CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Zoom Link

Announcements

- Last quiz on 12/7
- Writeup instructions on canvas
- Remember to email TAs if you are submitting a regrade of P1

Recap

- Adaboost answers the question: can a w___ l___ be boosted into a s___ l___?
- It maintains a w___ for each example.
- Each iteration it builds a c____ with the w___ e___. If the w___ t___ e___ of this classifier is __ or ___ it stops.
- Else, it updates the weight of each example. Correctly classified examples have their weights ____. Incorrect ones have their weights ____.
- The classifiers also have weights, which are i_____ p____ to their e_____.
- For a new example, the label is assigned through a w___ v___.
- Adaboost e____ d____ the training loss as a function of the number of t____.
- This still may not lead to overfitting because Adaboost can also m____ the m___. Alternatively, using s____ b____ c____ can prevent overfitting to noise.
- How do algorithms like naïve Bayes handle weighted data?
- What about SVMs?

Today

- Bias-Variance Analysis
- Feature Selection and Dimensionality Reduction

Analysis of Learning Algorithms

- Many different algorithms (trees, ANNs, SVMs, NB, LR) and statistical methods for evaluation and comparison
- Now, *theoretical analysis* of concept learnability
- Key question:
  - What are the sources of generalization error?

Bias-Variance Analysis

- Idea: try to decompose the generalization error of any concept class into components

- Gives quantitative insight into inductive bias and other sources of error in learned models

- But is not algorithm-specific

Problem Setup

- Given data $(\mathbf{x}_i, y_i)$ where $y_i = f(\mathbf{x}_i) + \varepsilon$, $\varepsilon \sim N(0, \sigma)$

- We produce a concept $h(x_i)$ to minimize squared loss
  - For illustration, we'll use a linear model (does not affect the analysis---this holds for *any* concept class)

