(DRAFT 0.1)Loss Functions for Regression and Classification

David Rosenberg

New York University

February 4, 2015

Loss Functions for Regression

- In general, loss function may take the form

$$(\hat{y}, y) \mapsto \ell(\hat{y}, y)$$

- Regression losses usually only depend on the **residual:**

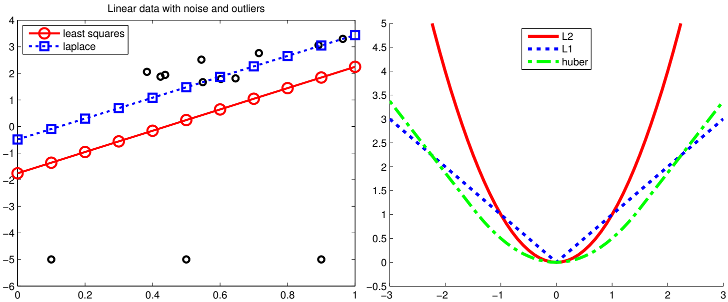
$$r = y - \hat{y}$$

$$(\hat{y}, y) \mapsto \ell(r) = \ell(y - \hat{y})$$

- When would you **not** want a translation-invariant loss?
  - Can you transform your response y so that the loss you want is translation-invariant?

Some Losses for Regression

- **Square** or ℓ_2 Loss: $\ell(r) = r^2$ (not robust to outliers, differentiable)
- **Absolute** or **Laplace** or ℓ_1 Loss: $\ell(r) = |r|$ (robust to outliers, not differentiable)
  - gives **median regression**
- **Huber** Loss: Quadratic for $|r| \leqslant \delta$ and linear for $|r| > \delta$ (robust and differentiable)



KPM Figure 7.6

The Classification Problem

- Action space $\mathcal{A} = \{-1, 1\}$ Output space $\mathcal{Y} = \{-1, 1\}$
- **0-1 loss** for $f : \mathcal{X} \to \{-1, 1\}$:

$$\ell(f(x), y) = 1(f(x) \neq y)$$

- But let's allow real-valued predictions $f : \mathcal{X} \to \mathbf{R}$:

$$f > 0 \implies \text{Predict } 1$$
$$f < 0 \implies \text{Predict } -1$$

The Classification Problem: Real-Valued Predictions

- Action space $\mathcal{A} = \mathbf{R}$ Output space $\mathcal{Y} = \{-1, 1\}$
- Prediction function $f : \mathcal{X} \to \mathbf{R}$

Definition

The value $f(x)$ is called the **score** for the input x. Generally, the magnitude of the score represents the **confidence of our prediction**.

Definition

The **margin** on an example (x, y) is $yf(x)$. The margin is a measure of how **correct** we are.

- Most classification losses depend only on the margin.
- We want to **maximize the margin**.

The Classification Problem: Real-Valued Predictions

- Empirical risk for $0-1$ loss:

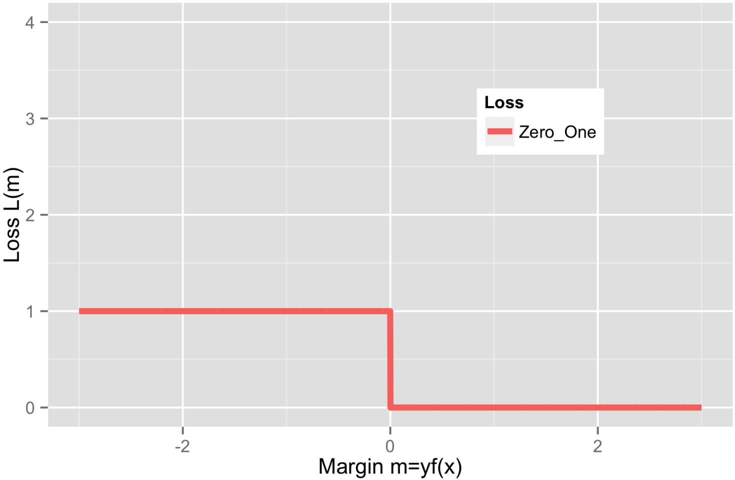
$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} 1(y_i f(x_i) \leqslant 0)$$

Minimizing empirical $0-1$ risk not computationally feasible

$\hat{R}_n(f)$ is non-convex, not differentiable (in fact, discontinuous!).
Optimization is **NP-Hard**.

