Lecture 5: Inference Problems:

\[ y \rightarrow x \rightarrow \text{predictor} \rightarrow \hat{y} \]

\[
\text{Label} \rightarrow \text{feature/observation}
\]

Example: \( X \): avacado or more precisely, the features of an avacado, e.g., the color of an avacado, \( X \in \mathbb{R} \)

\( y \): goodness of an avacado \( Y \in \mathbb{R} \)

Goal: Find a predictor \( h: X \rightarrow Y \) that can accurately predict \( Y \) from \( X \).

Data-Driven Approach:

**Data:** \( \{(x_i, y_i)\}_{i=1}^{n} \)  
E.g., 1 labeled avocados  
\( n \) labeled images.

**Hypothesis class:**

\( \mathcal{H} \): set of functions we can use to implement the predictor

\[ \mathcal{H} = \{ h_{a}: h_{a}(x) = \theta^{T} x \mid \theta \in \mathbb{R}^{d} \} \]

**Linear Predictors**

\[ h_{a}(x) = \sum_{i=1}^{d} \theta_{i} x_{i} \]
Example: \( \mathcal{H} = \{ h_\theta(x) = \theta_1 x^n + \theta_2 x^{n-1} + \ldots + \theta_n x + \theta_{n+1} \} \)  
set of polynomial predictors of degree \( \leq n \).

Example 2: \( \mathcal{H} = \{ h_\theta \} \): \( h_\theta \) is the set of functions we can implement with a neural network with a given structure and coefficients \( \theta \in \mathbb{R}^d \).

Neural network:

\[ h_1 = \text{ReLU}(a_{11} x_1 + a_{12} x_2 + a_{13} x_3) \]

\[ h_2 = \text{ReLU}(a_{21} x_1 + a_{22} x_2 + a_{23} x_3) \]

\[ h_3 = \text{ReLU}(a_{31} x_1 + a_{32} x_2 + a_{33} x_3) \]

\[ y = b_1 h_1 + b_2 h_2 + b_3 h_3 + b_4 h_4 \]

\( f: \mathbb{R}^3 \rightarrow \mathbb{R} \)

Loss function: \( l: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+ \) non-negative.

\( l(\hat{y}, y) \): takes two labels and quantifies how different the two labels are.

E.g.: \( y \in \mathbb{R} \) regression problems.

\[ l(\hat{y}, y) = (\hat{y} - y)^2 \] squared loss.

E.g.: \( y \in \{1, 2\} \) discrete set classification problems with \( k \) classes.

\[ l(\hat{y}_i, y_i) = \begin{cases} 1 & \text{if } y_i \neq \hat{y}_i \\ 0 & \text{otherwise} \end{cases} \]

0-1 loss.
The loss of a predictor \( h \in \mathcal{H} \) on a given sample \((x, y)\) will be given by:
\[
\ell(h(x), y) = \text{true label of } x \\
\downarrow
\text{the label predicted by } h \text{ for } x
\]

**Empirical Loss of a predictor \( h \):** (the average loss of a predictor \( h \) on our sample set \( \{(x_i, y_i)\}_{i=1}^{n} \))
\[
\mathcal{L}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)
\]

**Empirical Risk Minimization (ERM):**

Find \( h^* \in \mathcal{H} \) that has the smallest \( \mathcal{L}_n(h) \)

\[
\hat{h} = \arg \min_{h \in \mathcal{H}} \mathcal{L}_n(h) = \arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)
\]

**Ex:**

- **goodness**
- \( h(x) = ax + b \)
- \( \mathcal{H} = \{ h(a, b) : h(a, b) = ax + b \} \)

\[
\ell(h(x_i), y_i) = (h(x_i) - y_i)^2 = (ax_i + b - y_i)^2
\]

\[
\min_{h \in \mathcal{H}} \mathcal{L}_n(h) = \min_{a,b} \frac{1}{n} \sum_{i=1}^{n} (ax_i + b - y_i)^2
\]

**Linear Least Squares Regression**
How do we know that the performance of a chosen predictor \( h \) on the data set is representative of its performance in the real world/population?

The data may not be representative

**Memorization**
- Poor generalization
- Overfitting

Data generation model:
\[
\{ (x_i, y_i) \}_{i=1}^{n} \sim P \quad (P \text{ is the true distribution from which nature generates } (x,y))
\]

Population loss of a predictor \( h \):
\[
L(h) = \mathbb{E}[l(h(x), y)] = \int l(h(x), y) f_{x,y}(x,y) dx
\]

How can we ensure that \( L(\hat{h}) \approx L_{\text{true}}(\hat{h}) \):
\[
L(\hat{h}) - L_{\text{true}}(\hat{h}) = \text{generalization error}
\]

Uniform convergence: Show that if we take \( n \) iid samples from \( P \) and evaluate \( L_{n}(h) \) for all \( h \in \mathcal{H} \), then \( L_{n}(h) \approx L(h) \) for all \( h \in \mathcal{H} \) (regardless of \( P \)).

Then no matter what \( \hat{h} \in \mathcal{H} \) is chosen by the ERM algorithm, \( L_{n}(\hat{h}) \approx L(\hat{h}) \), i.e. the ERM solution will generalize well to the real world.