$$\hat{h} = \arg\min_h (y_i - h(x_i))^2$$

Example: 20 points

$$y = x + 2\sin(1.5x) + N(0,0.2)$$

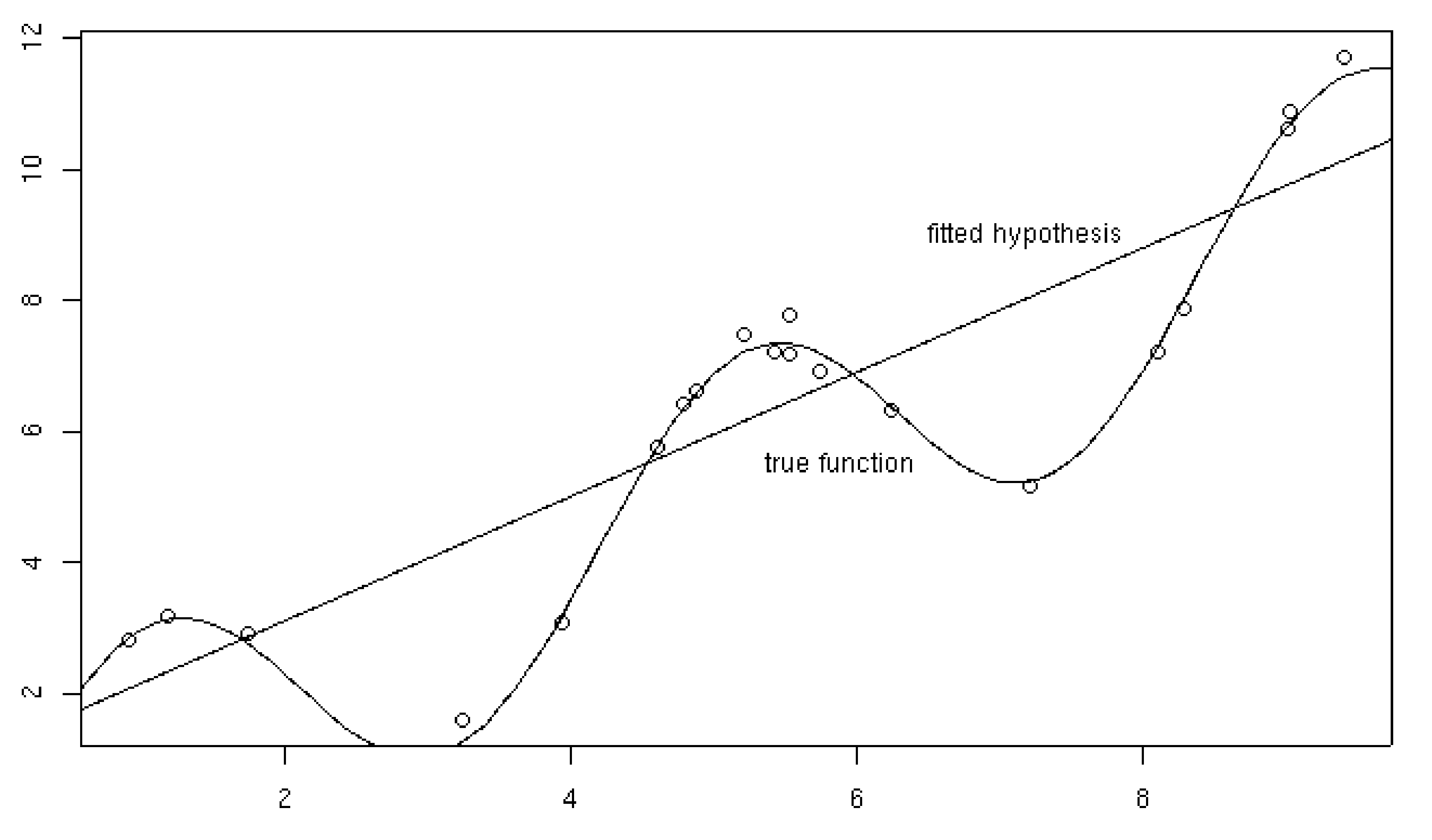

Figure due to Tom Dietterich, Oregon State U.

50 fits (20 examples each)

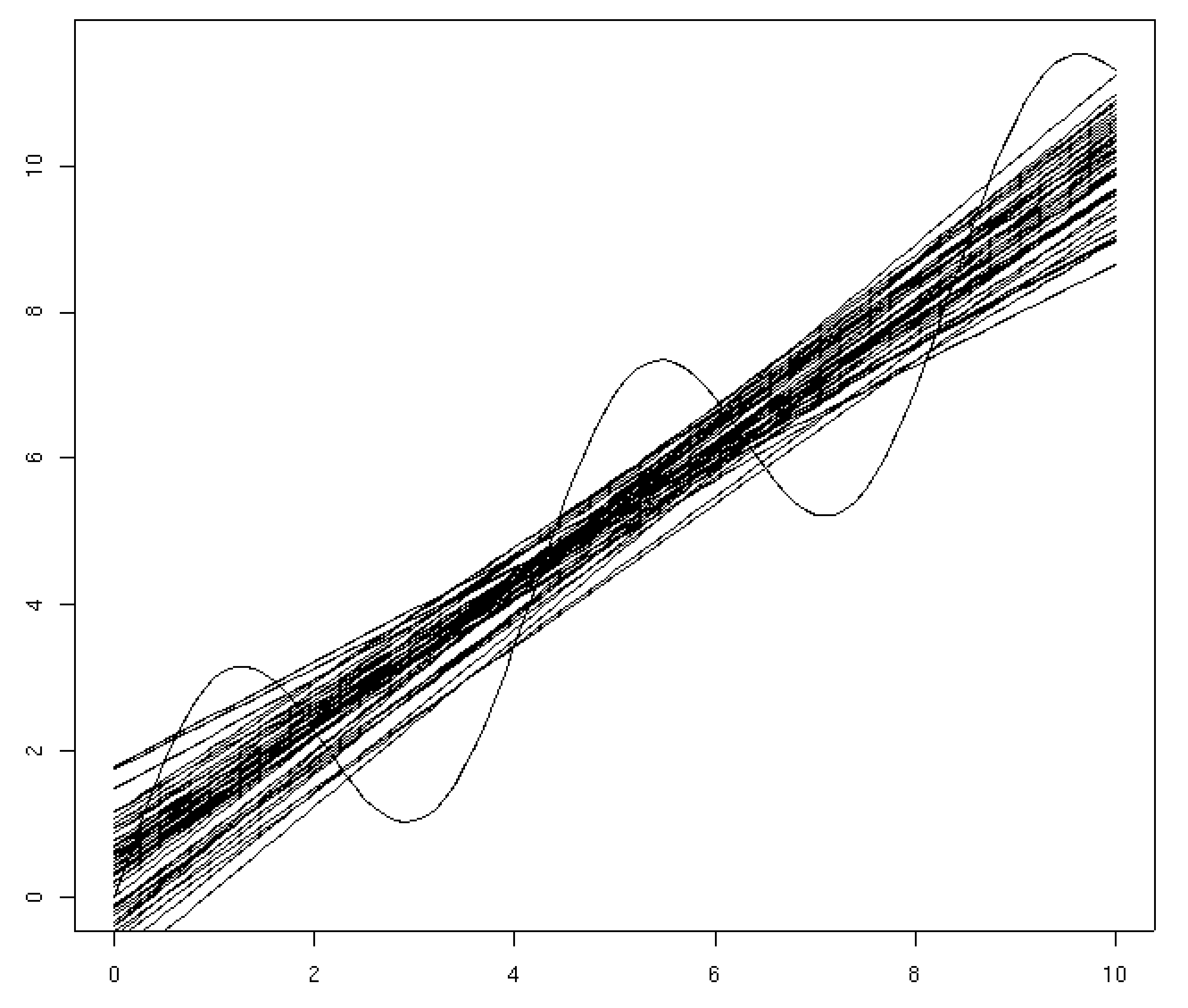

Figure due to Tom Dietterich, Oregon State U.

