CSci 5521: Machine Learning Fundamentals

- Supervised Learning

Slides from Prof. Catherine Qi Zhao

Announcements

- HW0 is available on canvas (due 1/27).
  - All HWs will have written and programming parts.
- I won't hold office hours today!
- If you need a permission number to register, fill out this form: https://z.umn.edu/5521_permission_number_request

(limited spots available)

Outline

- Formulating supervised learning
  - Classification
  - Regression
- Understanding features, model parameters, errors
- Model complexity
- Generalization and overfitting
- Evaluating generalizability

Supervised Learning

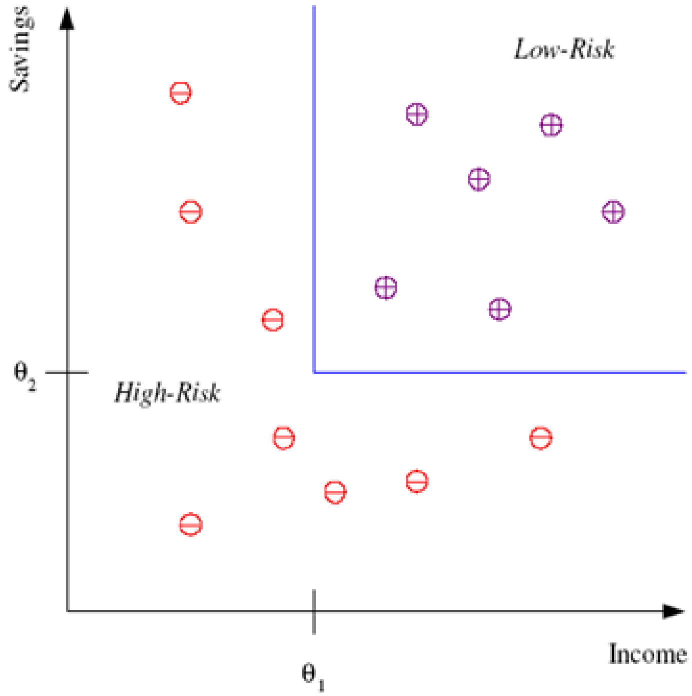
- We have access to:
  - Some input data samples
  - Expert assigned labels/outputs

- Examples
  - Images of animals and animal names
  - House locations and prices

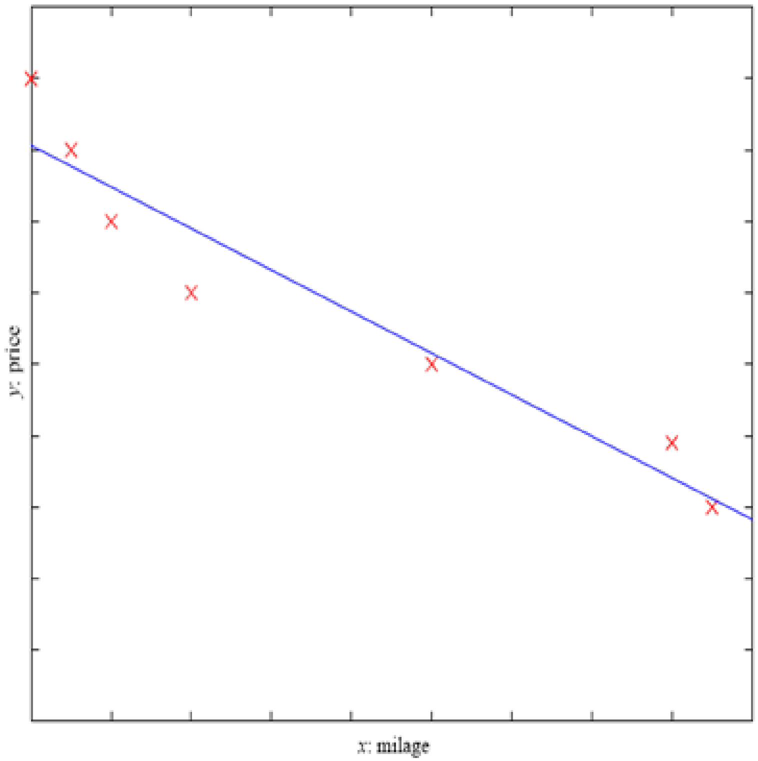
- Goal:
  - Learn a mapping between input data samples and labels/outputs

Supervised Learning

- Classification



- Regression



Output: discrete class label (e.g., {0, 1})

