机器学习第一次作业

汪隽立 2021012957

2023 年 11 月 2 日

注： 报告中只对作业要求的题目作出解答。代码填空部分除非特殊情况，不在报告中赘述。

解答 2.2.1.

$$J(\theta) = \frac{1}{m}\|y - X\theta\|^2 + \lambda\theta^\top\theta$$

其中 $X \in \mathbb{R}^{m\times(d+1)}$，$y \in \mathbb{R}^m$，$\theta \in \mathbb{R}^{d+1}$。

解答 2.2.3.

$$\frac{\partial J(\theta)}{\partial\theta} = \frac{2}{m}X^\top(X\theta - y) + 2\lambda\theta$$

其中用到了有关矩阵求导的公式：

$$\frac{\partial \mathbf{X}\theta}{\partial\theta} = \mathbf{X}^\top$$
$$\frac{\partial\|x\|^2}{\partial x} = 2x$$

解答 2.2.6.

在我们的处理中，偏置项 $b = \theta_{d+1}$（θ_{d+1} 表示 θ 的第 $i+1$ 个分量）。而

$$\frac{\partial J(\theta)}{\partial\theta_{d+1}} = \frac{2}{m}X^\top\left(X\theta - y\right)_{d+1} + 2\lambda\theta_{d+1}$$

如果将 x 添加的额外维度置为 B，那么 $X^\top$ 的第 $d+1$ 行为 $(B, B, \ldots, B)$，$X\theta - y = (A_1 + B\theta_{d+1}, A_2 + B\theta_{d+1}, \ldots, A_m + B\theta_{d+1})^\top$，其中 $A_i = \sum_{k=1}^d x_{ik}\theta_k$。故在偏导数的计算中，均方误差带来的梯度为

$$\frac{2}{m}X^\top(X\theta - y)_{d+1} = \sum_{i=1}^m B\times A_i + mB^2\theta_{d+1}$$

可以看出这一项的量级至少是 $O(B)$ 的。而正则化带来的梯度为 $2\lambda\theta_{d+1}$，一方面 $\|\theta_{d+1}\|$ 非常小，另一方面 B 非常大。我们有：

$$\|2\lambda\theta_{d+1}\| \ll \|\frac{2}{m}X^\top(X\theta - y)_{d+1}\|$$

也就是说，此时正则化对偏置项梯度的影响可以忽略不计。

解答 2.3.1.

$$J(\theta + \eta h) - J(\theta) \approx \eta \langle \nabla J(\theta), h \rangle$$

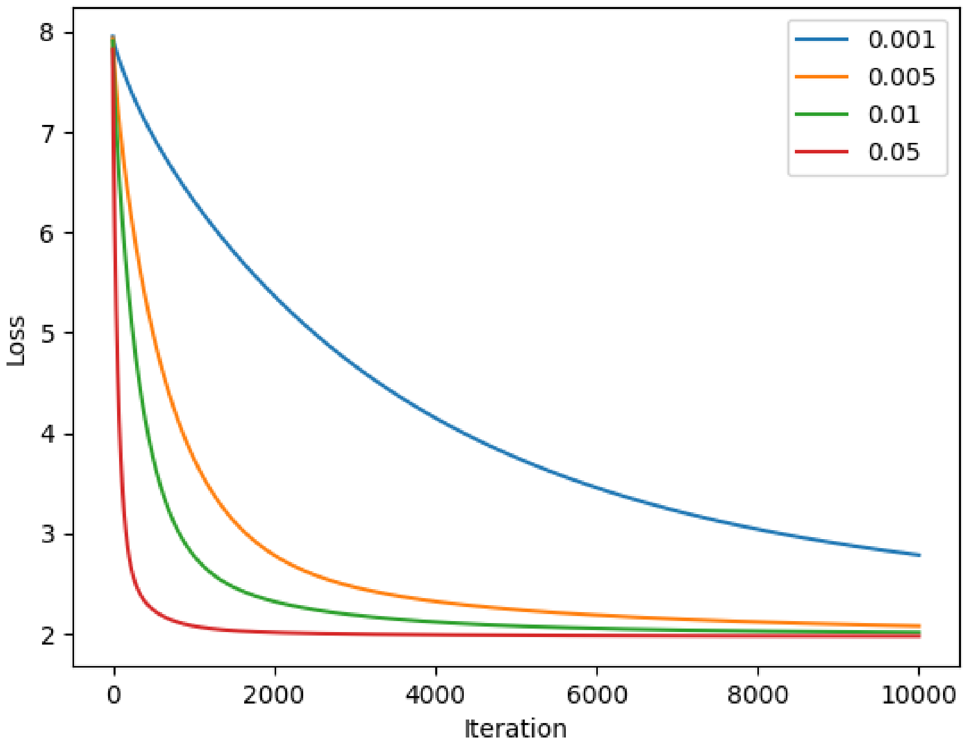
其中内积为 $\mathbb{R}^n$ 上的内积。可以看出当梯度与 h 反向时，函数减少得最多，所以 h 的前进方向应当为梯度的反方向。

解答 2.3.2.

