Loss Functions for Regression and Classification

David Rosenberg

New York University

February 7, 2017

Regression Loss Functions

Loss Functions for Regression

- In general, loss function may take the form

$$(\hat{y}, y) \mapsto \ell(\hat{y}, y) \in \mathbf{R}$$

- Regression losses usually only depend on the **residual** $r = y - \hat{y}$.
- Loss $\ell(\hat{y}, y)$ is called **distance-based** if it
  1. only depends on the residual:

$$\ell(\hat{y}, y) = \psi(y - \hat{y}) \quad \text{for some } \psi : \mathbf{R} \to \mathbf{R}$$

  2. loss is zero when residual is 0:

$$\psi(0) = 0$$

Distance-Based Losses are Translation Invariant

- Distance-based losses are translation-invariant. That is,

$$\ell(\hat{y}+a, y+a) = \ell(\hat{y}, y).$$

- When might you not want to use a translation-invariant loss?
- e.g. Sometimes relative error is a more natural loss (but not translation-invariant)
- Often you can transform response y so it's translation-invariant (e.g. log transform)
    - See homework or concept check questions.

Some Losses for Regression

- **Square** or ℓ_2 Loss: $\ell(r) = r^2$
- **Absolute** or **Laplace** or ℓ_1 Loss: $\ell(r) = |r|$

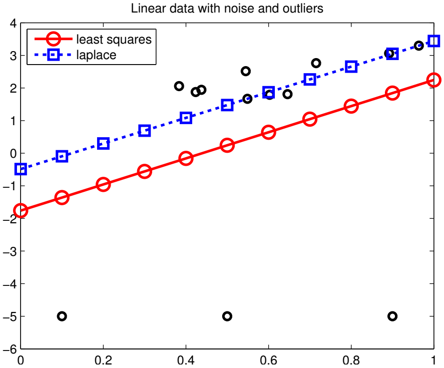
y	$\hat{y}$	$\lvert r\rvert = \lvert y - \hat{y}\rvert$	$r^2 = (y - \hat{y})^2$
1	0	1	1
5	0	5	25
10	0	10	100
50	0	50	2500

- Outliers typically have large residuals.
- Square loss much more affected by outliers than absolute loss.

KPM Figure 7.6

Loss Function Robustness

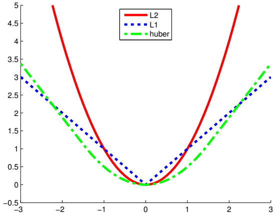
- **Robustness** refers to how affected a learning algorithm is by outliers.



KPM Figure 7.6

Some Losses for Regression

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



- x-axis is the residual $y - \hat{y}$.

KPM Figure 7.6

Classification Loss Functions

The Classification Problem

- Outcome space $\mathcal{Y} = \{-1, 1\}$
- Action space $\mathcal{A} = \{-1, 1\}$
- **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** $f : \mathcal{X} \to \mathbf{R}$:

$$\begin{aligned} f > 0 &\implies \text{Predict } 1 \\ f < 0 &\implies \text{Predict } -1 \end{aligned}$$

The Score Function

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

Definition

The value $f(x)$ is called the **score** for the input x.

- In this context, f may be called a **score function**.
- Intuitively, magnitude of the score represents the **confidence of our prediction**.

The Margin

Definition

The **margin** (or **functional margin**) on an example (x, y) is $yf(x)$.

- The margin is a measure of how **correct** we are.
- We want to **maximize the margin**.
- Most classification losses depend only on the margin.

(In Lab, we will discuss a related concept called the **geometric margin**.)