numeric response function $r^t \in \mathbb{R}$

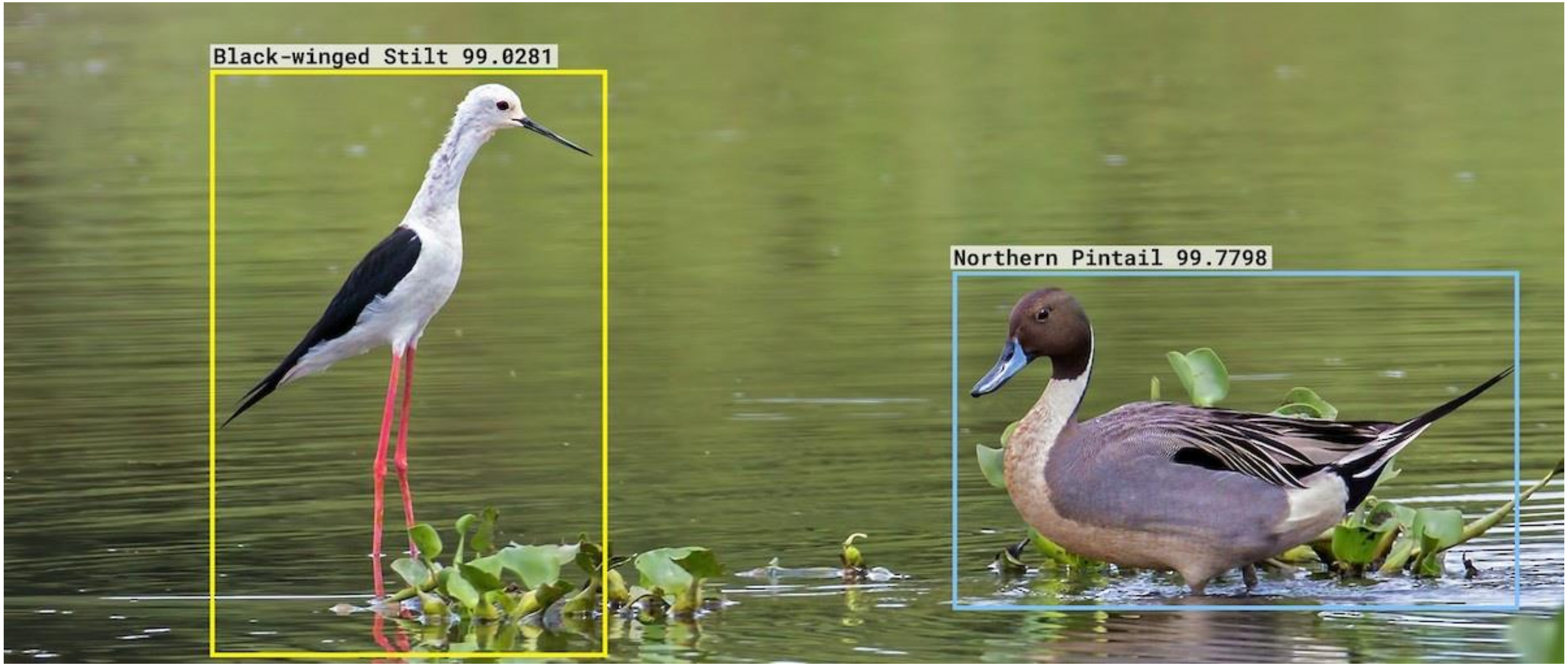
Features and Feature Space

- Feature vector: a n-dimensional vector to represent an object
  - Visualization is easier when n = 1,2,3
  - Dimensionality reduction techniques to reduce the dimension of the feature space
- Feature space: a n-dimensional space where feature vectors live

Feature Extraction

- Feature extraction:
  - Starts from an initial set of measured data and builds derived values (referred to as features, attributes)
  - This process can be automatic or hand-derived

- Desirable properties of features:
  - Informative
  - Non-redundant
  - Human interpretable



Black-winged Stilt 99.0281
Northern Pintail 99.7798

Definitions

- Training set $X = \{(\mathbf{x}^t, r^t)\}_{t=1}^N$
- Hypothesis class $\mathcal{H} = \{h\}$

$$h(\mathbf{x}) = \begin{cases} 1\,, & \text{if } h \text{ says } \mathbf{x} \text{ is positive} \\ -1\,, & \text{if } h \text{ says } \mathbf{x} \text{ is negative} \end{cases}$$

- Classes denoted as $\{0, 1\}, \{-1, +1\}, \{1, 2\}, \{C_1, C_2\}$
  - Equivalent representations
- Empirical error *rate*, on the training set

$$E(h|X) = \frac{1}{N} \sum_{t=1}^{N} \mathbb{1}(h(\mathbf{x}^t) \neq r^t)$$

- Generalization error rate: Performance on 'new' $(\mathbf{x}, r)$

Hypothesis Class

- Choice of hypothesis class $\mathcal{H}$
- Choice between hypothesis classes $\mathcal{H}_1, \ldots, \mathcal{H}_k$
- Realizable learning
  - Target function f belongs to hypothesis class $\mathcal{H}$
  - Can get empirical error $E(h|\mathcal{H}) = 0$ for some $h \in \mathcal{H}$
- Non-realizable learning
  - Target function f is not in $\mathcal{H}$
  - Do the best we can

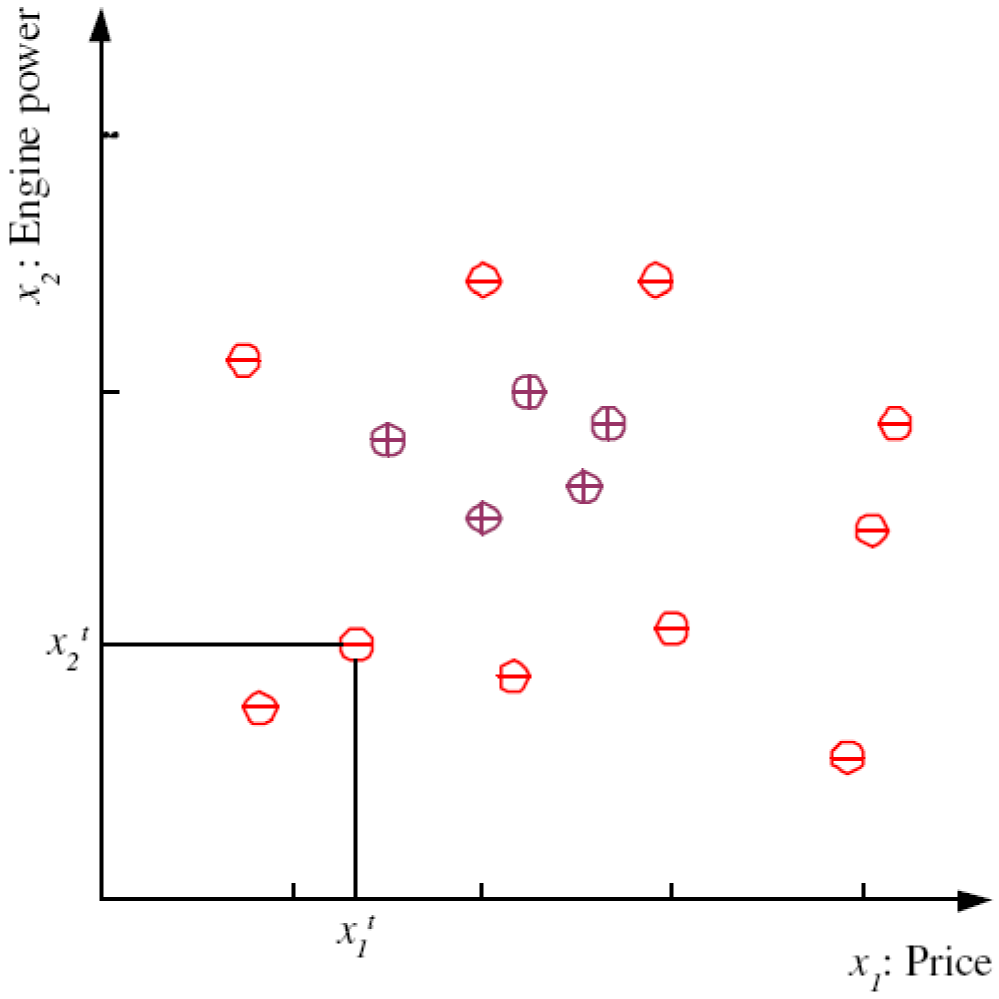
Learning a Class from Examples

- Class C of a “family car”
  - Prediction: Is car x a family car?
  - Knowledge extraction: What do people expect from a family car?
- Output:

  Positive (+) and negative (–) examples
- Input representation:

  x_1: price, x_2 : engine power

Training set X



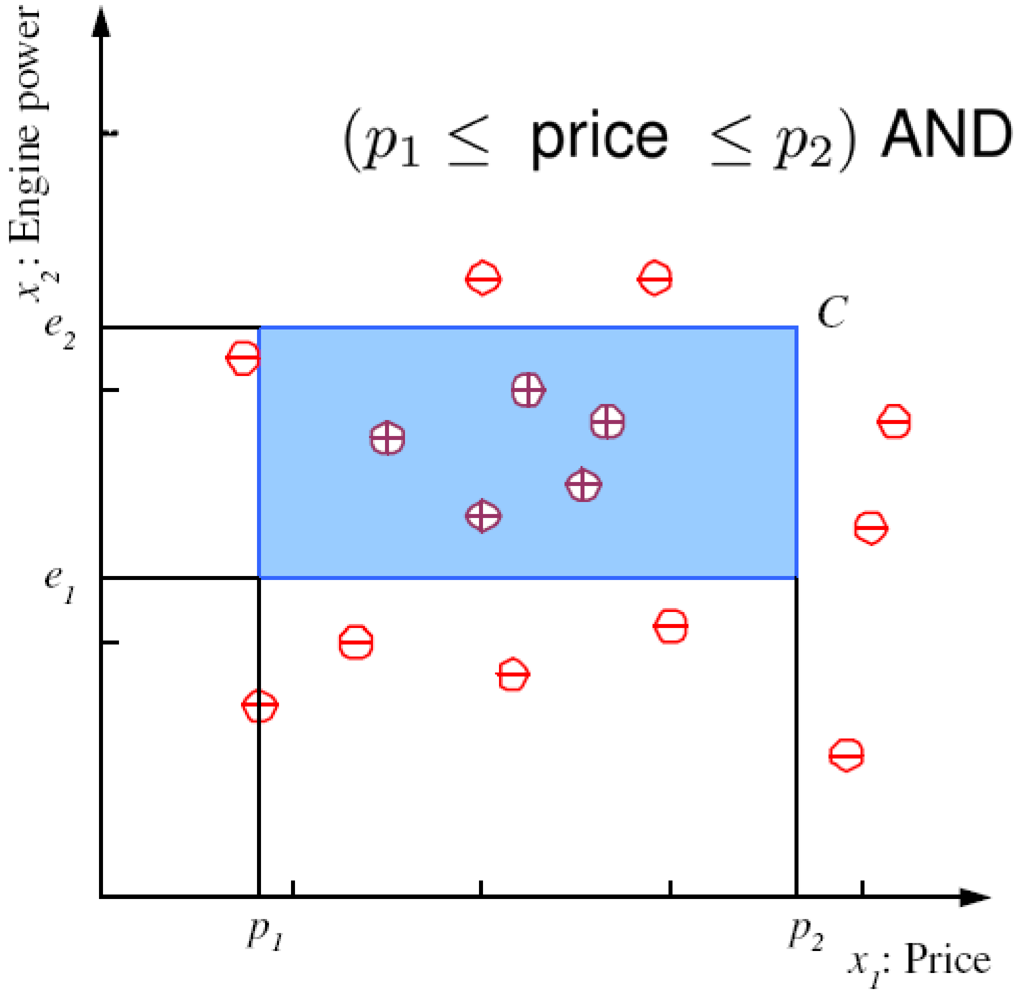
$$X = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

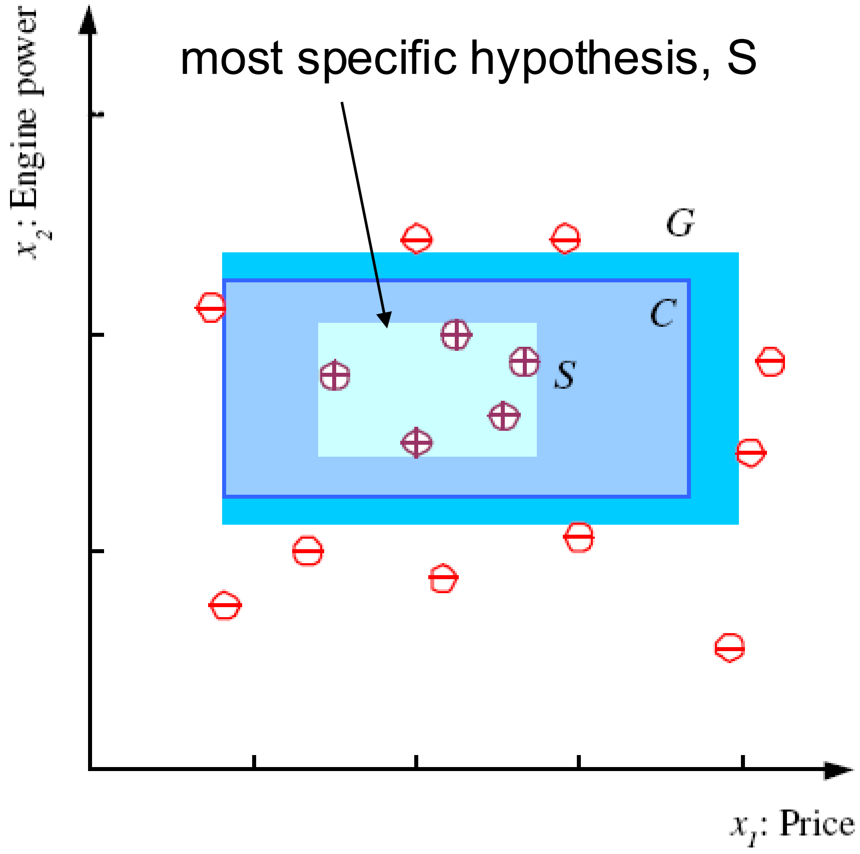
$$r = \begin{cases} 1, & \text{if } \mathbf{x} \text{ is positive} \\ 0, & \text{if } \mathbf{x} \text{ is negative} \end{cases}$$

Class C

$$(p_1 \leq \text{ price } \leq p_2) \text{ AND } (e_1 \leq \text{ engine power } \leq e_2)$$

