Assignment 1 for #70240413
"Statistical Machine Learning"

Kui XU, 2016311209

2017/03/23

1 Mathematics Basics

Choose one problem from the 1.1 and 1.2. A bonus would be given if you finished the both.

1.1 Calculus

The gamma function is defined by (assuming x > 0)

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du. \tag{1}$$

(1) Prove that $\Gamma(x+1) = x\Gamma(x)$.

(2) Also show that

$$\int_0^1 u^{a-1}(1-u)^{b-1} du = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}. \tag{2}$$

Solution: For Question (1), we can prove it by Using integration by parts, the steps are as follows:

$$\begin{aligned}
\Gamma(x+1) &= \int_0^\infty u^x e^{-u}\, du \\
&= \left[-u^x e^{-u}\right]_0^\infty + \int_0^\infty x u^{x-1} e^{-u}\, du \\
&= \lim_{u\to\infty} (-u^x e^{-u}) - (0e^{-0}) + x \int_0^\infty u^{x-1} e^{-u}\, du \\
&= x \int_0^\infty u^{x-1} e^{-u}\, du \\
&= x\Gamma(x)
\end{aligned} \tag{3}$$

As we know, when $u \to \infty$, $-u^x e^{-u} \to 0$, so the equation is proved.

Solution: For Question (2), we know that the left of the equation is a Beta function. From the definitions, we can express the equation which we want to prove as :

$$\Gamma(a+b)B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \tag{4}$$

It's a double integral, the expansion formula is as follows:

$$\begin{aligned}
\Gamma(a+b)B(a,b) &= \int_0^\infty u^{a+b-1} e^{-u} du \int_0^1 v^{a-1}(1-v)^{b-1} dv \\
&= \int_0^\infty \int_0^1 (uv)^{a-1}[u(1-v)]^{b-1} u e^{-u} du\, dv
\end{aligned} \tag{5}$$

Then we do a transformation $w = uv$, $z = u(1-v)$. The inverse transformation is $u = w+z$, $v = w/(w+z)$, the corresponding ranges of them are $w \in (0, \infty)$ and $u \in (0, \infty)$.
The absolute value of the Jacobian is

$$\left|\nabla \frac{\partial(u, v)}{\partial(w, z)}\right| = \frac{1}{(w+z)} \tag{6}$$

Next, we use the changed of variables to do a double integral, the equation above becomes:

$$\begin{aligned}
&\int_0^\infty \int_0^\infty w^{a-1} z^{b-1} (w+z) e^{-(w+z)} \frac{1}{w+z} dw\, dz \\
&= \int_0^\infty \int_0^\infty w^{a-1} z^{b-1} e^{-(w+z)} dw\, dz \\
&= \int_0^\infty w^{a-1} e^{-w} dw \int_0^\infty z^{b-1} e^{-z} dz \\
&= \Gamma(a)\Gamma(b)
\end{aligned} \tag{7}$$

Finally the equation is proved.

1.2 Optimization

Use the Lagrange multiplier method to solve the following problem:

$$\begin{aligned}
\min_{x_1, x_2} \quad & x_1^2 + x_2^2 - 1 \\
s.t. \quad & x_1 + x_2 - 1 = 0 \\
& 2x_1 - x_2 \geq 0
\end{aligned} \tag{8}$$

Solution: Consider the above equation is consist of inequality constraint functions and it is a nonlinear optimization problem, we can use the lagrange multiplier method with KKT condition to solve it. We construct the Lagrangian function for the problem:

$$\mathcal{L}(x, \lambda, \mu) = x_1^2 + x_2^2 - 1 + \lambda \cdot \Big(x_1 + x_2 - 1\Big) + \mu \cdot \Big(2x_1 - x_2\Big) \tag{9}$$

The certain conditions which are called KKT condition should satisfy,

$$\begin{aligned}
\frac{\partial(\mathcal{L})}{\partial(X)}|_X &= 0 \\
\lambda_j &\neq 0 \\
\mu_k &\geq 0 \\
\mu_k \cdot \Big(x_1^* + x_2^* - 1\Big) &= 0 \\
x_1^* + x_2^* - 1 &= 0 \\
2x_1^* - x_2^* &\leq 0
\end{aligned} \tag{10}$$

We set up the equations:

$$\begin{aligned}
\frac{\partial(\mathcal{L}, x, \lambda, \mu)}{\partial(x_1)} &= 2x_1 + \lambda + 2\mu = 0 \\
\frac{\partial(\mathcal{L}, x, \lambda, \mu)}{\partial(x_2)} &= 2x_2 + \lambda - \mu = 0 \\
\frac{\partial(\mathcal{L}, x, \lambda, \mu)}{\partial(\lambda)} &= x_1 + x_2 - 1 = 0 \\
\frac{\partial(\mathcal{L}, x, \lambda, \mu)}{\partial(\mu)} &= 2x_1 - x_2 = 0
\end{aligned} \tag{11}$$

We solve them:

$$\begin{aligned} x_1 &= \frac{1}{3} \\ x_2 &= \frac{2}{3} \\ \lambda &= \frac{2}{9} \\ \mu &= -\frac{10}{9} \end{aligned} \tag{12}$$

Choose one problem from the following 1.3 and 1.4. A bonus would be given if you finished the both.

