01.03.2019

Statistical Methods in AI (CSE/ECE 471)

Lecture-15: Bias-Variance, Model Selection, Regularization

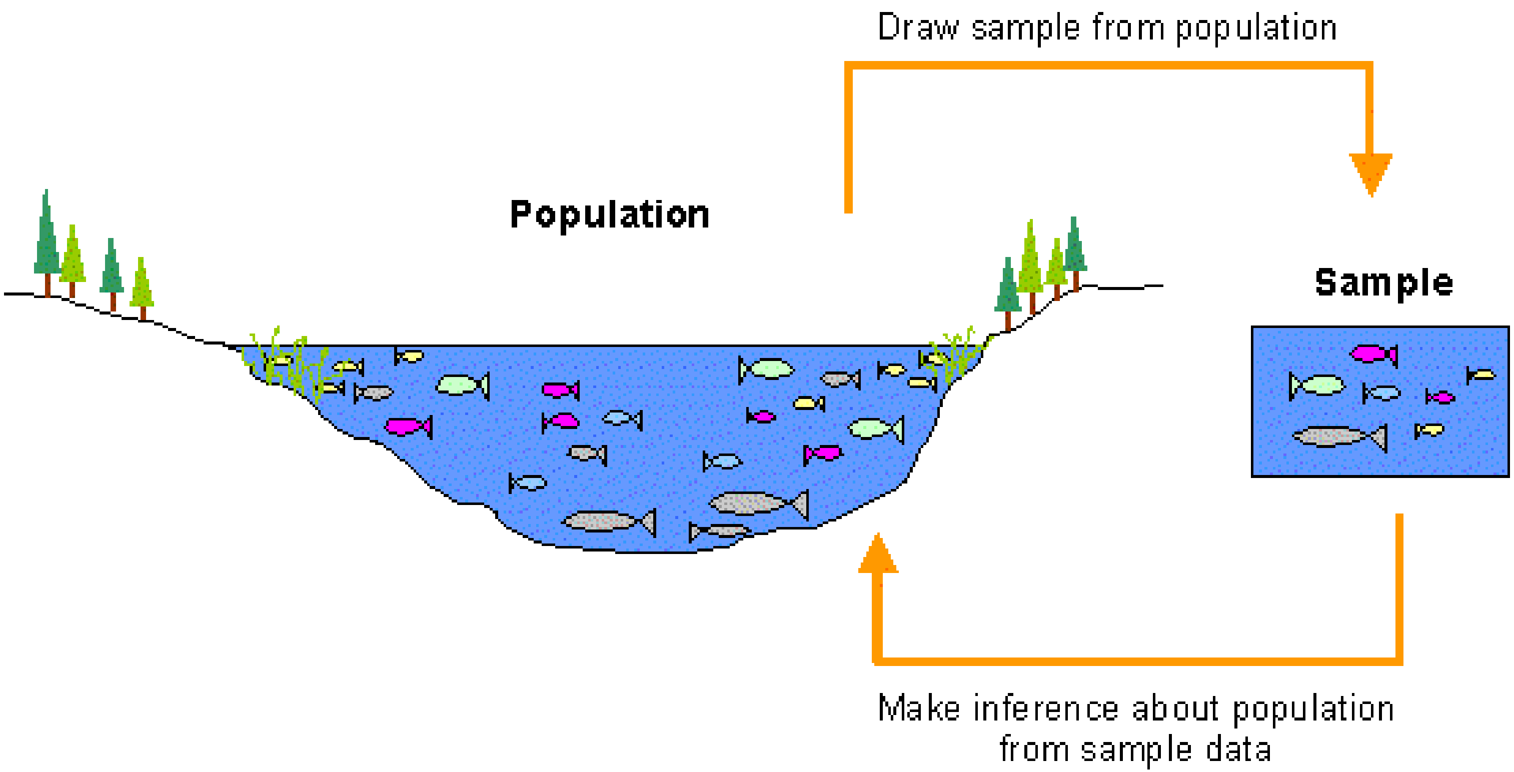
Ravi Kiran

Center for Visual Information Technology (CVIT), IIIT Hyderabad

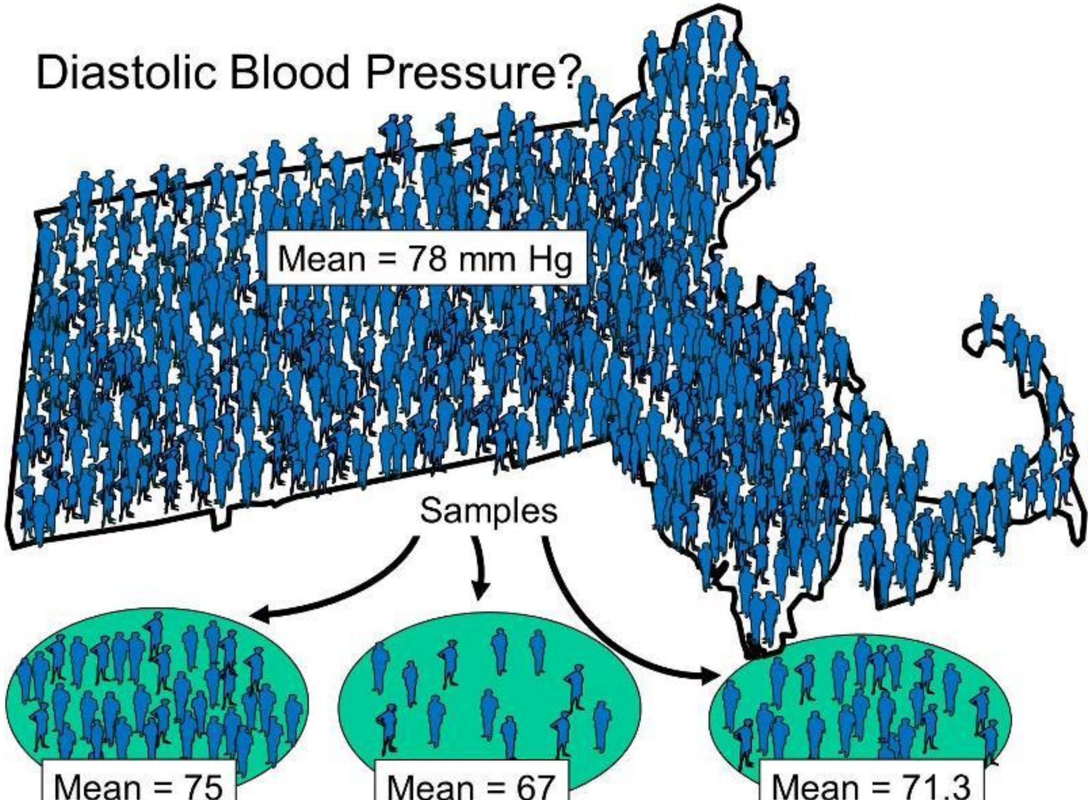
- ML workflow
  - Gather data
  - Select model (based on task)
  - Measure performance [using validation/test data]
  - Deploy model

- ML workflow
  - Gather data [“Data bias”]
  - Select model (based on task) [“Model bias”]
  - Measure performance [using validation/test data]
  - Deploy model