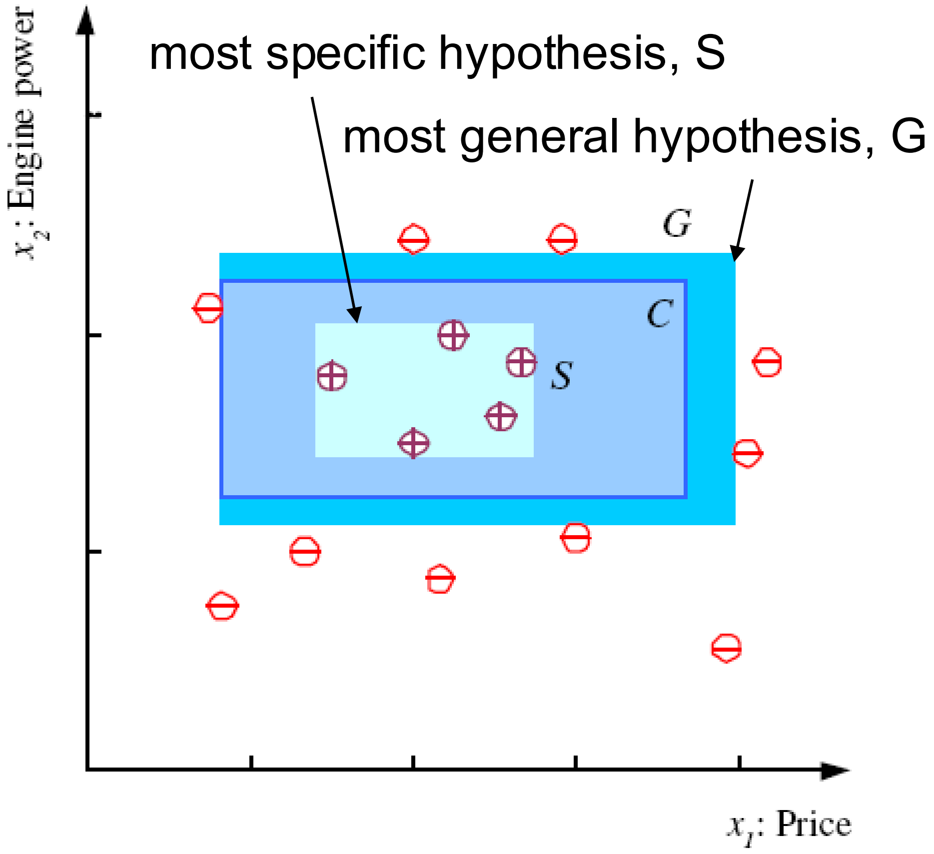


How to find the best h?



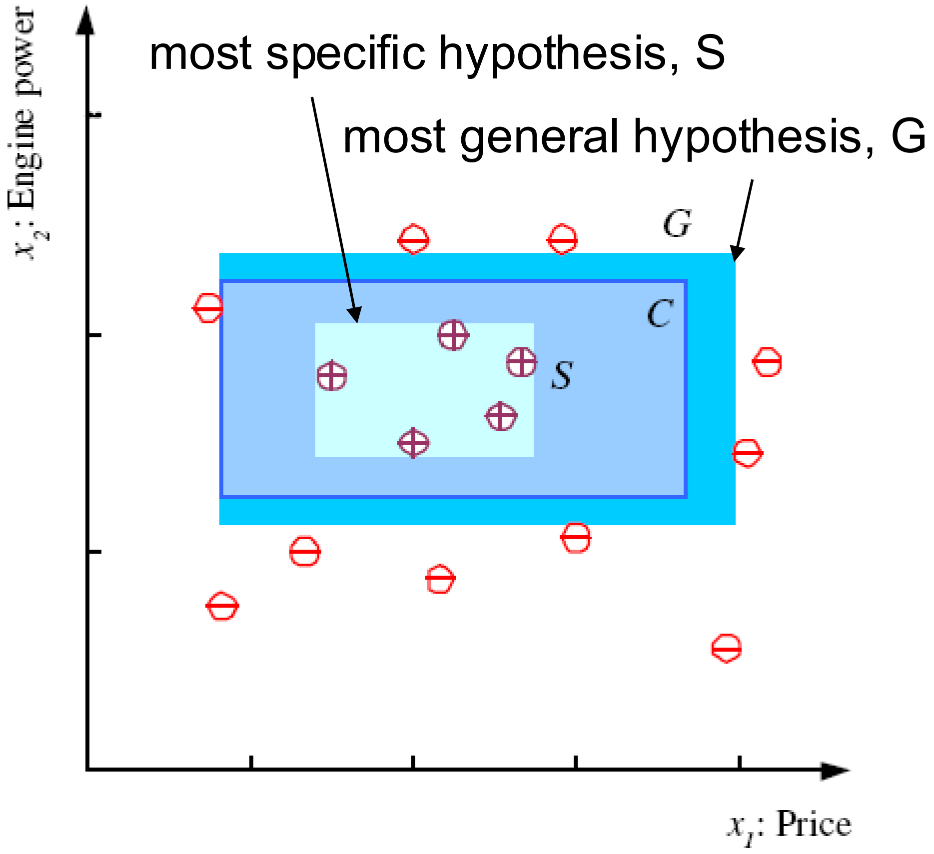
- Option 1: most specific hypothesis (S): the tightest rectangle that includes all the positive examples and none of the negative examples.
- Note that the actual class C may be larger than S but is never smaller.

How to find the best h?



- Option 2: most general hypothesis (G): the largest rectangle we can draw that includes all the positive examples and none of the negative examples

