1 Mathematics Basics

1.1 Optimization

Solution: The Lagrange function is

$$\mathcal{L}(x_1, x_2, w, v) = x_1^2 + x_2^2 - 1 - w(x_1 - 2x_2) - v(x_1 + x_2 - 1)$$

The KKT condition is

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial x_1} = 2x_1 - w - v = 0 \\ \dfrac{\partial \mathcal{L}}{\partial x_2} = 2x_2 + 2w - v = 0 \\ w(x_1 - 2x_2) = 0 \\ w \geq 0 \\ x_1 - 2x_2 \geq 0 \\ x_1 + x_2 - 1 = 0 \end{cases}$$

Note the condition $w(x_1 - 2x_2)$. If $w = 0$, then we have $2x_1 = v = 2x_2$ from the first two conditions. Combined with the last condition $x_1 + x_2 - 1 = 0$, we have $x_1 = x_2 = \frac{1}{2}$. However, this solution doesn't satisfy $x_1 - 2x_2 \geq 0$. So we can only have $x_1 - 2x_2 = 0$. Combined with the last condition $x_1 + x_2 - 1 = 0$, we have $x_1 = \frac{2}{3}, x_2 = \frac{1}{3}$. Substitute into the first two conditions, we get $w + v = \frac{4}{3}, 2w - v = -\frac{2}{3}$. Thus $w = \frac{2}{9}, v = \frac{10}{9}$. So $x_1 = \frac{2}{3}, x_2 = \frac{1}{3}, w = \frac{2}{9}, v = \frac{10}{9}$ satisfies all the conditions. Since the objective function is convex and the constraints are linear, KKT point is the global optimal solution. Thus, the solution is $x_1 = \frac{2}{3}, x_2 = \frac{1}{3}$, and the minimum value of objective function is $x_1^2 + x_2^2 - 1 = -\frac{4}{9}$.

1.2 Calculus

(1) **Proof:**

$$\begin{aligned} \Gamma(x+1) &= \int_0^\infty u^x e^{-u} \,\mathrm{d}u \\ &= -u^x e^{-u}\Big|_0^\infty + \int_0^\infty x u^{x-1} e^{-u} \,\mathrm{d}u \\ &= x \int_0^\infty u^{x-1} e^{-u} \,\mathrm{d}u \\ &= x\Gamma(x) \end{aligned}$$

(2) **Proof:**

$$\begin{aligned} \Gamma(a)\Gamma(b) &= \int_0^\infty u^{a-1} e^{-u} \,\mathrm{d}u \int_0^\infty v^{b-1} e^{-v} \,\mathrm{d}v \\ &= \int_0^\infty \int_0^\infty u^{a-1} v^{b-1} e^{-(u+v)} \,\mathrm{d}u \,\mathrm{d}v \end{aligned}$$

Let $u = xy, v = x(1-y)$, where $x \in [0, \infty), y \in [0, 1]$. The Jacobian matrix for this transformation is

$$J = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} = \begin{bmatrix} y & x \\ 1-y & -x \end{bmatrix}$$

Thus $|J| = -xy - x(1-y) = -x$. So

$$\begin{aligned} \Gamma(a)\Gamma(b) &= \int_0^1 \int_0^\infty (xy)^{a-1}(x(1-y))^{b-1} e^{-(xy+x(1-y))} |-x| \,\mathrm{d}x\,\mathrm{d}y \\ &= \int_0^1 \int_0^\infty x^{a+b-1} y^{a-1}(1-y)^{b-1} e^{-x} \,\mathrm{d}x\,\mathrm{d}y \\ &= \int_0^\infty x^{a+b-1} e^{-x} \,\mathrm{d}x \int_0^1 y^{a-1}(1-y)^{b-1} \,\mathrm{d}y \\ &= \Gamma(a+b) \int_0^1 y^{a-1}(1-y)^{b-1} \,\mathrm{d}y \end{aligned}$$

Thus

$$\int_0^1 y^{a-1}(1-y)^{b-1} \,\mathrm{d}y = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

