Bias-Variance & Learning Theory

CS 534: Machine Learning

Slides adapted from Lee Cooper, David Sontag, Carlos Guestrin, Luke Zettlemoyer, and Yan Liu

Review: Linear Classification

Bayes Classifier

- MAP classifier (maximum a posterior)

$$f(\mathbf{x}) = \mathrm{argmax}_{j=1,\ldots,K} \mathrm{Pr}(\mathbf{X} = \mathbf{x} | G = k)\pi_k$$

- Classifier is optimal — statistically minimizes the error rate
- Unrealistic — class conditional densities and prior probabilities must be known

Linear Discriminant Analysis (LDA)

- Assume each class density is a multivariate Gaussian

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \mathbf{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right)$$

- LDA assumes class have common covariance matrix

- Discriminant function:

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_k + \log \pi_k$$

Quadratic Discriminant Analysis (QDA)

- Covariances are not equal
- Quadratic discriminant functions:

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) + \log\pi_k$$

- Covariance matrix must be estimated for each class

LDA vs. QDA Decision Boundaries

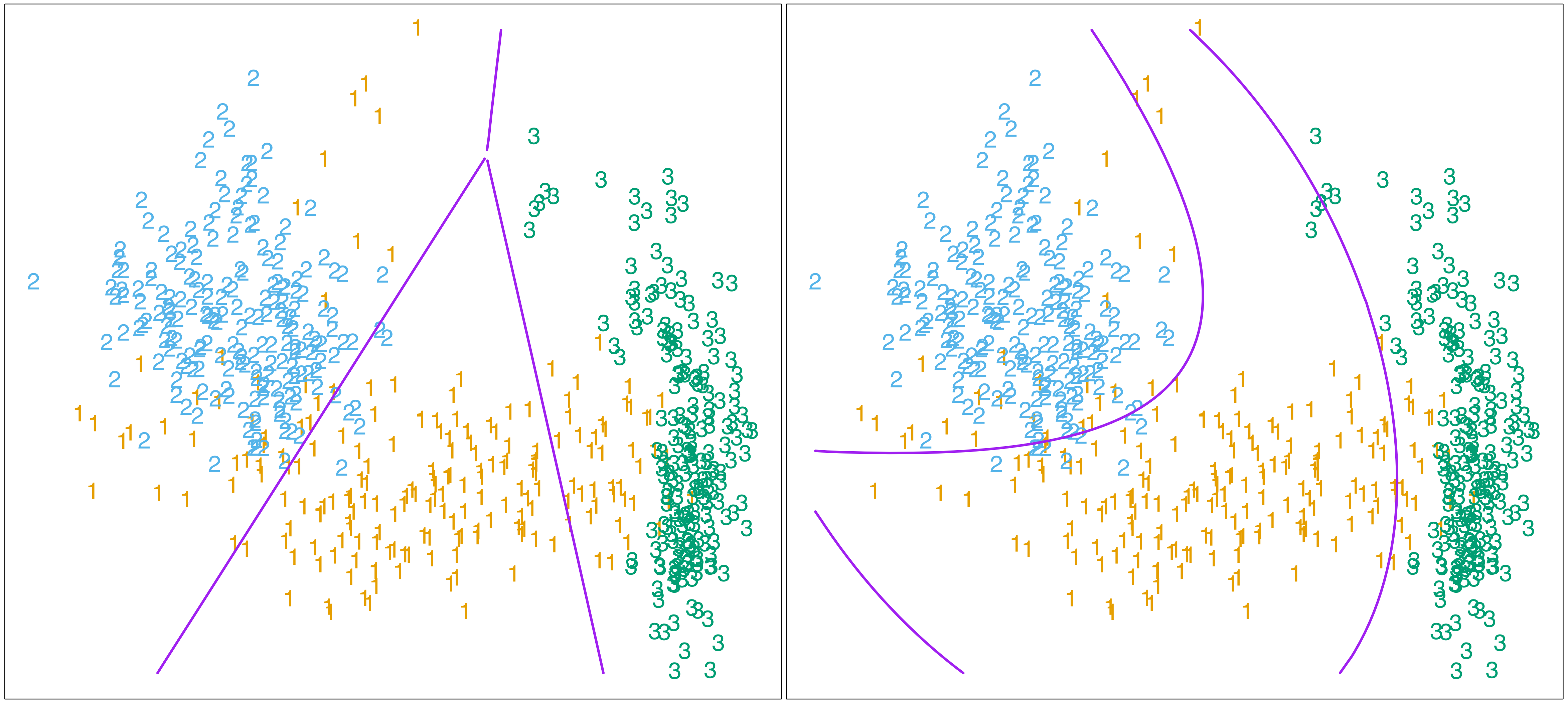


Figure 4.1 (Hastie et al.)

Logistic Regression

- Apply sigmoid to linear function of the input features

$$\Pr(G = 0|\mathbf{X}, \boldsymbol{\beta}) = \frac{1}{1 + \exp(\mathbf{X}\boldsymbol{\beta}^\top)}$$

$$\Pr(G = 1|\mathbf{X}, \boldsymbol{\beta}) = \frac{\exp(\mathbf{X}\boldsymbol{\beta}^\top)}{1 + \exp(\mathbf{X}\boldsymbol{\beta}^\top)}$$