Version Space

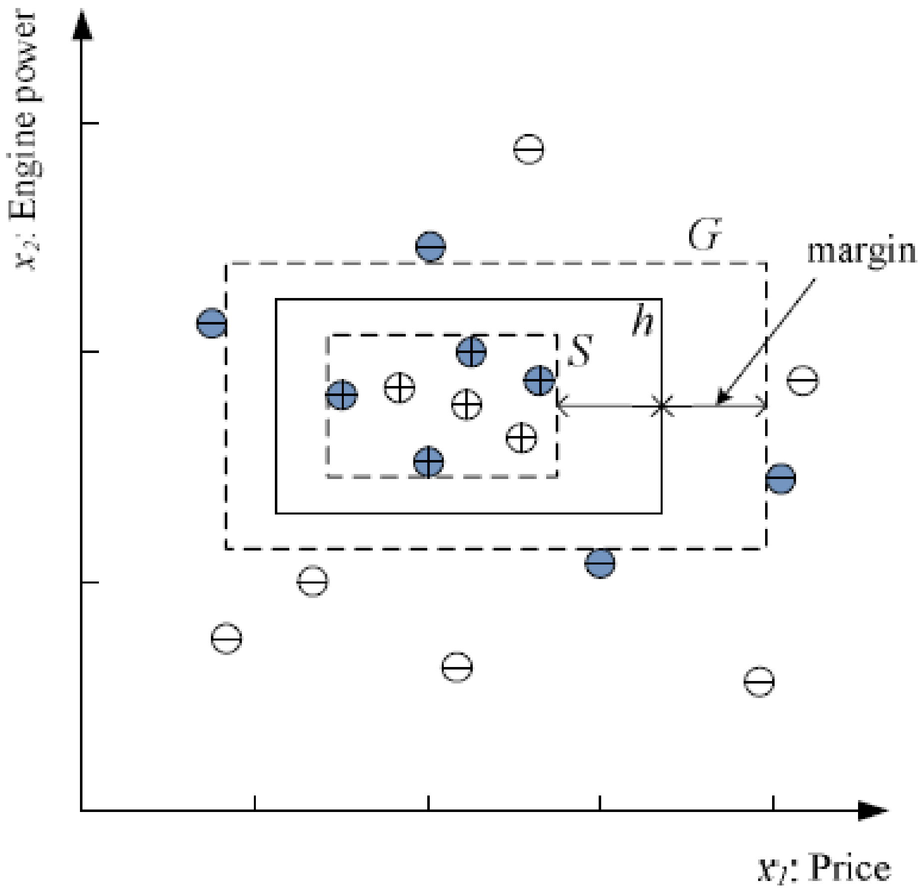


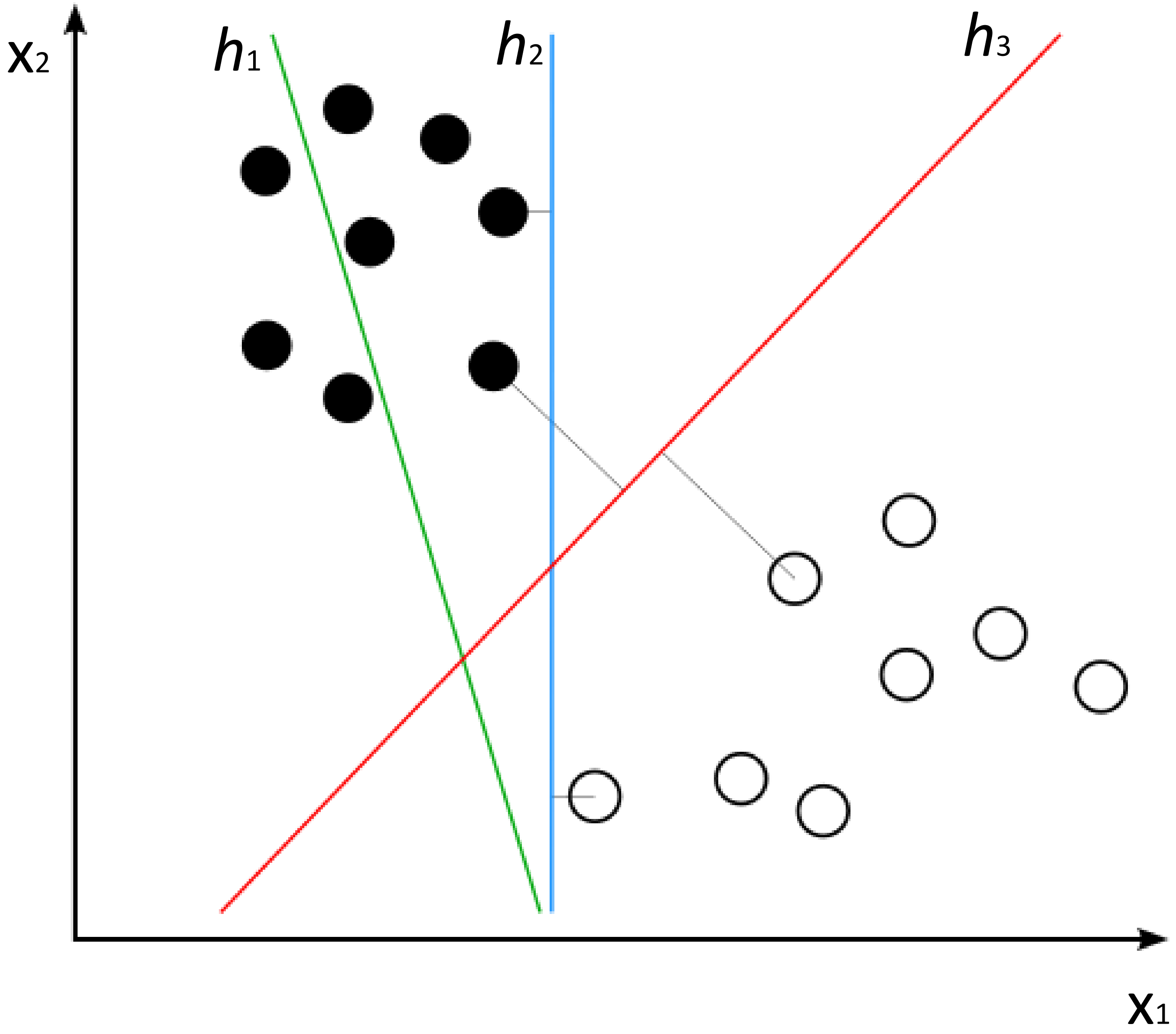
- $h \in H$, between S and G is consistent and make up the version space (Mitchell, 1997)

Margin

- Option 3: choose h with largest margin, that represents the largest separation of the classes

- Margin: the distance between the boundary and the instances closest to it



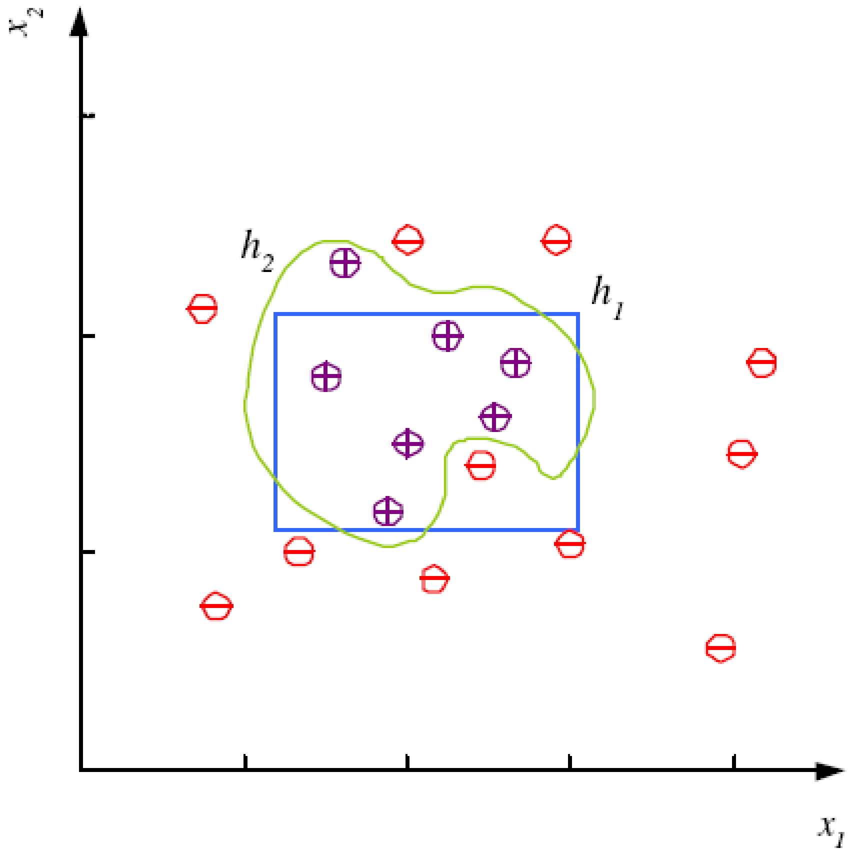


x_2
h_1
h_2
h_3
x_1

Noise and Model Complexity

Use the simpler one because

- Simpler to use (lower computational complexity)
- Easier to train (lower space complexity)
- Easier to explain (more interpretable)
- Generalizes better (lower variance)