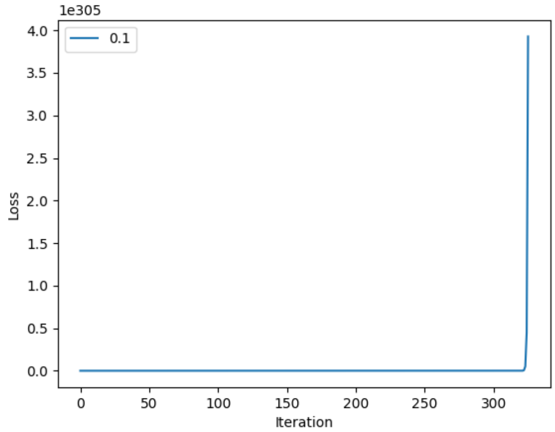
$$\theta' = \theta - \eta \nabla J(\theta)$$

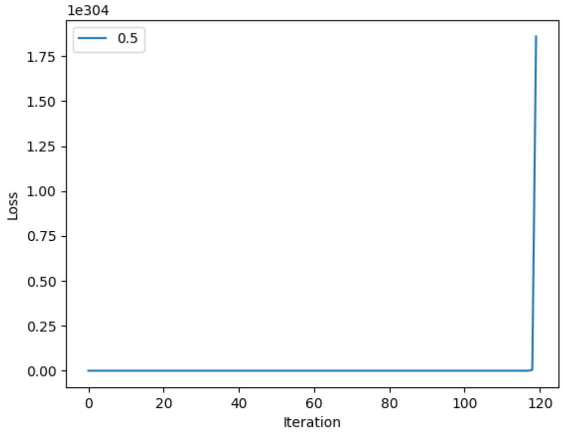
解答 2.3.4.

能使实验结果收敛的学习率如下，分别为 $0.001, 0.005, 0.01, 0.05$”：而当学习率为 $0.1, 0.5$



时，这时 $\lambda \gg O(\frac{1}{L})$，$L$ 为目标函数的 Lipschitz 常数，梯度下降算法不再收敛。





解答 2.4.1.

$$\nabla J_{SGD}(\theta) = \frac{2}{n}\sum_{k=1}^{n} x_{i_k}^\top(\theta^\top x_{i_k} - y_{i_k}) + 2\lambda\theta$$

解答 2.4.2.

$$\begin{aligned}
\mathbb{E}_{i_1,\dots,i_n}\left[\nabla J_{SGD}(\theta)\right] &= \mathbb{E}\left[\frac{2}{n}\sum_{k=1}^{n} x_{i_k}^\top(\theta^\top x_{i_k} - y_{i_k})\right] + 2\lambda\theta \\
\text{（期望的线性性）}\quad &= \frac{2}{n}\sum_{k=1}^{n}\mathbb{E}_{x_{i_k}\sim D}\left[x_{i_k}^\top(\theta^\top x_{i_k} - y_{i_k})\right] + 2\lambda\theta \\
\text{(i.i.d.)}\quad &= 2\mathbb{E}_{x\sim D}\left[x^\top(\theta^\top x - y)\right] + 2\lambda\theta \\
&= \frac{2}{m}\sum_{x_i\in\mathcal{X}}\left[x_i^\top(\theta^\top x_i - y_i)\right] + 2\lambda\theta \\
&= \nabla J(\theta)
\end{aligned}$$

解答 2.4.4.

选取学习率为 0.01，$batchsize = \{1, 10, 20, 50, 100\}$ 进行实验：可以看到，随着 $batchsize$ 的增大，训练误差有更好的稳定性。不过训练效率也有所下降，$batchsize = 100$ 即全批量训练。

解答 2.4.5.