Bias-Variance Analysis

- For a random new example $\mathbf{x}_{new}$, we want to understand the *expected prediction error*:

Label

Training set

$$E_{S,y_{new}}\left[(y_{new}-h(\mathbf{x}_{new}))^2\right]$$

$$y_{new}=f(\mathbf{x}_{new})+\varepsilon;\quad \varepsilon\sim N(0,\sigma)$$

$$y_{new}\sim N(f(\mathbf{x}_{new}),\sigma)$$

- Denote $E(R)$ as $\overline{R}$

Result

- For any random variable R

$$V(R) = E\left[(R-\bar{R})^2\right]$$

Def. of variance

$$= E\left[R^2 - 2R\bar{R} + \bar{R}^2\right]$$

$$= E\left[R^2\right] - 2E\left[R\bar{R}\right] + E\left[\bar{R}^2\right]$$

$$= E\left[R^2\right] - 2\bar{R}E\left[R\right] + \bar{R}^2$$

$\bar{R}$ is a constant

$$= E\left[R^2\right] - \bar{R}^2$$

$$E\left[R^2\right] = V(R) + \bar{R}^2$$

Bias-Variance Decomposition

$$E_{S,y}\left[(y-h(\mathbf{x}))^2\right] = E_{S,y}\left[h(\mathbf{x})^2 - 2yh(\mathbf{x}) + y^2\right]$$

$$= E_S\left[h(\mathbf{x})^2\right] - 2E_y(y)E_S(h(\mathbf{x})) + E_y(y^2)$$

Note $\mathbf{x}$, y=$\mathbf{x}_{new}$, y_{new}

$$= V(h(\mathbf{x})) + \overline{h(\mathbf{x})}^2 - 2f(\mathbf{x})\overline{h(\mathbf{x})} + V(y) + f(\mathbf{x})^2$$

From previous slide

Since $y \sim N(f(x),\sigma)$

From previous slide

Bias-Variance Decomposition

$$E\left[(y-h(\mathbf{x}))^2\right]$$

$$=V(h(\mathbf{x}))+\overline{h(\mathbf{x})}^2-2f(\mathbf{x})\overline{h(\mathbf{x})}+V(y)+f(\mathbf{x})^2$$

$$=V(h(\mathbf{x}))+V(y)+\left[\overline{h(\mathbf{x})}^2-2f(\mathbf{x})\overline{h(\mathbf{x})}+f(\mathbf{x})^2\right]$$

$$=V(h(\mathbf{x}))+V(y)+\left[\overline{h(\mathbf{x})}-f(\mathbf{x})\right]^2$$

$$=V(h(\mathbf{x}))+\sigma^2+\left[\overline{h(\mathbf{x})}-f(\mathbf{x})\right]^2$$

Bias-Variance Decomposition

Expected prediction error

$$E\left[(y-h(\mathbf{x}))^2\right] = V(h(\mathbf{x})) + \sigma^2 + \left[\overline{h(\mathbf{x})} - f(\mathbf{x})\right]^2$$

σ^2 — Noise error: Error in learned model's predictions due to noise in y

$V(h(\mathbf{x}))$ — Variance error: Error in learned model's predictions due to choice of training sample

$\left[\overline{h(\mathbf{x})} - f(\mathbf{x})\right]^2$ — Bias error: Systematic error in predictions due to choice of h as concept class

Bias, Variance, and Noise

- Variance describes how much the prediction error varies as h is trained using different training sets
- Bias describes the average error of h across all training sets
  - Using h, on average, we can't approximate $f(\mathbf{x})$ better than this
  - **This quantifies inductive bias**
- Noise describes how much y varies from $f(\mathbf{x})$

50 fits (20 examples each)