Multi-Class Classification

- Training set $\{(\mathbf{x}^t, r^t)\}_{t=1}^N$
- Class label r^t is one of k-classes
- Different ways of representing multi-class classification
  - Direct: $r^t \in \{1, \ldots, k\}$
  - One-vs-rest: Consider C_i vs rest, k 2-class problems

$$r_i^t = \begin{cases} 1\ , & \text{if } \mathbf{x}^t \in C_i \\ 0\ , & \text{if } \mathbf{x}^t \in C_j\ \ , j \neq i \end{cases}$$

  - Pairwise: Consider C_i, C_j, $\binom{k}{2}$ 2-class problems

$$r_{i,j}^t = \begin{cases} 1\ , & \text{if } \mathbf{x}^t \in C_i \\ 0\ , & \text{if } \mathbf{x}^t \in C_j \end{cases}$$

Regression

- Training set $X = \{\mathbf{x}^t, r^t\}$, where $r^t \in \mathbb{R}$
- (One possible) Regression Model

$$r^t = f(\mathbf{x}^t) + \epsilon^t$$

- Noise $\{\epsilon^t\}$ is (typically) assumed to be 'i.i.d.'
  - Independent and identically distributed
- Consider hypothesis class $\mathcal{H} = \{g\}$
- Empirical error on training set

$$E(g|X) = \frac{1}{N}\sum_{t=1}^{N}(r^t - g(\mathbf{x}^t))^2$$

Linear Regression

- Assume $\mathbf{x} \in \mathbb{R}^d$
- Hypothesis class is linear/affine functions in $\mathbb{R}^d$

$$g(\mathbf{x}) = w_1 x_1 + \ldots + w_d x_d + w_0 = \sum_{i=1}^{d} w_j x_j + w_0$$
$$= \mathbf{w}^T \mathbf{x} + w_0 = \langle \mathbf{w}, \mathbf{x} \rangle + w_0$$

- Parameters $\mathbf{w}^T = [w_1 \ \ldots \ w_d]$ and w_0
- If $d = 1$, $g(\mathbf{x}) = w_1 \mathbf{x} + w_0$, regression problem

$$E(w_1, w_0 | X) = \frac{1}{N} \sum_{t=1}^{N} (r^t - (w_1 \mathbf{x}^t + w_0))^2$$

Linear Regression (cont'd)

- If $d = 1$, $g(\mathbf{x}) = w_1\mathbf{x} + w_0$, regression problem

$$E(w_1, w_0|X) = \frac{1}{N}\sum_{t=1}^{N}(r^t - (w_1\mathbf{x}^t + w_0))^2$$

- Optimization problem, set gradient (derivative) to zero, solve

$$w_1 = \frac{\frac{1}{N}\sum_t \mathbf{x}^t r^t - \bar{\mathbf{x}}\,\bar{r}}{\frac{1}{N}\sum_t (\mathbf{x}^t)^2 - \bar{\mathbf{x}}^2}$$

$$w_0 = \bar{r} - w_1\bar{\mathbf{x}}$$

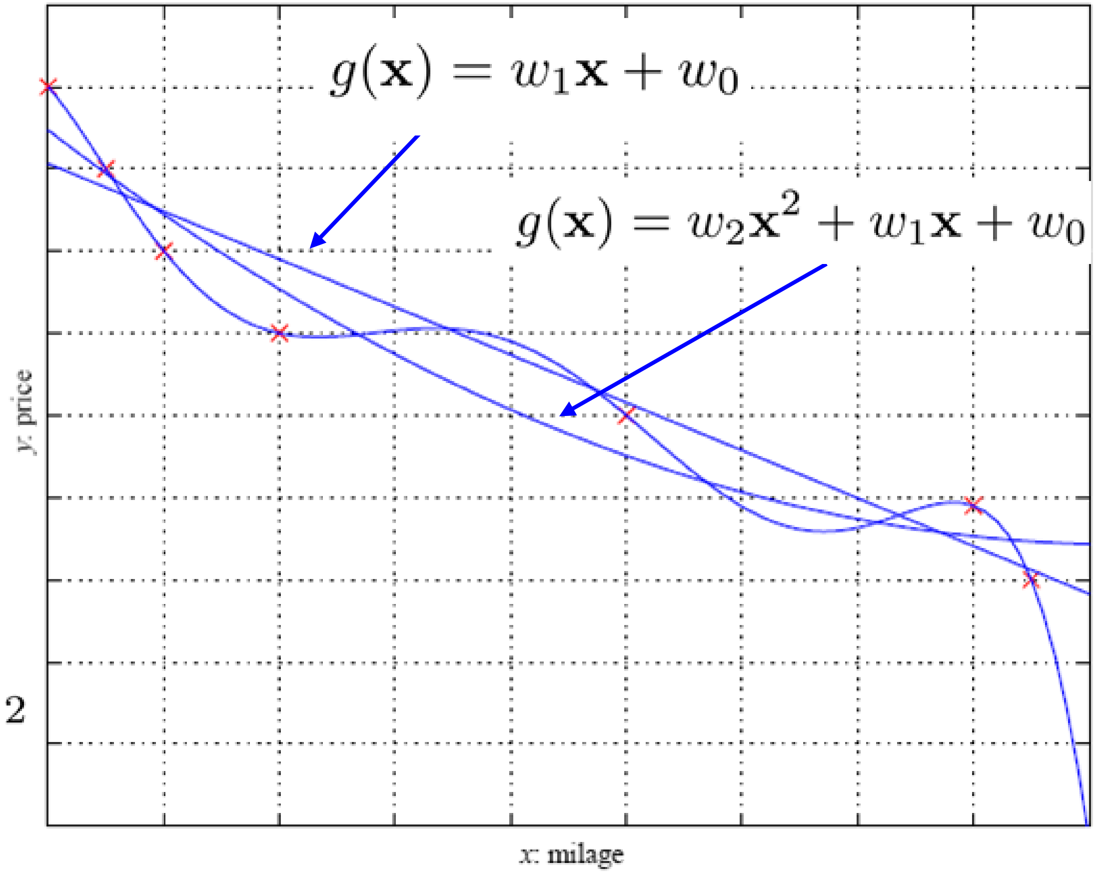
where $\bar{\mathbf{x}} = \frac{1}{N}\sum_t \mathbf{x}^t$ and $\bar{r} = \frac{1}{N}\sum_t r^t$