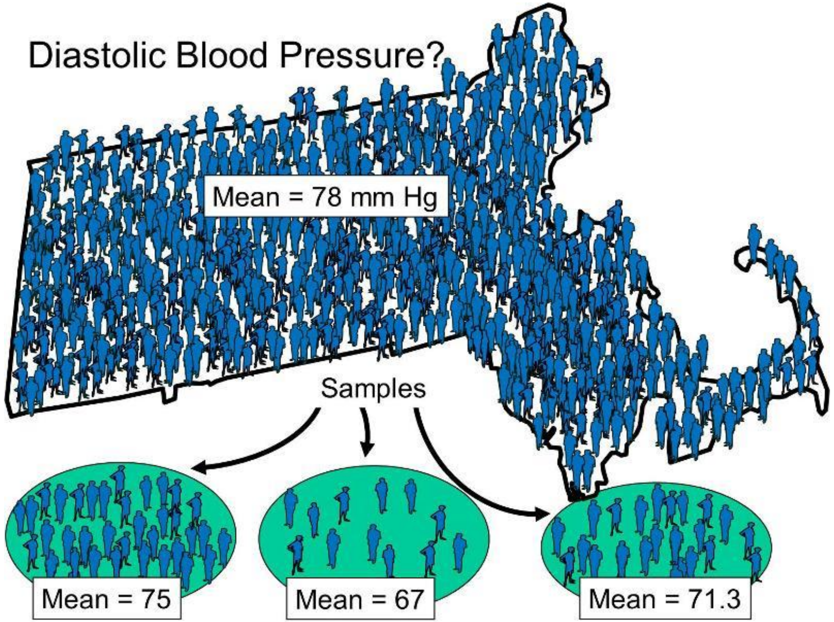
Sample vs Population

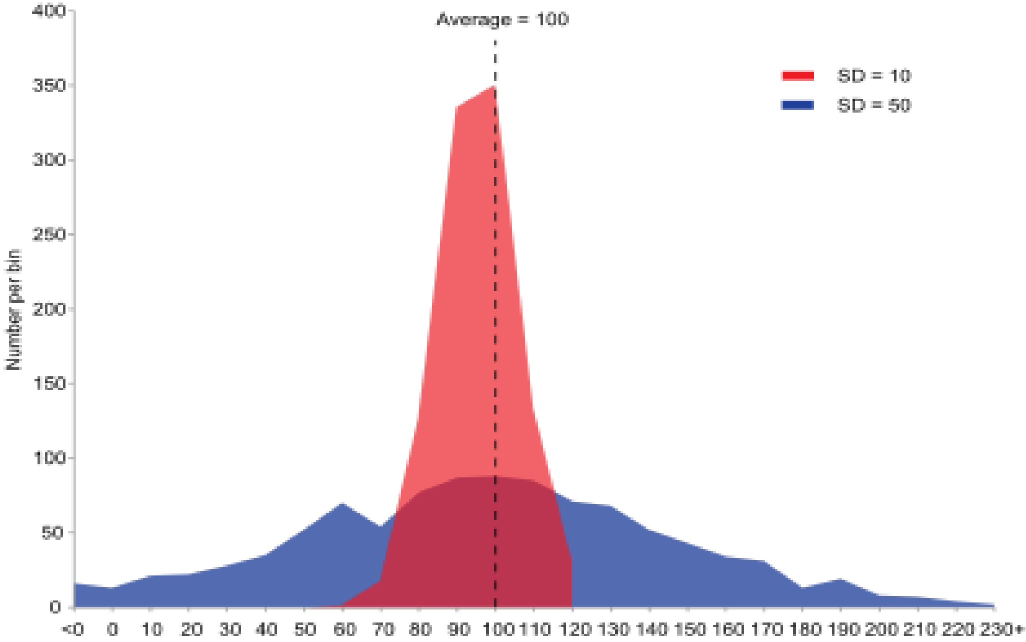


Statistics: Sample vs Population



Statistics: Sample vs Population





- A good model should generalize well

Data Collection Bias

- We place a huge amount of blind faith in the quality of datasets when training ML models.

- Quality
  - Coverage
    - Representative of Population
    - Label coverage
  - Noise
    - Input
    - Output [label noise]

Data Representation Bias

- Input representation
- Output representation

Model Bias

- Which ‘type’ of model ?
- Within the model ‘type’
  - Model-specific choices

Recall: Statistics 101

- Let X be a random variable with possible values $x_i, i = 1 \ldots n$ and with probability distribution $P(X)$
- The *expected value* or *mean* of X is:

$$E[X] = \sum_{i=1}^{n} x_i P(x_i)$$

- If X is continuous, roughly speaking, the sum is replaced by an integral, and the distribution by a density function
- The *variance* of X is:

$$\begin{aligned} Var[X] &= E[(X - E(X))^2] \\ &= E[X^2] - (E[X])^2 \end{aligned}$$

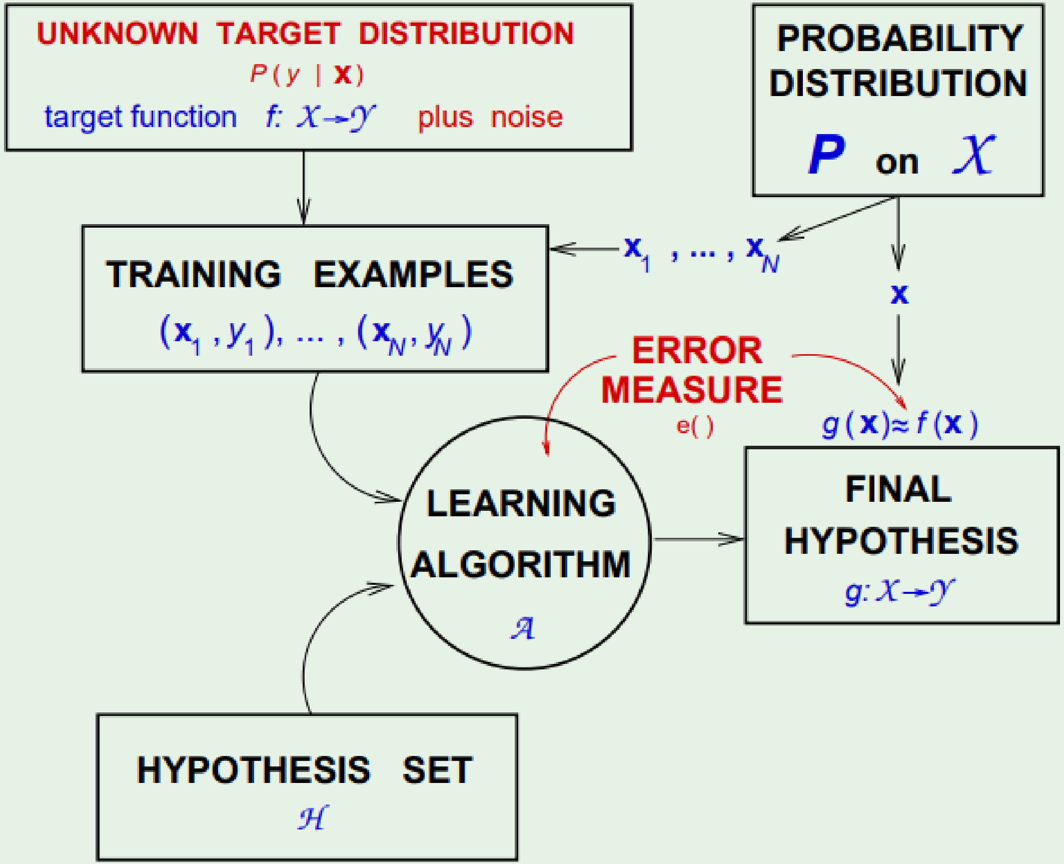
The variance lemma

$$
\begin{aligned}
Var[X] &= E[(X - E[X])^2] \\
&= \sum_{i=1}^{n} (x_i - E[X])^2 P(x_i) \\
&= \sum_{i=1}^{n} (x_i^2 - 2x_i E[X] + (E[X])^2) P(x_i) \\
&= \sum_{i=1}^{n} x_i^2 P(x_i) - 2E[X] \sum_{i=1}^{n} x_i P(x_i) + (E[X])^2 \sum_{i=1}^{n} P(x_i) \\
&= E[X^2] - 2E[X]E[X] + (E[X])^2 \cdot 1 \\
&= E[X^2] - (E[X])^2
\end{aligned}
$$