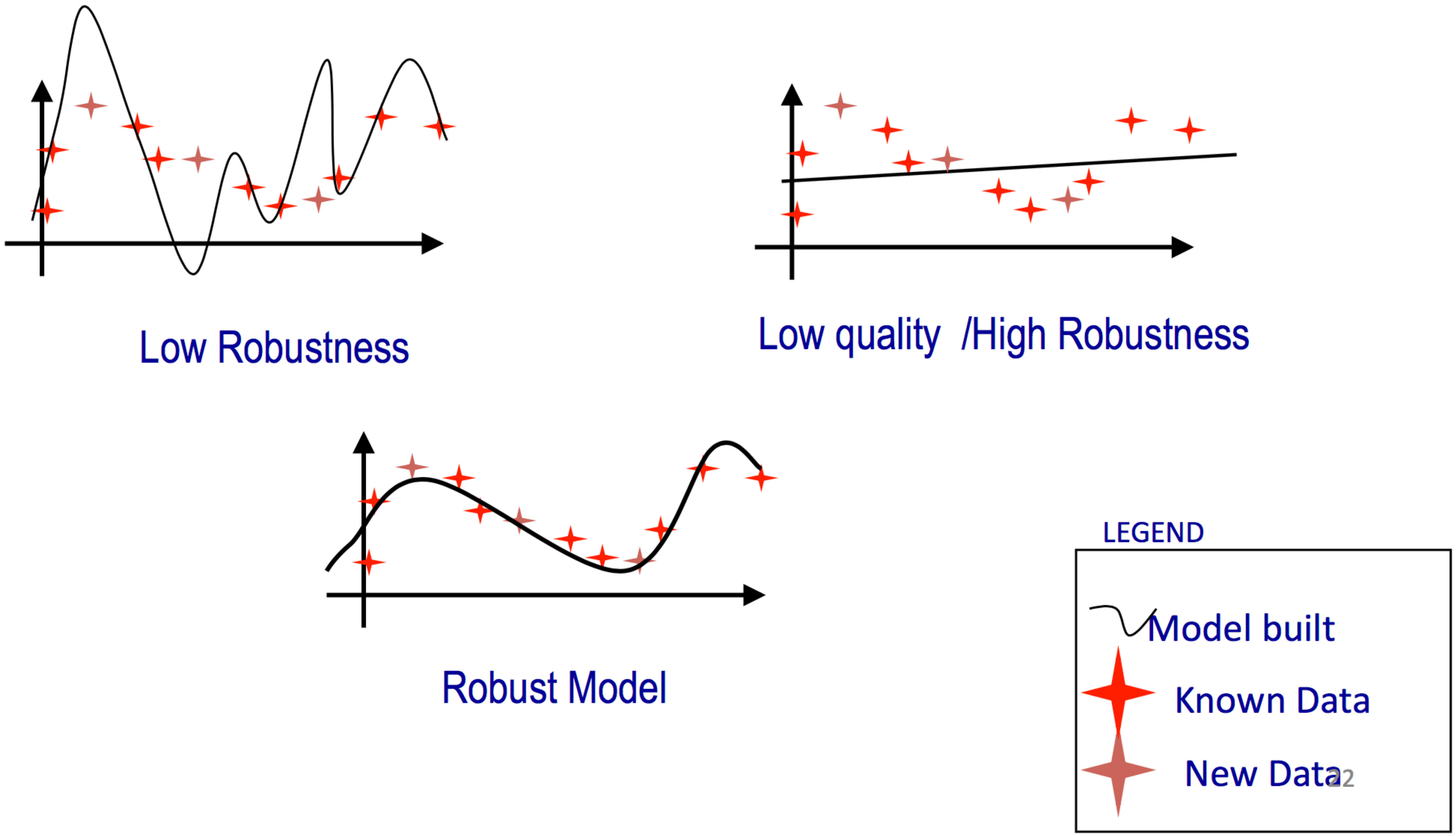
- Logistic regression estimates coefficients directly based on maximum likelihood (harder!)
- Parameters have useful interpretations
- Quite robust, well developed

Fundamental Questions

- Model selection: How to compare performance of multiple models to choose the best (identify the best parameters or methods)?
- Model Assessment: What is the performance of the model on data that it has not seen yet?

Model Assessment

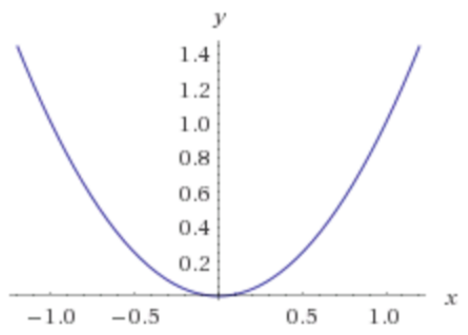
What is a “Good” Model?



http://www.cs.cmu.edu/~10601b/slides/learning_theory.pdf

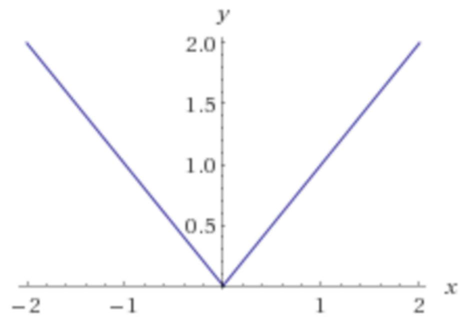
Review: Loss Functions

- Supervised learning: Find a function f(x) to predict true value y associated with x
- If a mistake is made, a loss is incurred $\ell(f(\mathbf{x}), y)$
- Examples for regression
  - Quadratic loss function (RSS)



$$\ell(f(\mathbf{x}), y) = (y - f(\mathbf{x}))^2$$

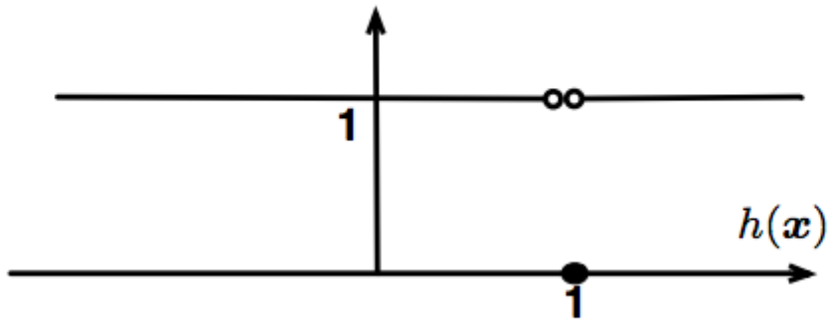
  - Absolute deviation



$$\ell(f(\mathbf{x}), y) = |y - f(\mathbf{x})|$$

Review: Loss Functions (2)

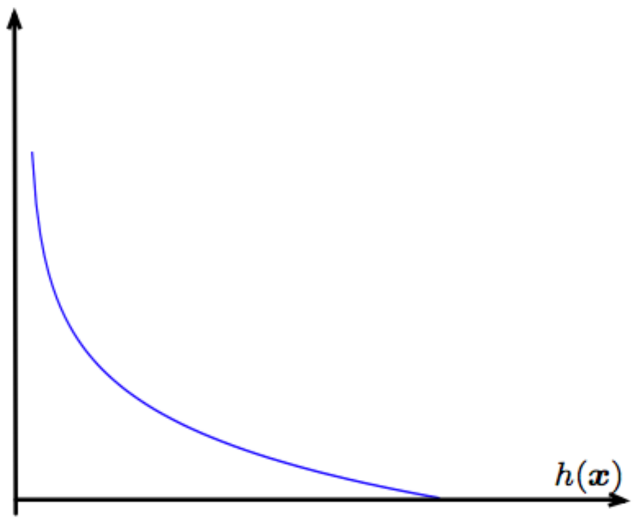
- Example for classification
  - 0/1 loss



$$\ell(f(\mathbf{x}), y) = \mathbb{1}_{y \neq f(\mathbf{x})}$$

  - Cross-entropy (logistic) loss

$$\ell(f(\mathbf{x}), y) = -y \log f(\mathbf{x}) \\ - (1-y)\log(1 - f(\mathbf{x}))$$



Measure of Predictor

- Assume we know the true distribution of the data p(x, y), the risk is

$$R[f(\mathbf{x})] = \int \ell(f(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x} dy$$

