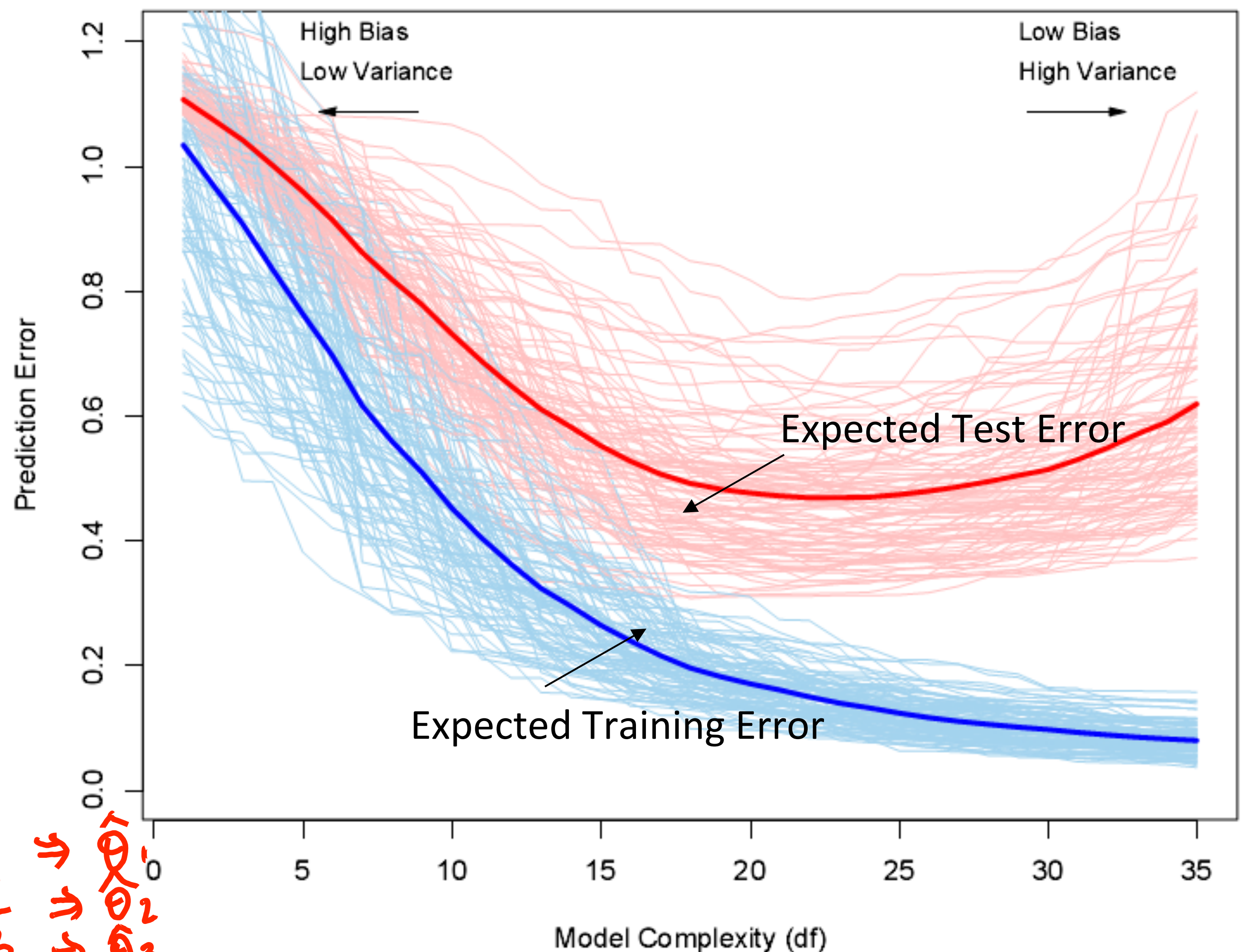
$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] = Bias^2(\hat{\theta}) + Var(\hat{\theta})$$

