Loss Functions in Machine Learning

Robert Moss
Stanford University, Stanford, CA 94305
mossr@cs.stanford.edu

2020-08-22

```
𝕀(b) = b ? 1 : 0 # indicator function
margin(x, y, 𝐰, ϕ) = (𝐰⋅ϕ(x))*y
```

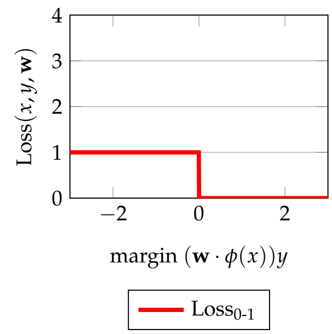
1 *Zero-One Loss*

The *zero-one loss* corresponds exactly to the notion of whether our predictor made a mistake or not. We can also write the loss in terms of the margin. Plotting the loss as a function of the margin, it is clear that the loss is 1 when the margin is negative and 0 when it is positive.

$$\text{Loss}_{0\text{-}1}(x, y, \mathbf{w}) = \mathbb{1}[\underbrace{(\mathbf{w} \cdot \phi(x))y}_{\text{margin}} \leq 0]$$

```
Loss_01(x, y, 𝐰, ϕ) = 𝕀(margin(x, y, 𝐰, ϕ) ≤ 0)
```

Figure 1. *Zero-one loss.*



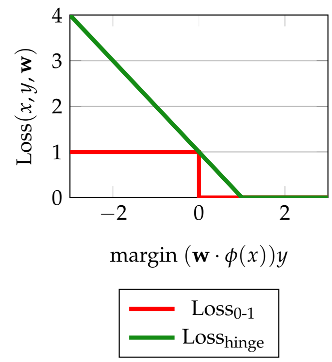
2 *Hinge Loss (SVMs)*

Hinge loss upper bounds $\text{Loss}_{0\text{-}1}$ and has a non-trivial gradient. The intuition is we try to increase the margin if it is less than 1. Minimizing upper bounds are a general idea; the hope is that pushing down the upper bound leads to pushing down the actual function.

$$\text{Loss}_{\text{hinge}}(x, y, \mathbf{w}) = \max\{1 - (\mathbf{w} \cdot \phi(x))y, 0\}$$

```
Loss_hinge(x, y, 𝐰, ϕ) = max(1 - margin(x, y, 𝐰, ϕ), 0)
```

Figure 2. *Hinge loss.*



3 *Logistic Loss*

Another popular loss function is the *logistic loss*. The intuition is we try to increase the margin even when it already exceeds 1. The main property of the logistic loss is no matter how correct your prediction is, you will have non-zero loss. Thus, there is still an incentive (although diminishing) to increase the margin. This means that you'll update on every single example.

$$\text{Loss}_{\text{logistic}}(x, y, \mathbf{w}) = \log(1 + e^{-(\mathbf{w} \cdot \phi(x))y})$$

```
Loss_logistic(x, y, 𝐰, ϕ) = log(1 + exp(-margin(x, y, 𝐰, ϕ)))
```

Figure 3. *Logistic loss.*