选取学习率为 0.01，$batchsize = 20$，$\lambda = \{10^{-7}, 10^{-5}, 10^{-3}, 10^{-1}, 1, 10, 100\}$ 进行 SGD 实验：可以看出，当 λ 过大时，**模型基本没有学习能力**。而适当的 $\lambda = \{10^{-5}, 10^{-3}\}$ 能够提

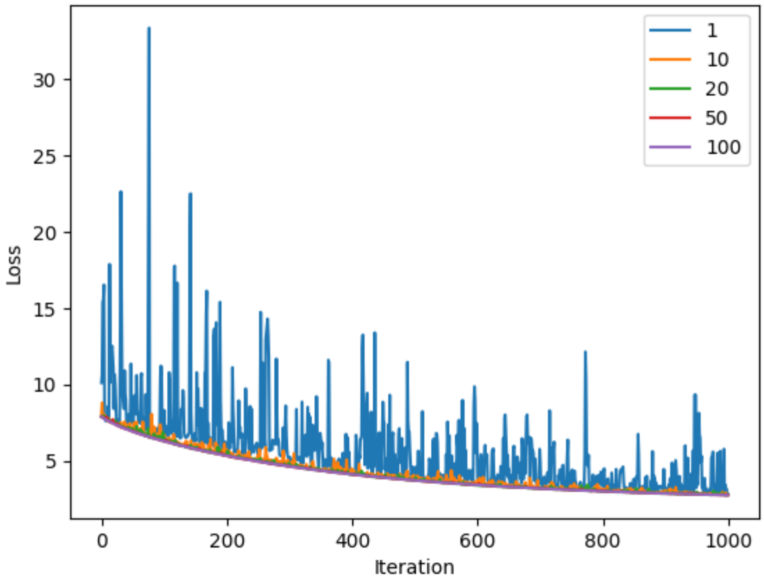


图 1: 不同 batchsize 对 SGD 的影响

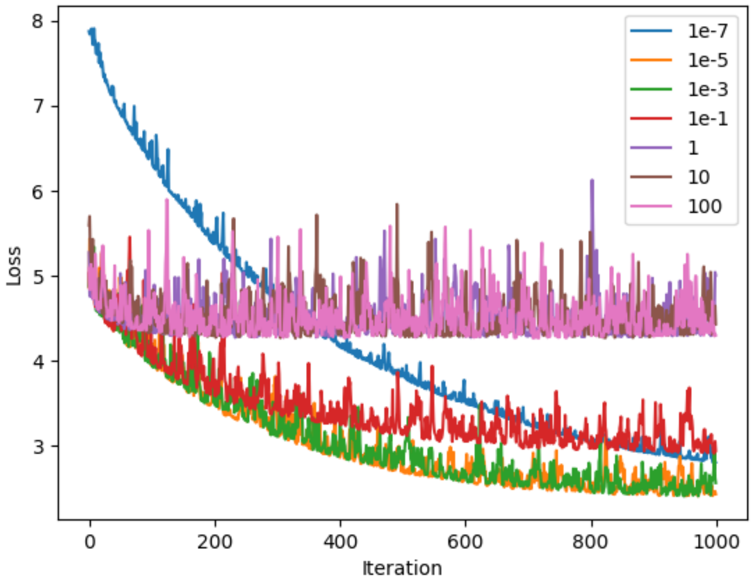


图 2: 不同 λ 对 SGD 的影响

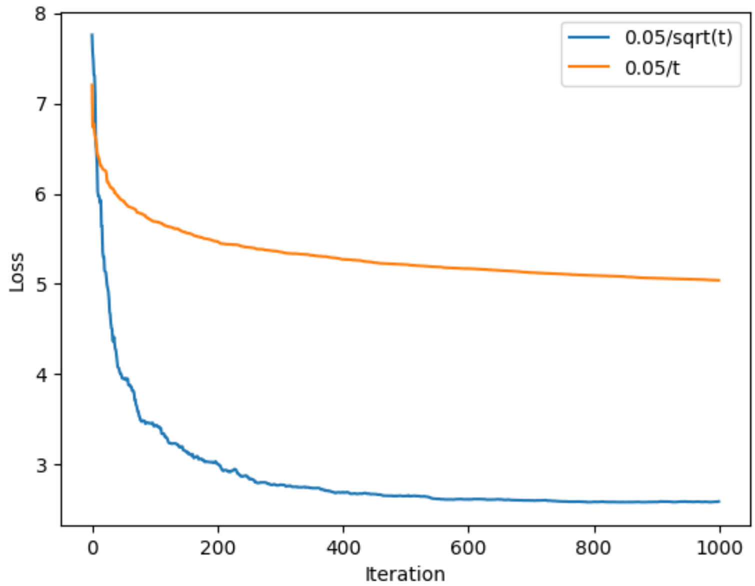
高模型的泛化能力。

解答 2.6.1.

$$H = \frac{2}{m} X^\top X + 2\lambda I$$

其中 $X^\top X$ 半正定，再加上一个正定阵，自然是正定阵，故奇异。

解答 2.6.2.



