Loss Functions for Regression and Classification

David Rosenberg

New York University

February 7, 2016

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:**

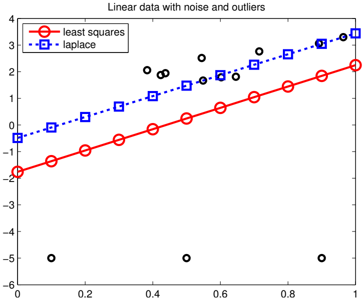
$$r = y - \hat{y}$$

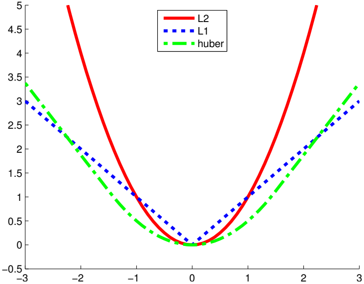
$$(\hat{y}, y) \mapsto \ell(r) = \ell(y - \hat{y})$$

- 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)
- **Absolute** or **Laplace** or ℓ_1 Loss: $\ell(r) = |r|$ (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.

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

The Classification Problem: Real-Valued Predictions

- Empirical risk for $0-1$ loss:

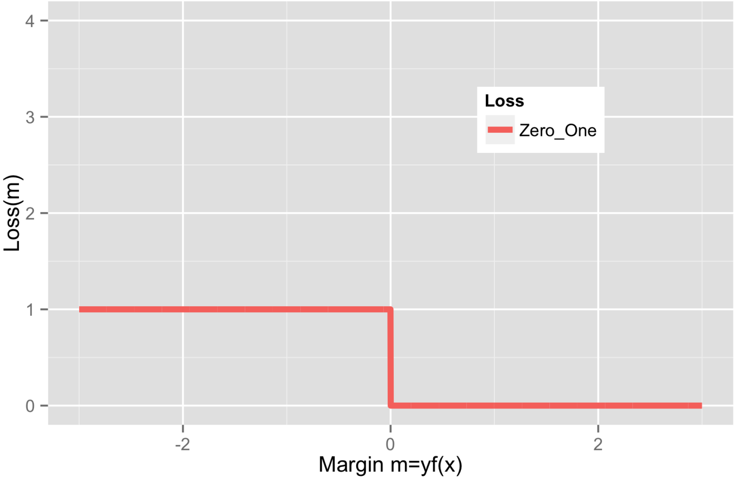
$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} 1(y_i f(x_i) \leqslant 0)$$

Minimizing empirical $0-1$ risk not computationally feasible

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

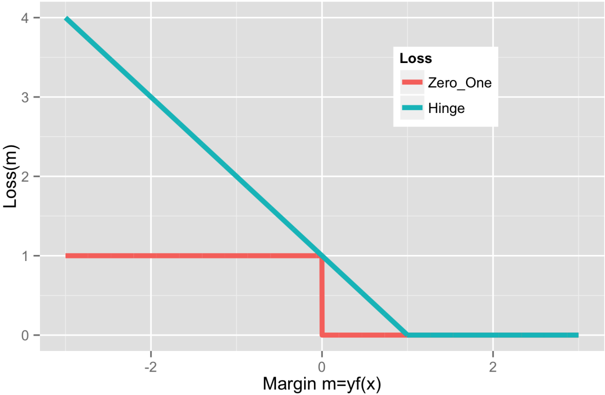
Classification Losses

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



Classification Losses

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

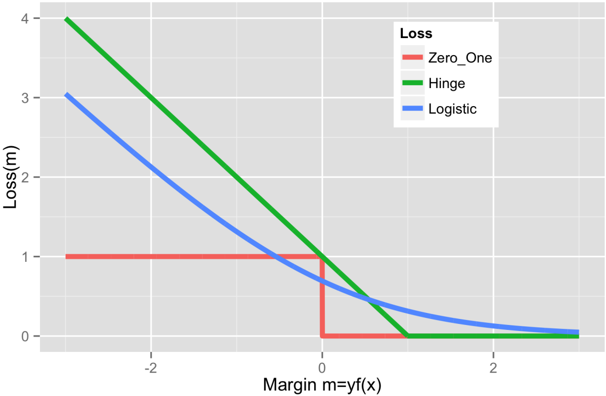


Hinge is a **convex**, **upper bound** on $0-1$ loss. Not differentiable at 1.

We have a **"margin error"** when $m < 1$.

Classification Losses

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



Logistic loss is differentiable. Never enough margin for logistic loss.

How many support vectors?

(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

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

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent

- initialize $w = 0$
- repeat
  - randomly choose training point $(x_i, y_i) \in \mathcal{D}_n$
  - $w \leftarrow w - \eta \underbrace{\nabla_w \ell(f_w(x_i), y_i)}_{\text{Grad(Loss on i'th example)}}$
- until stopping criteria met

SGD for Hinge Loss and Linear Predictors

- Consider linear hypothesis space: $f_w(x) = w^T x$.
- Gradient of hinge loss (x, y):

$$\nabla_w \ell_{\text{Hinge}}(yw^T x) = \begin{cases} -yx & \text{if } yf_w(x) < 1 \\ 0 & \text{if } yf_w(x) > 1 \\ \text{undefined} & \text{if } yf_w(x) = 1 \end{cases}$$

- A point with margin $m = yf_w(x) = 1$ is correctly classified.
  - We can skip SGD update for these points.
  - Rigorous approach: **subgradient descent**

SGD for Hinge Loss and Linear Predictors

- For step $t+1$ of SGD, we select a random training point (x,y) and set

$$w^{(t+1)} = \begin{cases} w^{(t)} + \eta^{(t)} yx & \text{if } yf_w(x) < 1 \\ w^{(t)} & \text{otherwise} \end{cases}$$

- $w^{(T)}$ is a linear combination of x_i's with margin error when selected.
- Any x_i in the expansion of $w^{(T)}$ is called a **support vector**.
- We can write:

$$\hat{w} = \sum_{i=1}^{s} a_i x^{(i)},$$

  where $x^{(1)}, \ldots, x^{(s)}$ are the support vectors.
- Having 0 gradient for $m > 1$ allows **sparse** support vectors.

Population Minimizers

The **population minimizer** is another name for risk minimizer.
It's the "infinite data" case.

Loss Function	$L[y, f(x)]$	Minimizing Function
Binomial Deviance	$\log[1 + e^{-yf(x)}]$	$f(x) = \log \frac{\Pr(Y=+1\|x)}{\Pr(Y=-1\|x)}$
SVM Hinge Loss	$[1 - yf(x)]_+$	$f(x) = \operatorname{sign}[\Pr(Y=+1\|x) - \frac{1}{2}]$
Squared Error	$[y - f(x)]^2 = [1 - yf(x)]^2$	$f(x) = 2\Pr(Y=+1\|x) - 1$

Hastie, Tibshirani, Friedman *The Elements of Statistical Learning*, 2nd Ed. Table 12.1