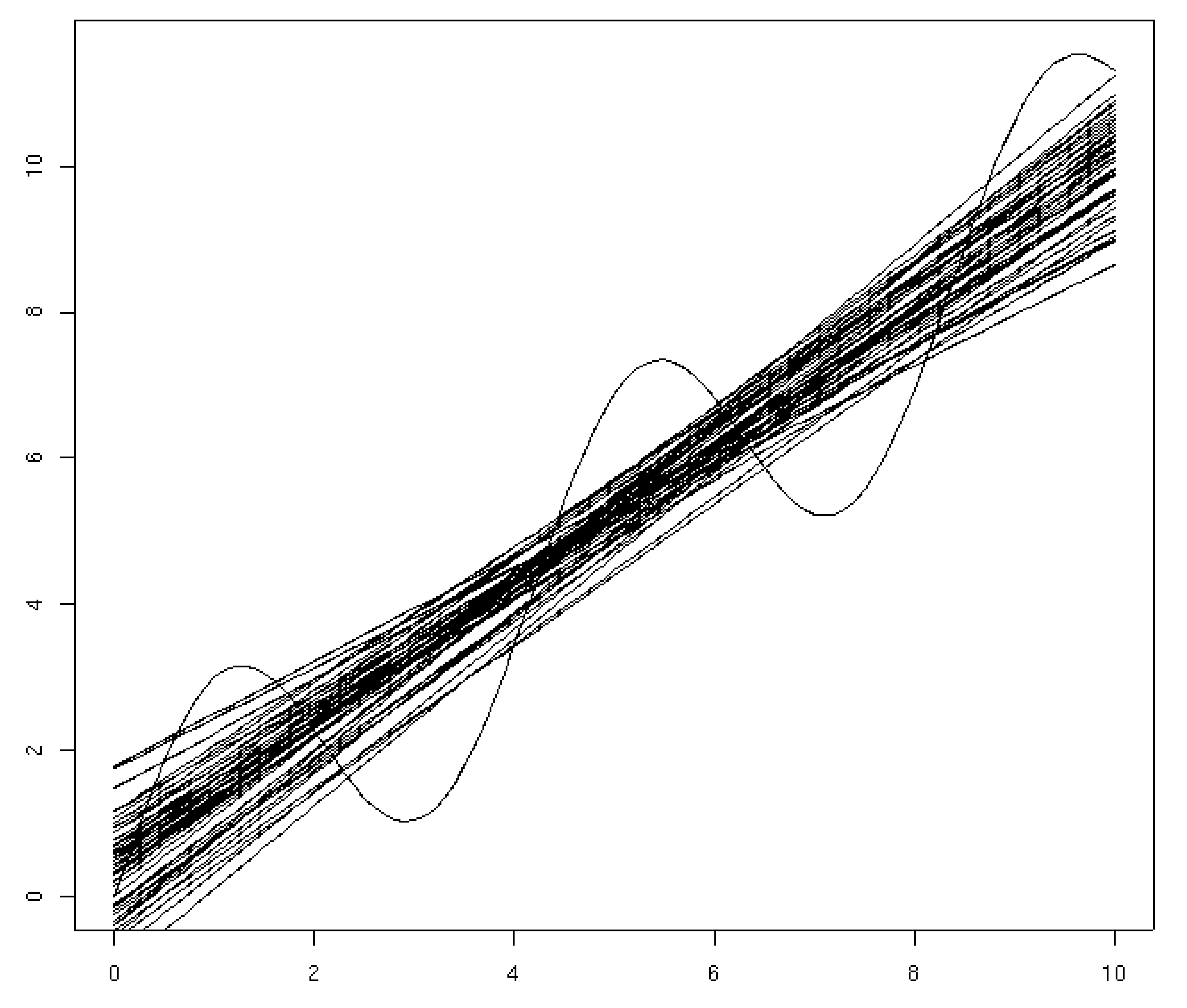

Figure due to Tom Dietterich, Oregon State U.