Regression

$$X = \{\mathbf{x}^t,\ r^t\}_{t=1}^N$$

$$r^t \in \mathbb{R}$$

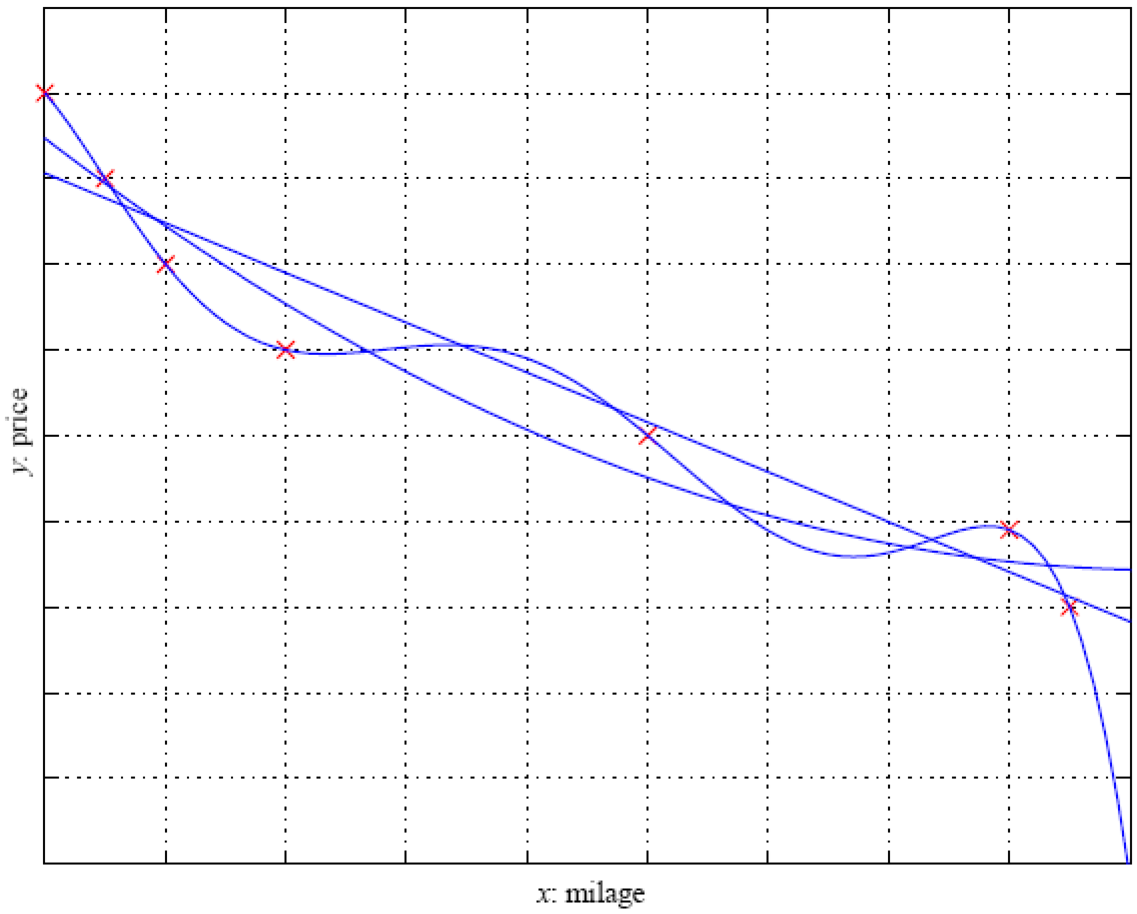
$$r^t = f(\mathbf{x}^t) + \epsilon$$



$$E(g|X) = \frac{1}{N}\sum_{t=1}^{N}[r^t - g(\mathbf{x}^t)]^2$$

$$E(w_1, w_0|X) = \frac{1}{N}\sum_{t=1}^{N}(r^t - (w_1\mathbf{x}^t + w_0))^2$$

Polynomial Regression



The six-order gives a perfect fit, but may be overfitting

Model Complexity

- Given a hypothesis class $\mathcal{H}$, 'best empirical error': $\min_{h \in \mathcal{H}} E(h|X)$
- Tradeoff in choosing $\mathcal{H}$
  - Bigger $\mathcal{H}$ will have lower 'best empirical error'
  - Bigger $\mathcal{H}$ will have higher 'complexity'
- Generalization error: Performance on 'new' $(\mathbf{x}, r)$
- At a high level
  Generalization error = Empirical error + Model Complexity
- Model selection: Goal is to get low generalization error
  - Choose $\mathcal{H}$: tradeoff between empirical error, model complexity

Generalization and Overfitting

- Generalization: how well a model performs on new data
  - If your model generalizes well, then it will perform well on new data that are similar in structure to the training data.
- Overfitting: If your model has very low training error but high generalization error
  - This means that the model has learned to model the noise in the training data, instead of learning the underlying structure of the data.