1.3 Stochastic Process

We toss a fair coin for a number of times and use H(head) and T(tail) to denote the two sides of the coin. Please compute the expected number of tosses we need to observe a first time occurrence of the following consecutive pattern

$$H, \underbrace{T, T, ..., T}_{k}. \tag{13}$$

Solution: we asume that E is the expection of the consecutive pattern $H, \underbrace{T, T, ..., T}_{k}$, and E_T^k is the expection of $\underbrace{T, T, ..., T}_{k}$. Consider an equivalent form of this pattern $H, \underbrace{T, T, ..., T}_{k-1}, T$, we have

$$\begin{cases} E = 1 + \frac{1}{2}E + \frac{1}{2}E_T^k, \\ E_T^k = E_T^{k-1} + 1 + \frac{1}{2}E_T^k + \frac{1}{2} \times 0. \quad E_T^1 = 2 \end{cases} \tag{14}$$

which $E = 1 + \frac{1}{2}E + \frac{1}{2}E_T^k$ shows the expection of the first toss. At the first time, you may get H or T with the $\frac{1}{2}$ probability. If you got H, OK, you succuced and then you will try to get k times T, the expection will be $\frac{1}{2}E_T^k$; If you got T, you fail and will restart to tosses and the expection will be $\frac{1}{2}E$.
which $E_T^k = E_T^{k-1} + 1 + \frac{1}{2}E_T^k + \frac{1}{2} \times 0$ shows the the expection of the $k-1$ times of T (E_T^{k-1}) and the last toss. At the last toss, as for the first time, you will get H or T with the $\frac{1}{2}$ probability. If you got H, you fail and you need to get k times T over again and the expection will be $\frac{1}{2}E_T^k$. If you got T, OK, you win the game, the expection will be $\frac{1}{2}E_T^k$;
Next, we solve the recursive function above

$$E_T^k = 2^{k+1} - 2 \tag{15}$$

$\Rightarrow$

$$E = 1 + \frac{1}{2}E + \frac{1}{2}(2^{k+1} - 2) \tag{16}$$

$\Rightarrow$

$$E = 2^{k+1} \tag{17}$$

So the expected number of tosses is 2^{k+1}.

1.4 Probability

Suppose $p \sim Beta(p|\alpha, \beta)$ and $x|p \sim Bernoulli(x|p)$. Show that $p|x \sim Beta(p|\alpha + x, \beta + 1 - x)$, which implies that the Beta distribution can serve as a conjugate prior to the Bernoulli distribution.

Solution: Consider calculating the posterior $p|x$, and we know the likelihood function $x|p$ and the prior p, here we use Bayes' theorem:

$$\begin{aligned} P(p|x) &= \frac{P(x|p)P(p)}{P(x)} \\ &= \frac{P(x|p)P(p)}{\int P(x|p')P(p')dp'} \end{aligned} \tag{18}$$

From the definition, $P(p) \sim Beta(p|\alpha, \beta)$ and $P(x|p) \sim Bernoulli(x|p)$, and the Beta function is

$$Beta(p|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{19}$$

so $P(p|x)$ should be

$$\begin{aligned} P(p|x) &= \frac{P(x|p)P(p)}{\int_0^1 P(x|p')P(p')dp'} \\ &= \frac{\binom{m}{n}p^m(1-p)^{n-m}\frac{1}{B(\alpha,\beta)}p^{\alpha-1}(1-p)^{\beta-1}}{\int_0^1 \binom{m}{n}p^m(1-p)^{n-m}\frac{1}{B(\alpha,\beta)}p^{\alpha-1}(1-p)^{\beta-1}dp} \\ &= \frac{p^{\alpha+m-1}(1-p)^{\beta-1+n-m}}{\int_0^1 p^{\alpha+m-1}(1-p)^{\beta-1+n-m}dp} \\ &= \frac{p^{\alpha+m-1}(1-p)^{\beta-1+n-m}}{B(\alpha+m, \beta+n-m)} \\ &= Beta(p|\alpha+m, \beta+n-m) \end{aligned} \tag{20}$$

So, it implies that the Beta distribution can serve as a conjugate prior to the Bernoulli distribution.