Bias-Variance Tradeoff / Model Selection



(1) Training vs Test Error

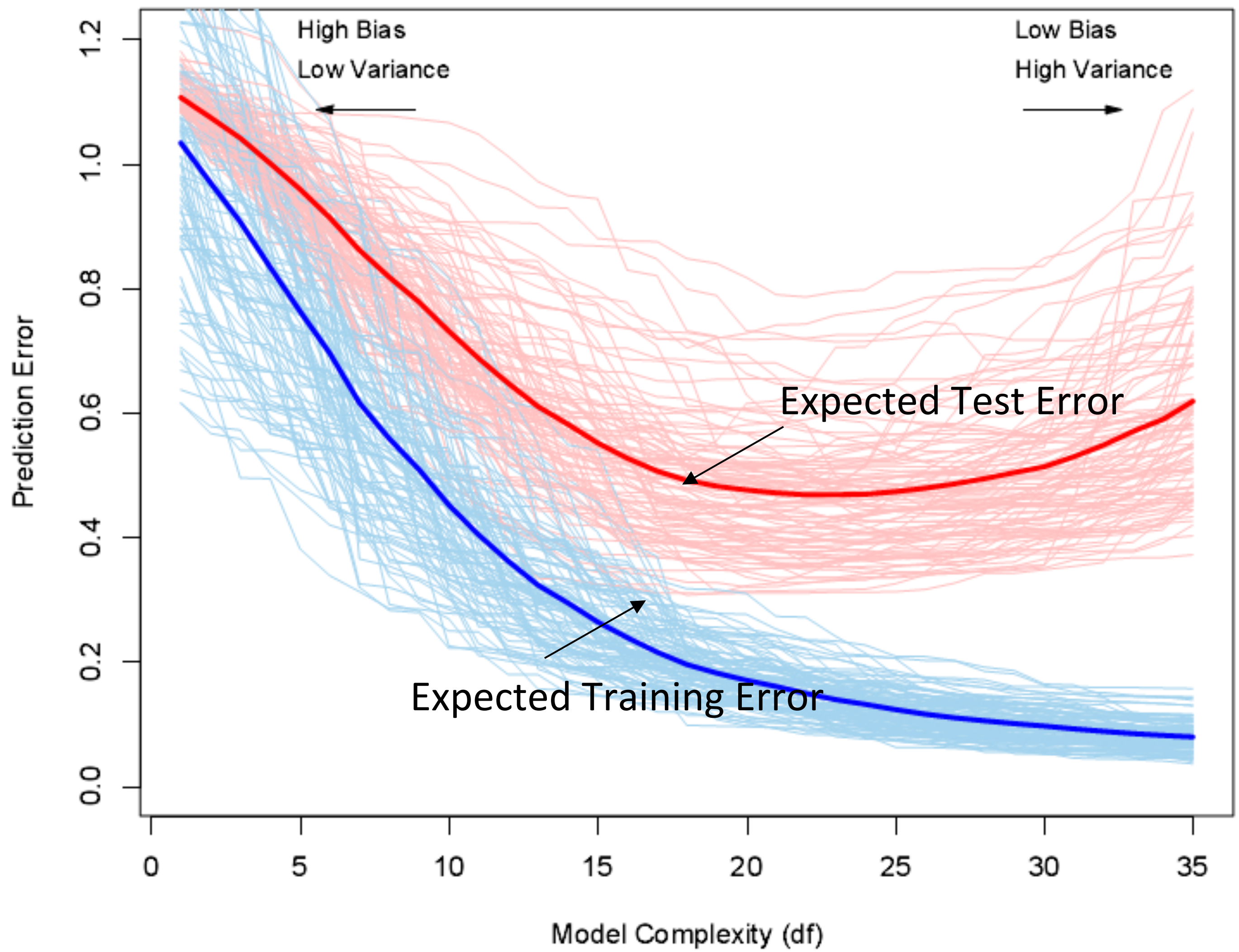
randomness of train / test set



training set _ random 1 → $\hat{\theta}_1$
training set - random 2 → $\hat{\theta}_2$
...... 3 → $\hat{\theta}_3$

(1) Training vs Test Error

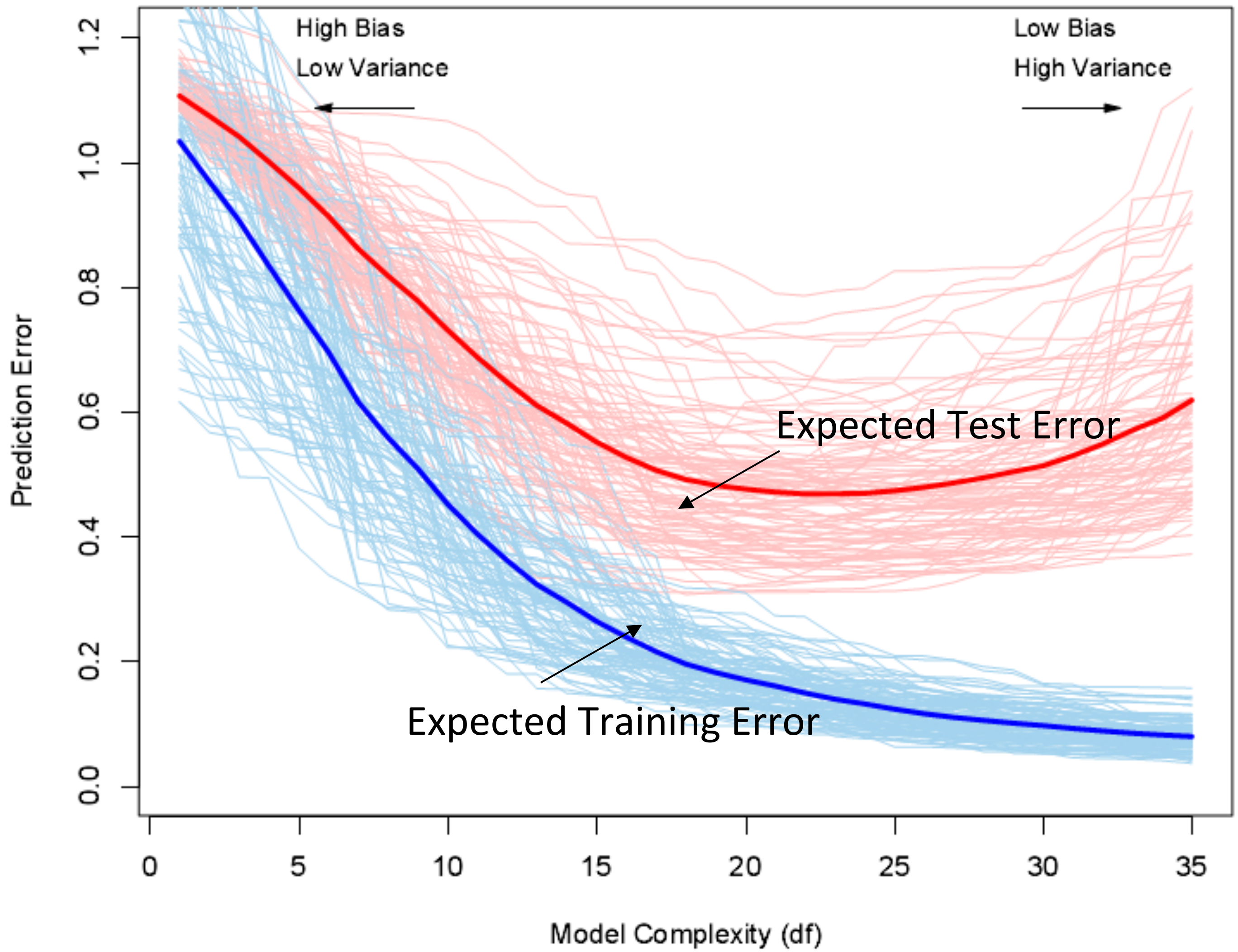
- Training error can always be reduced when increasing model complexity,



(1) Training vs Test Error

- Training error can always be reduced when increasing mod complexity,

- Expected Test error and CV error ➔ good approximation of EPE



Statistical Decision Theory (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