We will use the form:

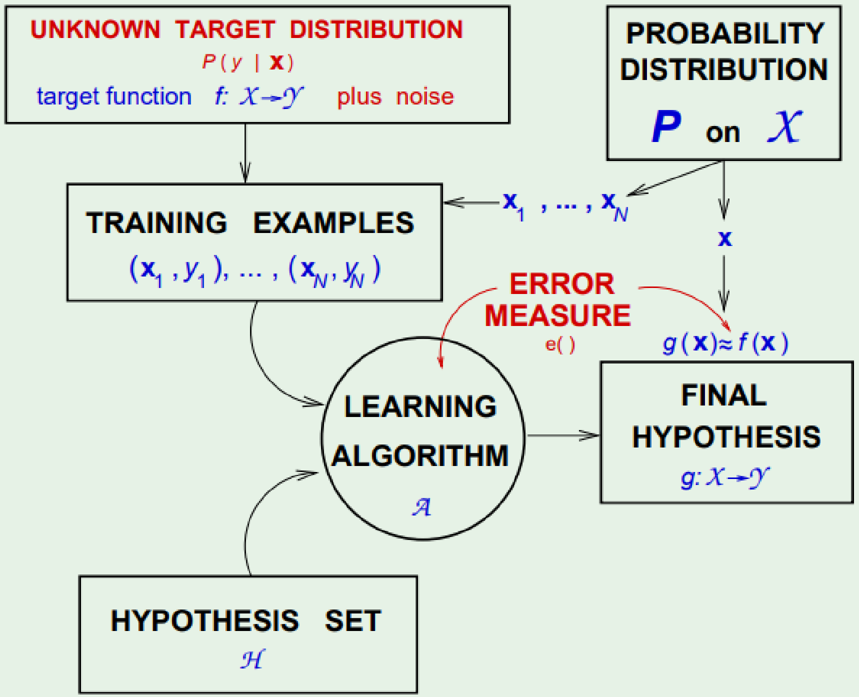
$$E[X^2] = (E[X])^2 + Var[X]$$

The learning diagram - including noisy target



Bias-Variance analysis

The learning diagram - including noisy target



1. How well $\mathcal{H}$ can approximate f

2. How well we can zoom in on a good $h \in \mathcal{H}$

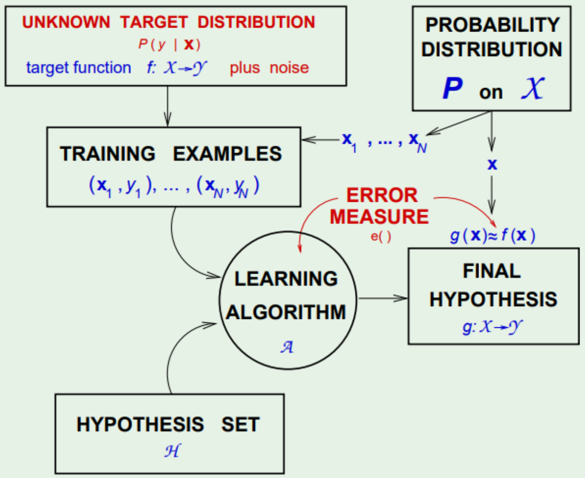
Applies to **real-valued targets** and uses **squared error**

$$E_{\text{out}}(g^{(\mathcal{D})}) \qquad \left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2$$

$$E_{\text{out}}(g^{(\mathcal{D})}) \qquad \left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2$$

$$E_{\text{out}}(g^{(\mathcal{D})}) = \mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

The learning diagram - including noisy target



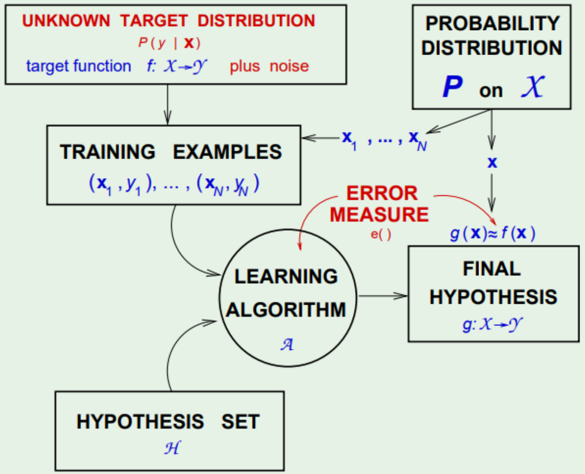
$$E_{\text{out}}(g^{(\mathcal{D})}) \qquad (g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2$$

$$E_{\text{out}}(g^{(\mathcal{D})}) = \mathbb{E}_{\mathbf{x}}\left[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2\right]$$

$$\mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right] = \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathbf{x}}\left[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2\right]\right]$$

$$= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2\right]\right]$$

The learning diagram - including noisy target



Bias-Variance analysis

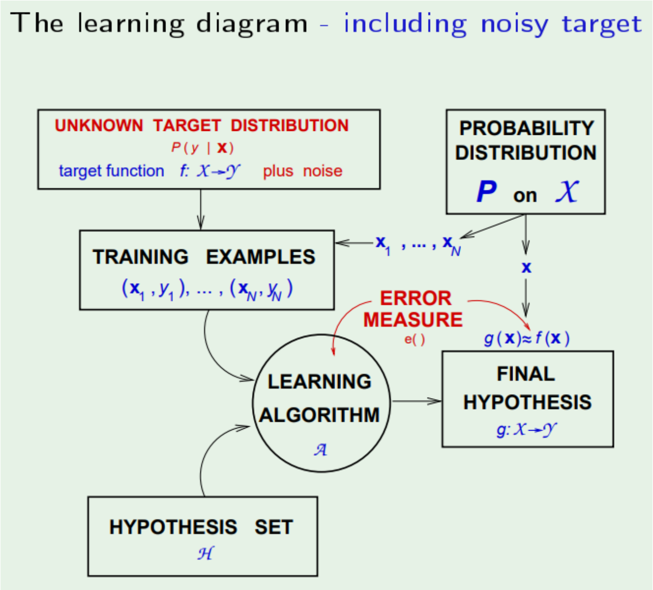
$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

we define the 'average' hypothesis $\bar{g}(\mathbf{x})$:

$$\bar{g}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}(\mathbf{x})\right]$$

Imagine **many** data sets $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_K$

$$\bar{g}(\mathbf{x}) \approx \frac{1}{K}\sum_{k=1}^{K} g^{(\mathcal{D}_k)}(\mathbf{x})$$



Bias-Variance analysis

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\text{var}(\mathbf{x})} + \underbrace{\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2}_{\text{bias}(\mathbf{x})}$$

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right] &= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right] \\
&= \mathbb{E}_{\mathbf{x}}[\text{bias}(\mathbf{x}) + \text{var}(\mathbf{x})] \\
&= \quad \text{bias} \quad + \quad \text{var}
\end{aligned}$$

Bias-Variance analysis

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\text{var}(\mathbf{x})} + \underbrace{\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2}_{\text{bias}(\mathbf{x})}$$

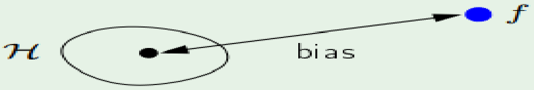
$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right] &= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right] \\
&= \mathbb{E}_{\mathbf{x}}[\text{bias}(\mathbf{x}) + \text{var}(\mathbf{x})] \\
&= \quad \text{bias} \quad + \quad \text{var}
\end{aligned}$$