- Since we cannot compute risk in practice, we use empirical risk on a training dataset

$$R_{\text{EMP}}[f(\mathbf{x})] = \frac{1}{N} \sum_{n} \ell(f(\mathbf{x_n}), y_n)$$

$$\text{As } N \to +\infty, R_{\text{EMP}}[f(\mathbf{x})] \to R[f(\mathbf{x})]$$

Empirical Risk Minimization

- Turns out we have been doing empirical risk minimization
  - Linear regression:

$$f(\mathbf{x}) = \beta^\top \mathbf{x}, \ell(f(\mathbf{x}), y) = (y - f(\mathbf{x}))^2$$

  - Logistic regression:

$$f(\mathbf{x}) = \sigma(\beta^\top \mathbf{x}),$$
$$\ell(f(\mathbf{x}), y) = -y \log f(\mathbf{x}) - (1 - y) \log(1 - f(\mathbf{x}))$$

Potential Problem with ERM

- If our function (hypothesis) is complicated enough, the empirical risk will approach 0

$$R_{\mathrm{EMP}}[f(\mathbf{x})] \to 0$$

- What is wrong with this?
    - What about new data that is outside the training dataset?

Generalization & Overfitting

- Generalization — model performance of a model on independent / future unseen data (data not used in training)
- Overfitting — model is specific to the training set and is learning the noise from the data instead of generalizable rule

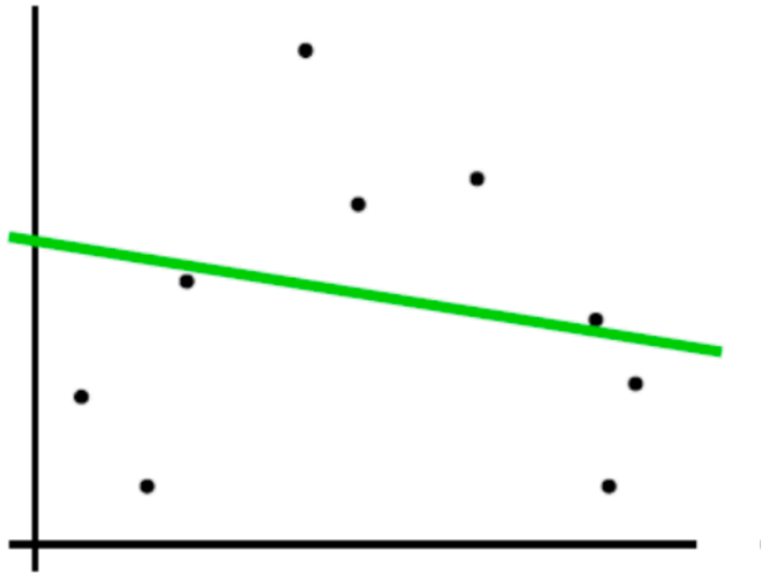
Bias and Variance: Conceptually

- Error due to bias: Difference between expected (or average) prediction of our model and the correct value we are trying to predict
  - How far off are the models if we repeat the process on new data several times?
- Error due to variance: Variability of the model prediction for a given data point
  - How different are the predictions for a given point between various realizations of the model?

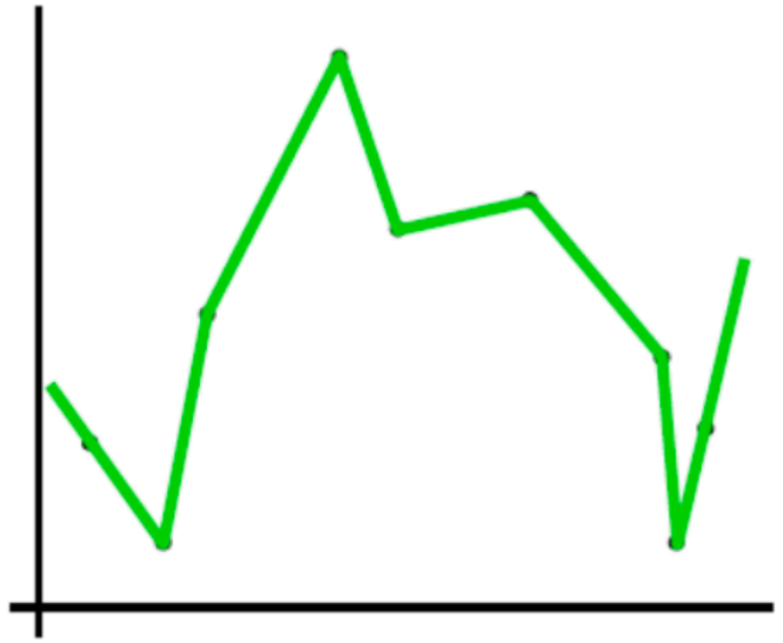
http://scott.fortmann-roe.com/docs/BiasVariance.html

Bias-Variance Tradeoff: Intuition

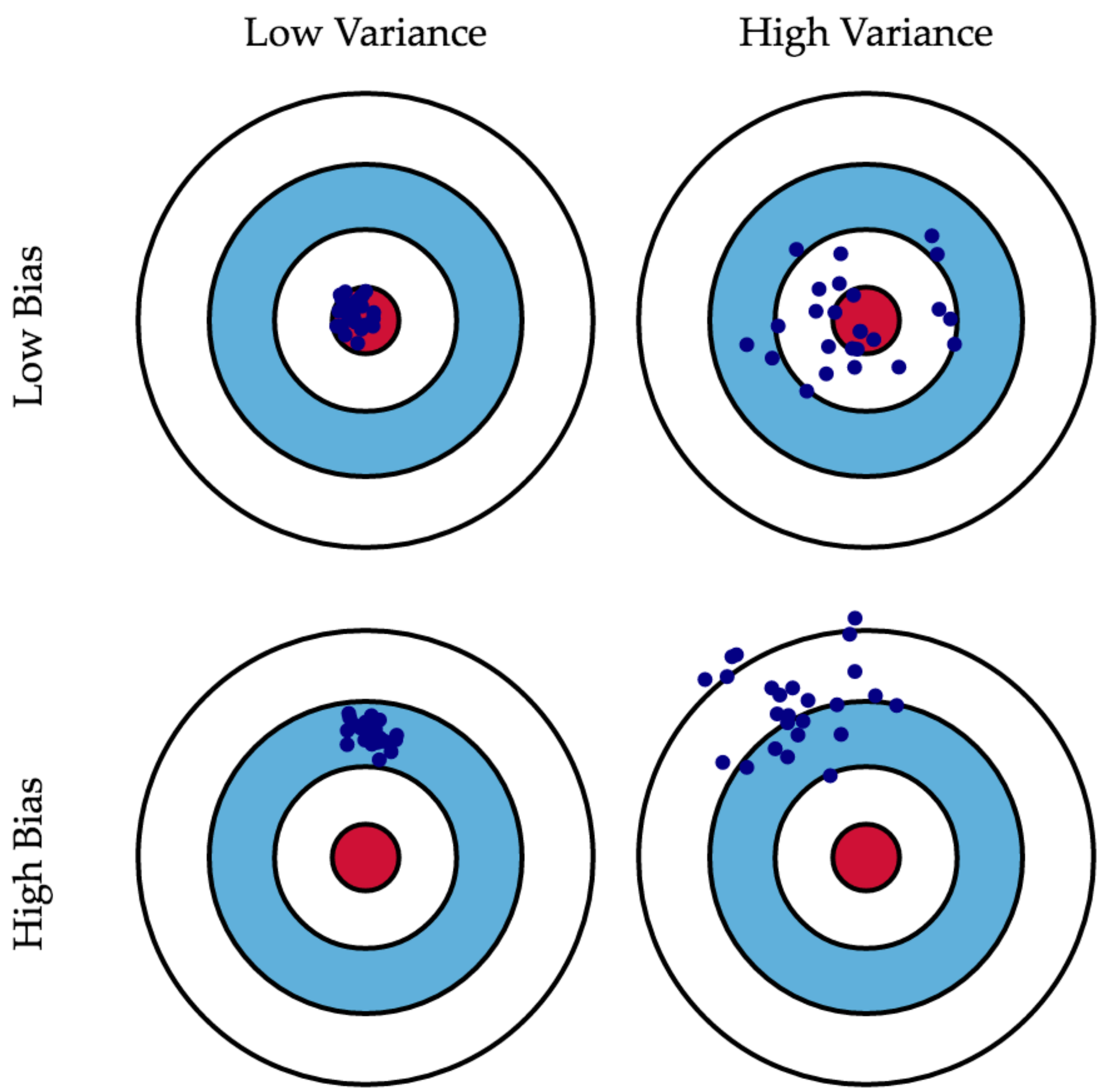
- Too “simple” model —> does not fit data well (biased solution)