(2) Bias-Variance Trade-off

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

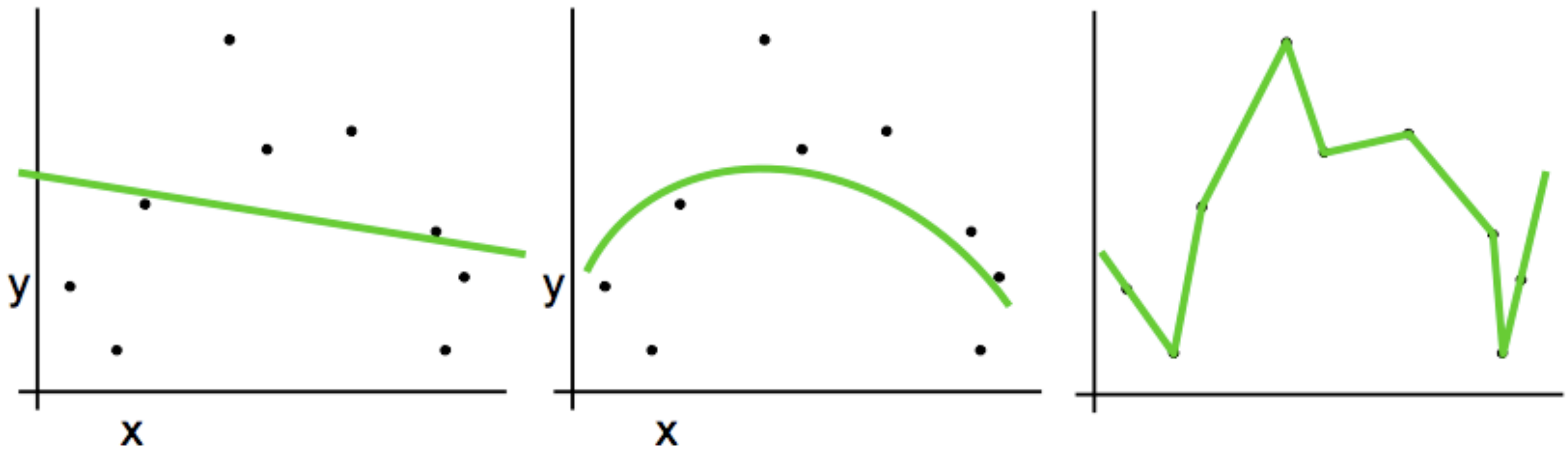
poly regression: d small
KNN: K large

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

poly regression: d large
KNN: K small

Slide credit: D. Hoiem

(2.1) Regression: Complexity versus Goodness of Fit

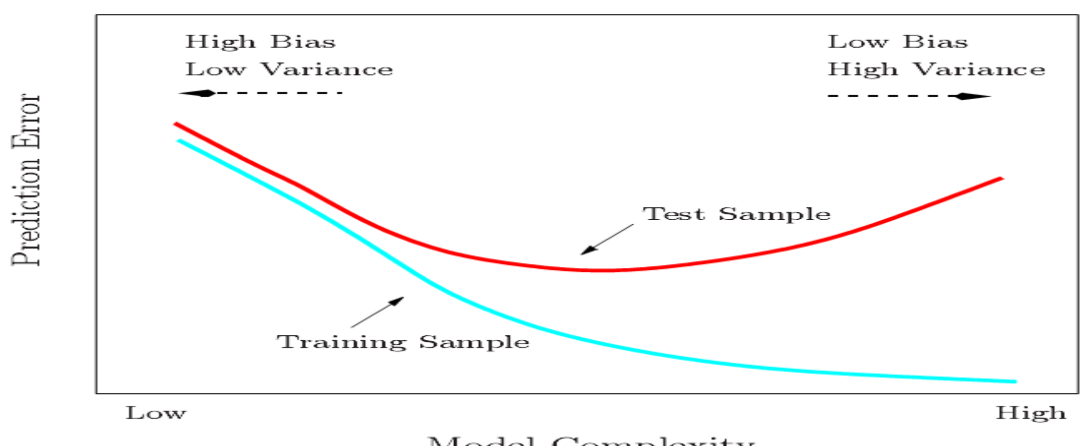


Highest Bias
Lowest variance
Model complexity = low

Medium Bias
Medium Variance
Model complexity = medium

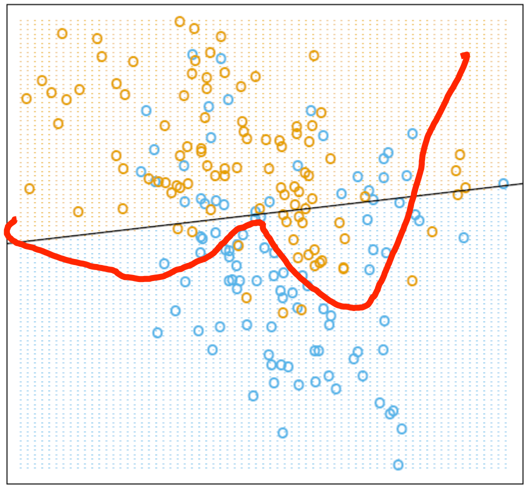
Smallest Bias
Highest variance
Model complexity = high