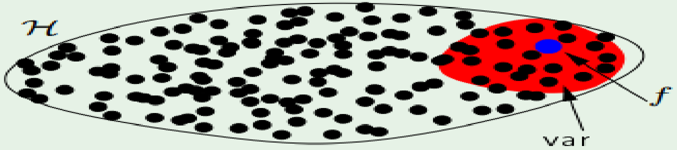
$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\text{var}(\mathbf{x})} + \underbrace{\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2}_{\text{bias}(\mathbf{x})}$$

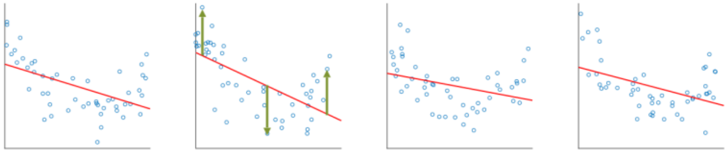
$$\mathbb{E}_{\mathbf{x}}[\text{bias}(\mathbf{x}) + \text{var}(\mathbf{x})]$$

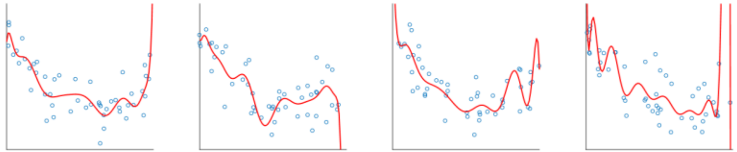
The tradeoff

$$\text{bias} = \mathbb{E}_{\mathbf{x}}\left[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \qquad \text{var} = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]\right]$$