1.3 Probability

Since $\lambda \sim \Gamma(\lambda|\alpha, \beta)$,

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}}{\Gamma(\alpha)}.$$

Since $x|\lambda \sim \text{Poisson}(x|\lambda)$,

$$p(x|\lambda) = \frac{\lambda^x}{x!} e^{-\lambda}$$

Thus

$$\begin{aligned} p(x|\alpha, \beta) &= \int_0^\infty p(\lambda|\alpha, \beta) p(x|\lambda) \,\mathrm{d}\lambda \\ &= \int_0^\infty \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}}{\Gamma(\alpha)} \frac{\lambda^x}{x!} e^{-\lambda} \,\mathrm{d}\lambda \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)x!} \int_0^\infty \lambda^{\alpha+x-1} e^{-(\beta+1)\lambda} \,\mathrm{d}\lambda \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)x!} \int_0^\infty \left(\frac{u}{\beta+1}\right)^{\alpha+x-1} e^{-u} \,\mathrm{d}\left(\frac{u}{\beta+1}\right) \\ &= \frac{\beta^\alpha}{(\beta+1)^{\alpha+x}\Gamma(\alpha)x!} \int_0^\infty u^{\alpha+x-1} e^{-u} \,\mathrm{d}u \\ &= \frac{\beta^\alpha \Gamma(\alpha+x)}{(\beta+1)^{\alpha+x}\Gamma(\alpha)x!} \end{aligned}$$

Thus

$$\begin{aligned} p(\lambda|x,\alpha,\beta) &= \frac{p(\lambda|\alpha,\beta)p(x|\lambda,\alpha,\beta)}{p(x|\alpha,\beta)} \\ &= \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}}{\Gamma(\alpha)} \cdot \frac{\lambda^x e^{-\lambda}}{x!} \cdot \frac{(\beta+1)^{a+x}\Gamma(\alpha)x!}{\beta^\alpha \Gamma(\alpha+x)} \\ &= \frac{(\beta+1)^{\alpha+x}\lambda^{\alpha+x-1}e^{-(\beta+1)\lambda}}{\Gamma(\alpha+x)} \end{aligned}$$

Thus $\lambda|x \sim \Gamma(\lambda|\alpha+x, \beta+1)$.

1.4 Stochastic Process

Let $x_1, x_2, x_3, \dots$ be a sequence of coins tosses, where $x_i \in \{H, T\}, \forall i \geq 1$. Let y be the average number of tosses needed for the first occurrence of pattern

$$H, \underbrace{T, \dots, T}_{k}$$

in the sequence **given** $x_1 = H$.

We first calculate y. All possible sequences given $x_1 = H$ are in one of the following forms,

$$\begin{aligned} &H, H, \dots \\ &H, T, H, \dots \\ &H, T, T, H, \dots \\ &H, T, T, T, H, \dots \\ &\quad\vdots \\ &H, \underbrace{T, \dots, T}_{k-1}, H, \dots \\ &H, \underbrace{T, \dots, T, T}_{k}, \dots \end{aligned}$$

Note that all these forms are exclusive with each other. We call $H, \underbrace{T, \dots, T}_{k}$ the *target pattern*.

First consider the first form $H, H, \dots$. In this case the target pattern can only appear after the first H, thus the average number of tosses for the first occurrence of target pattern is $y + 1$.

Then we consider the second form $H, T, H, \dots$. In this case the target pattern can only appear after the H, T, thus the average number of tosses for the first occurrence of target pattern is $y + 2$. This process repeats for all other forms.

Let Y be the random variable which records the number of tosses for the first occurrence of target pattern. Then $y = \mathbb{E}[Y|x_1 = H]$. We can formulate the idea above as follows,

$$\begin{aligned}
&\mathbb{E}\left[Y|x_1 = H, x_2 = H\right] = \mathbb{E}[Y|x_1 = H] + 1\\
&\mathbb{E}\left[Y|x_1 = H, x_2 = T, x_3 = H\right] = \mathbb{E}[Y|x_1 = H] + 2\\
&\qquad\vdots\\
&\mathbb{E}\left[Y|x_1 = H, x_2 = T, \dots, x_k = T, x_{k+1} = H\right] = \mathbb{E}[Y|x_1 = H] + k\\
&\mathbb{E}\left[Y|x_1 = H, x_2 = T, \dots, x_k = T, x_{k+1} = T\right] = k + 1
\end{aligned}$$

