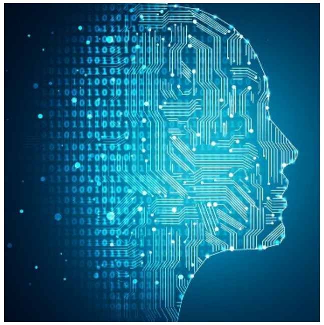
Machine Learning

Empirical Risk Minimization

Kevin Moon (kevin.moon@usu.edu)

STAT/CS 5810/6655



Big Picture

- Many machine learning methods can be cast in a common framework
- This general framework makes it possible to understand several different methods at once

Outline

1. Loss and Risk
2. Empirical Risk Minimization
3. Surrogate Losses

Loss and Risk

- Consider a supervised learning problem with jointly distributed $(\boldsymbol{X}, Y)$
- Let $\mathcal{Y}$ denote the output space
  - Regression: $\mathcal{Y} = \mathbb{R}$
  - Binary classification: $\mathcal{Y} = \{-1,1\}$
- A *loss* is a function $L: \mathcal{Y} \times \mathbb{R} \to [0, \infty)$
  - $L(y, t) =$ cost of predicting t when y is the true output
- Let $f: \mathbb{R}^d \to \mathbb{R}$ be a prediction function. The *risk* of f is:

$$R_L(f) \coloneqq \mathbb{E}_{\boldsymbol{X}Y}\left[L\left(Y, f(\boldsymbol{X})\right)\right]$$

  - I.e., the expected loss of f

Loss and Risk: Regression

- For regression problems, f is a regression function
- **Example**: L is the *squared error* loss

$$L(y,t) = (y-t)^2,$$

$$R_L(f) = \mathbb{E}_{\boldsymbol{X}Y}\left[\left(Y - f(\boldsymbol{X})\right)^2\right]$$

  - $R_L(f)$ is the *mean squared error*
- **Example**: L is the *absolute deviation* loss

$$L(y,t) = |y-t|$$

$$R_L(f) = \mathbb{E}_{\boldsymbol{X}Y}[|Y - f(\boldsymbol{X})|]$$

  - $R_L(f)$ is the *mean absolute error*

Loss and Risk: Binary Classification

- For binary classification problems, f is called a *decision function*. The predicted label is usually

$$\hat{y} = \text{sign}\{f(\boldsymbol{x})\}$$

- Linear classifier example: $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$
- **Example**: L is the *0-1* loss

$$L(y,t) = \begin{cases} 1, & \text{if } y \neq \text{sign}(t) \\ 0, & \text{otherwise} \end{cases} = \mathbf{1}_{\{y \neq \text{sign}(t)\}}$$

$$R_L(f) = \mathbb{E}_{XY}\left[\mathbf{1}_{\{Y \neq \text{sign}(f(\boldsymbol{X}))\}}\right] = \Pr\big(Y \neq f(\boldsymbol{X})\big)$$

  - $R_L(f)$ is the *probability of error*
- What is another interesting loss besides 0-1?

$$0 < \alpha < 1, \qquad L(y,t) = \begin{cases} \alpha & y = -1, t \geq 0 \\ 1-\alpha & y = 1, t < 0 \\ 0 & \text{otherwise} \end{cases}$$

Loss and Risk: Binary Classification

- What about the following:

$$L_{-1,1}(y,t) := \begin{cases} 1 & y \neq \text{sign}(t) \\ -1 & y = \text{sign}(t) \end{cases}$$

  - How will the minimizer be different from that of the 0-1 loss?

- It won't be:

$$L_{-1,1}(y,t) = 2L_{0,1}(y,t) - 1$$

$$R_{L_{-1,1}}(f) = 2R_{L_{0,1}}(f) - 1$$

  - The minimizer will be the same

Empirical Risk Minimization

- Given: training data $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)$ for regression or binary classification
- The *empirical risk* of f:
$$\hat{R}(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n} L\big(y_i, f(\boldsymbol{x}_i)\big)$$
- *Regularized empirical risk minimization* learns f by solving
$$\min_{f\in\mathcal{F}} \hat{R}(f) + \lambda\Omega(f)$$
  - $\mathcal{F}$ is the set of candidate functions
    - **Example**: linear function $\boldsymbol{w}^T\boldsymbol{x} + b$
  - $\Omega(f)$ is the regularizer that measures the complexity of f
    - **Examples**: $||\boldsymbol{w}||_2^2$ or $||\boldsymbol{w}||_1$
  - $\lambda \geq 0$ is user-specified (a tuning parameter)

ERM vs. Risk minimization

- Does minimizing $\hat{R}(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n} L\big(y_i, f(x_i)\big)$ also minimize $R_L(f) \coloneqq \mathbb{E}_{\boldsymbol{X}Y}\big[L\big(Y, f(\boldsymbol{X})\big)\big]$?
- Under certain assumptions on $P_{\boldsymbol{X}Y}$ and $\mathcal{F}$ (the set of candidate functions), it can be shown that minimizing $\hat{R}$ converges to minimizing R as $n \to \infty$
- The field of Statistical Learning Theory studies this problem
  - I.e., what assumptions guarantee convergence and at what rate?
  - Could teach a whole class on this

ERM Examples: Regression

- Least squares loss: $L(y, t) = (y - t)^2$
  - $\mathcal{F}$ = linear regression functions