- Too complex model —> small changes to the data changes the solution a lot (high variance solution)



Bias and Variance: Graphically



http://scott.fortmann-roe.com/docs/BiasVariance.html

Bias-Variance Decomposition

$$
\begin{aligned}
\mathrm{Err}(\mathbf{x}_0) &= E[(Y - \hat{f}(\mathbf{x}_0))^2 | \mathbf{x} = \mathbf{x}_0] \\
&= E[(f(\mathbf{x}_0) + \epsilon - \hat{f}(\mathbf{x}_0))^2 | \mathbf{x} = \mathbf{x}_0] \\
&= (E[\hat{f}(\mathbf{x}_0)] - f(\mathbf{x}_0))^2 + E[\hat{f}(\mathbf{x}_0) - E[\hat{f}(\mathbf{x}_0)]]^2 + \sigma_\epsilon^2 \\
&= \mathrm{Bias}^2(\hat{f}(\mathbf{x}_0)) + \mathrm{Var}(\hat{f}(\mathbf{x}_0)) + \sigma_\epsilon^2
\end{aligned}
$$

Effect of Finite Samples

- Every training sample D is a sample from the true joint distribution
- Prediction function $f_D(x)$ is a random function with respect to this distribution
- Risk: $R[f_D(\mathbf{x})] = \int_{\mathbf{x}} \int_y (f_D(\mathbf{x}) - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy$

Average Over Training Set Distribution

- Averaged risk to remove randomness with respect to D

$$E_D[R[f_D(\mathbf{x})]] = \int_D \left(\int_{\mathbf{x}} \int_y (f_D(\mathbf{x}) - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy \right) P(D) dD$$

- Averaged prediction

$$E_D[f_D(\mathbf{x})] = \int_D f_D(\mathbf{x}) P(D) dD$$

With many training datasets, use the average of the predicted functions learned on each dataset

Bias/Variance Detailed Analysis

$$
\begin{aligned}
E_D[R[f(_D\mathbf{x})]] &= \int_D \left(\int_{\mathbf{x}} \int_y (f_D(\mathbf{x}) - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy \right) P(D) dD \\
&= \int_D \int_{\mathbf{x}} \int_y [f_D(\mathbf{x}) - E_D[f_D(\mathbf{x})] \\
&\qquad + E_D[f_D(\mathbf{x})] - y]^2 p(\mathbf{x}, y) d\mathbf{x} dy P(D) dD \\
&= \int_D \int_{\mathbf{x}} \int_y [f_D(\mathbf{x}) - E_D[f_D(\mathbf{x})]]^2 p(\mathbf{x}, y) d\mathbf{x} dy P(D) dD \\
&\quad + \int_D \int_{\mathbf{x}} \int_y [E_D[f_D(\mathbf{x})] - y]^2 p(\mathbf{x}, y) d\mathbf{x} dy P(D) dD \\
&\quad + \int_D \int_{\mathbf{x}} \int_y (f_D(\mathbf{x}) - E_D[f_D(\mathbf{x})])(E_D[f_D(\mathbf{x})] - y) p(\mathbf{x}, y) d\mathbf{x} dy P(D) dD
\end{aligned}
$$

variance

cross-term

Cross-Term = 0

$$\int_D \int_{\mathbf{x}} \int_y (f_D(\mathbf{x}) - E_D[f_D(\mathbf{x})])(E_D[f_D(\mathbf{x})] - y) p(\mathbf{x}, y) d\mathbf{x} dy P(D) dD$$

$$\int_{\mathbf{x}} \int_y \underbrace{\left\{ \int_D f_D(\mathbf{x}) - E_D[f_D(\mathbf{x})] P(D) dD \right\}}_{=0} (E_D[f_D(\mathbf{x})] - y) p(\mathbf{x}, y) d\mathbf{x} dy$$

Variance Analysis: Sources

$$\int_D \int_{\mathbf{x}} \int_y [f_D(\mathbf{x}) - E_D[f_D(\mathbf{x})]]^2 p(\mathbf{x}, y) d\mathbf{x} dy P(D) dD$$

- Noise in labels or features
- Training sample too small
- "Too local" algorithms that easily fit data
- Randomness in learning algorithm
  (i.e., non-convex algorithms)

High variance —>
overfitting the data

Variance Analysis: Reduction

$$\int_D \int_{\mathbf{x}} \int_y [f_D(\mathbf{x}) - E_D[f_D(\mathbf{x})]]^2 p(\mathbf{x}, y) d\mathbf{x} dy P(D) dD$$

- Use a lot of data (increase size of D)
- Use a simple function so that $f_D(x)$ does not vary much across different training sets
  (e.g., f(x) = c)