Thus

$$\begin{aligned}
y = \mathbb{E}[Y|x_1 = H] &= \mathbb{E}\left[Y|x_1 = H, x_2 = H\right] P(x_2 = H|x_1 = H)\\
&+ \mathbb{E}\left[Y|x_1 = H, x_2 = T, x_3 = H\right] P(x_2 = T, x_3 = H|x_1 = H)\\
&\ \vdots\\
&+ (k+1)P(x_2 = T, \dots, x_k = T, x_{k+1} = T|x_1 = H)\\
&= \sum_{i=1}^{k} \frac{y+i}{2^i} + \frac{k+1}{2^k}\\
&= \frac{1}{2}y + \frac{1}{2^2}y + \cdots + \frac{1}{2^k}y + \sum_{i=1}^{k}\frac{i}{2^i} + \frac{k+1}{2^k}\\
&= (1 - \frac{1}{2^k})y + (2 - \frac{1}{2^{k-1}} - \frac{k}{2^k}) + \frac{k+1}{2^k} = (1 - \frac{1}{2^k})y + 2 - \frac{1}{2^k}
\end{aligned}$$

Which yields $y = 2^{k+1} - 1$.

Let $z = \mathbb{E}[Y|x_1 = T]$. Using similar idea as we calculate y,

$$\begin{aligned}
z = \mathbb{E}[Y|x_1 = T] &= \mathbb{E}[Y|x_1 = T, x_2 = H]P(x_2 = H|x_1 = T)+\\
&= \mathbb{E}[Y|x_1 = T, x_2 = T]P(x_2 = T|x_1 = T)\\
&= \frac{1}{2}(y+1) + \frac{1}{2}(z+1)
\end{aligned}$$

Thus $z = y + 2 = 2^{k+1} + 1$. Finally,

$$\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}[Y|x_1 = T]P(x_1 = T) + \mathbb{E}[Y|x_1 = H]P(x_1 = H)\\
&= \frac{1}{2}y + \frac{1}{2}z = 2^{k+1}
\end{aligned}$$

Thus on average, it takes 2^{k+1} steps for the first occurrence of the target pattern.

2 SVM

Let $\mathbf{1} = (1, ..., 1)^T \in \mathbb{R}^N$, $\boldsymbol{\xi} = (\xi_1, ..., \xi_N)^T \in \mathbb{R}^N$, $\hat{\boldsymbol{\xi}} = (\hat{\xi}_1, ..., \hat{\xi}_N)^T \in \mathbb{R}^N$, $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^T \in \mathbb{R}^{N\times d}$, and $\mathbf{y} = (y_1, ..., y_N)^T \in \mathbb{R}^N$. Let $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4 \in \mathbb{R}_+^N$ be the Lagrange multipliers.

Then the Lagrange function is

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = & \frac{1}{2}\|\mathbf{w}\|^2 + C\mathbf{1}^T\boldsymbol{\xi} + C\mathbf{1}^T\hat{\boldsymbol{\xi}} \\ & - \mathbf{u}_1^T\mathbf{X}\mathbf{w} - b\mathbf{1}^T\mathbf{u}_1 - \epsilon\mathbf{1}^T\mathbf{u}_1 - \boldsymbol{\xi}^T\mathbf{u}_1 + \mathbf{y}^T\mathbf{u}_1 \\ & + \mathbf{u}_2^T\mathbf{X}\mathbf{w} + b\mathbf{1}^T\mathbf{u}_2 - \epsilon\mathbf{1}^T\mathbf{u}_2 - \hat{\boldsymbol{\xi}}^T\mathbf{u}_2 - \mathbf{y}^T\mathbf{u}_2 \\ & - \boldsymbol{\xi}^T\mathbf{u}_3 - \hat{\boldsymbol{\xi}}^T\mathbf{u}_4\end{aligned}$$

Rearrange the order,

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = & \frac{1}{2}\|\mathbf{w}\|^2 + (\mathbf{u}_2^T\mathbf{X} - \mathbf{u}_1^T\mathbf{X})\mathbf{w} \\ & + (\mathbf{1}^T\mathbf{u}_2 - \mathbf{1}^T\mathbf{u}_1)b \\ & + (C\mathbf{1} - \mathbf{u}_1 - \mathbf{u}_3)^T\boldsymbol{\xi} \\ & + (C\mathbf{1} - \mathbf{u}_2 - \mathbf{u}_4)^T\hat{\boldsymbol{\xi}} \\ & + (\mathbf{y} - \epsilon\mathbf{1})^T\mathbf{u}_1 \\ & - (\epsilon\mathbf{1} + \mathbf{y})^T\mathbf{u}_2\end{aligned}$$