2 SVM

2.1 From Primal to Dual

Consider the binary classification problem with training data $\{(x_i, y_i)\}_{i=1}^N (x_i \in \mathbb{R}^d, y_i \in \{0, 1\})$. Derive the dual problem of the following primal problem of linear SVM:

$$\begin{aligned} &\min_{w,b,\xi} \frac{\lambda}{2}\|w\|^2 + \sum_{i=1}^N \xi_i \\ s.t. \qquad &y_i(w^\top x_i + b) \geq 1 - \xi_i = 0 \qquad \forall i = 1, ..., N \\ &\xi_i \geq 0 \qquad \forall i = 1, ..., N \end{aligned} \tag{21}$$

(Hint: Please note that we explicitly include the offset b here, which is a little different from the simplified expressions in the slides.)

Solution: The Lagrangian functional of the the primal problem of linear SVM above is:

$$\mathcal{L}(w, b, \xi, \alpha, \mu) = \frac{\lambda}{2}\|w\|^2 + \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i[y_i(w^\top x_i + b) - 1 + \xi_i] - \sum_{i=1}^N \mu_i \xi_i \tag{22}$$

and the KKT conditions are:

$$
\begin{aligned}
&0 \in \partial\mathcal{L}(w, b, \xi, \alpha, \mu) \\
&\alpha_i[y_i(w^\top x_i + b) - 1 + \xi_i] = 0 \qquad \forall i \\
&y_i(w^\top x_i + b) - 1 + \xi_i \geq 0 \qquad \forall i \\
&\mu_i \xi_i = 0 \qquad \forall i \\
&\mu_i \geq 0 \qquad \forall i \\
&\alpha_i \geq 0 \qquad \forall i
\end{aligned} \tag{23}
$$

The Lagrange problem:

$$
(\hat{w}, \hat{b}, \hat{\xi}, \hat{\alpha}, \hat{\mu}) = arg \min_{w,b,\xi} \max_{\alpha,\mu} \mathcal{L}(w, b, \xi, \alpha, \mu) \tag{24}
$$

Solve the Lagrange problem:

$$
\begin{aligned}
\frac{\partial\mathcal{L}}{\partial(w)} \mid_{\hat{w}} &= \lambda\hat{w} - \sum_i \alpha_i y_i x_i = 0 \\
\hat{w} &= \frac{1}{\lambda}\sum_i \alpha_i y_i x_i \\
\frac{\partial\mathcal{L}}{\partial(b)} \mid_{\hat{b}} &= \sum_i \alpha_i y_i = 0 \\
\frac{\partial\mathcal{L}}{\partial(\xi)} \mid_{\hat{\xi}} &= 1 - \mu - \alpha = 0 \\
\mu &= 1 - \alpha \\
\alpha_i &\geq 0 \\
\mu &\geq 0
\end{aligned} \tag{25}
$$

then the dual problem:

$$
\mathcal{L}(\hat{w}, b, \xi, \alpha) = \frac{\lambda}{2}\|\frac{1}{\lambda}\sum_i \alpha_i y_i x_i\|^2 + \sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\frac{1}{\lambda}\sum_i \alpha_i y_i x_i)^\top x_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\mu_i\xi_i \tag{26}
$$

and the KKT conditions of the dual problem are:

$$
\begin{aligned}
&0 \in \partial\mathcal{L}(\hat{w}, b, \xi, \alpha) \\
&\alpha_i[y_i(w^\top x_i + b) - 1 + \xi_i] = 0 \qquad \forall i \\
&y_i(w^\top x_i + b) - 1 + \xi_i \geq 0 \qquad \forall i \\
&\alpha_i \geq 0 \qquad \forall i
\end{aligned} \tag{27}
$$

Solve the dual problem:

$$
\begin{aligned}
\mathcal{L}(\hat{w}, b, \xi, \alpha) &= \frac{\lambda}{2}\|\frac{1}{\lambda}\sum_i \alpha_i y_i x_i\|^2 + \sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\frac{1}{\lambda}\sum_i \alpha_i y_i x_i)^\top x_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\mu_i\xi_i \\
&= -\frac{1}{\lambda}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i x_j - b\sum_i \alpha_i y_i + \sum_i \alpha_i(1 - \xi_i) + \sum_i (1 - \mu_i)\xi_i, \\
&= \sum_i (\alpha_i - \alpha_i\xi_i + \xi_i - \mu_i\xi_i) - \frac{1}{\lambda}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i x_j \\
&= \sum_i (\alpha_i - \alpha_i\xi_i + \xi_i - (1 - \alpha_i)\xi_i) - \frac{1}{\lambda}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i x_j \\
&= \boldsymbol{\alpha}^\top - \frac{1}{\lambda}\boldsymbol{\alpha}^\top Y G Y \boldsymbol{\alpha}
\end{aligned} \tag{28}
$$