Bias

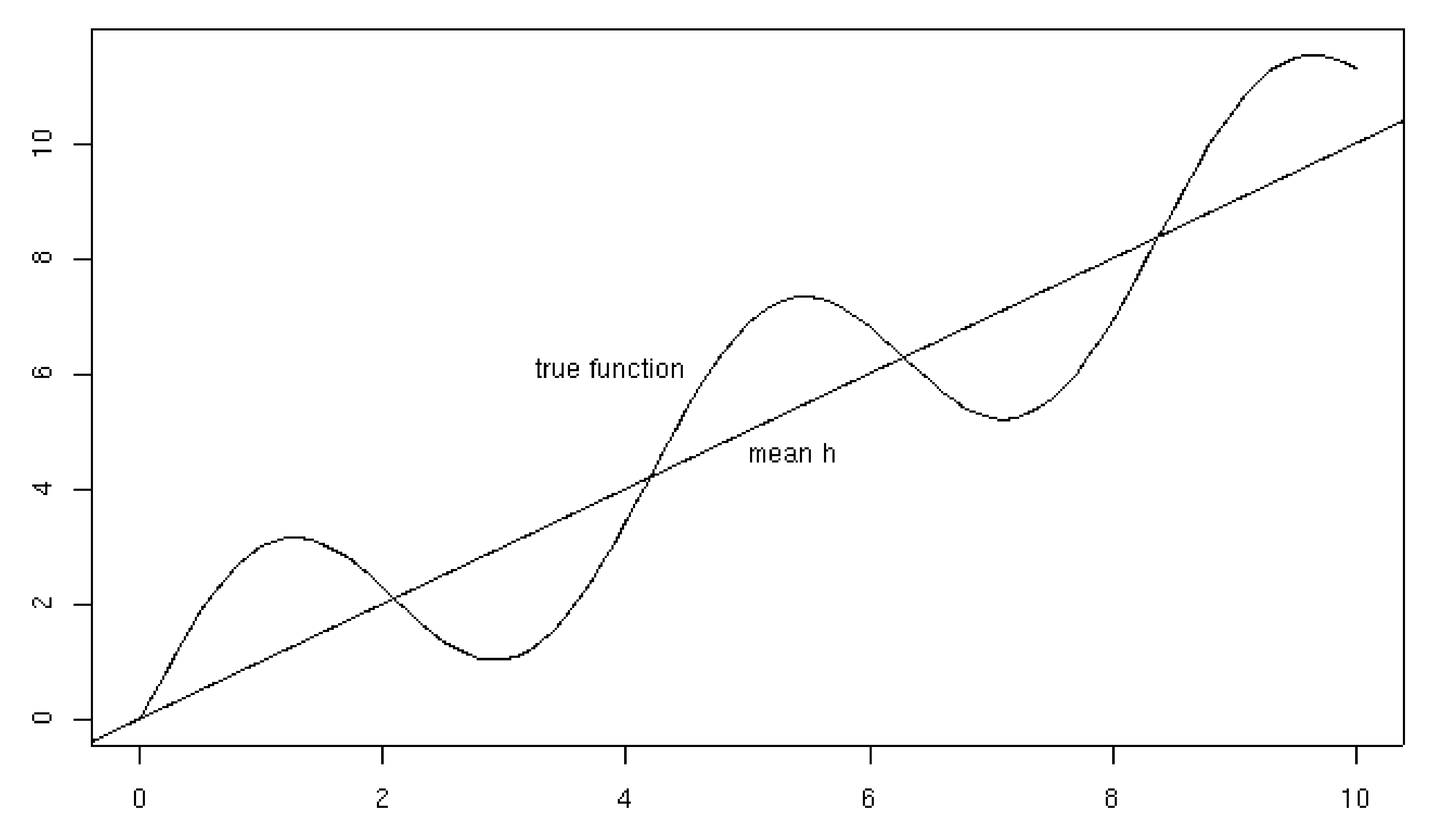

Figure due to Tom Dietterich, Oregon State U.

Variance

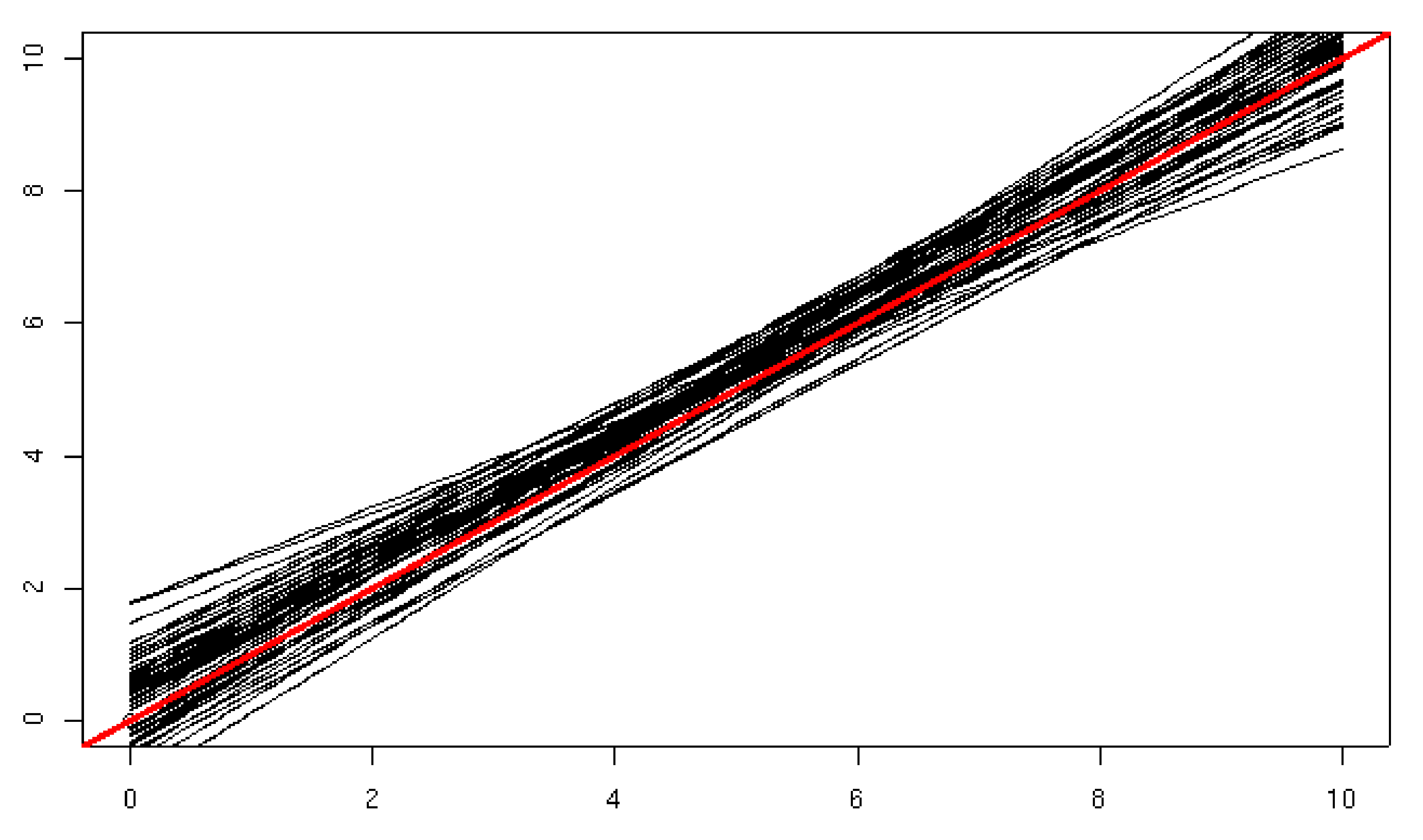

Figure due to Tom Dietterich, Oregon State U.