The Classification Problem: Real-Valued Predictions

- Empirical risk for $0-1$ loss:

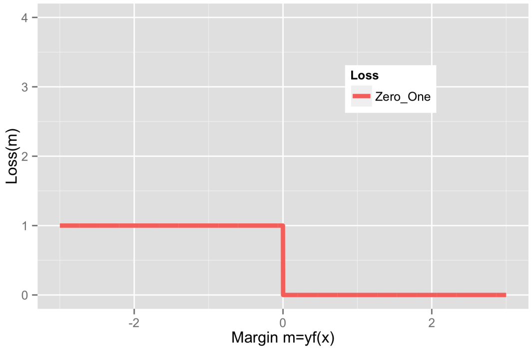
$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} 1(y_i f(x_i) \leqslant 0)$$

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

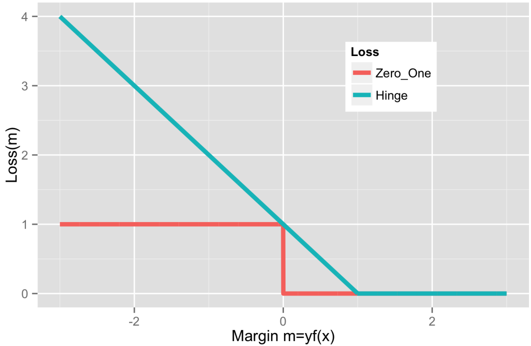
Zero-One loss: $\ell_{0\text{-}1} = 1(m \leqslant 0)$



- x-axis is **margin**: $m > 0 \iff$ correct classification

Classification Losses

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



Hinge is a **convex**, **upper bound** on $0-1$ loss. Not differentiable at $m=1$.
We have a **"margin error"** when $m<1$.

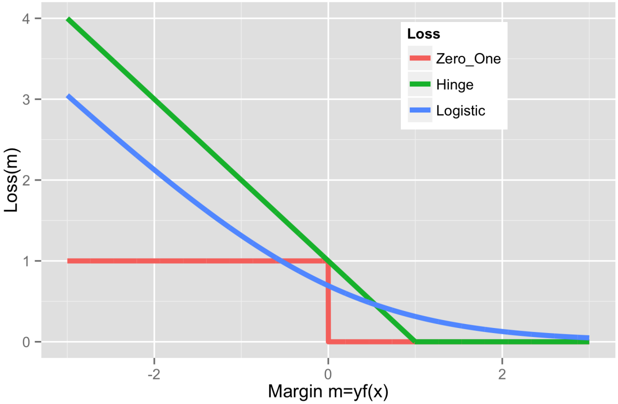
(Soft Margin) Linear Support Vector Machine

- Hypothesis space $\mathcal{F} = \left\{ f(x) = w^T x \mid w \in \mathbf{R}^d \right\}$.
- Loss $\ell(m) = (1-m)_+$
- ℓ_2 regularization

$$\min_{w \in \mathbf{R}^d} \sum_{i=1}^{n} \left(1 - y_i f_w(x_i)\right)_+ + \lambda \|w\|_2^2$$

Classification Losses

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



Logistic loss is differentiable. Logistic loss always wants more margin (loss never 0).

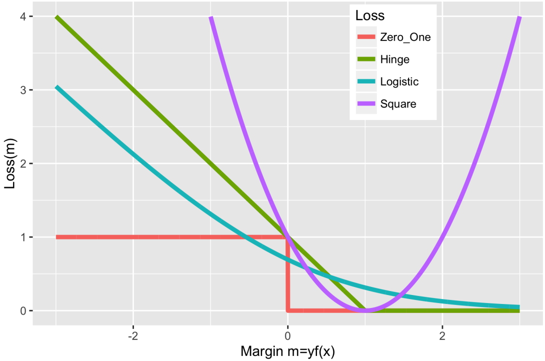
What About Square Loss for Classification?

- Action space $\mathcal{A} = \mathbf{R}$ Output space $\mathcal{Y} = \{-1, 1\}$
- Loss $\ell(f(x), y) = (f(x) - y)^2$.
- Turns out, can write this in terms of margin $m = f(x)y$:

$$\ell(f(x), y) = (f(x) - y)^2 = (1 - m)^2$$

- Prove using fact that $y^2 = 1$, since $y \in \{-1, 1\}$.

What About Square Loss for Classification?



Heavily penalizes outliers.

Seems to have higher sample complexity (i.e. needs more data) than hinge & logistic[1].

[1] Rosasco et al's "Are Loss Functions All the Same?" http://web.mit.edu/lrosasco/www/publications/loss.pdf