2.2 Finding Support Vectors (Optional)

As you get the dual problem using KKT conditions. Now please argue from KKT conditions why the following hold:

$$\begin{aligned} \alpha_i = 0 &\Rightarrow y_i(w^\top x_i + b) \geq 1 \\ 0 < \alpha_i < C &\Rightarrow y_i(w^\top x_i + b) = 1 \\ \alpha_i = C &\Rightarrow y_i(w^\top x_i + b) \leq 1 \end{aligned} \tag{29}$$

Solution: The equation in section 2.1 does not have a C parameter, but this question is trying to discuss conditions based on C, so I add C into the equation in section 2.1, where $C = \frac{1}{\lambda}$

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \\ s.t. \qquad & y_i(w^\top x_i + b) \geq 1 - \xi_i = 0 \qquad \forall i = 1, ..., N \\ & \xi_i \geq 0 \qquad \forall i = 1, ..., N \end{aligned} \tag{30}$$

The KKT conditions for the above equation,

$$\begin{aligned} &0 \in \partial \mathcal{L}(w, b, \xi, \alpha, \mu) \\ &\alpha_i[y_i(w^\top x_i + b) - 1 + \xi_i] = 0 \qquad \forall i \\ &y_i(w^\top x_i + b) - 1 + \xi_i \geq 0 \qquad \forall i \\ &\mu_i \xi_i = 0 \qquad \forall i \\ &\xi_i \geq 0 \qquad \forall i \\ &\alpha_i \geq 0 \qquad \forall i \end{aligned} \tag{31}$$

So $\forall i$, we always have $\alpha_i = 0$ or $y_i(w^\top x_i + b) = 1 - \xi_i$,
When $\alpha_i = 0$, the samples will have no influence on $y_i(w^\top x_i + b)$
When $\alpha_i > 0$, $y_i(w^\top x_i + b) = 1 - \xi_i$ is always right, the samples should be the support vector.

Now, we can solve the above problem to get the detail range of α

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial(\xi)} \mid_{\hat{\xi}} = C - \mu_i - \alpha_i = 0 \\ \mu_i = C - \alpha_i \\ \alpha_i \geq 0 \\ \mu_i \geq 0 \end{aligned} \tag{32}$$

So, the range of α is

$$0 \leq \alpha_i \leq C \tag{33}$$

Now we discuss the different condition by different value of α
When $\alpha_i = 0$,

$$\begin{aligned} y_i(w^\top x_i + b) \geq 1 - \xi_i \\ \mu_i &= C - \alpha_i \\ &= C \\ &\Rightarrow \xi_i = 0 \qquad (\mu_i \xi_i = 0) \\ &\Rightarrow y_i(w^\top x_i + b) \geq 1 \end{aligned} \tag{34}$$

When $0 < \alpha_i < C$, the sample is just right on the margin.

$$
\begin{aligned}
y_i(w^\top x_i + b) = 1 - \xi_i \\
\mu_i &= C - \alpha_i = 0 \\
\Rightarrow & \xi_i \geq 0 \qquad (\mu_i \xi_i = 0) \\
\Rightarrow & y_i(w^\top x_i + b) \leq 1
\end{aligned}
\tag{35}
$$

When $\alpha_i = C$, the sample is in the gap.

$$
\begin{aligned}
y_i(w^\top x_i + b) = 1 - \xi_i \\
\mu_i &= C - \alpha_i < C \\
\Rightarrow & \xi_i = 0 \qquad (\mu_i \xi_i = 0) \\
\Rightarrow & y_i(w^\top x_i + b) = 1
\end{aligned}
\tag{36}
$$

3 IRLS for Logistic Regression

For a binary classification problem $\{(x_i, y_i)\}_{i=1}^N (x_i \in \mathbb{R}^d, y_i \in \{0,1\})$, the probabilistic decision rule according to "logistic regression" is

$$
P_w(y|x) = \frac{exp(y\boldsymbol{w}^\top \boldsymbol{x})}{1 + exp(\boldsymbol{w}^\top \boldsymbol{x})}
\tag{37}
$$

And hence the log-likelihood is

$$
\begin{aligned}
\mathcal{L}(w) &= \log \prod_{i=1}^N P_w(y|x) \\
&= \sum_{i=1}^N (y_i \boldsymbol{w}^\top \boldsymbol{x} - log(1 + exp(\boldsymbol{w}^\top \boldsymbol{x_i})))
\end{aligned}
\tag{38}
$$

Please implement the IRLS algorithm to estimate the parameters of logistic regression

$$
\max_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w})
\tag{39}
$$

and the L2-norm regularized logistic regression

$$
\max_{\boldsymbol{w}} -\frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 + \mathcal{L}(\boldsymbol{w})
\tag{40}
$$

where λ is the positive regularization constant.