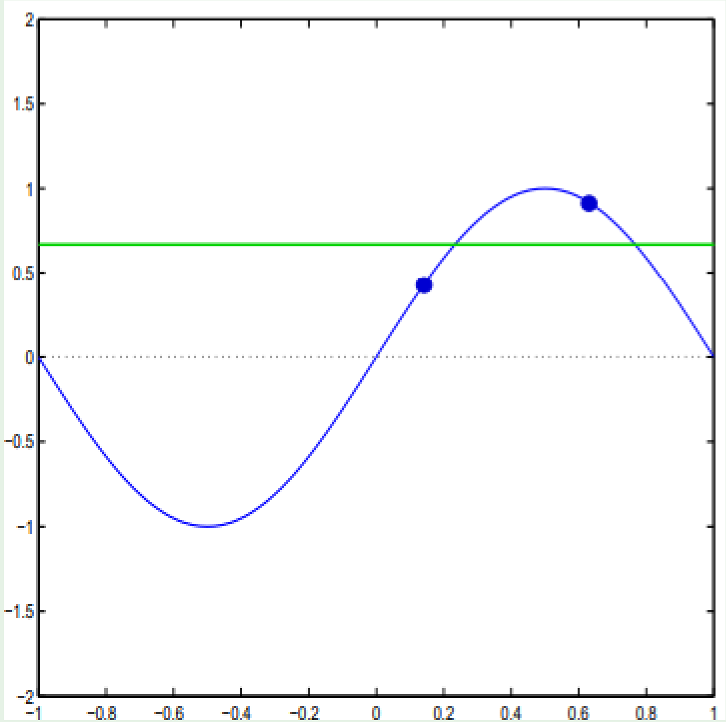


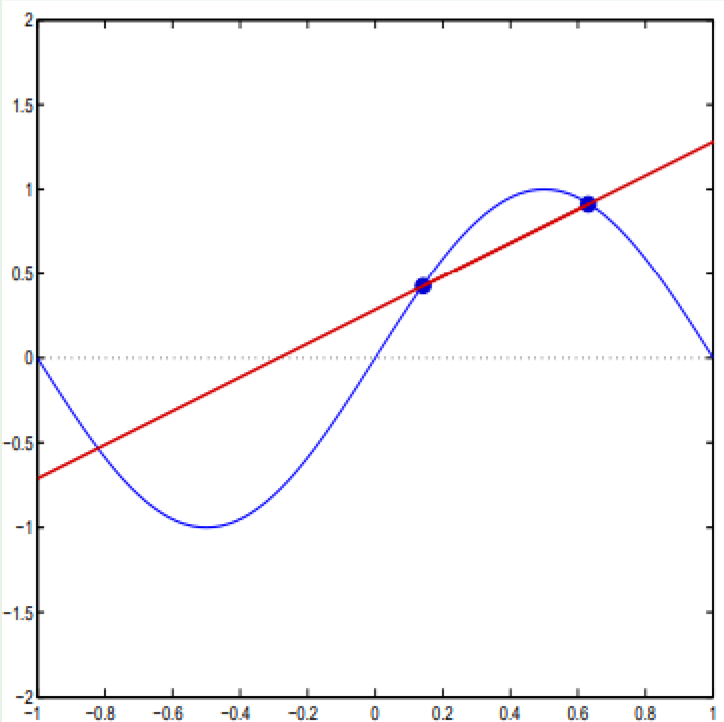




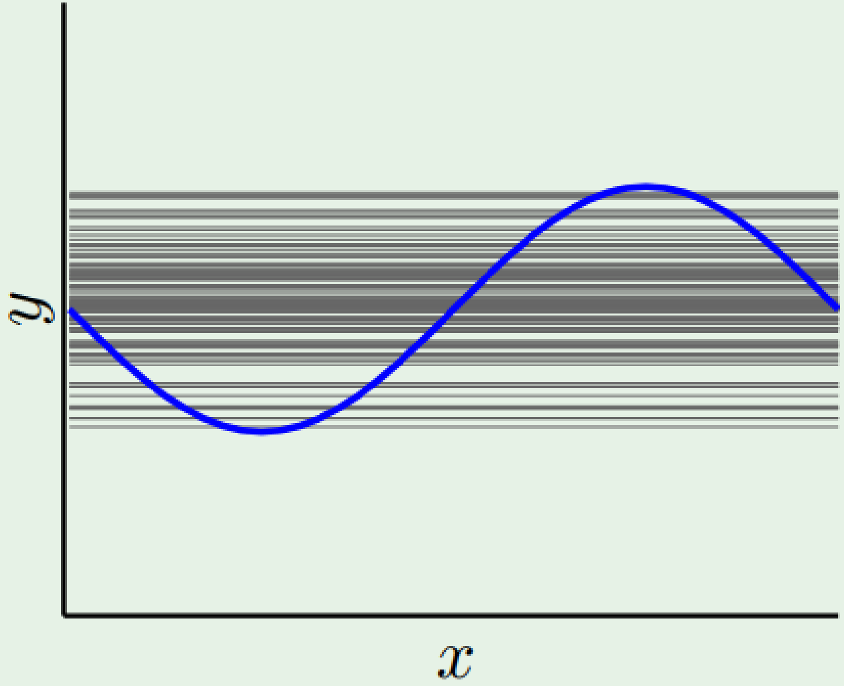


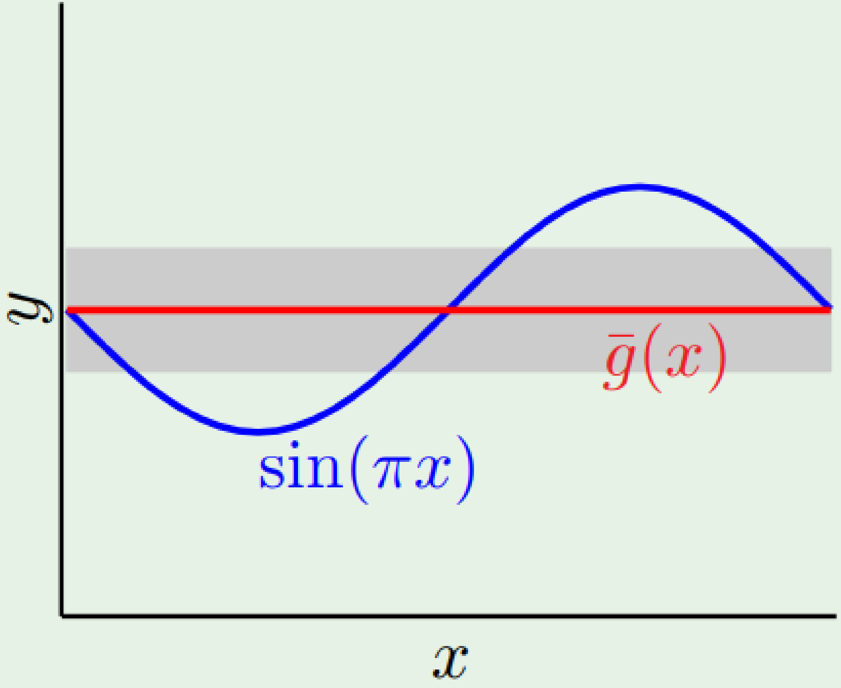
$\mathcal{H}_0$ $\mathcal{H}_1$



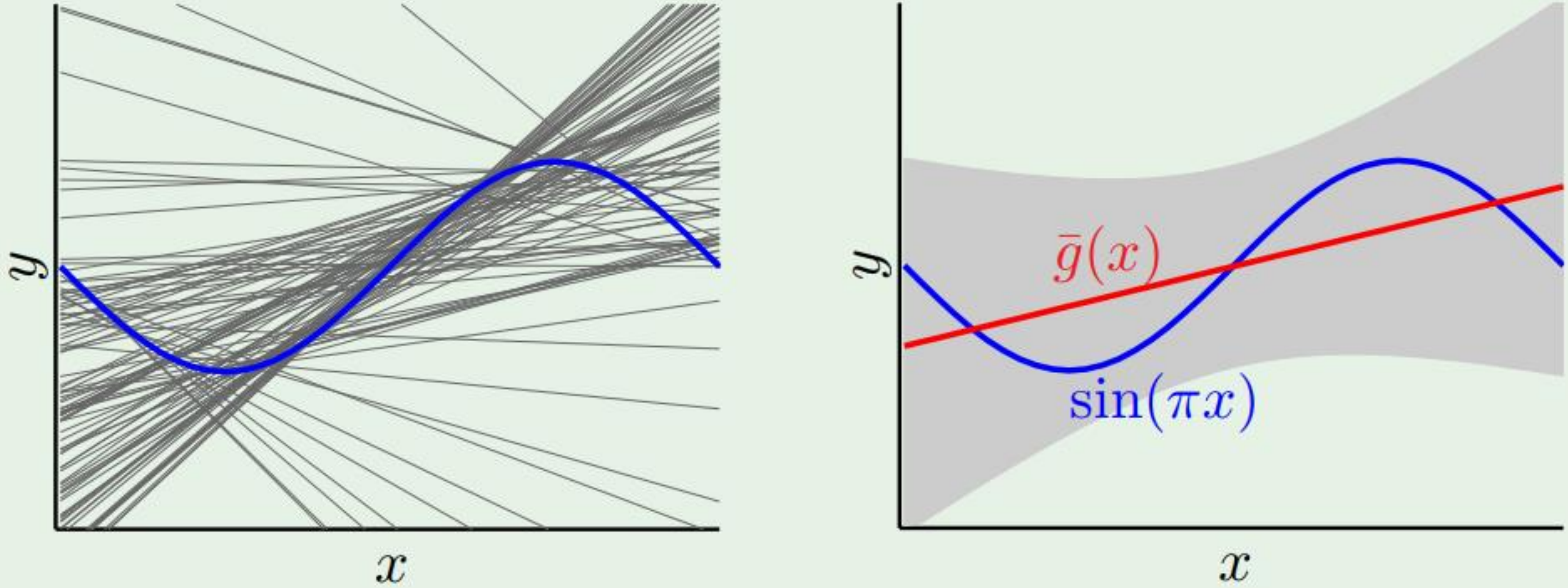


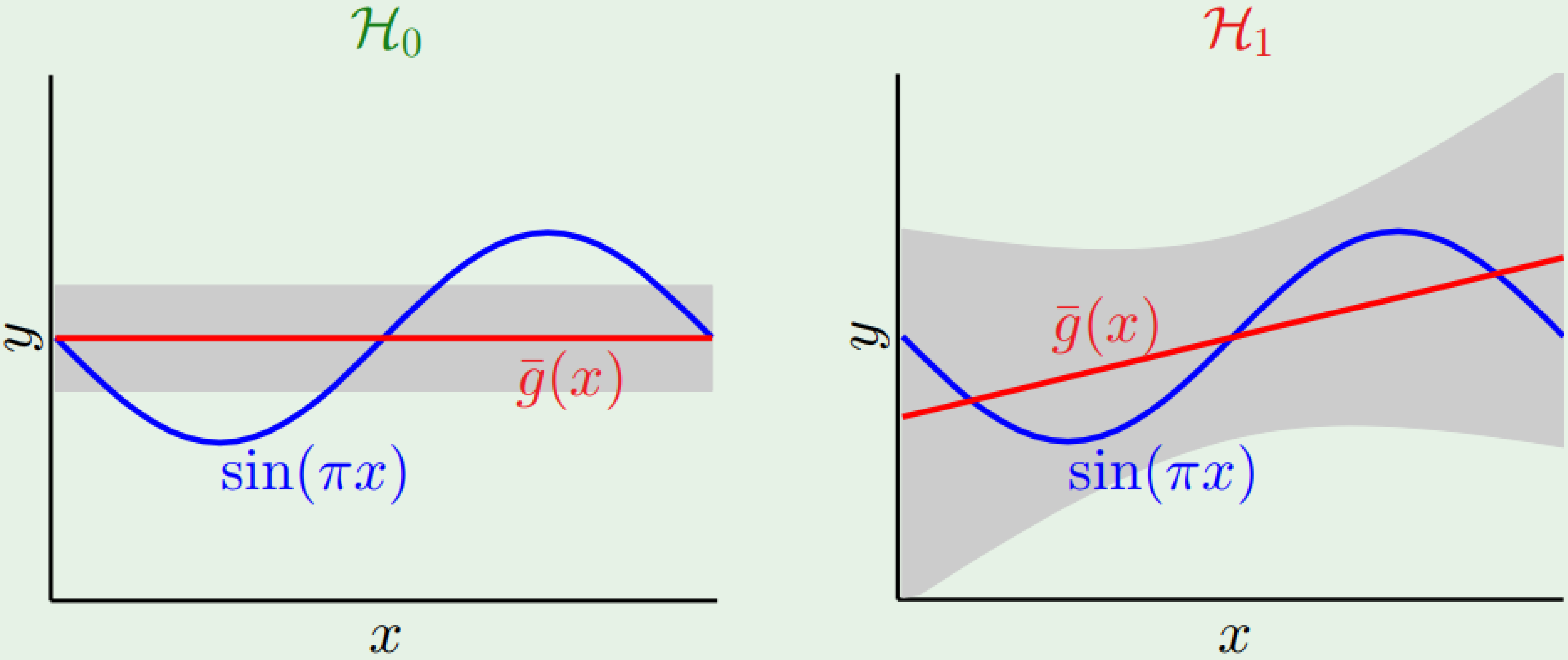
Bias and variance - $\mathcal{H}_0$





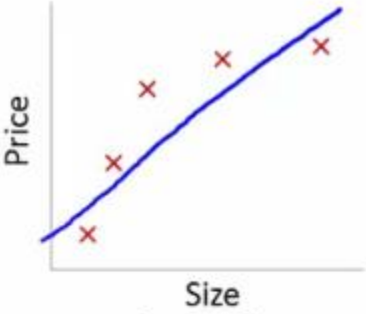
Bias and variance - $\mathcal{H}_1$





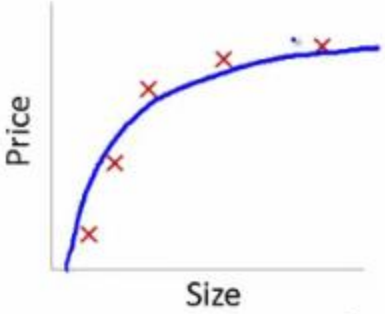
bias $= \mathbf{0.50}$ var $= \mathbf{0.25}$

