COMPSCI 689
Lecture 2: Optimization-Based Supervised Learning

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Supervised Learning Definitions

Basic Definitions:

In supervised learning, our goal is to predict the output $\mathbf{y}$ that corresponds to an input $\mathbf{x}$ using a prediction function $f(\mathbf{x})$. We assume samples $(\mathbf{x}, \mathbf{y})$ are drawn from $P_*(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$.

- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- True Joint Distribution: $P_*(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}) = P_*(\mathbf{x}, \mathbf{y})$
- Prediction Function: $f : \mathcal{X} \to \mathcal{Y}$

Distributional Supervised Learning Problem

Question

Given the joint distribution $P_*(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$, what is the best choice of prediction function f?

Prediction Loss Functions

Prediction Loss Function: A prediction loss function $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a real-valued function that is bounded below (typically at 0), and that satisfies $L(\mathbf{y}, \mathbf{y}) \leq L(\mathbf{y}, \mathbf{y}')$ for all $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$.

Examples:

- Squared Loss: $L_{sqr}(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|_2^2$
- Absolute Loss: $L_{abs}(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|_1$
- 0/1 Loss: $L_{01}(\mathbf{y}, \mathbf{y}') = [\mathbf{y} \neq \mathbf{y}']$

Given a loss function L, a sample $(\mathbf{x}, \mathbf{y})$, and a prediction function f, we compute the loss of f on $(\mathbf{x}, \mathbf{y})$ as $L(y, f(\mathbf{x}))$.

Given the joint distribution $P_*(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$ and a loss function L, what is the best choice of prediction function f?

Optimization-Based Distributional Supervised Learning

Optimization-Based Distributional Supervised Learning

$$f_* = \arg\min_f \mathbb{E}_{P_*(\mathbf{x},\mathbf{y})} \left[L(\mathbf{y}, f(\mathbf{x})) \right]$$
$$= \arg\min_f \int_{\mathcal{X}} \int_{\mathcal{Y}} L(\mathbf{y}, f(\mathbf{x})) P_*(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

Question: What is the problem with this formulation of the supervised learning problem?

Data Set

- We denote a data set consisting of N samples $(\mathbf{x}_n, \mathbf{y}_n)$ drawn from $P_*(\mathbf{x}, \mathbf{y})$ by: $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n) | (\mathbf{x}_n, \mathbf{y}_n) \sim P_*(\mathbf{x}, \mathbf{y}), 1 \leq n \leq N\}$
- Question: How can we use a data set $\mathcal{D}$ to approximate the solution to the distributional supervised learning problem?
- Two options: Approximate P_* using some model and learn its parameters (density estimation) or introduce a model for f and learn its parameters (function approximation).

Probability Distributions

Defn: A probability density P over a random variable Z with support set $\mathcal{Z}$ is a function that satisfies Kolmogorov's axioms:

- Normalization: $\int_{\mathcal{Z}} P(\mathbf{Z} = \mathbf{z}) d\mathbf{z} = 1$
- Non-Negativity: $\forall \mathbf{z} \in \mathcal{Z}\ P(\mathbf{Z} = \mathbf{z}) \geq 0$

Defn: A parametric probability density P over a support set $\mathcal{Z}$ is a probability density satisfying Kolmogorov's axioms for a specific parameter value θ:

- Normalization: $\int_{\mathcal{Z}} P(\mathbf{Z} = \mathbf{z}|\theta) d\mathbf{z} = 1$
- Non-Negativity: $\forall \mathbf{z} \in \mathcal{Z}\ P(\mathbf{Z} = \mathbf{z}|\theta) \geq 0$

Probability Modeling

- Defn: A continuous probability model $\mathbb{P}$ over a support set $\mathcal{Z}$ is a **set of valid probability densities**.
- Defn: A continuous parametric probability model $\mathbb{P}(\Theta)$ with parameter space Θ is generated by a parametric probability density function P, which must yield a valid density for all $\theta \in \Theta$:

$$\mathbb{P}(\Theta) = \{P(\mathbf{Z}|\theta)|\theta \in \Theta\}$$

Example: Probability Models

Consider a biased coin. Let θ be the probability that the coin comes up heads. Let $Z \in \{0, 1\}$ be the value of the coin flip (1 for heads, 0 for tails). We thus have:

- Support set: $\mathcal{Z} = \{0, 1\}$
- Parameter Space: $\Theta = [0, 1]$
- Parametric Distribution Function: $P(Z = z|\theta) = \theta^z(1 - \theta)^{(1-z)}$
- Parametric Model: $\mathbb{P}(\Theta) =$

$$\left\{\theta^z(1 - \theta)^{(1-z)} \mid \theta \in \Theta\right\}$$

Question: Given a data set $\mathcal{D}$, how do we choose the element of $\mathbb{P}(\Theta)$ that best represents $\mathcal{D}$?