You may refer to the lecture slides for derivation details but you are more encouraged to derive the iterative update equations yourself.

Please compare the results of the two models on the "UCI a9a" dataset1. The suggested performance metrics to investigate are e.g. prediction accuracies (both on training and test data), number of IRLS iterations, L2-norm of $\|\boldsymbol{w}_2$, etc. You may need to test a range of λ values with e.g. cross validation for the regularized logistic regression.

Hint: You can use the convergence curves as shown in the lecture slides to show the convergence properties of these two methods.

3.1 Derivation

Solution: For the L2-norm regularized logistic regression

$$\begin{aligned}\mathcal{L}_{L2}(w) &= -\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \mathcal{L}(\boldsymbol{w}) \\ &= \sum_{i=1}^{N}(y_i\boldsymbol{w}^\top\boldsymbol{x} - log(1 + exp(\boldsymbol{w}^\top\boldsymbol{x_i}))) - \frac{\lambda}{2}\|\boldsymbol{w}\|^2\end{aligned} \tag{41}$$

We need to solve the w^* such that

$$\begin{aligned}\nabla\mathcal{L}_{L2}(w^*) &= 0 \\ \nabla\mathcal{L}_{L2}(w_t) &= \sum_i (y_i - \mu_i)x_i - \lambda w_t \quad = X(y-\mu) - \lambda w_t \\ \mu_i &= \psi(w_t^\top x_i)\end{aligned} \tag{42}$$

The Hessian matrix is:

$$\begin{aligned}H_{L2} &= \nabla^2\mathcal{L}_{L2}(w^*)\|w_t \\ &= -\sum_i (\mu_i(1-\mu_i))x_i x_i^\top - \lambda I \\ &= -XRX^\top - \lambda I\end{aligned} \tag{43}$$

where $\quad R_{ii} = \mu_i(1-\mu_i)$

Now, we can solve w_{t+1} for the L2-norm regularized logistic regression

$$\begin{aligned}w_{t+1} &= w_t - H^{-1}\nabla_w\mathcal{L}_{L2}(w^t) \\ &= w_t - (-XRX^\top - \lambda I)^{-1}(X(y-\mu) - \lambda w_t) \\ &= w_t + (XRX^\top + \lambda I)^{-1}(X(y-\mu) - \lambda w_t) \\ &= (XRX^\top + \lambda I)^{-1}\{(XRX^\top + \lambda I)w_t + (X(y-\mu) - \lambda w_t)\} \\ &= (XRX^\top + \lambda I)^{-1}\{XRX^\top w_t + \lambda I w_t + X(y-\mu) - \lambda w_t\} \\ &= (XRX^\top + \lambda I)^{-1}\{XRX^\top w_t + X(y-\mu)\} \\ &= (XRX^\top + \lambda I)^{-1}XRz\end{aligned} \tag{44}$$

where $z = X^\top w_t + R^{-1}(y-\mu)$

3.2 Implemention

I implemented the algorithm using Python. IRLS can be run just by **import IRLS** and less than 5 lines code, and the Loss and accuracy curve on training and testing data can be ploted with only one line.

examples of running **IRLS.py**

```
import IRLS

x_train, y_train  = IRLS.loadDotData("a9a/a9a")
x_test,  y_test   = IRLS.loadDotData("a9a/a9a.t")
```

```
# Run IRLS
trainHistory = IRLS.train(x_train, y_train, x_test, y_test)
IRLS.test(x_test, y_test, regularizer="")
IRLS.plotLossAcc(trainHistory,"IRLS on a9a data")

# Run IRLS with L2 norm regularized
trainHistoryL2 = IRLS.train(x_train, y_train, x_test, y_test, regularizer="L2", w_lambda= 0.3)
IRLS.test(x_test, y_test, regularizer="L2")
IRLS.plotLossAcc(trainHistory,"IRLS-L2 on a9a data")
```