从结果上看，牛顿迭代法大大减少了收敛所需要的迭代步数。不论是 GD 还是 SGD，其需要的步数都大约为 1000 步，而牛顿迭代法基本上只需要 100 步就可收敛。

然而，牛顿迭代法也牺牲了一定的计算复杂度。在每一步迭代中，牛顿迭代法牺牲了更多的时间计算 Hessian 矩阵的逆矩阵。而 GD 和 SGD 只需要计算梯度，计算复杂度较低。实验表明，在迭代次数为 10000 时，牛顿迭代法所需要的时间为 1.48 秒，而 SGD 需要 0.22 秒（$batchsize = 20$）。总的来说牛顿迭代法在时间上能使收敛速度变快，但其单步的速度实际上是不如 SGD 的。

解答 3.1.1.

$$\partial J(w) = \begin{cases} -yx & \text{if } yw^\top x < 1 \\ 0 & \text{if } yw^\top x > 1 \\ \text{any number between 0 and } -yx & \text{if } yw^\top x = 1 \end{cases}$$

解答 3.1.2.

假设 f 不凸，i.e. 存在 $x_0, y_0 \in dom(f)$，$\lambda \in [0,1]$，使得

$$\lambda f(x_0) + (1-\lambda) f(y_0) < f(\lambda x_0 + (1-\lambda) y_0)$$

因为 f 的次梯度处处存在，所以在 $\lambda x_0 + (1-\lambda) y_0$ 处有

$$f(x_0) \geq f(\lambda x_0 + (1-\lambda) y_0) - (1-\lambda) g^\top (y_0 - x_0) \tag{1}$$

$$f(y_0) \geq f(\lambda x_0 + (1-\lambda) y_0) + \lambda g^\top (y_0 - x_0) \tag{2}$$

$\lambda \times (1) + (1-\lambda) \times (2)$，得

$$\lambda f(x_0) + (1-\lambda) f(y_0) \geq f(\lambda x_0 + (1-\lambda) y_0)$$

这与 f 不凸的假设矛盾。故 f 为凸函数。

解答 3.2.1.

选择合适的次梯度为

$$g = \begin{cases} -y_i x_i & \text{if } y_i w^\top x_i < 0 \\ 0 & \text{Others} \end{cases}$$

这样，当 SSGD 算法随机采样到 (x_i, y_i) 时，w 的更新如下：

$$w^{(k+1)} = \begin{cases} w^{(k)} & \text{if } y_i w^\top x_i > 0 \text{（分类正确）} \\ w^{(k)} + y_i x_i & \text{if } y_i w^\top x_i < 0 \text{（分类错误）} \end{cases}$$

这和感知机算法的更新是一致的。假设 SSGD 迭代的轮次足够大，可以认为 w 和感知机算法得到的结果一样，都能正确地线性区分数据。

解答 3.2.2.

由于 w 的初始值为 $(0, \ldots, 0)$，而每一步更新 w 用到的公式为

$$w^{(k+1)} = \begin{cases} w^{(k)} & \text{if } y_i w^\top x_i > 0 \text{（分类正确）} \\ w^{(k)} + y_i x_i & \text{if } y_i w^\top x_i < 0 \text{（分类错误）} \end{cases}$$

其中 $y_i \in \{-1, +1\}$。故 w 自然是 $\{x_i\}_{i=1}^n$ 的线性组合，α_i 即为更新过程中加上 $y_i x_i$ 的次数（或者可以用数学归纳法/表示定理说明这件事）。

解答 3.3.1.

$$L(w,b,\xi,\alpha,\mu)=\frac{\lambda}{2}\|w\|^2+\frac{1}{m}\sum_{i=1}^{m}\xi_i+\sum_{i=1}^{m}\alpha_i(1-y_i(w^\top x_i+b)-\xi_i)-\sum_{i=1}^{m}\mu_i\xi_i$$
$$\text{where}\quad \alpha_i>0,\mu_i>0,i=1,2,\dots,n$$

解答 3.3.2.

在 3.3.1 的拉格朗日方程中，令

$$\frac{\partial L}{\partial w}=\lambda w-\sum_{i=1}^{m}\alpha_i y_i x_i=0$$
$$\frac{\partial L}{\partial b}=-\sum_{i=1}^{m}\alpha_i y_i=0$$
$$\frac{\partial L}{\partial \xi_i}=\frac{1}{m}-\alpha_i-\mu_i=0$$

得

$$w=\frac{1}{\lambda}\sum_{i=1}^{m}\alpha_i y_i x_i$$
$$\sum_{i=1}^{m}\alpha_i y_i=0$$
$$\alpha_i=\frac{1}{m}-\mu_i$$

带入拉格朗日方程，得

$$\Pi(\alpha,\mu)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2\lambda}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle$$

Dual Problem 即为

$$\begin{aligned}&\max_{\alpha}\Pi(\alpha,\mu)\\ s.t.\quad &0\le\alpha_i\le\frac{1}{m}\\ &\sum_{i=1}^{m}\alpha_i y_i=0,\quad i=1,2,\dots,m\end{aligned}$$

解答 3.3.3.

