1 Minimize losses

1.1 Loss minimization of logistic regression

Let’s revisit the negative log-likelihood, which is minimized in optimization of logistic regression model. Let’s drop instance indices for a while for notation simplicity. Remember, \( \eta \) is the score given by the linear product, and \( y \) is the true label. We use \( \ell \) to denote the negative log-likelihood of a single instance with \( \log_2 \), and we call it loss. Loss minimization is only reasonable by its name now, but we will see it is actually another way to formulate classifiers.

\[
\ell = y \log(1 + \exp(-\eta)) + (1 - y) \log(1 + \exp(\eta))
\]

Now we plot the function out for two cases, \( y = 1 \) and \( y = 0 \).

![Figure 1: Loss function of logistic regression model. Blue for \( y = 0 \); red for \( y = 1 \)](image)

Now let’s consider the binary loss we have used to evaluate our classifier. How different values of \( \eta \) brings different binary losses?

![Figure 2: Loss function of logistic regression model. Blue for \( y = 0 \); red for \( y = 1 \)](image)

You can see that the two loss functions for \( y = 1 \) and \( y = 0 \) are symmetric to each other. Now let’s consider only the case \( y = 0 \). Plot the two loss functions – logistic loss and binary loss together. You can see logistic loss is an upper bound of the binary loss.

From the perspective of optimization, a hard-to-treat objective can be minimized through minimizing its upper bound (with multiplication to a constant \( \log 2 \)).
Logistic regression is essential doing this – minimizing the binary loss by minimizing its upper bound.

\[
\min_{\mathbf{w}, w_0} \sum_{i=1}^{N} \ell_{lr}(\mathbf{w}, w_0) \geq \log 2 \cdot \sum_{i=1}^{N} \ell_{bin}(\mathbf{w}, w_0) \tag{1}
\]

Here \( \ell_{lr}(\mathbf{w}, w_0) \) and \( \ell_{bin}(\mathbf{w}, w_0) \) are logistic loss and binary loss incurred by model parameters \( \mathbf{w} \) and \( w_0 \).

You may have this question: why don’t we directly minimize the binary loss – as minimizing upper bounds does not necessarily get even a local minimum of \( \mathbf{w} \) and \( w_0 \) for the binary loss, which is what we really want to minimize.

There are two reasons. First, it is not possible to find a global minimum of the binary loss. Second, the function itself consists plateaus at different levels, which does not admit gradient-based optimization algorithms. [There are work on direction optimization of binary losses, but these algorithms do not scale to large problems in general]. There is trade-off here: inexact solution in the sense of solving only an upper bound versus inexact solution in the sense of solving a hard problem. It seems that the first one win out.

### 1.2 Support Vector Machine

Are these better losses? We want the bound to be as tight as possible, and we also want the loss function to be convex. The hinge loss is your choice.

Let’s follow the tradition of SVM and denote class labels as \( y = 1 \) and \( y = -1 \) instead of \( y = 1 \) and \( y = 0 \) for logistic regression.
Hinge loss for the two cases, $y = -1$ and $y = 1$, can be expressed as

$$\ell^{\text{hin}} = \max(0, 1 + \eta) \text{ for } y = -1 \quad (2)$$

$$\ell^{\text{hin}} = \max(0, 1 - \eta) \text{ for } y = 1 \quad (3)$$

Write the two cases together we get

$$\ell^{\text{hin}} = \max(0, 1 - y\eta) \quad (4)$$

Then the optimization problem for SVM becomes

$$\min_{w, w_0} \frac{1}{N} \sum_{i=1}^{N} \ell^{\text{hin}}_i + \frac{\lambda}{2} \|w\|_2^2 \quad (5)$$

Here we still have the regularization term. Expand the objective, then we have,

$$\min_{w, w_0} \frac{1}{N} \sum_{i=1}^{N} \max(0, 1 - y_i(w^\top x_i + w_0)) + \frac{\lambda}{2} w^\top w \quad (6)$$

One can solve this optimization problem with subgradient descent algorithms. One equivalent form (why?) is

$$\min_{w, w_0} \frac{1}{N} \sum_{i=1}^{N} \xi_i + \frac{\lambda}{2} w^\top w \quad (7)$$

$$y_i w^\top x_i + w_0 - 1 \geq \xi_i \quad (8)$$

$$\xi_i \geq 0 \quad (9)$$

[This is the primal form of SVM. We will talk about dual form later.] People have studied this form for many years to speed up the optimization of SVM. The CIML book and many other books also discuss the case where one use hard margin.

The hard margin formulation

$$\min_{w, w_0} \frac{\lambda}{2} w^\top w \quad (10)$$

$$y_i (w^\top x_i + w_0) - 1 \geq 0 \quad (11)$$

But this form does not admit a feasible solution when the data is not linearly separable.