Remaining Term

$$\int_D \int_{\mathbf{x}} \int_y [E_D[f_D(\mathbf{x})] - y]^2 p(\mathbf{x}, y) d\mathbf{x} dy P(D) dD$$

$$= \int_{\mathbf{x}} \int_y [E_D[f_D(\mathbf{x})] - y]^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

$$= \int_{\mathbf{x}} \int_y [E_D[f_D(\mathbf{x})] - E_y[y] + E_y[y] - y]^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

$$= \int_{\mathbf{x}} \int_y [E_D[f_D(\mathbf{x})] - E_y[y]]^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

bias

$$+ \int_{\mathbf{x}} \int_y [E_y[y] - y]^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

noise

Noise Analysis

$$\int_{\mathbf{x}} \int_{y} [E_y[y] - y]^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

- Nothing we can do!
- Quantity depends on joint distribution only, choosing function or training dataset has no effect

Bias Analysis: Sources

$$\int_{\mathbf{x}}\int_{y}[E_D[f_D(\mathbf{x})] - E_y[y]]^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

- Inability to represent certain decision boundaries
- Incorrect assumptions
- Classifiers are "too global" (e.g., single linear separator)

High bias —> underfitting the data

Bias Analysis: Reduction

$$\int_{\mathbf{x}}\int_{y}[E_D[f_D(\mathbf{x})] - E_y[y]]^2 p(\mathbf{x}, y)d\mathbf{x}dy$$

- More complex models
  - Function as flexible as possible
  - Better function approximates $E_y[y]$ —> smaller bias

Bias, Variance, and Model Complexity

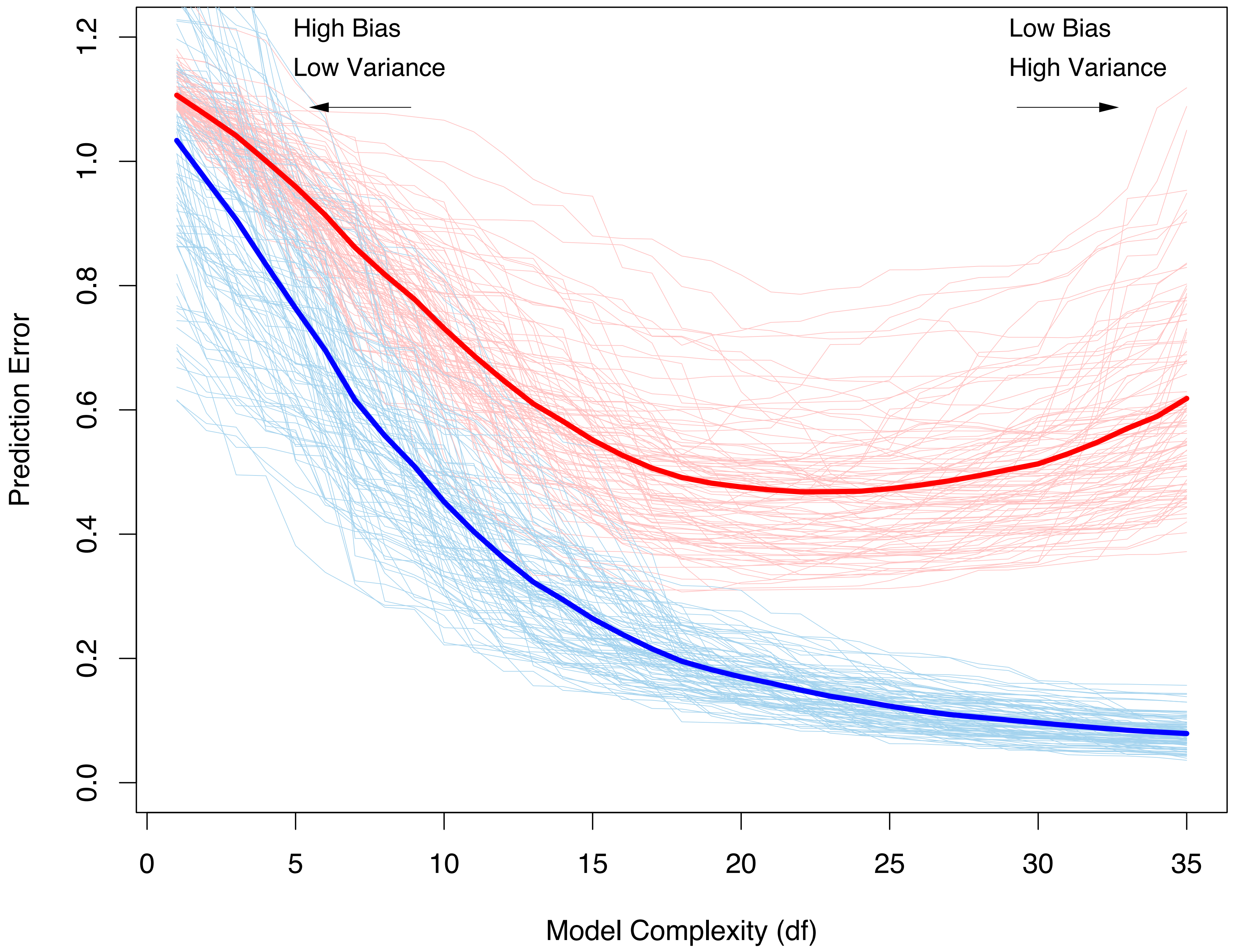
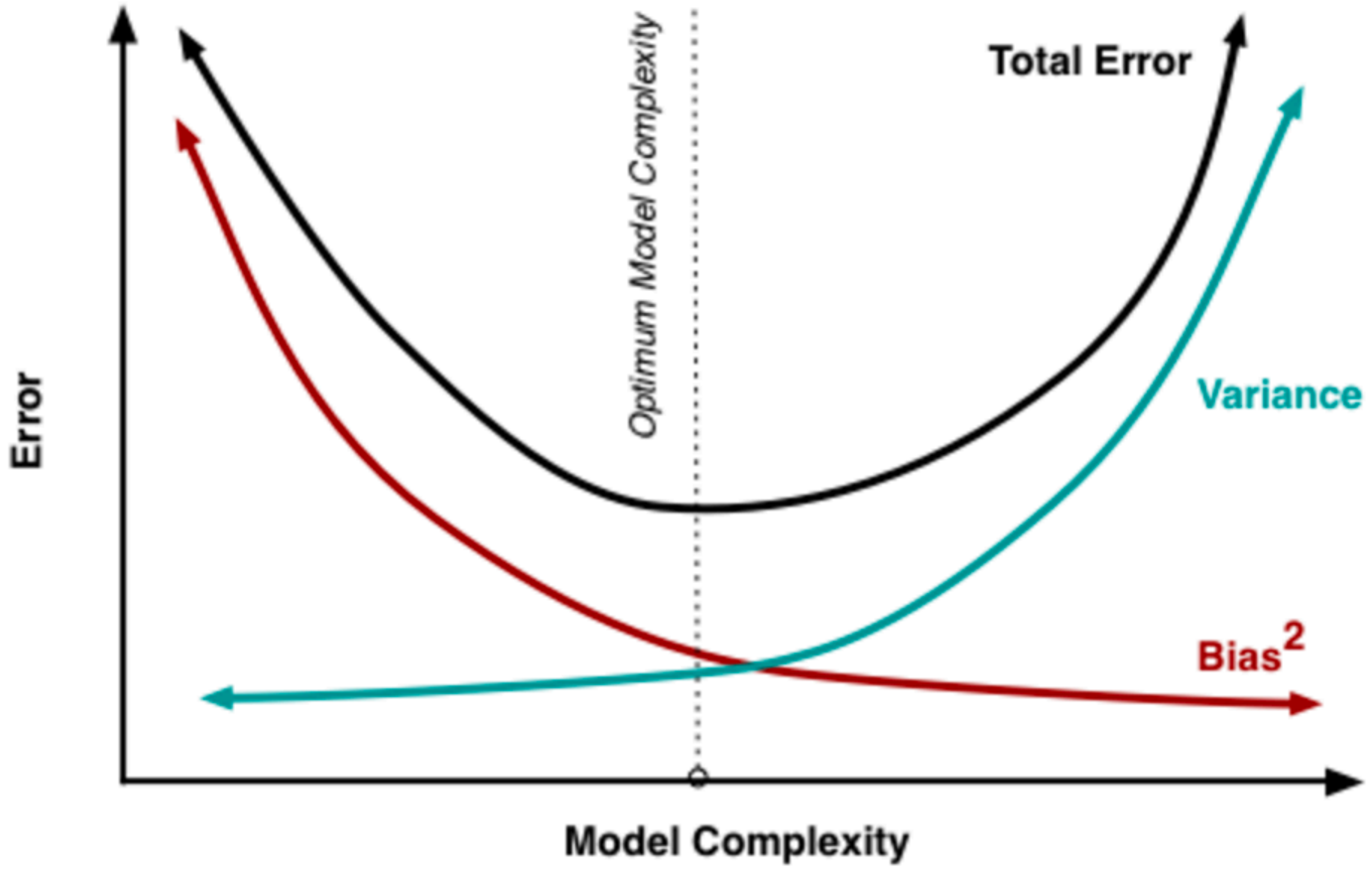


