COMPSCI 689
Lecture 8: Expected Risk Minimization and Regression

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

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

Distributional Supervised Learning Problem

Question

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

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: $L_{01}(\mathbf{y}, \mathbf{y}') = [\mathbf{y} \neq \mathbf{y}']$

Optimization-Based Distributional Supervised Learning

Given P_* and L, we can define the optimal f to be the function that minimizes the expected loss:

Optimization-Based Distributional Supervised Learning

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

Of course, we never actually get to see P_*. We only see samples from it.

Data Sets and the Empirical Distribution

- We denote a data set consisting of N samples $(\mathbf{x}_n, \mathbf{y}_n)$ drawn from $P_*(\mathbf{x}, \mathbf{y})$ by: $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n) | (\mathbf{x}_n, \mathbf{y}_n) \sim P_*(\mathbf{x}, \mathbf{y}), 1 \leq n \leq N\}$
- We call $P_\mathcal{D}$ the empirical distribution induced by the data set $\mathcal{D}$:

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

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

- Given a data set we have two options: Solve for the theoretically optimal f_* in terms of P_* and then approximate P_* using a model, or approximate P_* with $P_\mathcal{D}$ and then optimize f_*.

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 f satisfying $f\colon \mathcal{X} \times \Theta \to \mathcal{Y}$.
- For simplicity, we will use the notation $f(\mathbf{x}, \theta)$ to refer to the model $F(\Theta)$ generated by f and Θ.

Expected Risk Minimization

The ERM framework for parametric prediction models corresponds to finding the optimal prediction function within the model by minimizing it's expected loss under the empirical distribution. The expected loss of the function is called it's risk.

Expected Risk Minimization

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

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

ERM For GLMs

- So far, we have focused on approximating the distributional supervised learning problem by using a (generalized) linear probability model to approximate P_*.
- We can also solve this problem within the empirical risk minimization framework by directly approximating f_* with a transformed linear model: $f(\mathbf{x}, \mathbf{w}) = g^{-1}(\mathbf{w}\mathbf{x}^T)$ (assuming the bias has already been absorbed).
- For regular linear regression problems, it is common to apply the squared loss, resulting in the following optimization problem:

$$\mathbf{w}_* = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} (y_n - \mathbf{w}\mathbf{x}_n^T)^2$$

ERM For Linear Regression

- It's easy to see that this optimization problem has exactly the same solution as we found for the linear Gaussian model!:

$$\mathbf{w}_*^T = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- We've thus shown two different ways to obtain the same algorithm for learning linear regression parameters that give exactly the same solution: one based on approximating P_* and one based on approximating f_*.

- Importantly, the solution obtained via ERM has the same disadvantages as the MLE. In particular, it may not give robust estimates when N is small.

Regularization

- Within the ERM framework, the way to deal with small N is to apply an idea called regularization.
- The idea is to penalize the complexity of the function that is used to approximate f_*. In general, the more data we have, the more complex a function we will allow.
- In the case of linear regression, the usual notion of complexity is the squared norm of weights $\mathbf{w}$. Penalizing $\|\mathbf{w}\|_2^2$ leads to the following optimization problem:

$$\mathbf{w}_*^T = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} (y_n - \mathbf{w}\mathbf{x}_n^T)^2 + \lambda \|\mathbf{w}\|_2^2$$

- Here, λ is a parameter that controls how much penalization is applied.

Regularization

- It's not hard to see that this problem is equivalent to penalized conditional maximum likelihood estimation.
- The optimal regularized weights are thus given by:

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

- In general, the addition of regularization to the ERM framework is referred to as *regularized risk minimization* (RRM).
- If we have a pair of models where MLE and ERM give equivalent solutions for approximating f_*, RRM and penalized maximum likelihood will give equivalent solutions if they use the same regularization/penalty terms.

Regression with Other Losses

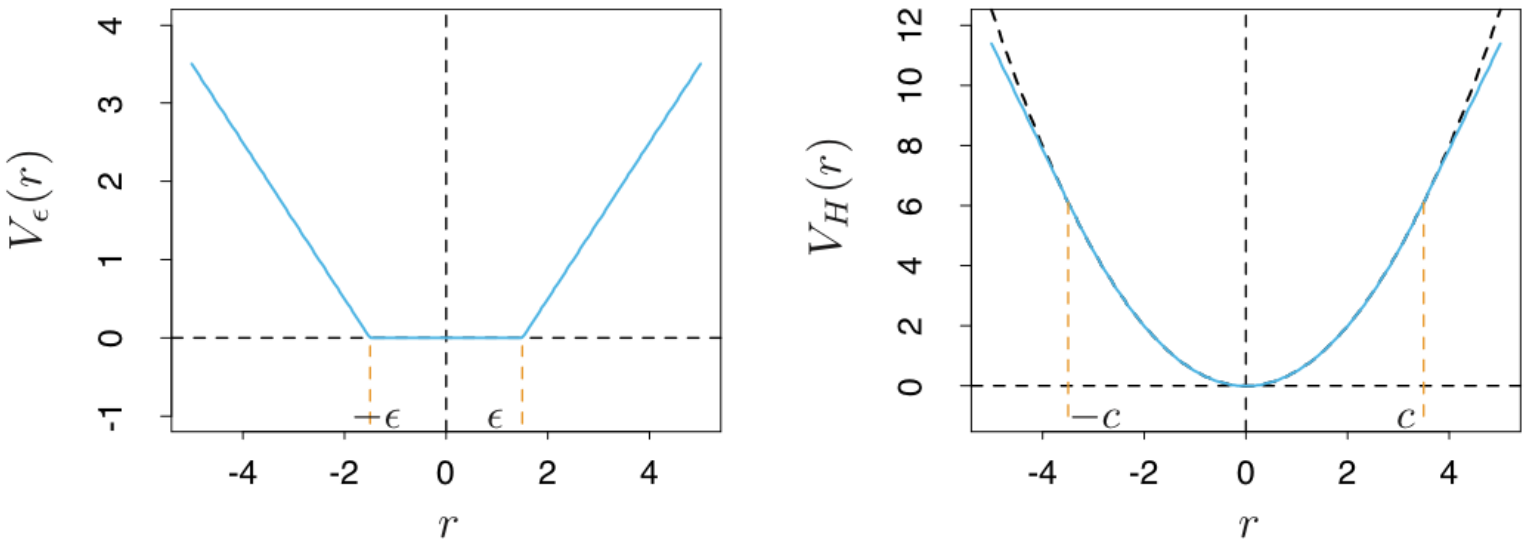
- The strength of the ERM framework is that we are free to choose any loss L we like. For linear regression, this leads to the problem:

$$\mathbf{w}_* = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} L(y_n, \mathbf{w}\mathbf{x}_n^T)$$

- In order for the ERM problem to be solvable efficiently, it's typical to select a convex loss.

Regression with Other Losses

Common alternative losses for regression include the epsilon insensitive loss and the Huber loss.



Both of these losses are specifically designed to limit the influence of outliers on the model fit.

The specific combination of the epsilon insensitive loss with a squared ℓ_2 norm regularizer is referred to as *support vector regression*.

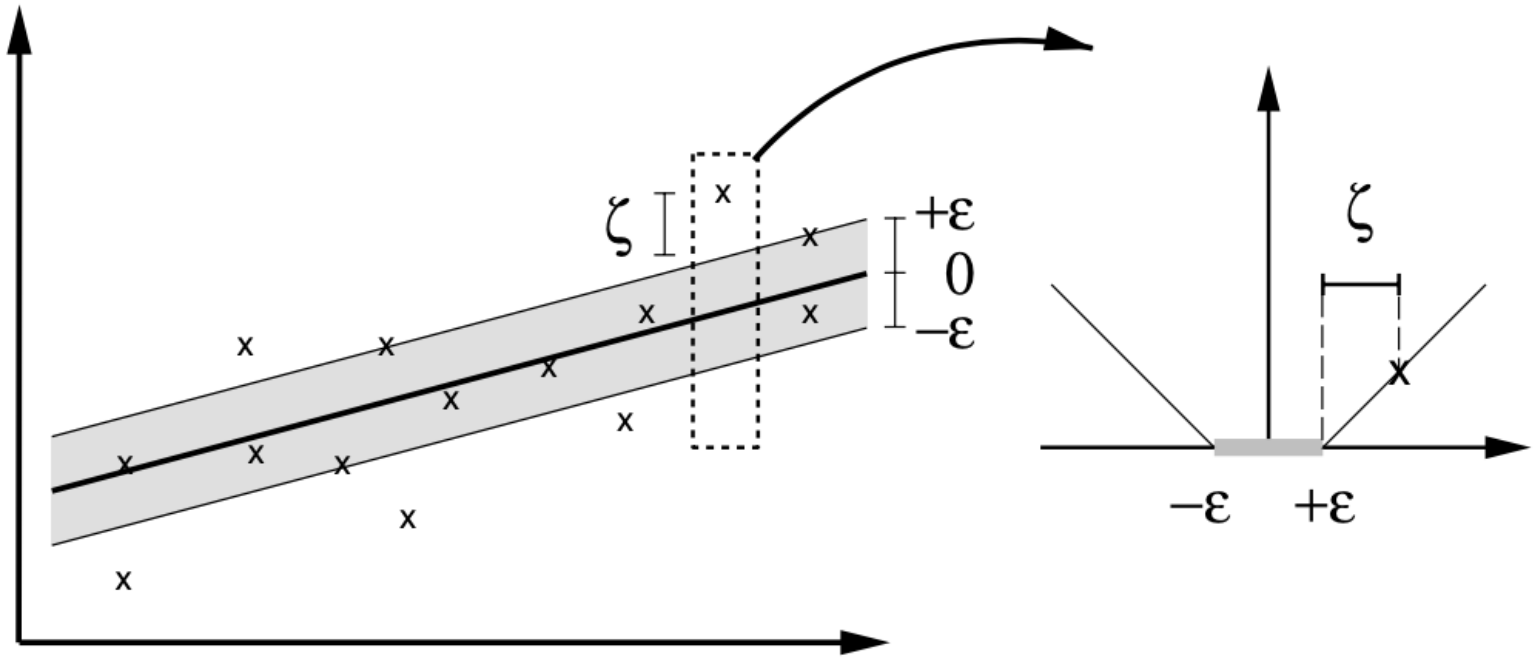
Support Vector Regression

$$f_{SVR}(\mathbf{x}, \mathbf{w}) = \left(\sum_{d=1}^{D} w_d x_d \right) = \mathbf{w}\mathbf{x}^T$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} C \sum_{n=1}^{N} L_\epsilon(y_n, \mathbf{w}\mathbf{x}_n^T) + ||\mathbf{w}||_2^2$$

$$L_\epsilon(y, y') = \begin{cases} 0 & \dots \text{ if } |y - y'| < \epsilon \\ |y - y'| - \epsilon & \dots \text{ otherwise} \end{cases}$$

Support Vector Regression



This model is called *support* vector regression because only the data cases that fall outside of the epsilon insensitive region determine the model parameters. The optimal parameters are thus *supported* by a subset of all data instances.