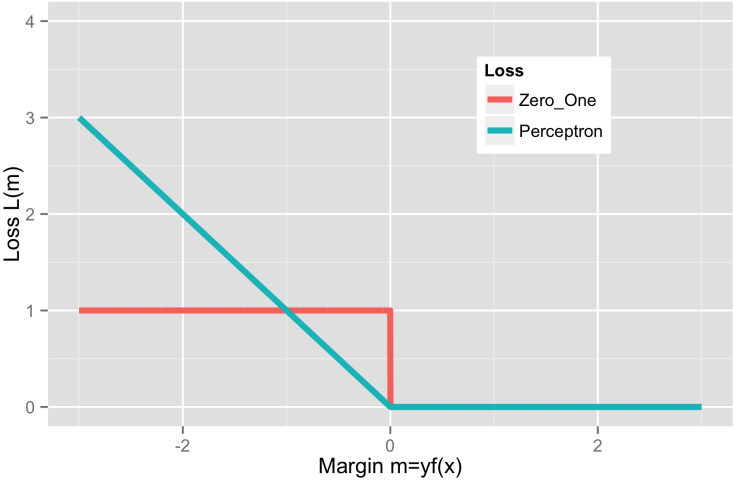
Classification Losses

- Zero-One loss: $\ell_{0\text{-}1} = \max\{1 - m, 0\}$



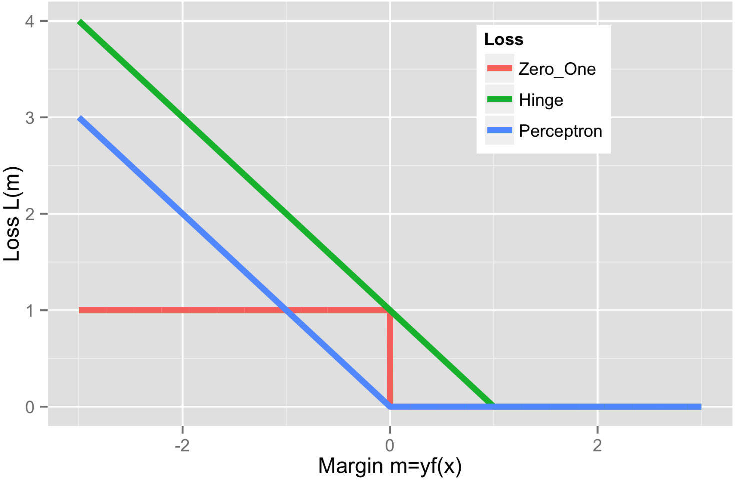
Classification Losses

- Perceptron loss: $\ell_{\text{Perceptron}} = \max\{-m, 0\}$



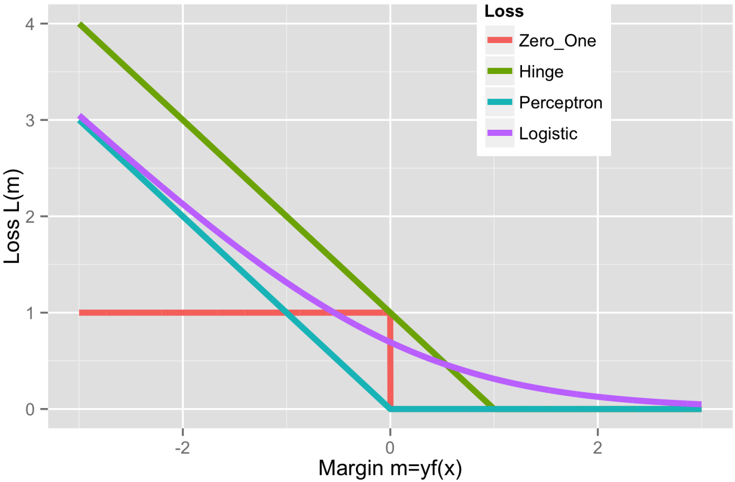
Classification Losses

- SVM/Hinge loss: $\ell_{\text{Hinge}} = \max\{1-m, 0\}$



Classification Losses

- Logistic/Log loss: $\ell_{\text{Logistic}} = \log\left(1 + e^{-m}\right)$



Classification Losses

- Logistic/Log loss: $\ell_{\text{Logistic}} = \log\left(1 + e^{-m}\right)$