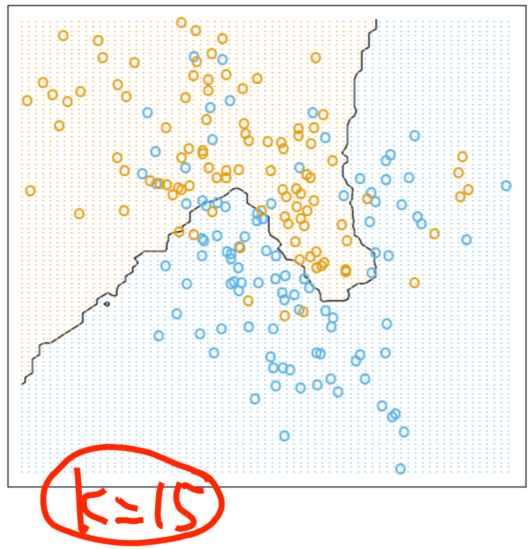
Low Variance / High Bias



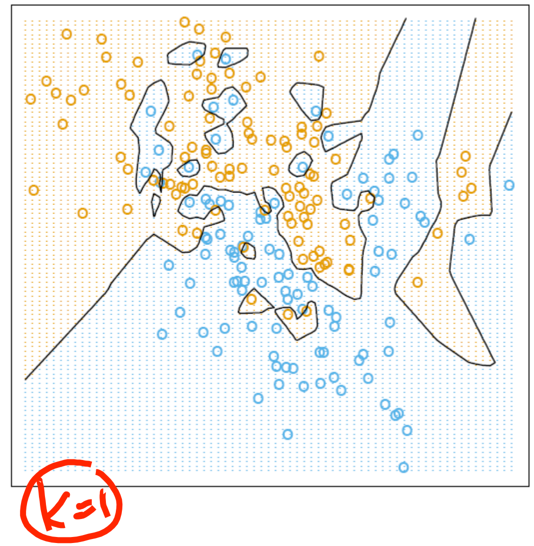
Low Bias / High Variance

(2.2) Classification, Decision boundaries in global vs. local models





15-nearest neighbor



1-nearest neighbor

Low Variance / High Bias

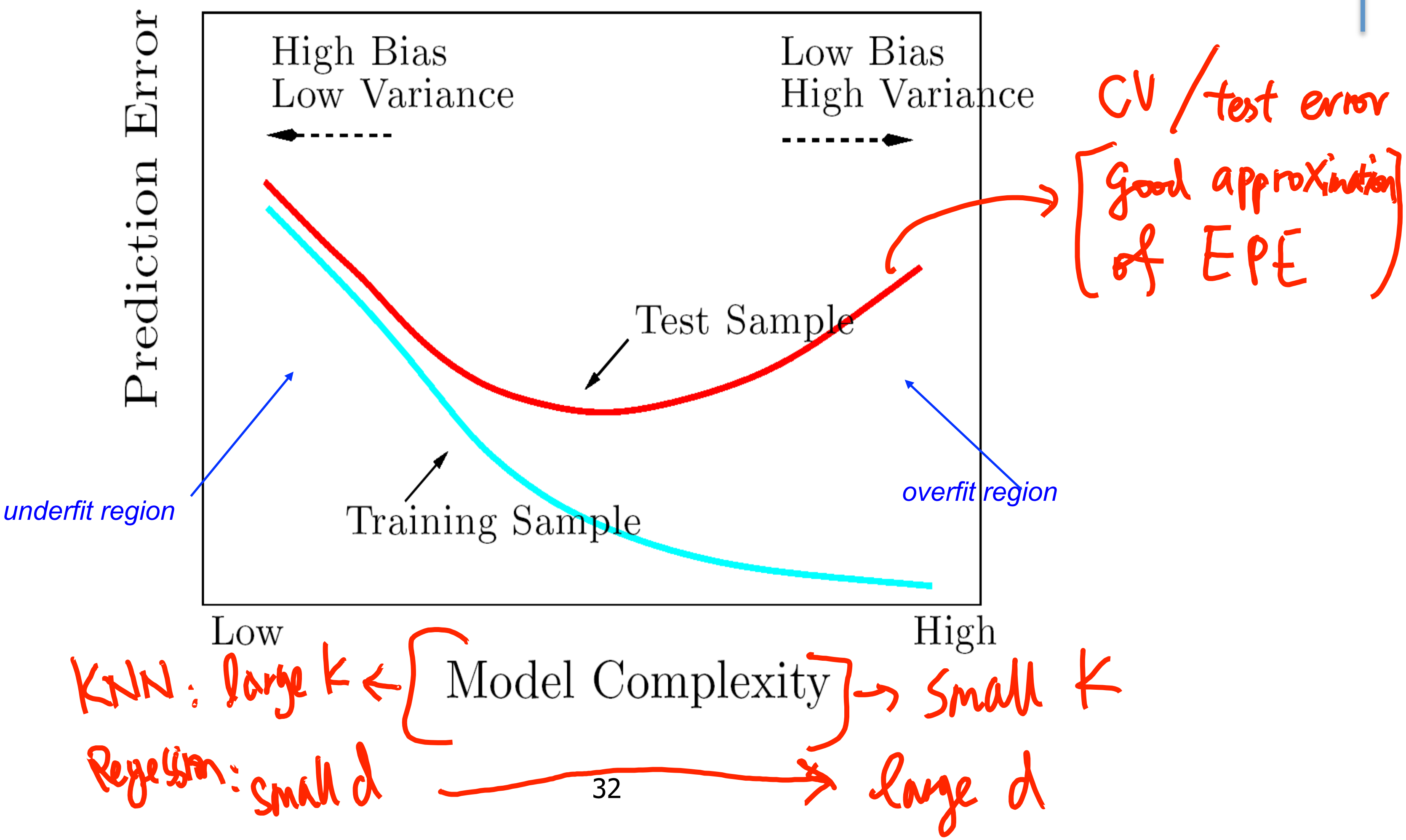
linear regression
- global
- stable
- can be inaccurate

KNN
- local
- accurate
- unstable

Low Bias / High Variance

What ultimately matters: ***GENERALIZATION***

Bias-Variance Tradeoff / Model Selection



Model "bias" & Model "variance"

- Middle RED:
  - TRUE function θ

[middle red]