bias $= \mathbf{0.21}$ var $= \mathbf{1.69}$



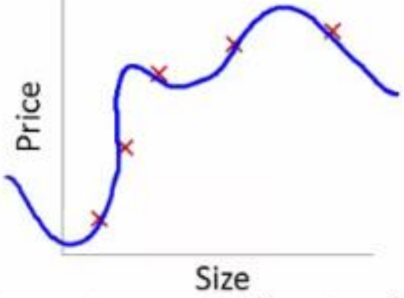
$$\theta_0 + \theta_1 x$$

High bias
(underfit)



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

"Just right"



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High variance
(overfit)

- Bias-Variance decomposition provides insight into model complexity issue
- Limited practical value since it is based on ensembles of data sets
  - In practice there is only a single observed data set
  - If there are many training samples then combine them
    - which would reduce over-fitting for a given model complexity

Regularization

- Remember the intuition: complicated hypotheses lead to overfitting
- Idea: change the error function to *penalize hypothesis complexity*:

$$J(\mathbf{w}) = J_D(\mathbf{w}) + \lambda J_{pen}(\mathbf{w})$$

  This is called *regularization* in machine learning and *shrinkage* in statistics
- λ is called *regularization coefficient* and controls how much we value fitting the data well, vs. a simple hypothesis

Regularization for linear models

- A squared penalty on the weights would make the math work nicely in our case:

$$\frac{1}{2}(\mathbf{\Phi w}-\mathbf{y})^T(\mathbf{\Phi w}-\mathbf{y})+\frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

- This is also known as *L_2 regularization*, or *weight decay* in neural networks
- By re-grouping terms, we get:

$$J_D(\mathbf{w})=\frac{1}{2}(\mathbf{w}^T(\mathbf{\Phi}^T\mathbf{\Phi}+\lambda\mathbf{I})\mathbf{w}-\mathbf{w}^T\mathbf{\Phi}^T\mathbf{y}-\mathbf{y}^T\mathbf{\Phi w}+\mathbf{y}^T\mathbf{y})$$

- Optimal solution (obtained by solving $\nabla_{\mathbf{w}}J_D(\mathbf{w})=0$)

$$\mathbf{w}=(\mathbf{\Phi}^T\mathbf{\Phi}+\lambda I)^{-1}\mathbf{\Phi}^T\mathbf{y}$$

What L_2 regularization does

$$\arg\min_{\mathbf{w}} \frac{1}{2}(\mathbf{\Phi w} - \mathbf{y})^T(\mathbf{\Phi w} - \mathbf{y}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi} + \lambda I)^{-1}\mathbf{\Phi}^T\mathbf{y}$$

- If $\lambda = 0$, the solution is the same as in regular least-squares linear regression
- If $\lambda \to \infty$, the solution $\mathbf{w} \to 0$
- Positive λ will cause the magnitude of the weights to be smaller than in the usual linear solution
- This is also called *ridge regression*, and it is a special case of Tikhonov regularization
- A different view of regularization: we want to optimize the error while keeping the L_2 norm of the weights, $\mathbf{w}^T\mathbf{w}$, bounded.

L_2 Regularization for linear models revisited

- Optimization problem: minimize error while keeping norm of the weights bounded

$$\begin{aligned} \min_{\mathbf{w}} J_D(\mathbf{w}) &= \min_{\mathbf{w}} (\mathbf{\Phi w} - \mathbf{y})^T (\mathbf{\Phi w} - \mathbf{y}) \\ \text{such that } \mathbf{w}^T \mathbf{w} &\leq \eta \end{aligned}$$

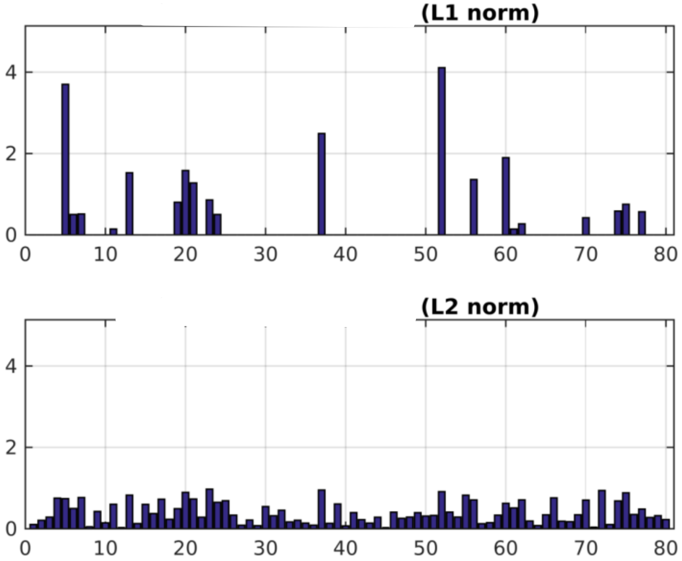
- The Lagrangian is:

$$L(\mathbf{w}, \lambda) = J_D(\mathbf{w}) - \lambda(\eta - \mathbf{w}^T\mathbf{w}) = (\mathbf{\Phi w} - \mathbf{y})^T (\mathbf{\Phi w} - \mathbf{y}) + \lambda \mathbf{w}^T \mathbf{w} - \lambda\eta$$

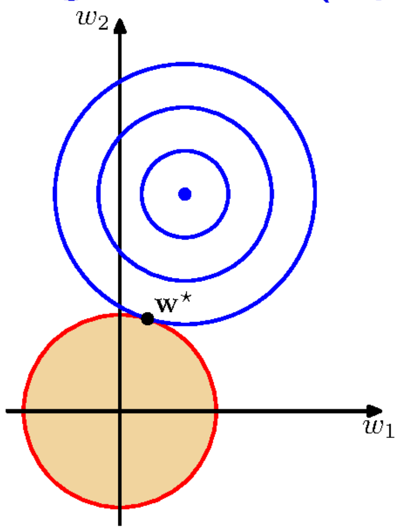
- For a fixed λ, and $\eta = \lambda^{-1}$, the best $\mathbf{w}$ is the same as obtained by weight decay

Pros and cons of L_2 regularization

- If λ is at a "good" value, regularization helps to avoid overfitting
- Choosing λ may be hard: cross-validation is often used
- If there are irrelevant features in the input (i.e. features that do not affect the output), L_2 will give them small, but non-zero weights.
- Ideally, irrelevant input should have weights exactly equal to 0.