Figure 7.1 (Hastie et al.)

Underfitting vs Overfitting



Example: Linear Regression

- Gauss-Markov Theorem: Least squares estimate has the minimum variance among all linear unbiased estimates
- Truth: $f(\mathbf{x}) = \mathbf{X}\boldsymbol{\beta}$
- Observed: $y = f(\mathbf{x}) + \epsilon, \ E[\epsilon] = 0$
- Bias: $f(\mathbf{x}_0) - E[\hat{f}(\mathbf{x}_0]$

$$
\begin{aligned}
&= \mathbf{x}_0\boldsymbol{\beta} - E[\mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta} + \epsilon)] \\
&= \mathbf{x}_0\boldsymbol{\beta} - E[\mathbf{x}_0^\top\boldsymbol{\beta} + \mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\epsilon] \\
&= \mathbf{x}_0\boldsymbol{\beta} - \mathbf{x}_0^\top\boldsymbol{\beta} + \mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top E[\epsilon] = 0
\end{aligned}
$$

Example: Linear Regression (2)

- Variance:

$$
\begin{aligned}
&E[(\hat{f}(\mathbf{x}_0 - E[\hat{f}(\mathbf{x}_0])^2] \\
&= E[(\hat{f}(\mathbf{x}_0 - f(\mathbf{x}_0))^2] \\
&= E[(\mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta} + \epsilon) - \mathbf{x}_0^\top\boldsymbol{\beta})^2] \\
&= E[(\mathbf{x}_0^\top\boldsymbol{\beta} + \mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}\epsilon) - \mathbf{x}_0^\top\boldsymbol{\beta})^2] \\
&= E[(\mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}\epsilon)^2] \\
&= \sigma_\epsilon^2 \frac{p}{N}
\end{aligned}
$$

Refined Decomposition of Bias

- Bias = model bias + estimation bias

$$
\begin{aligned}
&E_{\mathbf{x}_0}[f(\mathbf{x}_0) - E[\hat{f}(\mathbf{x}_0)]]^2 \\
&\quad = E_{\mathbf{x}_0}[f(\mathbf{x}_0) - \mathbf{x}_0^\top \beta_*]^2 + E_{\mathbf{x}_0}[\mathbf{x}_0^\top \beta_* - E[\mathbf{x}_0^\top \hat{\beta}_\alpha]]^2 \\
&\quad = \text{Ave}[\text{Model Bias}]^2 + \text{Ave}[\text{Estimation Bias}]^2
\end{aligned}
$$

- Model bias: price for choosing linear functions to model data

- Estimation bias: difference between optimal model and estimated model

Regularized Linear Regression Tradeoff

- For regularized regression, estimation bias becomes positive compared to zero for least squares

$$E_{\mathbf{x}_0}[\mathbf{x}_0^\top \beta_* - E[\mathbf{x}_0^\top \hat{\beta}_\alpha + \lambda ||\beta||_p]]^2$$

- No longer unbiased estimate

- However, variance can be reduced

$$\mathrm{Var}(\hat{f}(\mathbf{x}_0)) = ||\mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{x}_0||^2 \sigma_\epsilon^2$$

Bais-Variance Tradeoff: Key in ML

- Choice of hypothesis class introduces learning bias
    - More complex class —> less bias
    - More complex class —> more variance