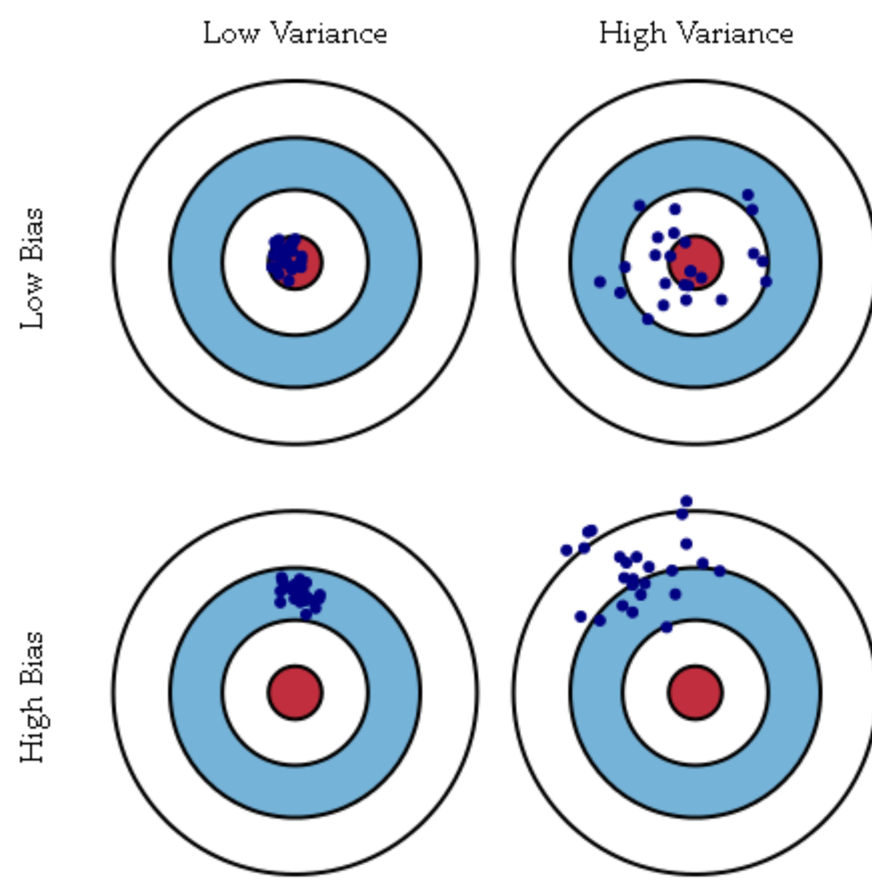
- Error due to bias:
  - How far off in general from the middle red

$$E(\theta - \underbrace{\bar{\theta}}_{\text{mean of } \hat{\theta}})$$

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

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

$\{\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \ldots\}$ Blue dots



[middle red dot]

Model “bias” & Model “variance”

- Middle RED:
  - TRUE function

θ [middle red]