Visualizing regularization (2 parameters)



$$\mathbf{w}^* = (\mathbf{\Phi}^T\mathbf{\Phi} + \lambda I)^{-1}\mathbf{\Phi}\mathbf{y}$$

L_1 Regularization for linear models

- Instead of requiring the L_2 norm of the weight vector to be bounded, make the requirement on the L_1 norm:

$$\begin{aligned}\min_{\mathbf{w}} J_D(\mathbf{w}) &= \min_{\mathbf{w}} (\mathbf{\Phi w} - \mathbf{y})^T (\mathbf{\Phi w} - \mathbf{y}) \\ \text{such that } \sum_{i=1}^{n} |w_i| &\leq \eta\end{aligned}$$

- This yields an algorithm called Lasso (Tibshirani, 1996)

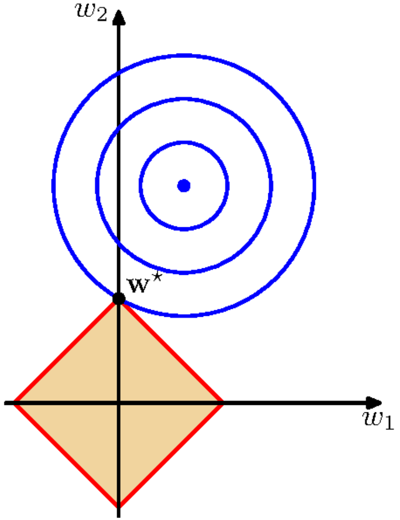
Solving L_1 regularization

- The optimization problem is a quadratic program
- There is one constraint for each possible sign of the weights (2^n constraints for n weights)
- For example, with two weights:

$$\begin{array}{rcl} \min\limits_{w_1,w_2} & & \sum\limits_{j=1}^{m}(y_j - w_1x_1 - w_2x_2)^2 \\ \text{such that } w_1 + w_2 & \leq & \eta \\ w_1 - w_2 & \leq & \eta \\ -w_1 + w_2 & \leq & \eta \\ -w_1 - w_2 & \leq & \eta \end{array}$$

- Solving this program directly can be done for problems with a small number of inputs

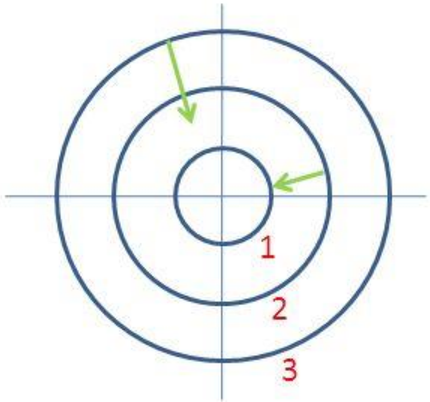
Visualizing L_1 regularization



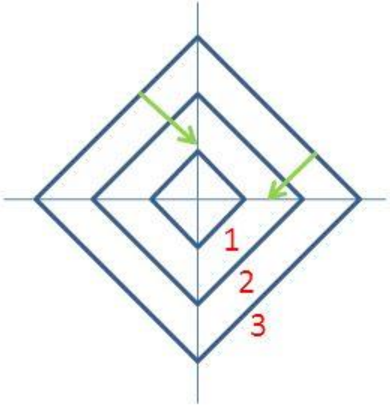
- If λ is big enough, the circle is very likely to intersect the diamond at one of the corners
- This makes L_1 regularization much more likely to make some weights *exactly* 0

A nice property of L_1

- Gradients of L_2- and L_1-norm



Negative gradient of L_2-norm always points directly toward 0

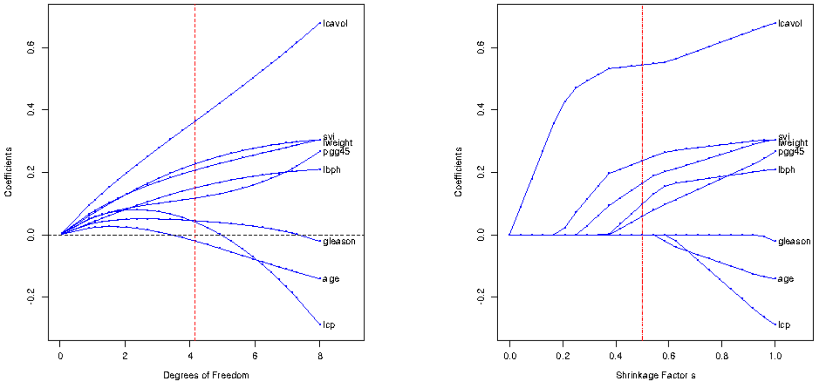


"Negative gradient" of L_1-norm (direction of steepest descent) points toward coordinate axes

Pros and cons of L_1 regularization

- If there are irrelevant input features, Lasso is likely to make their weights 0, while L_2 is likely to just make all weights small
- Lasso is biased towards providing *sparse solutions* in general
- Lasso optimization is computationally more expensive than L_2
- More efficient solution methods have to be used for large numbers of inputs (e.g. least-angle regression, 2003).
- L_1 methods of various types are very popular

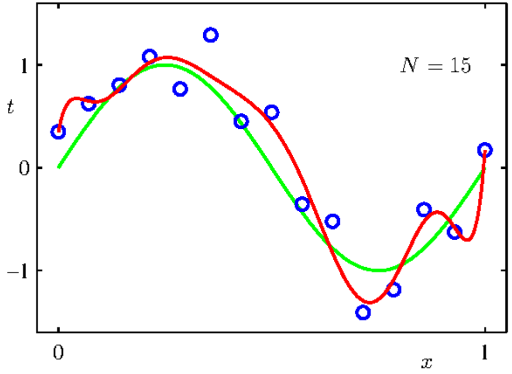
Example of L1 vs L2 effect

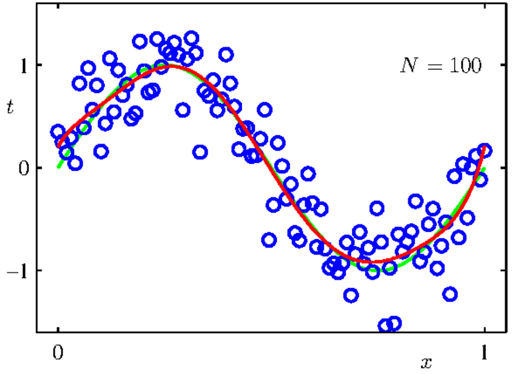


- Note the sparsity in the coefficients induces by L_1
- Lasso is an efficient way of performing the L_1 optimization