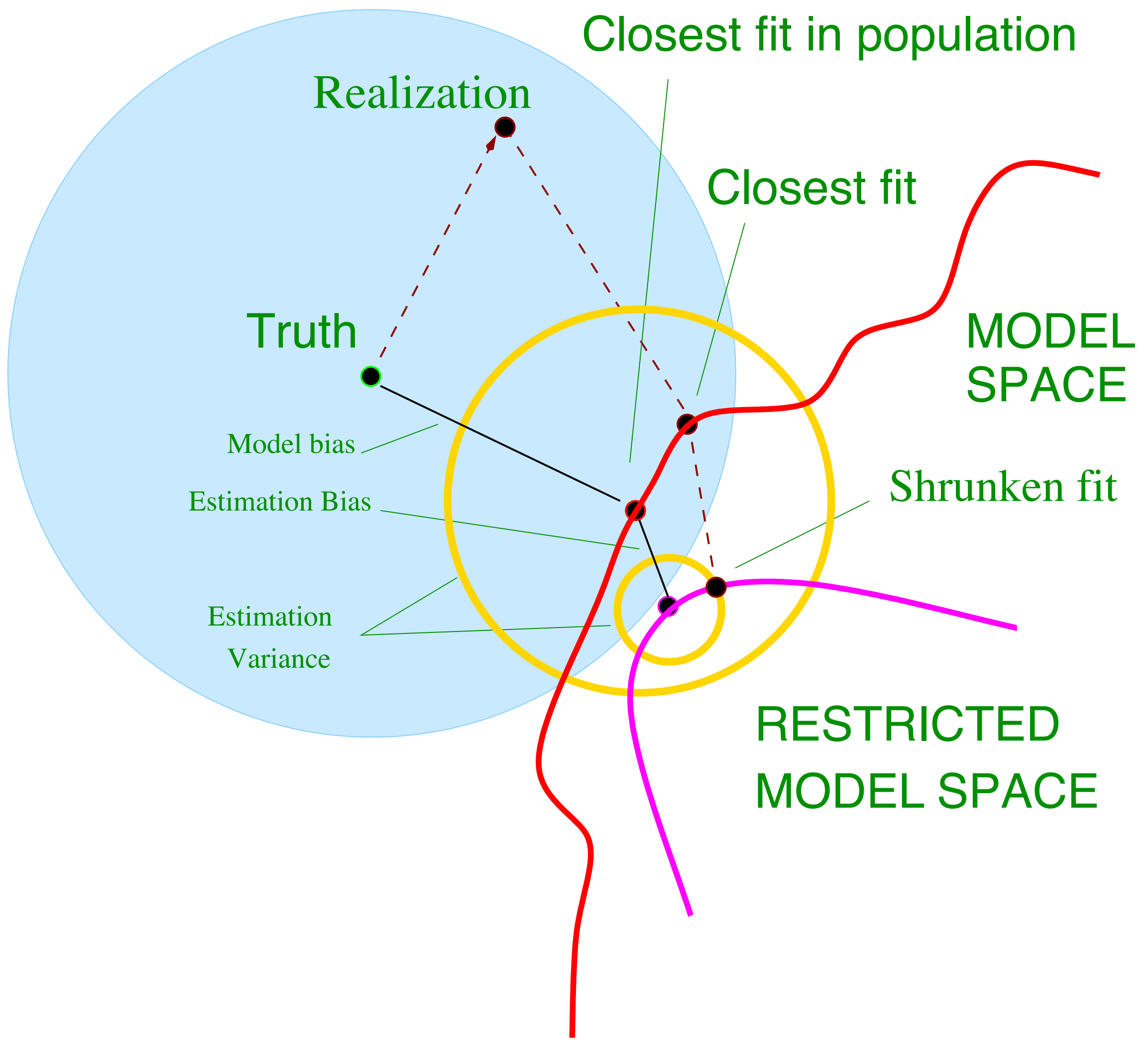


Figure 7.2 (Hastie et al.)

Learning Theory: An Introduction

Background: Union Bound

Let A_1, A_2, …, A_k be k different events (need not be independent)

$$P(A_1 \cup \cdots \cup A_k) \leq P(A_1) + \cdots + P(A_k)$$

Probability of any of k events happening is at most the sum of the probabilities of the k events

Background: Hoeffding Inequality

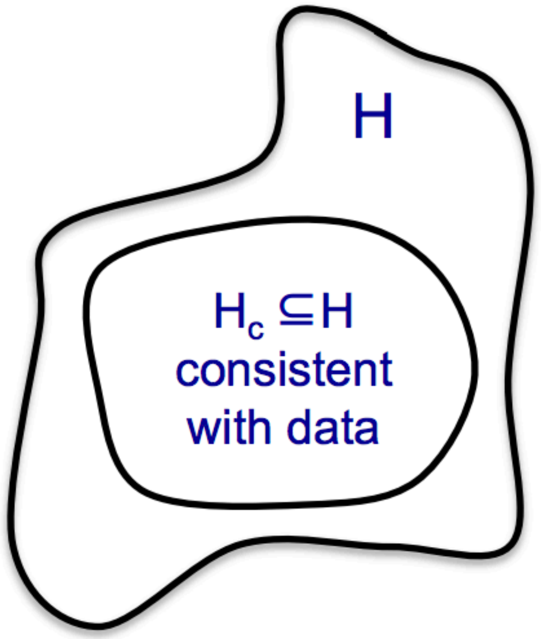
Let Z_1, Z_2, …, Z_m be m iid random variables drawn from a Bernoulli distribution, Bernoulli(ϕ). Let $\hat{\phi}$ represent the mean of these random variables, and let any $\gamma > 0$ be fixed. Then

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2\exp(-2\gamma^2 m)$$

Also known as Chernoff bound tells us the probability of how far our estimate of the parameter is from the true value

Simple Setting

- Classification problem
  - m data points
  - Finite number of possible hypothesis (e.g., 40 spam classifiers)
- A learner finds a hypothesis h that is consistent with training data
  - Gets zero error in training (i.e., one of the classifier gets 100% accuracy on the m emails)



Simple Setting: Notation

- Training Error: fraction of training examples misclassified

$$\hat{\epsilon}(h) = \frac{1}{m} \sum_i \mathbb{1}_{\{h(x_i) \neq y_i\}}$$

- Generalization Error: probability of drawing a new sample from distribution D and f will misclassify it

$$\epsilon(h) = P_{(x,y)\sim D}(h(x) \neq y)$$

- Hypothesis class H to be the set of all classifiers considered in the algorithm
  - Linear: $\mathcal{H} = \{h_\theta : h_\theta(x) = \mathbb{1}_{\{\theta^\top x \geq 0\}}, \theta \in \mathbb{R}^{n+1}\}$

Finite Hypothesis Case

- k hypothesis
- ERM selects the one that has the smallest training error
- Training error:

$$\hat{\epsilon}(h_i) = \frac{1}{m}\sum_j Z_j,\ Z_j = \mathbb{1}_{\{h_i(x_j) \neq y_j\}}$$

Finite Hypothesis: Bounds

- For a particular hypothesis, training error and generalization error bound

$$P(|\epsilon(h_i) - \hat{\epsilon}(h_i)| > \gamma) \leq 2\exp(-2\gamma^2 m)$$

- For the entire hypothesis class

$$P(\exists h \in \mathcal{H} \mid |\epsilon(h_i) - \hat{\epsilon}(h_i)| > \gamma) \leq 2k\exp(-2\gamma^2 m)$$
$$P(\neg\exists h \in \mathcal{H} \mid |\epsilon(h_i) - \hat{\epsilon}(h_i)| > \gamma) \geq 1 - 2k\exp(-2\gamma^2 m)$$

Generalization Error [Haussler, 1998]

- Theorem:

  If the hypothesis space H is finite and D is a sequence of m ≥ 1 independent random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to H and D is not ϵ-exhausted its less than $|H| \exp(-m\epsilon)$

Bounds the probability that any consistent learner will output a hypothesis with error(h) greater than or equal to epsilon

Sample Complexity

- How large must m be before we can guarantee that with probability at least $1 - \delta$ and $\delta > 0$, the training error will be within γ of generalization error?

$$m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$$

- Bound tells us how many training examples are needed to achieve a certain performance (aka sample complexity)