Now consider the value of $\inf_{\mathbf{w},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$. Since $b \in \mathbb{R}$, so when $\mathbf{1}^T\mathbf{u}_2 - \mathbf{1}^T\mathbf{u}_1 \neq 0, \inf_{\mathbf{w},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}} \mathcal{L} = -\infty$. Since $\boldsymbol{\xi} \geq 0$, when $C\mathbf{1} - \mathbf{u}_1 - \mathbf{u}_3 \ngeq \mathbf{0}$, $\inf_{\mathbf{w},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}} \mathcal{L} = -\infty$. Similarly, when $C\mathbf{1} + \mathbf{u}_2 - \mathbf{u}_4 \ngeq \mathbf{0}$, $\inf_{\mathbf{w},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}} \mathcal{L} = -\infty$. When $\mathbf{1}^T\mathbf{u}_2 - \mathbf{1}^T\mathbf{u}_1 = 0, C\mathbf{1} - \mathbf{u}_1 - \mathbf{u}_3 \geq \mathbf{0}$ and $C\mathbf{1} + \mathbf{u}_2 - \mathbf{u}_4 \geq \mathbf{0}$, to minimize $\mathcal{L}$, we should set $b = 0, \boldsymbol{\xi} = \mathbf{0}$, and $\hat{\boldsymbol{\xi}} = \mathbf{0}$. In this case, we have

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = \mathbf{w} + \mathbf{X}^T\mathbf{u}_2 - \mathbf{X}^T\mathbf{u}_1$$

Since $\mathcal{L}$ is a convex function of $\mathbf{w}$, so we can get the optimal value of $\mathbf{w}$ by setting $\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 0$, which yields $\mathbf{w} = \mathbf{X}^T\mathbf{u}_1 - \mathbf{X}^T\mathbf{u}_2$. Substitute this into $\mathcal{L}$, we get

$$\begin{aligned}\mathcal{L}(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) &= \inf_{\mathbf{w},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) \\ &= -\frac{1}{2}\|\mathbf{X}^T(\mathbf{u}_1 - \mathbf{u}_2)\|^2 + (\mathbf{y} - \epsilon\mathbf{1})^T\mathbf{u}_1 - (\epsilon\mathbf{1} + \mathbf{y})^T\mathbf{u}_2\end{aligned}$$

To satisfy the condition $C\mathbf{1} - \mathbf{u}_1 - \mathbf{u}_3 \geq \mathbf{0}$, we have $C\mathbf{1} \geq \mathbf{u}_1 + \mathbf{u}_3 \geq \mathbf{u}_1$. Similarly $C\mathbf{1} \geq \mathbf{u}_2$. Thus the dual problem is

$$\max_{\mathbf{u}_1,\mathbf{u}_2} -\frac{1}{2}\|\mathbf{X}^T(\mathbf{u}_1 - \mathbf{u}_2)\|^2 + (\mathbf{y} - \epsilon\mathbf{1})^T\mathbf{u}_1 - (\epsilon\mathbf{1} + \mathbf{y})^T\mathbf{u}_2$$

$$\begin{aligned} s.t.\quad & \mathbf{0} \leq \mathbf{u}_1 \leq C\mathbf{1} \\ & \mathbf{0} \leq \mathbf{u}_2 \leq C\mathbf{1} \\ & \mathbf{1}^T\mathbf{u}_1 = \mathbf{1}^T\mathbf{u}_2\end{aligned}$$

3 IRLS for Logistic Regression

3.1 Derivation

First derive the formula for IRLS. Let

$$\mathbf{p} = \left[\frac{e^{\mathbf{w}^T\mathbf{x}_1}}{1+e^{\mathbf{w}^T\mathbf{x}_1}}, \dots, \frac{e^{\mathbf{w}^T\mathbf{x}_N}}{1+e^{\mathbf{w}^T\mathbf{x}_N}}\right]^T$$

$$\mathbf{H} = \mathrm{diag}\left(\frac{e^{\mathbf{w}^T\mathbf{x}_1}}{(1+e^{\mathbf{w}^T\mathbf{x}_1})^2}, \dots, \frac{e^{\mathbf{w}^T\mathbf{x}_N}}{(1+e^{\mathbf{w}^T\mathbf{x}_N})^2}\right)$$

$$\mathbf{X} = \left[\mathbf{x}_1, \dots \mathbf{x}_N\right], \mathbf{y} = \left[y_1, \dots, y_N\right]^T$$