Noise

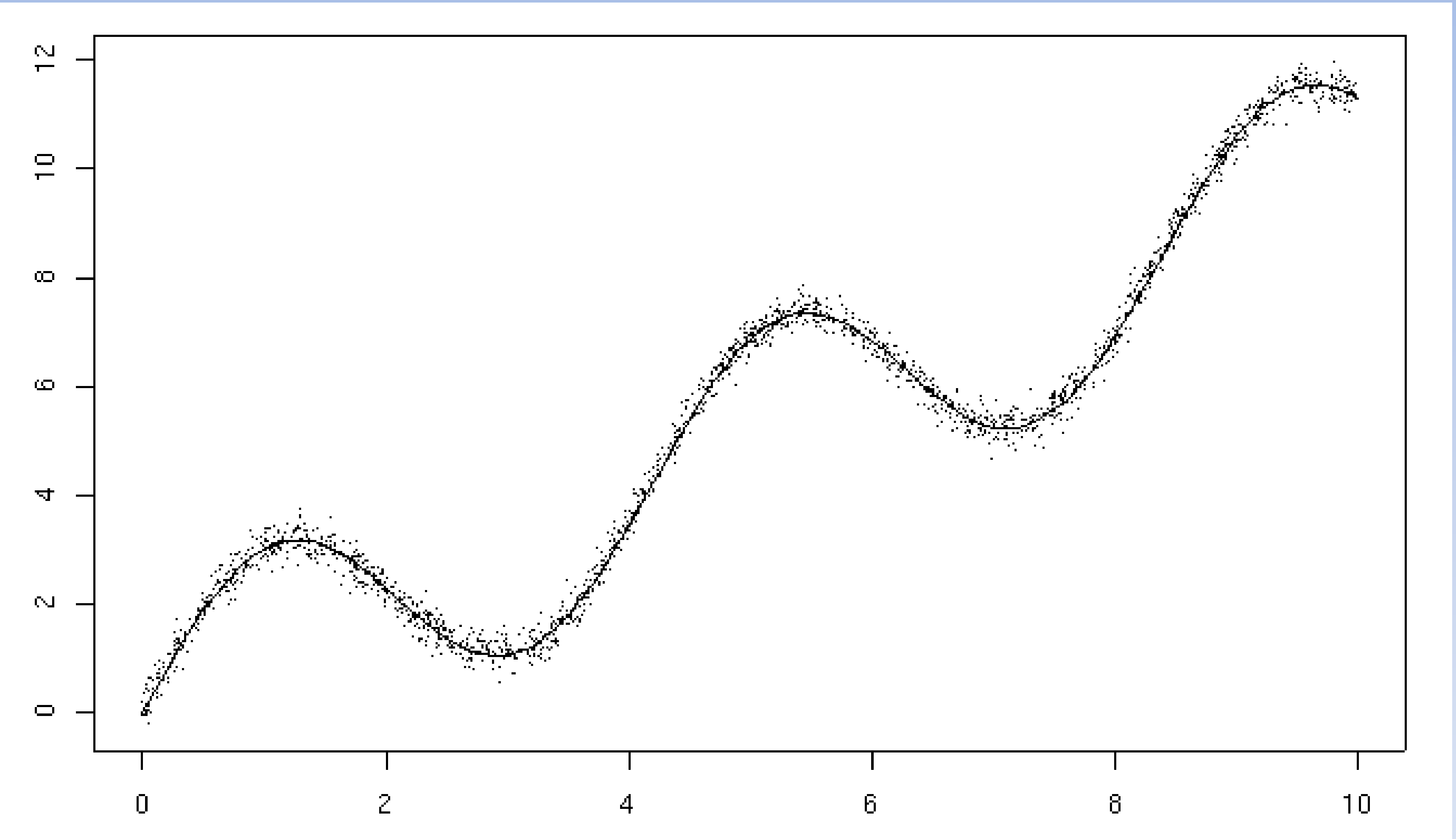

Figure due to Tom Dietterich, Oregon State U.