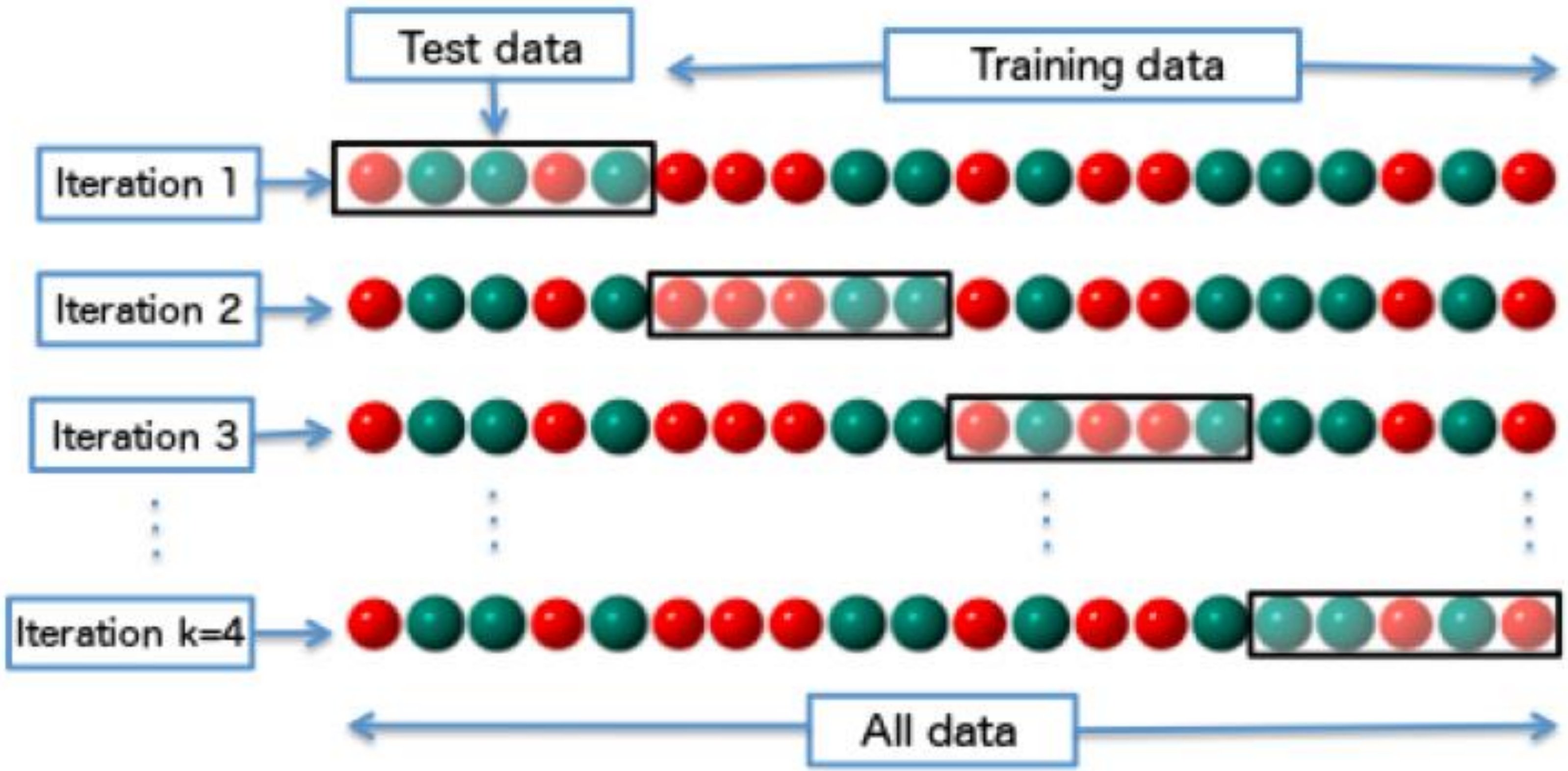
Validation Set

- Split dataset X intro two parts X_T, X_V
- Training set X_T
  - Train different models ($\mathcal{H}$) on X_T
- Validation set X_V
  - Evaluate Train different learned models on X_V
- Pick the best model based on validation set performance

Test Set

- Split dataset into three parts X_T, X_V, X_F
- Train on X_T and validate on X_V as before
- Report results on the fresh set X_F
  - Has never been seen or used in deciding the final model
  - Gives a good sense of generalization error
- Downside: Reduced data for training
- Real life: Adaptive Data Analysis
  - Train, Test, Change, Train, Test, Change, ...
  - Aka "Cheating"
- Machine learning competitions, e.g., Kaggle

K-Fold Cross-Validation



Components of Supervised Learning

- Input: Training set $X = \{(\mathbf{x}^t, r^t)\}_{t=1}^N$ assumed to be 'i.i.d.'
  - Input (representation) $\mathbf{x}^t$, different for different problems
  - Output (representation) r^t, different for different problems
- Model $g(\mathbf{x}|\theta)$, parameters θ, $g(\cdot|\theta) \in \mathcal{H}$
  - Examples: Linear functions, decision trees, neural networks, etc.
- Loss function, for comparing r^t and $g(x^t|\theta)$
  - Computes empirical error

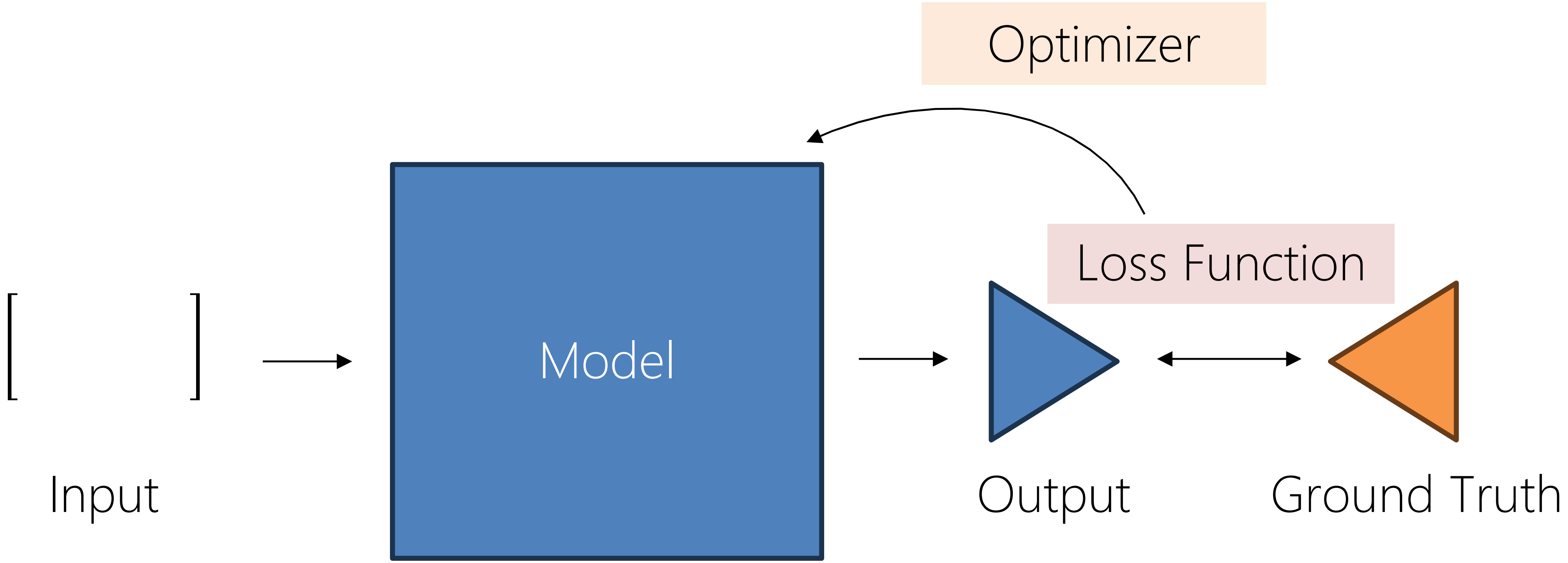
$$E(\theta|X) = \sum_t L(r^t, g(\mathbf{x}^t|\theta))$$

  - Examples: 0-1 loss, squared loss, log loss, etc.
- Learning algorithm
  - Most are based on optimization

$$\theta^* = \underset{\theta}{\operatorname{argmin}} E(\theta|X)$$

  - Alternatives: Bayesian learning, local learning, ensembles, etc.

Components of Supervised Learning



Some materials credit to former 5521, Introduction to Machine Learning, by Ethem Alpaydin, and other online resources