Compute the gradient and Hessian,

$$\begin{aligned}\nabla\left(-\frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \mathcal{L}(\mathbf{w})\right) &= -\lambda\mathbf{w} + \nabla\mathcal{L}(\mathbf{w}) \\ &= -\lambda\mathbf{w} + \sum_{i=1}^N y_i\mathbf{x}_i - \sum_{i=1}^N \frac{e^{\mathbf{w}^T\mathbf{x}_i}}{1+e^{\mathbf{w}^T\mathbf{x}_i}}\mathbf{x}_i \\ &= -\lambda\mathbf{w} + \mathbf{X}(\mathbf{y}-\mathbf{p})\end{aligned}$$

Thus

$$\begin{aligned}\nabla^2\left(-\frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \mathcal{L}(\mathbf{w})\right) &= -\lambda\mathbf{I} - \sum_{i=1}^N \mathbf{x}_i \frac{e^{\mathbf{w}^T\mathbf{x}_i}(1+e^{\mathbf{w}^T\mathbf{x}_i}) - e^{2\mathbf{w}^T\mathbf{x}_i}}{(1+e^{\mathbf{w}^T\mathbf{x}_i})^2}\mathbf{x}_i^T \\ &= -\lambda\mathbf{I} - \sum_{i=1}^N \mathbf{x}_i \frac{e^{\mathbf{w}^T\mathbf{x}_i}}{(1+e^{\mathbf{w}^T\mathbf{x}_i})^2}\mathbf{x}_i^T \\ &= -\lambda\mathbf{I} - \mathbf{X}\mathbf{H}\mathbf{X}^T\end{aligned}$$

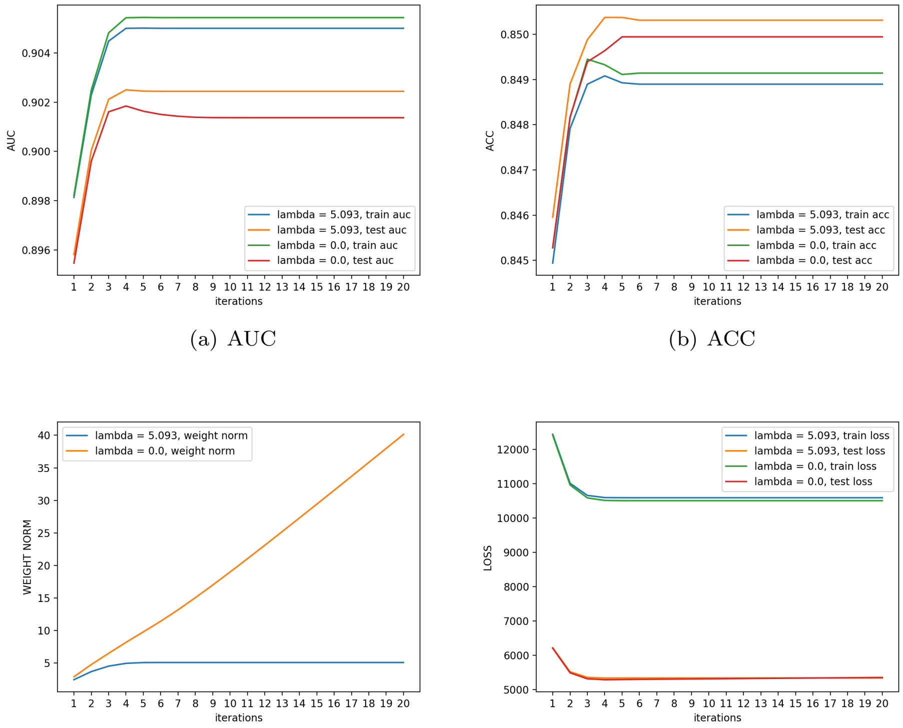
With one Newton step, the new parameter $\mathbf{w}'$ is