Density Estimation

Let $\mathcal{L}(\mathcal{D}, P)$ be a loss function that measures how “close" a data set D is to a distribution P.

Parametric Density Estimation

$$P = \arg\min_{P' \in \mathbb{P}} \mathcal{L}(\mathcal{D}, P')$$

$$\theta_* = \arg\min_{\theta \in \Theta} \mathcal{L}(\mathcal{D}, P(\theta))$$

Density Estimation

- If $\mathbb{P} = \mathbb{P}(\Theta)$ is a parametric probability model and

$$\mathcal{L}(\mathcal{D}, \theta) = -\sum_{n=1}^{N} \log P(\mathbf{Z} = \mathbf{z_n}|\theta)$$

  then then θ_* is called the the *maximum likelihood* estimate of the unknown parameter θ.
- Maximum likelihood is *the* method used for density estimation in statistics.
- We will investigate maximum likelihood estimation in detail next class.

Supervised Learning By Density Estimation

Given a data set $\mathcal{D}$ and a probability model $\mathbb{P}(\Theta)$, we use density estimation to approximate $P_*(\mathbf{x}, \mathbf{y})$ by $P(\mathbf{x}, \mathbf{y}|\theta_*)$.
We then use $P(\mathbf{x}, \mathbf{y}|\theta_*)$ in place of $P_*(\mathbf{x}, \mathbf{y})$ in the distributional supervised learning problem:

$$\theta_* = \arg\min_{\theta \in \Theta} \mathcal{L}(\mathcal{D}, P(\theta))$$

$$\hat{f}_* = \arg\min_f \int_{\mathcal{X}} \int_{\mathcal{Y}} L(\mathbf{y}, f(\mathbf{x})) P(\mathbf{x}, \mathbf{y}|\theta_*) d\mathbf{x} d\mathbf{y}$$

$$\hat{f}_*(\mathbf{x}) = \arg\min_{\mathbf{y}'} \int_{\mathcal{Y}} L(\mathbf{y}, \mathbf{y}') P(\mathbf{y}|\mathbf{x}, \theta_*) d\mathbf{y}$$

For example, if $L(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|_2^2$, then $\hat{f}_*(\mathbf{x}) = \mathbb{E}_{P(\mathbf{y}|\mathbf{x}, \theta_*)}[\mathbf{y}]$

Empirical Probability Distribution

- We can instead explicitly model $f(x)$, but this requires making a default choice for approximating P_*.
- The standard choice is the *empirical distribution* $P_{\mathcal{D}}$:

$$P_{\mathcal{D}}(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\|\mathbf{x} - \mathbf{x}_n\|)\delta(\|\mathbf{y} - \mathbf{y}_n\|)$$

where $\delta(z)$ is the Dirac delta function $\delta(z) = \begin{cases} \infty & z = 0 \\ 0 & z \neq 0 \end{cases}$

- Expectations under $P_{\mathcal{D}}$ have the following important property:

$$\mathbb{E}_{P_{\mathcal{D}}(\mathbf{x},\mathbf{y})}\left[g(\mathbf{y}, \mathbf{x})\right] = \frac{1}{N} \sum_{n=1}^{N} g(\mathbf{y}_n, \mathbf{x}_n)$$

Prediction Models

- A prediction model F is a set of functions $F = \{f \mid f\colon \mathcal{X} \to \mathcal{Y}\}$.
- A parametric prediction model $F(\Theta)$ with parameter vector $\theta \in \Theta$ is generated by a fixed function g satisfying $g\colon \mathcal{X} \times \Theta \to \mathcal{Y}$. We write $F(\Theta) = \{f \mid f(\mathbf{x}) = g(\mathbf{x}, \theta), \theta \in \Theta\}$.
- **Example**: Let $F(\mathbb{R}^D) = \{f \mid f(\mathbf{x}) = \theta^T \mathbf{x}, \theta \in \mathbb{R}^D\}$ for $\mathbf{x} \in \mathbb{R}^D$.

Supervised Learning By Function Optimization

Given a data set $\mathcal{D}$ and a prediction model $F(\Theta)$, we first approximate P_* by $P_{\mathcal{D}}$, and then minimize L with respect to θ.

$$\hat{f}_* = \arg\min_{f \in F(\Theta)} \mathbb{E}_{P_{\mathcal{D}}(\mathbf{x},\mathbf{y})} \left[L(\mathbf{y}, f(\mathbf{x})) \right]$$

$$\hat{\theta}_* = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} L(\mathbf{y}_n, f(\mathbf{x}_n, \theta))$$

Optimization and Learning

In either the probability modeling or prediction function modeling frameworks, we identify the optimal model parameters via the solution to an optimization problem.

Next class, we will turn to the study of optimization itself.