The output log:

```
===== IRLS  ====
Max Iteration:  50
Early Stopping: 10

[ 0/50] best model from 99999.000000 to 0.669441
 | trainLoss 0.6931   trainAcc 0.7592   valLoss 0.6694   valAcc 0.8455
[ 1/50] best model from 0.669441 to 0.646152
 | trainLoss 0.6696   trainAcc 0.8452   valLoss 0.6462   valAcc 0.8482
[ 2/50] best model from 0.646152 to 0.623514
 | trainLoss 0.6464   trainAcc 0.8482   valLoss 0.6235   valAcc 0.8493
[ 3/50] best model from 0.623514 to 0.608796
...
[ 14/50] best model from 0.602329 to 0.602253
 | trainLoss 0.6028   trainAcc 0.8491   valLoss 0.6023   valAcc 0.8499
[ 15/50] best model from 0.602253 to 0.602187
 | trainLoss 0.6027   trainAcc 0.8491   valLoss 0.6022   valAcc 0.8499
...
[ 25/50] best model from 0.601798 to 0.601796
 | trainLoss 0.6023   trainAcc 0.8491   valLoss 0.6018   valAcc 0.8499
[ 26/50] model is not improved
...
Stoped by earlystoping, best model loss: 0.6018 in Iteration 25

 Testing IRLS  Loading Best model...
 * testACC: 0.8499  - testAUC: 0.9014 - testAP: 0.7450 - testF1: 0.6531 -
 testPrecision: 0.7194 - testRecall: 0.5980 - L2norm: 38.5209
```

Experiment setting:

1. The maximum iteration is set to be 50, which can be set to be any positive value.
2. The training will stop when the the model has not yet improved since sevral iteration, we can call this EarlyStopping.
3. The initial W is set to be 0.
4. A very samll value 1e-09 is added into when calculating the inverse of a matrix to avoid inversing a singularity matrix which can not calculating the inverse.
5. I import a loss function below to evaluate the training.

$$J(w) = \sum_i y_i \log\left(\frac{1}{1+e^{-w^\top x}}\right) + (1-y_i)\log\left(1-\frac{1}{1+e^{-w^\top x}}\right) \tag{45}$$

6. In testing step, metrics (Accuracy / AUC / Average Precision / F1 score / Precision / Recall) are calculated.
7. The best model(minimum loss) is saved into **model/** directory.

I test a set of $\lambda \in (1e-06, 3)$ on the training set("UCI a9a/a9a") and validate the training model on the testing set("UCI a9a/a9a.t"), if I have more slack times, I will do a 10-fold cross validation. From equation44, we know when $\lambda = 0$, the algorithm is IRLS without L2 normalization, when $\lambda > 0$, the algorithm is IRLS with L2 normalization.
In my implementation, A very samll value 1e-09 is added into when calculating the inverse of a matrix to avoid inversing a singularity matrix(maybe it is the data that makes a singularity matrix), so in a broad sense, my implementation of IRLS is always a L2 normalized IRLS. But in a narrow sense, as long as the $\lambda to 0$, the algorithm is IRLS without L2 normalization.

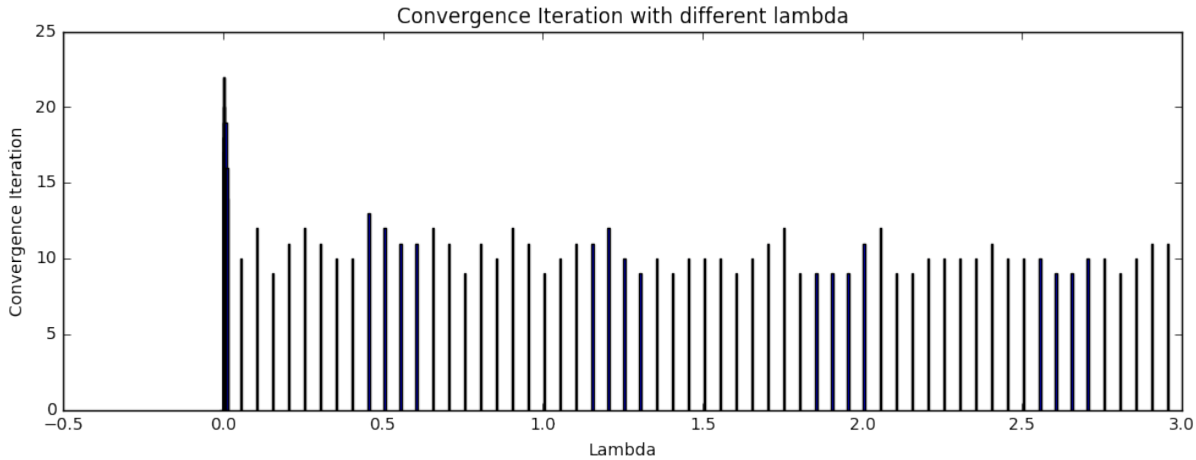

Figure 1: Convergence Iteration with a set of λ on a9a with IRLS(L2).

Figure 1 shows the Convergence Iteration of a set of λ. So the We can see that, when $\lambda \approx 0$, the convergence iteration is more than 20 times, when $\lambda \geq 0.1$, the convergence iteration is about 10 times, it shows that the L2 normalized IRLS has a fast convergence speed than IRLS.