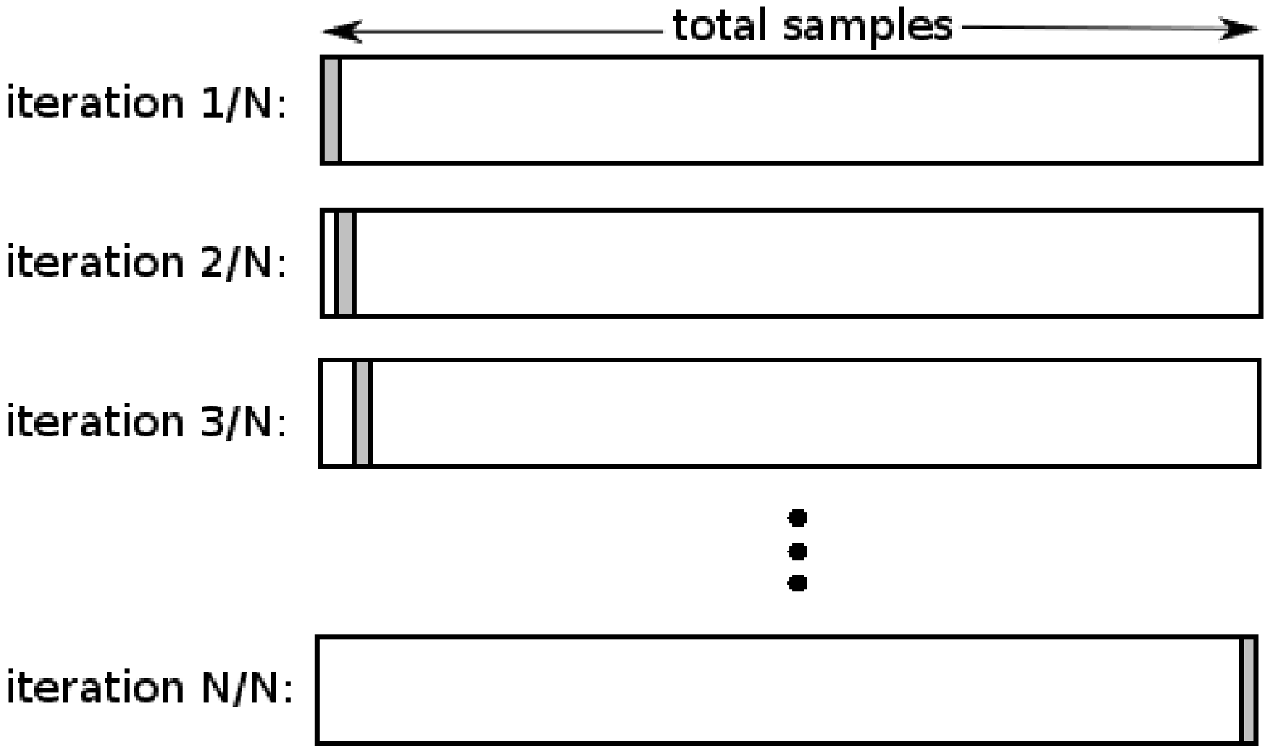
More on overfitting

- Overfitting depends on the amount of data, relative to the complexity of the hypothesis
- With more data, we can explore more complex hypotheses spaces, and still find a good solution



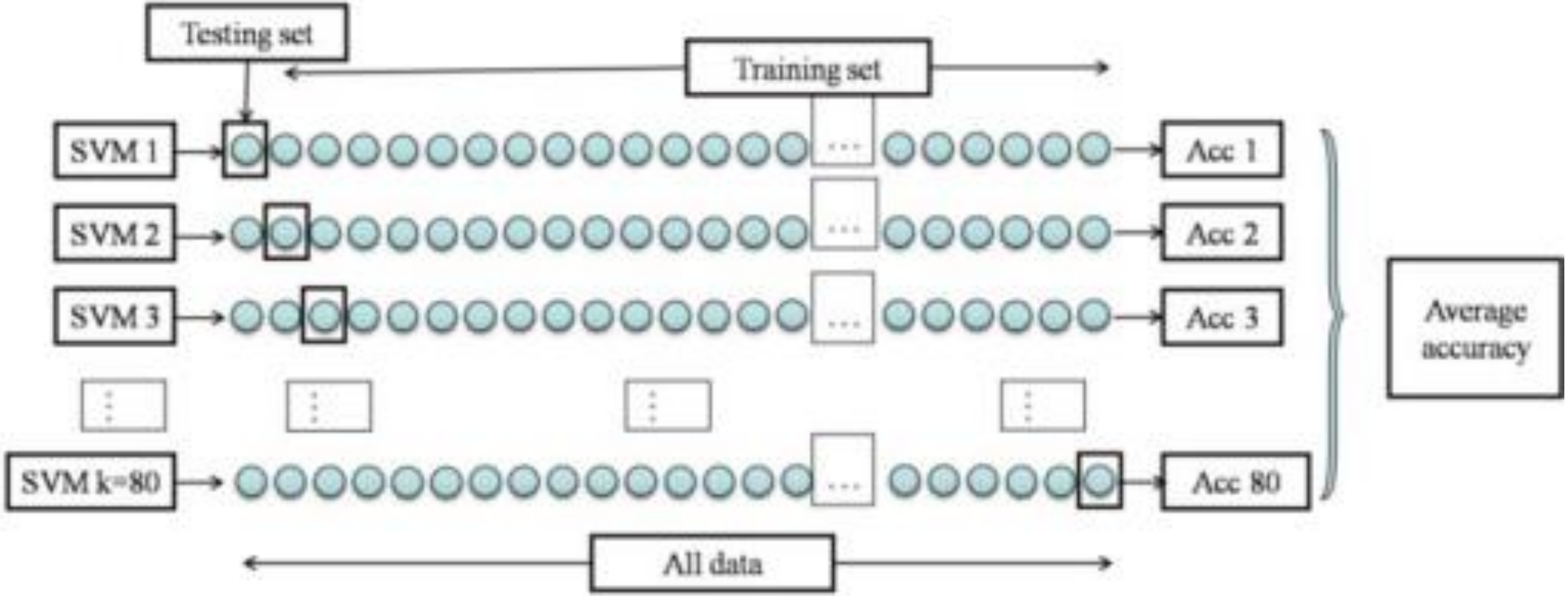


Leave One Out Cross Validation



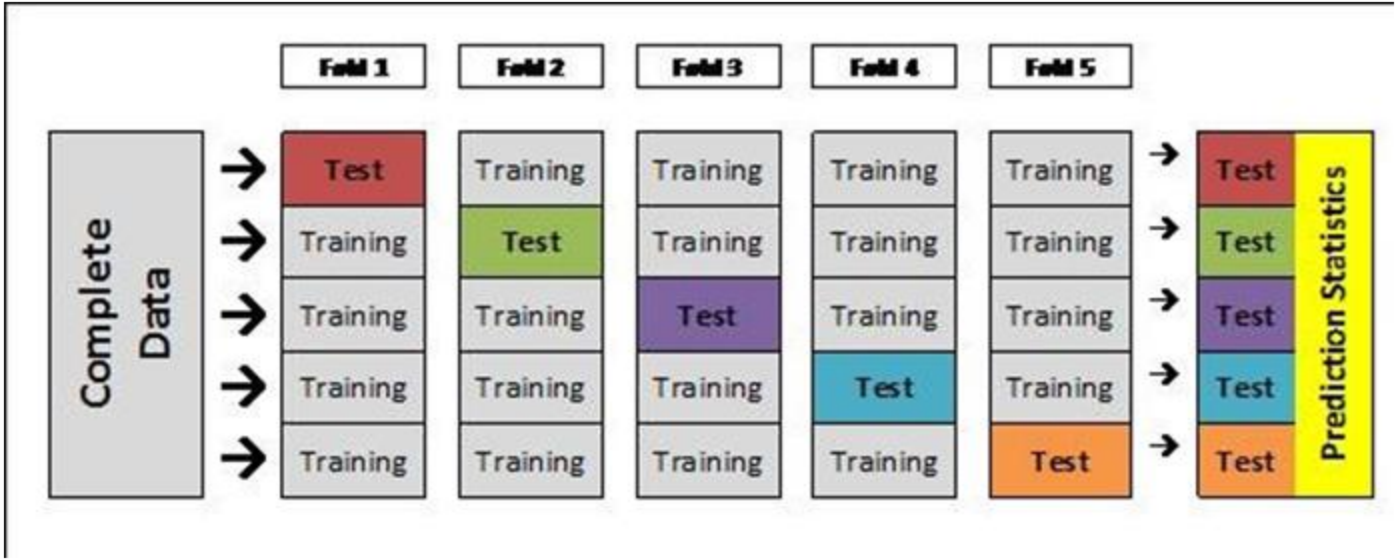
Leave One Out Cross Validation

High variance



K-fold Cross Validation

Fold → bias



Assess model's bias-variance characteristics using k-fold Cross Validation?

- With *k*-fold CV , you get *k* different estimates of your model's error- $e_1, e_2, e_3, \ldots, e_k$
- Mean(errors) ~ 0 ➔ low bias
- Variance(errors) ~ 0 ➔ low variance