Distribution of predictions at $x=2.0$

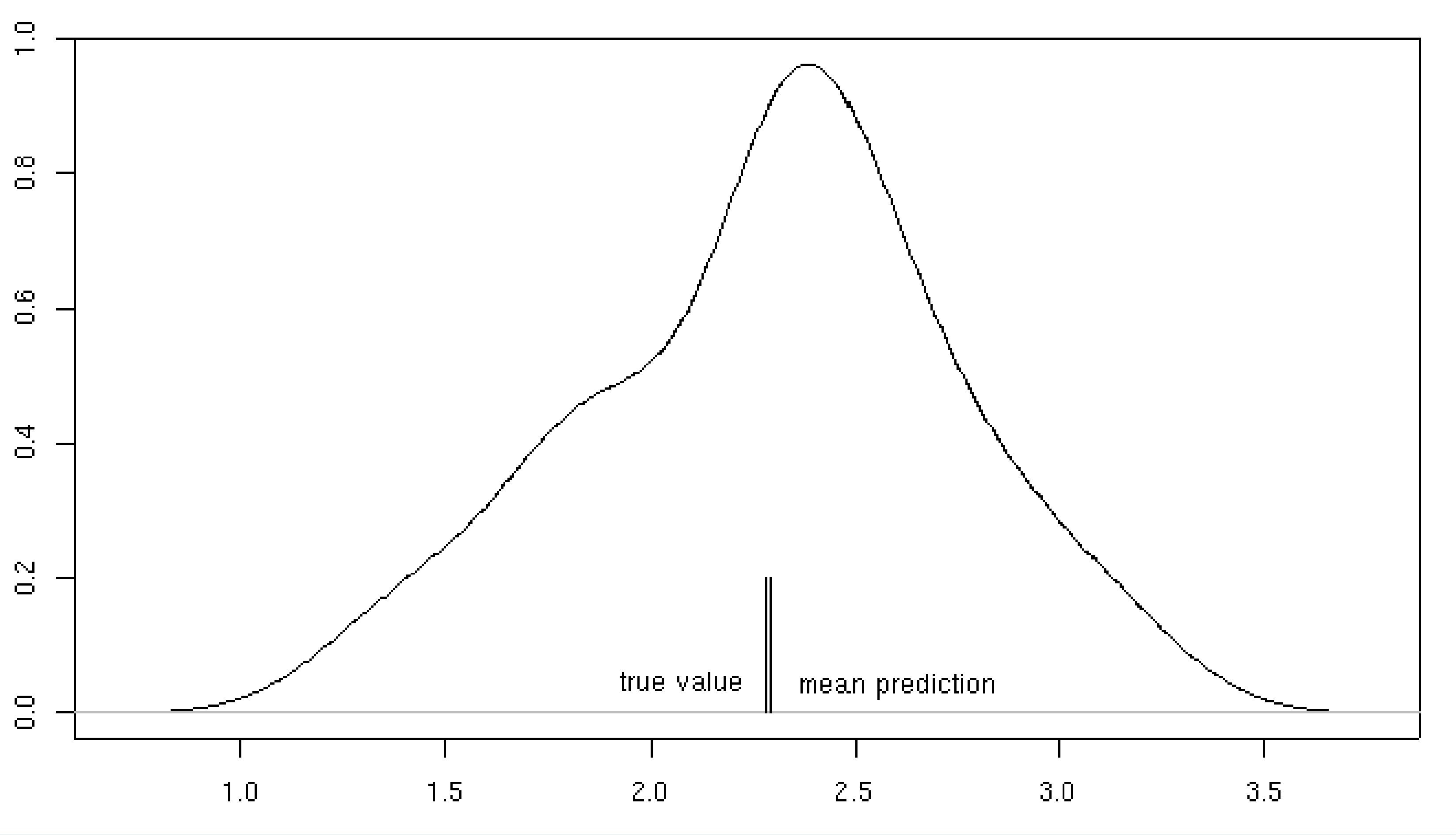

Figure due to Tom Dietterich, Oregon State U.