- Error due to bias:
  - How far off in general from the middle red

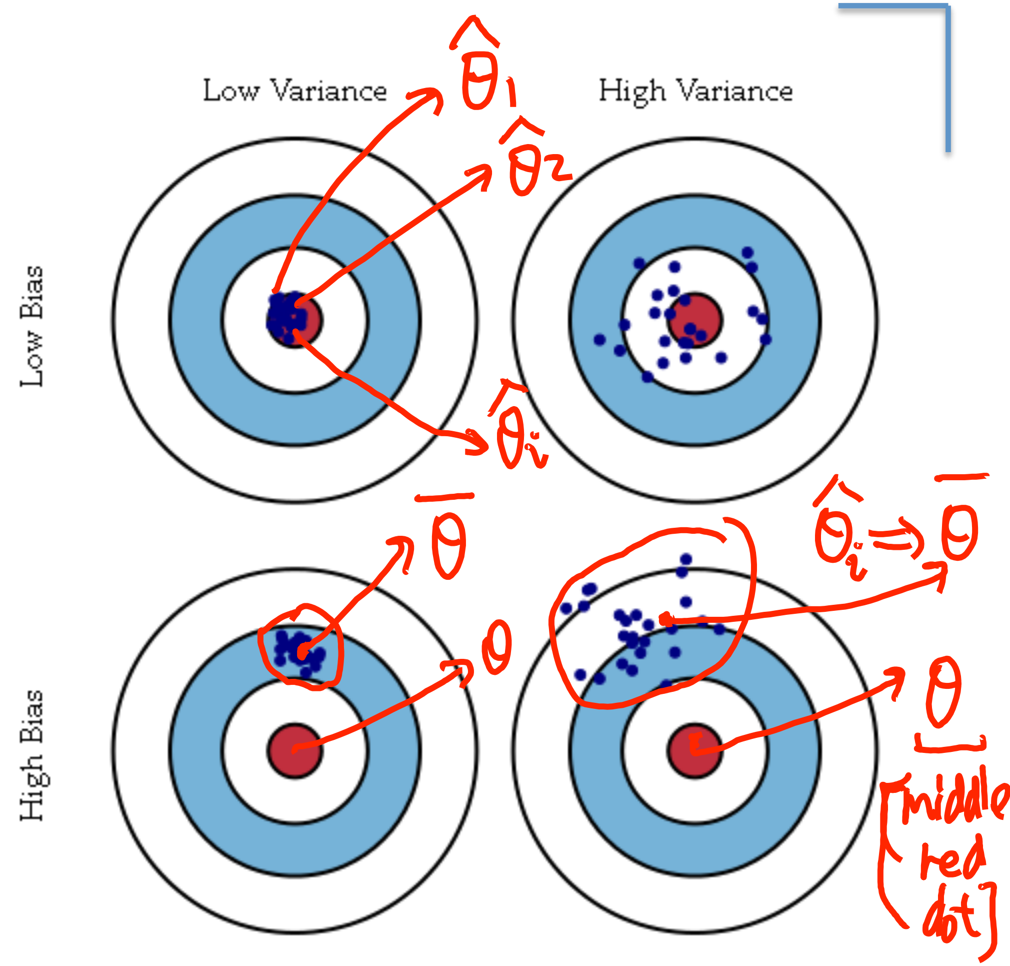
$$E(\theta - \bar{\theta})$$

mean of $\hat{\theta}$

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

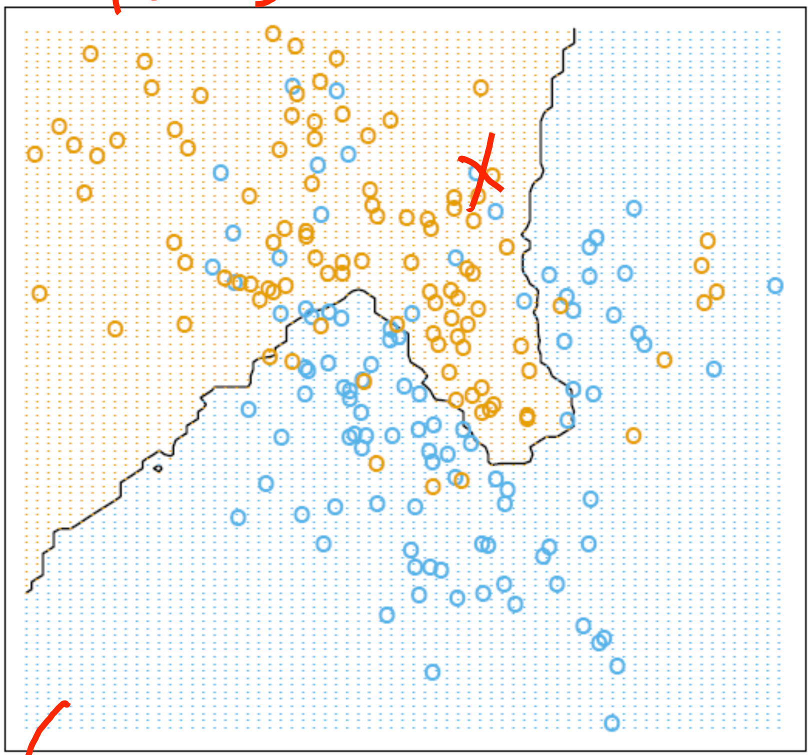
$$E((\hat{\theta} - \bar{\theta})^2)$$

$\{\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \dots\}$ Blue dots

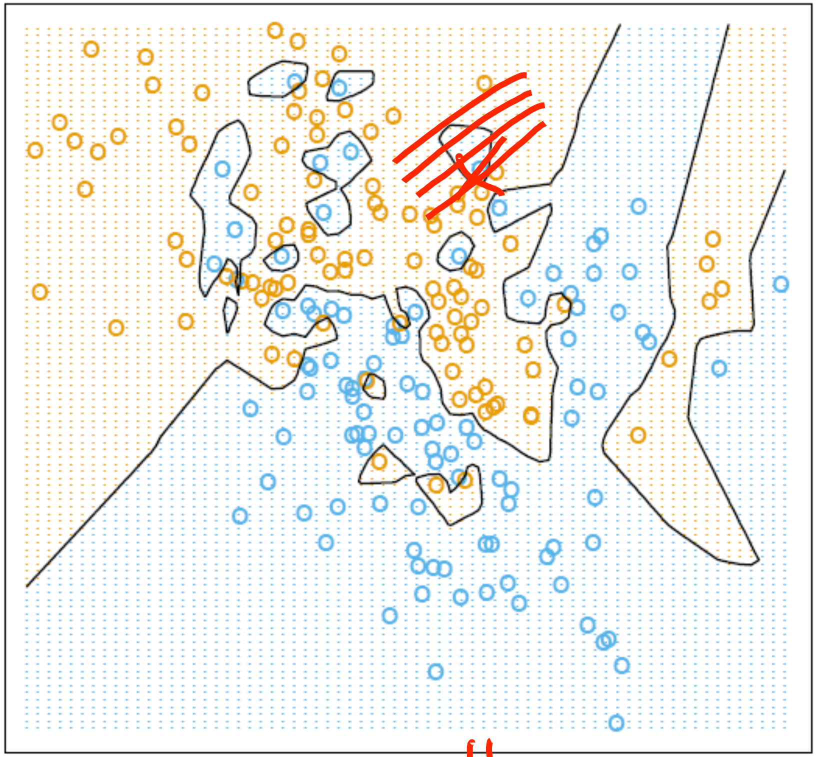


Randomness of Train Set => Variance of Models, e.g.,

K=15



K=1



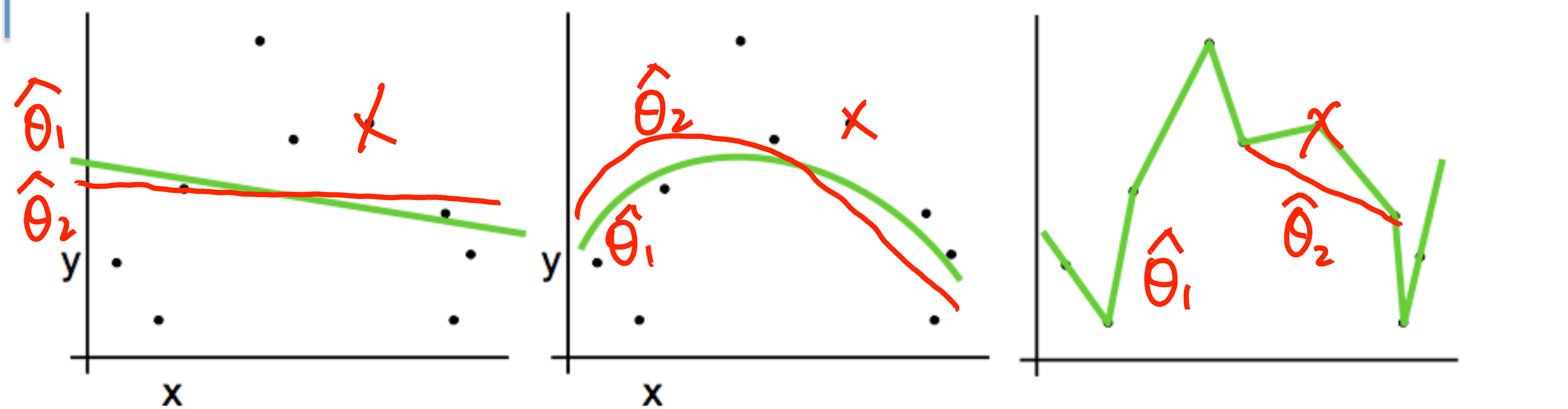
e.g removing one train sample

[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 ↗

need to make assumptions that are able to generalize

- Components
  - **Bias:** how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - **Variance:** how much models estimated from different training sets differ from each other

Slide credit: L. Lazebnik

Today :

- ✓ K-nearest neighbor
- ✓ Model Selection / Bias Variance Tradeoff
  - ✓ Bias-Variance tradeoff
  - ✓ High bias ? High variance ? How to respond ?