PAC Bound & Bias-Variance

- What if we hold number of samples and probability fixed and want to solve for distance of generalization error?
- Theorem: With probability at least $1 - \delta$,

$$\epsilon(\hat{h}) \leq \underbrace{\left(\min_{h \in \mathcal{H}} \epsilon(h)\right)}_{\text{bias}} + \underbrace{2\sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}}_{\text{variance}}$$

Infinite Hypothesis Class

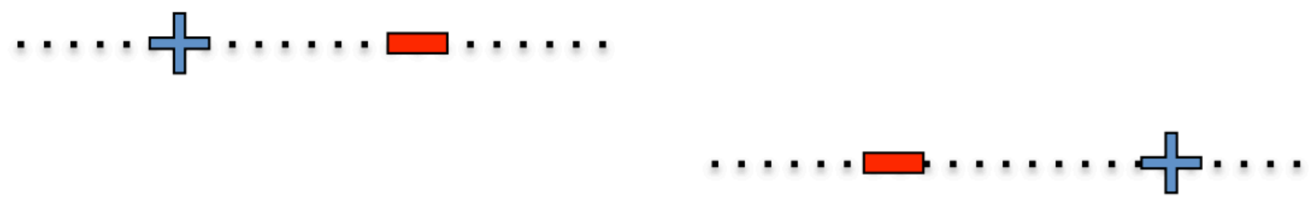
- How can we generalize the bounds to infinite number of functions (i.e., linear classification parameterized by real numbers)?
- Variance is obviously not infinite…
- Idea: only care about the maximum number of points that can be classified exactly

Vapnik-Chernovenkis (VC) Dimension

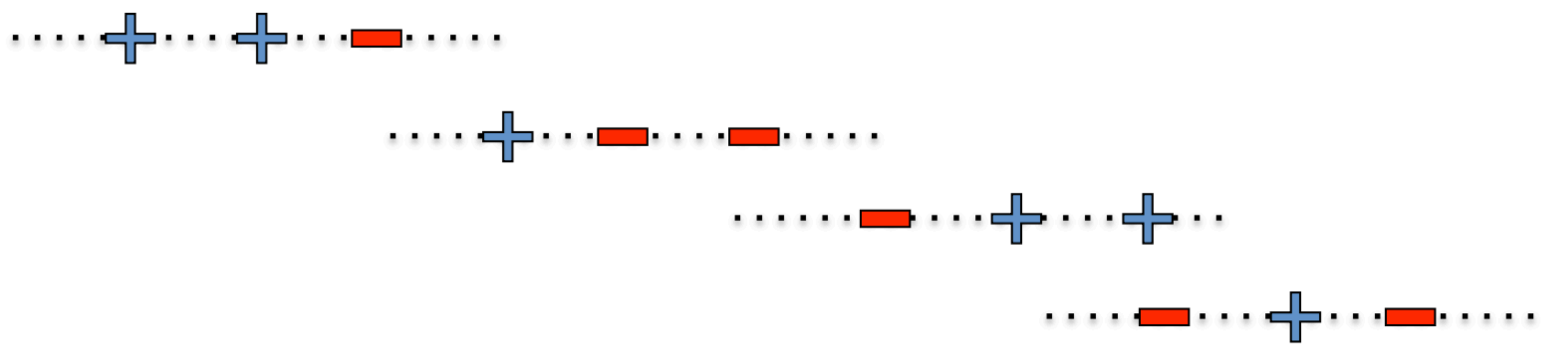
- Classic measure of complexity of infinite hypothesis classes
- Answers the question of whether we can find a hypothesis that correctly classifies the data no matter how the data points were labeled
- Maximum number of points K so that you can always find the correct

Example: VC Dimension 1-D

- How many points can a linear boundary classify in one dimension?
    - 2 points?

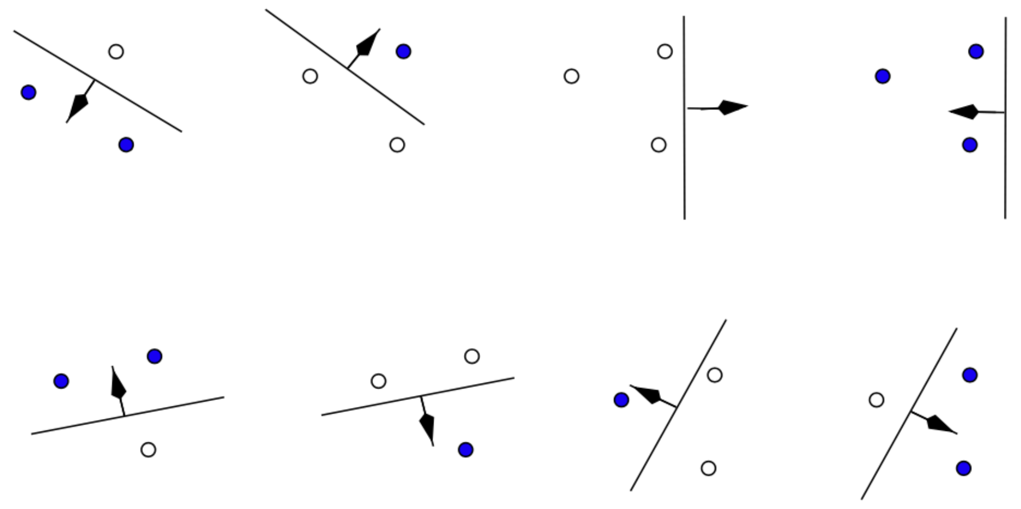


    - 3 points?

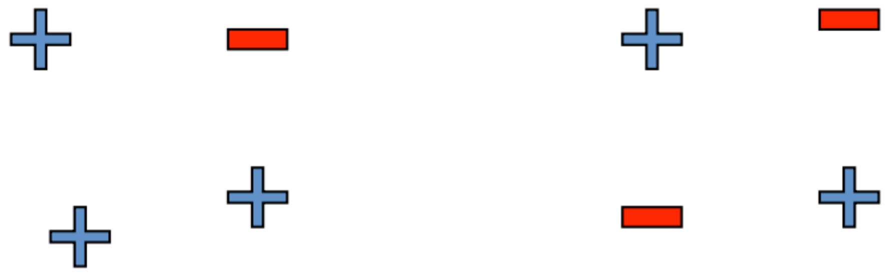


Example: VC Dimension 2-D

- How many points can a linear boundary classify in two dimension?
  - 3 points?



  - 4 points?



PAC Bound via VC Dimension

- VC dimension: Measures relevant size of hypothesis space

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h)\right) + O\left(\sqrt{\frac{VC(\mathcal{H})}{m} \log \frac{m}{VC(\mathcal{H})} + \frac{1}{m} \log \frac{1}{\delta}}\right)$$

- Same bias/variance tradeoff as before, now just a function of VC

- Theory is for binary classification — can be generalized for multi-class and regression