$$\begin{aligned}\mathbf{w}' &= \mathbf{w} + (\lambda\mathbf{I} + \mathbf{X}\mathbf{H}\mathbf{X}^T)^{-1}(-\lambda\mathbf{w} + \mathbf{X}(\mathbf{y}-\mathbf{p})) \\ &= (\lambda\mathbf{I} + \mathbf{X}\mathbf{H}\mathbf{X}^T)^{-1}(\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{w} + \mathbf{X}(\mathbf{y}-\mathbf{p})) \\ &= (\lambda\mathbf{I} + \mathbf{X}\mathbf{H}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{H}\mathbf{z}\end{aligned}$$

where $\mathbf{z} = \mathbf{X}^T\mathbf{w} + \mathbf{H}^{-1}(\mathbf{y}-\mathbf{p})$.

3.2 Experiment

We record the training and testing accuracy. We also record the 2-norm of $\mathbf{w}$ and the loss value. For the regularized model, we select the parameter λ using 5-fold cross validation. We randomly sample 20 values between 10^{-5} and 10^5 for λ and choose the best one according to the average prediction accuracy on the 5 runs. We find that $\lambda = 5.093$ is the best one among all 20 trials. Table 1 lists the results after 20 iterations. Table 2 lists parts of the parameters we tried in the 5-fold cross validation and their average prediction AUC. Figure 1 plots the AUC, accuracy rate, weight norm and loss value $-\mathcal{L}(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$ on both training and testing datasets w.r.t. the number of iterations, for both $\lambda = 0$ and $\lambda = 5.093$. We have following observations



(a) AUC

(b) ACC

(c) 2-norm of $\mathbf{w}$

(d) Loss

Figure 1: Results for $\lambda = 0$ and $\lambda = 5.093$

Table 1: Results After 20 Iterations

Model	Train AUC	Test AUC	Train ACC	Test ACC	$\|\mathbf{w}\|_2$
$\lambda = 5.093$	0.90500	0.90244	0.84889	0.85031	5.059
$\lambda = 0$	0.90544	0.90137	0.84914	0.84994	40.15

1. With regularization, we can reduce overfitting. In (a) of Figure 1, the training AUC with $\lambda = 0$ is higher than $\lambda = 5.093$. Without the regularization term, we only maximize the likelihood and fit the training data

Table 2: Part of Results of 5-Fold Cross Validation

λ	5.5×10^{-3}	1.226	5.093	18.181	173.763	732.445
avg. AUC	0.90307	0.90337	**0.90339**	0.90306	0.89865	0.89059

better. However, the testing AUC of $\lambda = 0$ is worse than that of $\lambda = 5.093$. With regularization, we can control the complexity of the model, making it generalize better to unseen data. The same effect can be observed in Figure 1(b). The only difference between Figure 1(a) and Figure 1(b) is that, in (a) the training AUC is higher, while in (b) the testing accuracy is higher (Though it's a little strange, but I've verified carefully for this). This perhaps indicates that AUC is a better metric for binary classification.

2. The regularization term $\frac{\lambda}{2}\|\mathbf{w}\|_2^2$ do limit the norm of the weights. In Figure 1(c), $\|\mathbf{w}\|_2$ grows linearly with the number of iterations when $\lambda = 0$. However, when $\lambda = 5.093$, $\|\mathbf{w}\|_2$ is bounded at around 5.0. Increasing $\|\mathbf{w}\|_2$ has little effect on the final prediction, which can be seen by comparing Figure 1(a) and Figure 1(c). On the other hand, larger $\|\mathbf{w}\|_2$ makes the classifier sensitive to little perturbation in data. Since we are classifying using $\mathbf{w}^T\mathbf{x}$, when $\mathbf{w}$ is large, small change in $\mathbf{x}$ can have large effect on the prediction outcome. This makes the model less robust, which accounts for the worse generalization for $\lambda = 0$.

3. IRLS converges very quickly, in about 5 iterations. This is partly because IRLS is a second-order method. Also, IRLS do not have a learning rate as other optimizers like SGD of Adam. Thus it is more likely to be captured in a bad local optimum.

4. In Table 2, when λ grows very large, the AUC drops severely. In these cases the complexity of the model is too small to fit the data, resulting in high bias.