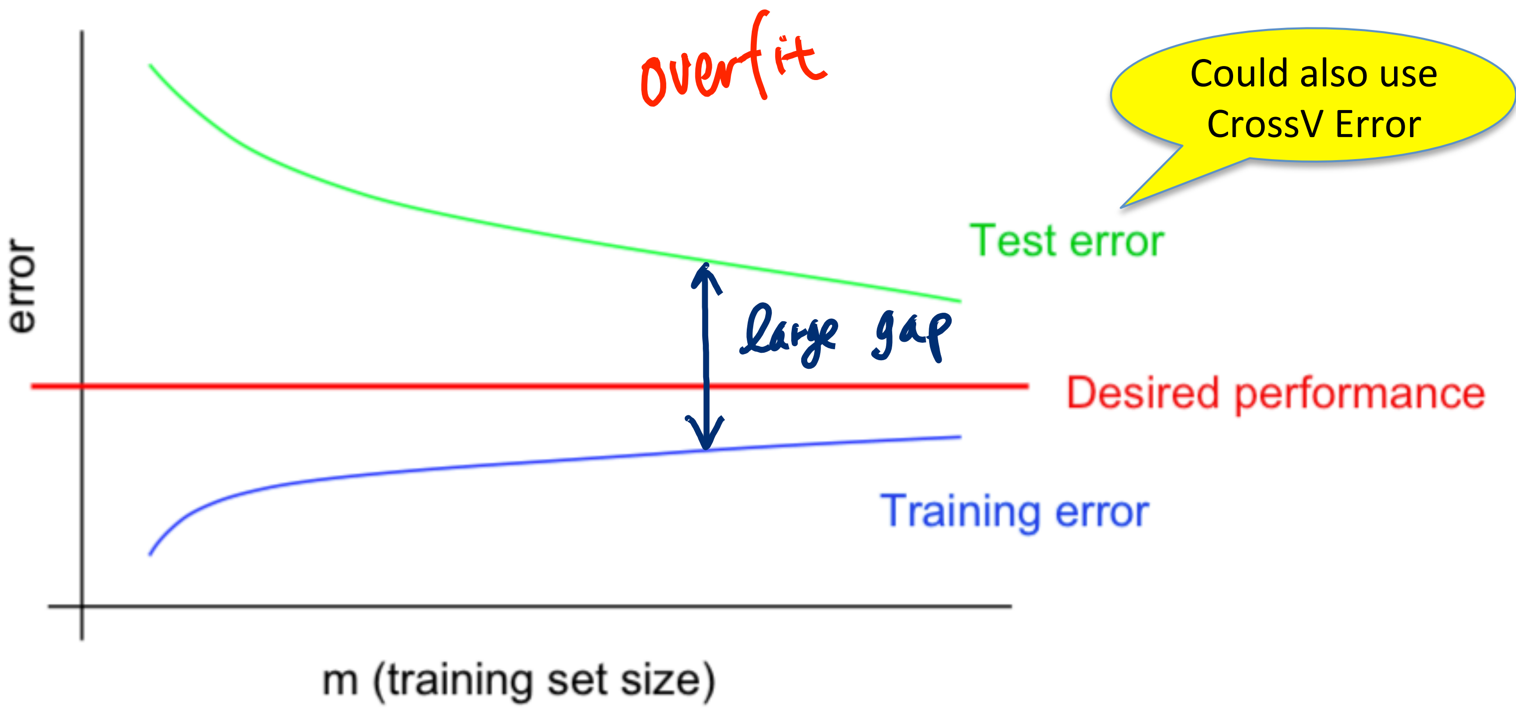
need to make assumptions that are able to generalize

- **Underfitting:** model is too “simple” to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Slide credit: L. Lazebnik

(1) High variance

Typical learning curve for high variance:



- Test error still decreasing as m increases. Suggests larger training set will help.
- Large gap between training and test error.
- Low training error and high test error

Slide credit: A. Ng

How to reduce variance?

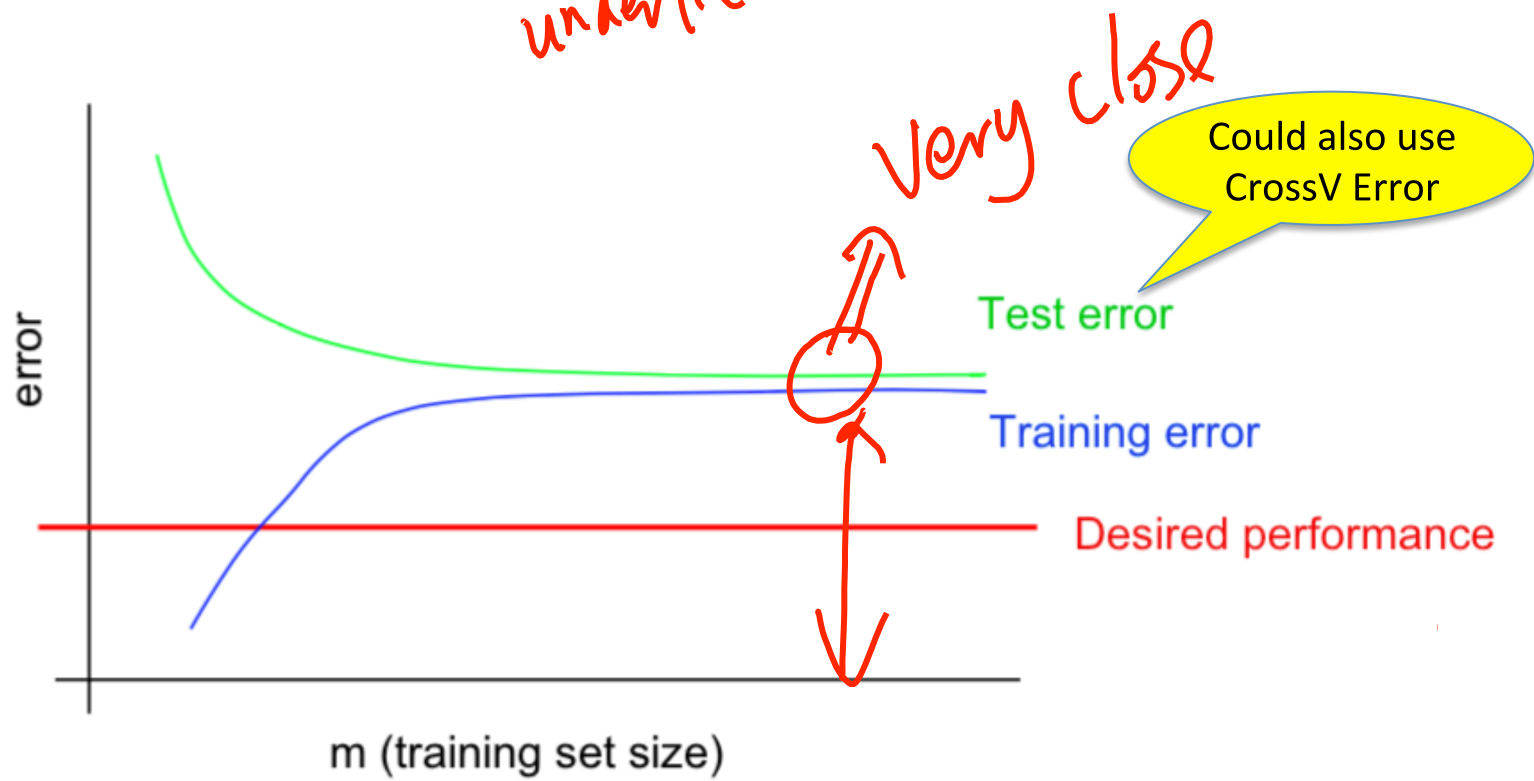
- Choose a simpler classifier
- Regularize the parameters
- Get more training data
- Try smaller set of features