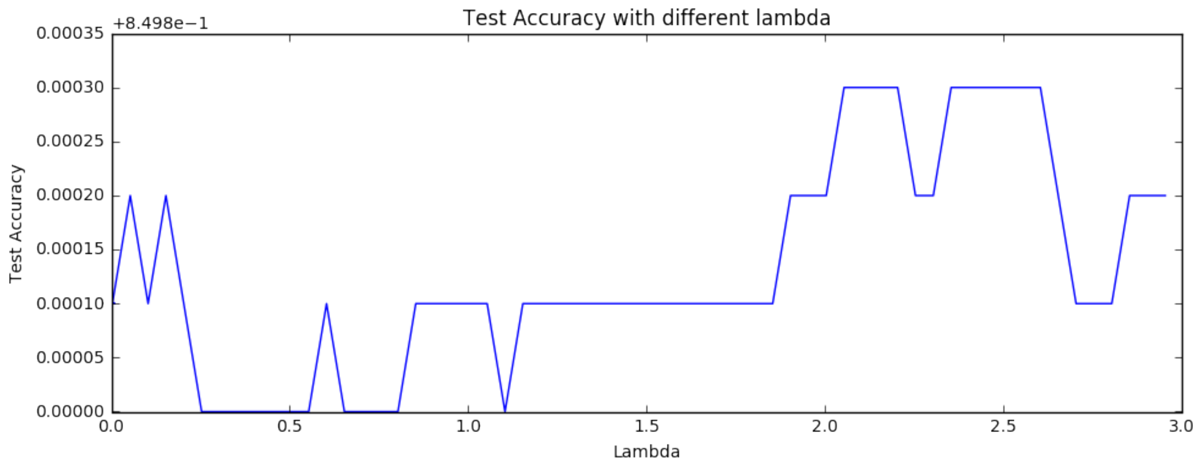

Figure 2: Test Accuracy with a set of λ on a9a with IRLS(L2).

Figure 2 shows the Test Accuracy of a set of λ. So the We can see that, when $\lambda \approx 0$, the convergence iteration is more than 20 times, when $\lambda \geq 0.1$, the convergence iteration is about 10 times, it shows that the L2 normalized IRLS has a fast convergence speed than IRLS.

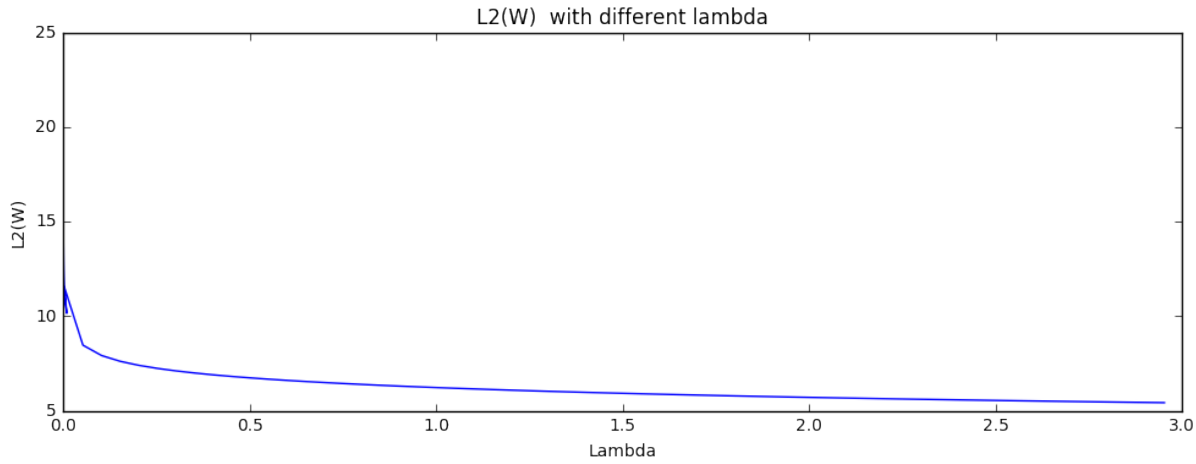

Figure 3: The L2 norm value of W with a set of λ on a9a.

Figure 3 shows the L2 norm value of W with a set of λ. The W is decreased when λ is larger.

