1 Generalization and Stability

In general, most classification problems are an optimization over the objective

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

over some specified loss function and a possible regularization term (sometimes we may set $\lambda = 0$). The most simple loss function that we can optimize is the 0-1 **step loss**:

$$L_{\text{STEP}}(y, \mathbf{w}^\top \mathbf{x} - b) = \begin{cases} 1 & y(\mathbf{w}^\top \mathbf{x} - b) < 0 \\ 0 & y(\mathbf{w}^\top \mathbf{x} - b) \geq 0 \end{cases}$$

The 0-1 loss is 0 if $\mathbf{x}$ is correctly classified and 1 otherwise. Minimizing $\frac{1}{n}\sum_{i=1}^{n} L(y_i, \mathbf{w}^\top \mathbf{x}_i - b)$ directly minimizes classification error on the training set. However, the 0-1 loss is difficult to optimize: it is neither convex nor differentiable (see Figure 1). Furthermore, if we exclude the regularization term, we do not penalize the classifier for being close to the training points, which leads to generalization issues.

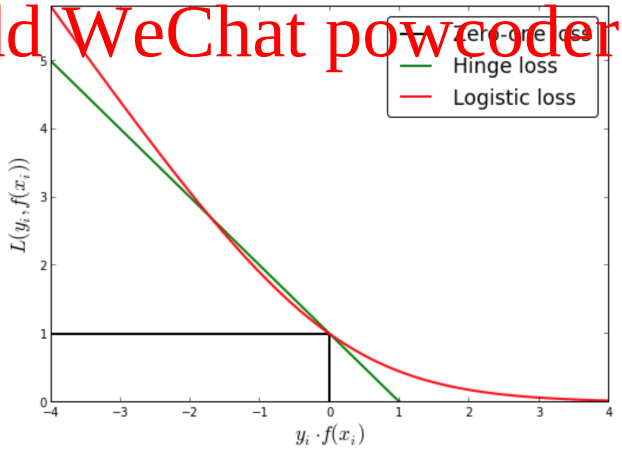

Figure 1: Step (0-1) loss, hinge loss, and logistic loss. Logistic loss is convex and differentiable, hinge loss is only convex, and step loss is neither.

Another loss function that we have seen is the **logistic loss**, which is used in logistic regression:

$$L_{\text{LR}}(y, \mathbf{w}^\top \mathbf{x} - b) = y \ln\left(\frac{1}{s(\mathbf{w}^\top \mathbf{x} - b)}\right) + (1-y)\ln\left(\frac{1}{1 - s(\mathbf{w}^\top \mathbf{x} - b)}\right)$$

The logistic loss is convex and differentiable, and is optimized using gradient descent methods. The logistic loss is the basis for logistic regression, and it works well without regularization:

$$\min_{\mathbf{w},b} \frac{1}{n} \sum_{i=1}^{n} L_{\text{LR}}(y_i, \mathbf{w}^\top \mathbf{x}_i - b)$$

The **hinge loss** modifies the 0-1 loss to be convex. The points with $y(\mathbf{w}^\top \mathbf{x} - b) \geq 0$ should remain at 0 loss, but we may consider allowing a linear penalty "ramp" for misclassified points. This leads us to the hinge loss, as illustrated in Figure 1:

$$L_{\text{HINGE}}(y, \mathbf{w}^\top \mathbf{x} - b) = \max(1 - y(\mathbf{w}^\top \mathbf{x} - b), 0)$$

The ramp ensures that misclassified points that are close to the boundary are penalized less than misclassified points that are far from the boundary. The perceptron algorithm optimizes over the sum of hinge losses contributed from all of the training points:

$$\min_{\mathbf{w},b} \frac{1}{n} \sum_{i=1}^{n} L_{\text{HINGE}}(y_i, \mathbf{w}^\top \mathbf{x}_i - b)$$

The SVM formulation is an optimization over the same problem, with the addition of a regularization term:

$$\min_{\mathbf{w},b} \frac{1}{n} \sum_{i=1}^{n} L_{\text{HINGE}}(y_i, \mathbf{w}^\top \mathbf{x}_i - b) + \lambda \|\mathbf{w}\|^2$$

The regularization term allows for better generalization, in this case by penalizing choices of $\mathbf{w}$ for which the margin is small.