Distribution of predictions at $x=5.0$

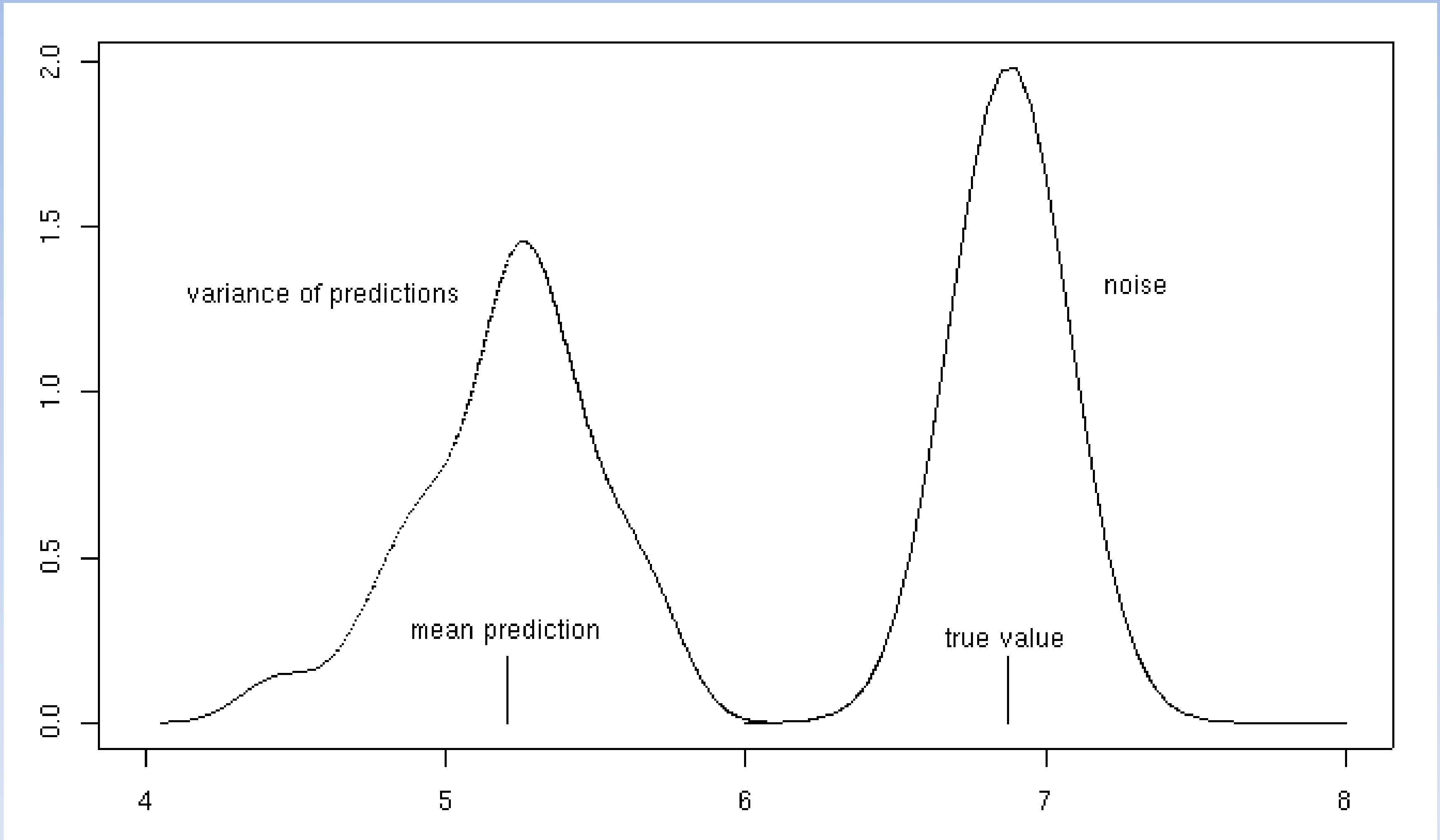

Figure due to Tom Dietterich, Oregon State U.

Decomposition for Classification

- Define the *main prediction*:

$$y_m(\mathbf{x}) = \arg\min_y E_S\left[L(y, h(\mathbf{x}))\right]$$

- For each $\mathbf{x}$, the prediction y that on average across all training sets minimizes the loss

Decomposition for Classification

- For squared loss, the main prediction is

$$y_m(\mathbf{x}) = \overline{h(\mathbf{x})}$$

- For 0/1 loss, the main prediction is the most common output of $h(\mathbf{x})$ (over training sets S)

Bias-Variance Decomposition

- Then for an arbitrary loss function L we define:
  - Bias: $B(\mathbf{x}) = L(y_m, f(\mathbf{x}))$
  - Variance: $V(\mathbf{x}) = E_S\left[L(y_m, h(\mathbf{x}))\right]$
  - Noise: $N(\mathbf{x}) = E_y\left[L(y, f(\mathbf{x}))\right]$

Bias-Variance decomposition

- With these definitions, for many loss functions, (ask for paper)

$$E(L(y,h(\mathbf{x})) = c_1 N(\mathbf{x}) + c_2 V(\mathbf{x}) + B(\mathbf{x})$$

- For squared loss, $c_1 = c_2 = 1$

- For 0/1 loss, $c_1 = 2I(h(\mathbf{x}) = f(\mathbf{x})) - 1$ and $c_2 = 1$ if $y_m = f(\mathbf{x})$ and $c_2 = -1$ otherwise

Summary and Lessons

- Expected prediction errors can be due to choice of concept class (bias) and choice of training sample (variance)
- We must try to balance the two sources of error
- Usually, low bias=richer, more complex concept class=higher variance, so there is a tradeoff
- High variance leads to overfitting. But controlling for overfitting, e.g. using a penalty term, introduces bias
- Even if we have a good idea about the target concept, it may be useful to choose a concept class with high bias if our training sample is small/unrepresentative, to control the variance

Feature Selection and Dimensionality Reduction

Feature Selection

- In propositional supervised learning, examples are represented through feature vectors
- Generally, features are overgenerated
  - Missing information is hard to compensate for
- So not all features might be *relevant* for a classification problem

Relevance

- A feature is *relevant* iff the target concept or best approximation in the hypothesis space uses the feature to make predictions

Feature Selection

- If there are irrelevant features, they can still be used by learning algorithms
  - Lead to overfitting
  - Increase computational complexity
  - Increase sample complexity
  - Classifiers harder to interpret

- **Question: Can we remove these before learning?**

Problem Statement

- Given: A set of examples $(\mathbf{x}_i, y_i)$ over D features
- Do: Find a *subset* S of features ($|S| \leq D$) so that, for any other subset $S' \neq S$, a learned concept that uses S *generalizes better* than a learned concept that uses S'