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} (y_i - \boldsymbol{w}^T \boldsymbol{x}_i - b)^2 + \lambda \|\boldsymbol{w}\|^2$$

  - $\lambda > 0 \Rightarrow$ ridge regression
- Robust loss, $\lambda = 0$

$$L(y, t) = \rho(y - t) = \sqrt{1 + (y - t)^2} - 1$$

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} \rho(y_i - \boldsymbol{w}^T \boldsymbol{x}_i - b)$$

  - A smooth version of absolute value

ERM Examples: Binary Classification

- 0-1 loss

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{y_i \neq \mathrm{sign}(\boldsymbol{w}^T \boldsymbol{x}_i + b)\}}$$

- Unfortunately, this is *intractable* even for linear classifiers
  - Impossible to solve a reasonable sized problem in a reasonable amount of time
- Motivates the use of *surrogate losses*

Surrogate losses

- A surrogate loss takes the place of another loss, usually because of nicer computational properties (convexity, differentiability, etc.)
- Some common surrogate losses for binary classification:
  - Logistic loss

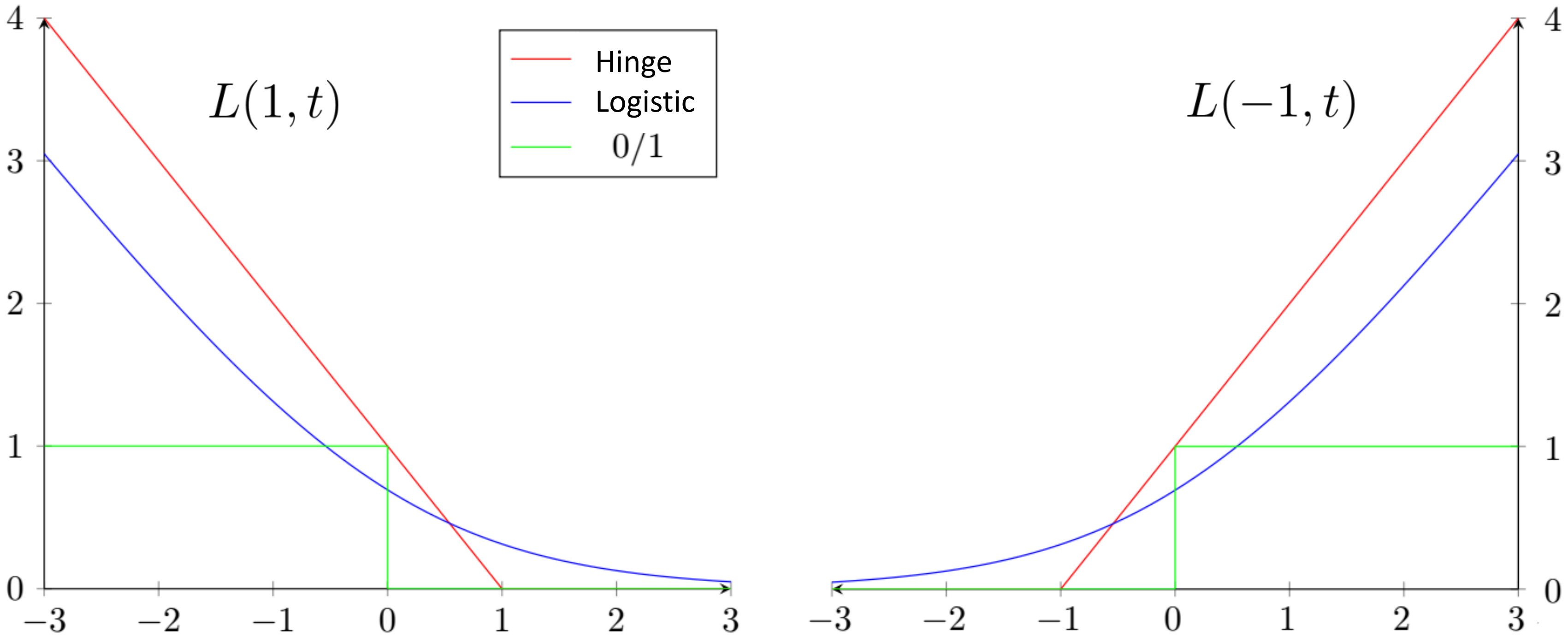
$$L(y,t) = \log(1 + \exp(-yt))$$

  - Hinge loss

$$L(y,t) = \max(0,1 - yt)$$

Surrogate losses

- Graphical representation of these losses



Logistic Regression

- On a homework assignment, you will show that

$$-\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} L\big(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\big)$$

where $\ell(\boldsymbol{\theta})$ is the logistic regression log-likelihood, L is the logistic loss, $y_i \in \{-1,1\}$, and $f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \boldsymbol{\theta}^T \widetilde{\boldsymbol{x}}_i$

- Take home message: Logistic regression can be derived from two different perspectives
  - A plug-in estimator solved via maximum likelihood
  - ERM with logistic loss

Big Picture

- *(Regularized) empirical risk minimization* learns f by solving

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, f(\boldsymbol{x}_i)\big) + \lambda \Omega(f)$$

- Different choices of $L, \mathcal{F}, \Omega$ give rise to different methods
- We will see several other examples including support vector machines, neural networks, and boosting and decision trees
- One advantage of this framework is it makes it easier to compare and contrast different methods
- Another is that there are optimization strategies for solving large classes of ERM methods