Slide credit: D. Hoiem

(2) High bias

Typical learning curve for high bias:

underfit



- Even training error is unacceptably high.
- Small gap between training and test error.

High training error and high test error

How to reduce Bias ?

- E.g.

- Get additional features
- Try adding basis expansions, e.g. polynomial
- Try more complex learner

(3) 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

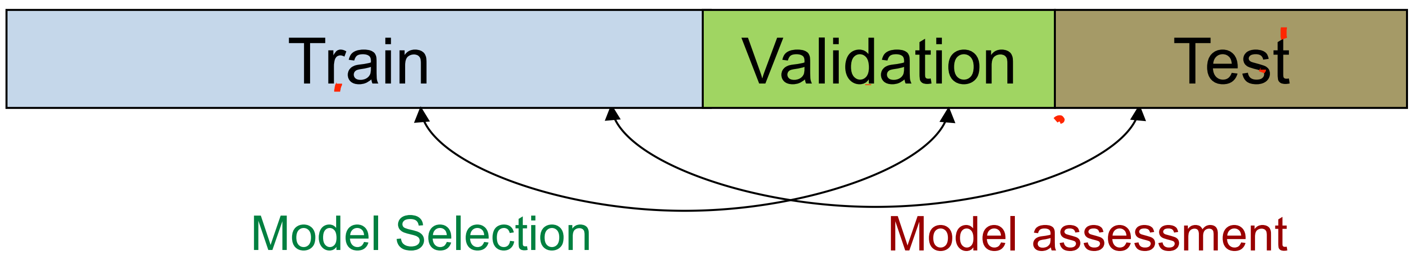
(4) Model Selection and Assessment

- Model Selection
  - Estimating performances of different models to choose the best one

- Model Assessment
  - Having chosen a model, estimating the <u>prediction error</u> on new data

Model Selection and Assessment (Extra)

- When Data Rich Scenario: Split the dataset



- When Insufficient data to split into 3 parts
  - Approximate validation step analytically
    - AIC, BIC, MDL, SRM
  - Efficient reuse of samples
    - Cross validation, bootstrap

Today Recap:

- ✓ K-nearest neighbor
- ✓ Model Selection / Bias Variance Tradeoff
  - ✓ Bias-Variance tradeoff
  - ✓ High bias ? High variance ? How to respond ?

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.

Statistical Decision Theory (Extra)

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

- Expected prediction error (EPE):
- $$\mathrm{EPE}(f) = \mathrm{E}(L(Y, f(X))) = \int L(y, f(x)) \Pr(dx, dy)$$
  $$\text{e.g.} = \int (y - f(x))^2 \Pr(dx, dy)$$

Consider population distribution

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

Expected prediction error (EPE)

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

Consider joint distribution

- For L2 loss: 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)

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

$\hat{f}(X) = C_k$ if

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

Bayes Classifier

EXPECTED PREDICTION ERROR for L2 Loss

- 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 $\Pr(X, Y) = \Pr(Y \mid X)\Pr(X)$, 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:

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

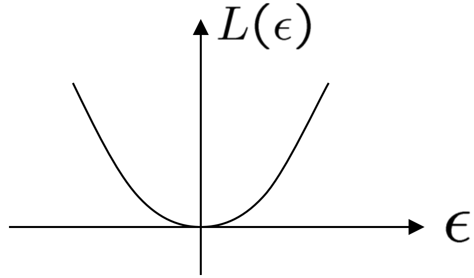
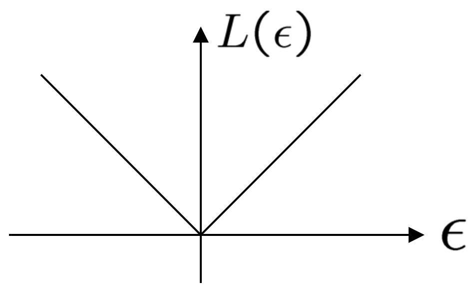
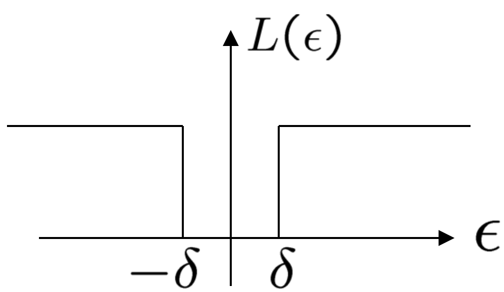
KNN FOR MINIMIZING EPE

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

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

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

Review : WHEN EPE USES DIFFERENT LOSS

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

Decomposition of EPE

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

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

MSE component of f-hat in estimating f

Irreducible / Bayes error

Bias-Variance Trade-off for EPE:

EPE (x_0) = $noise^2 + bias^2 + variance$

Unavoidable error

Error due to incorrect assumptions

Error due to variance of training samples

Slide credit: D. Hoiem

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

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

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:

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

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:

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

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

MSE of Model, aka, Risk

- 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), $i = 1, \dots n$ 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

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

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 tradeof*

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

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,

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

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