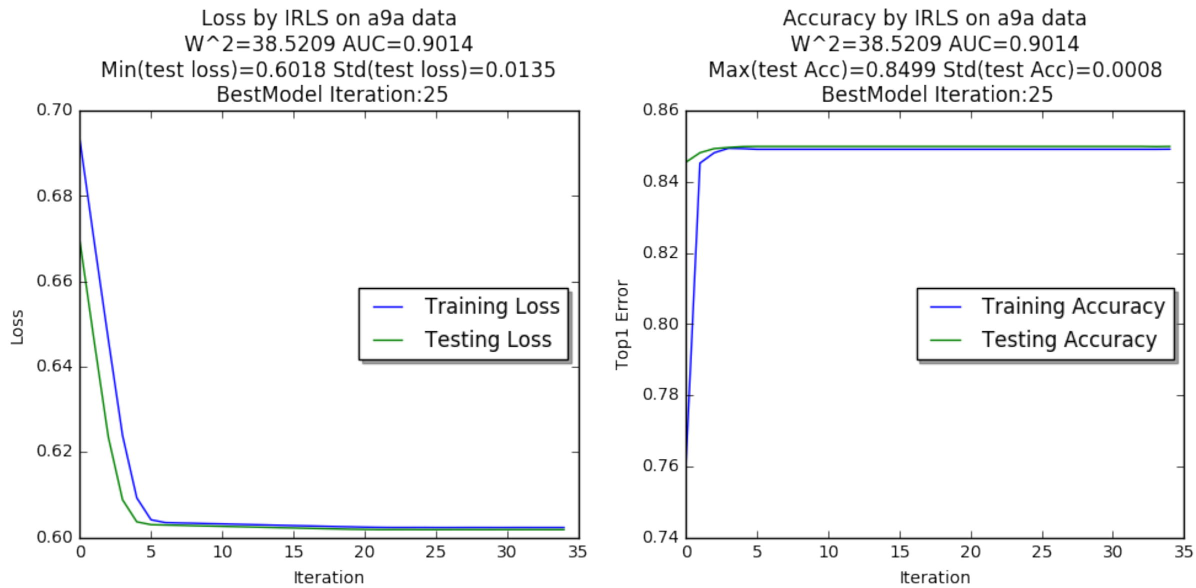

Figure 4: Training performance with IRLS on "UCI a9a" dataset.

Figure 4 shows the loss and the accuracy curve by IRLS with the earlystoping on "UCI a9a" dataset. The best model (validation loss is minimum) is at iteration 25, and the loss ($J(W)$) is 0.6018, $\|W\|_2^2$ is 38.5209, the best test accuracy is 0.8499, the AUC is 0.9014.

Figure 5 shows the loss and the accuracy curve by L2 normalized IRLS with the earlystoping on "UCI a9a" dataset. The best model (validation loss is minimum) is at iteration 9, and the loss ($J(W)$) is 0.6033, $\|W\|_2^2$ is 7.1097, the best test accuracy is 0.8499, the AUC is 0.9020.

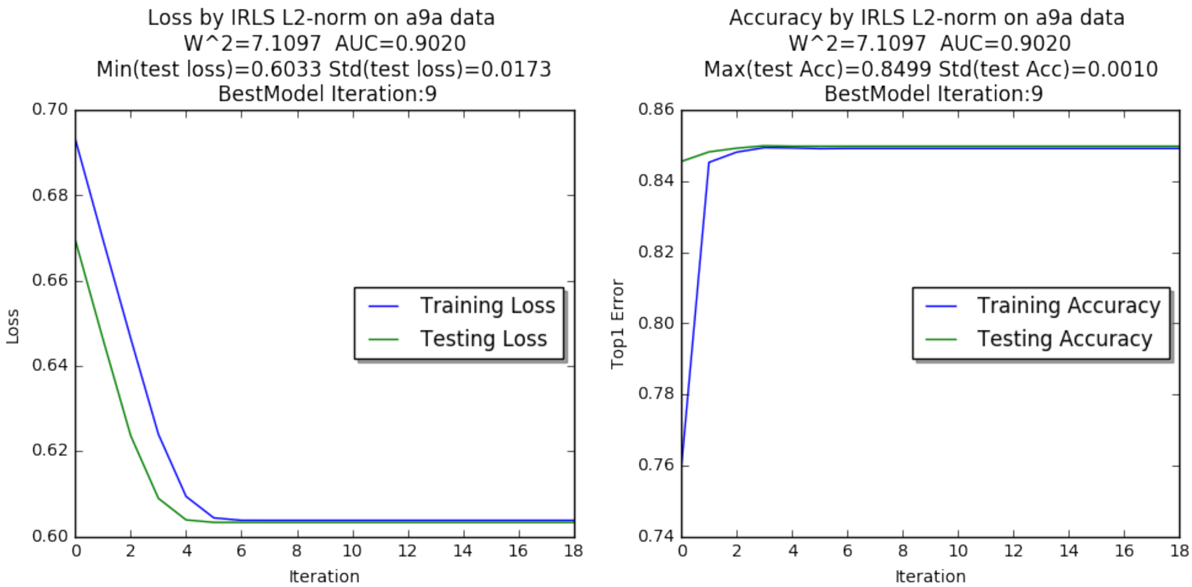

Figure 5: Training performance with L2 normalized IRLS on "UCI a9a" dataset.