Primal Problem:

$$\begin{aligned}&\min_{w,b,\xi}\frac{\lambda}{2}\|w\|^2+\frac{1}{m}\sum_{i=1}^{m}\xi_i\\ s.t.\quad &y_i(w^\top\Phi(x_i)+b)\ge 1-\xi_i\\ &\xi_i\ge 0,\quad i=1,2,\dots,m\end{aligned}$$

Dual Problem:

$$\begin{aligned}\max_{\alpha}\ & \sum_{i=1}^{m}\alpha_i-\frac{1}{2\lambda}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j k(x_i,x_j)\\ s.t.\quad & 0\le\alpha_i\le\frac{1}{m}\\ & \sum_{i=1}^{m}\alpha_i y_i=0,\quad i=1,2,\dots,m\end{aligned}$$

解答 3.3.4.

记 $f(w)=\frac{\lambda}{2}\|w\|^2$。由于 f 关于 w 是凸函数，故 f 的梯度即为其次梯度。又由前我们已经计算出 Hinge Loss 的次梯度为：

$$\partial(\max\{0,1-y_i(w^\top x_i+b)\})_w=\begin{cases}-y_i x_i & \text{if } y_i w^\top x_i<1\\ 0 & \text{if } y_i w^\top x_i>1\end{cases}$$

$$\partial(\max\{0,1-y_i(w^\top x_i+b)\})_b=\begin{cases}-y_i & \text{if } y_i w^\top x_i<1\\ 0 & \text{if } y_i w^\top x_i>1\end{cases}$$

而次梯度的和仍然是次梯度，$\partial J_{i|w}=\lambda w+\partial(\max\{0,1-y_i(w^\top x_i+b)\})_w$，$\partial J_{i|b}=\partial(\max\{0,1-y_i(w^\top x_i+b)\})_b$。此即要证明的等式。

解答 3.3.5.

见 Algorithm 1。

Algorithm 1 SSGD for SVM

输入：训练集 $(x_1,y_1),\dots,(x_n,y_n)\in\mathbb{R}^d\times\{-1,1\}$

Initialize w, b, BS = Batch Size, η = Learning Rate, T = Iterations

for $t=1,2,\dots,T$ **do**

  Randomly sample a minibatch $\{(x_i,y_i)\}_{i=1}^{BS}$

  Calculate subgradient on minibatch:

  $\partial J_w=\frac{1}{BS}\sum_{i=1}^{BS}\partial J_{i|w}$, $\partial J_b=\frac{1}{BS}\sum_{i=1}^{BS}\partial J_{i|b}$

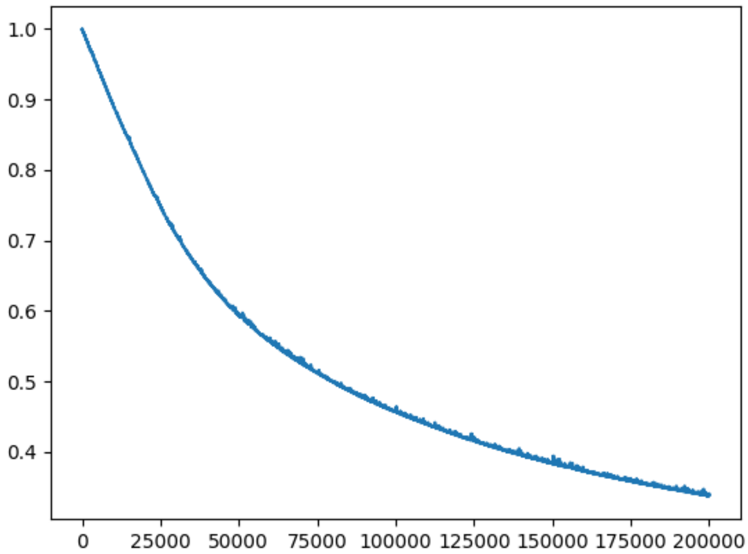
  $w=w-\eta\partial J_w$, $b=b-\eta\partial J_b$

end for

输出：w, b

解答 3.3.6.

图 3: SGD 训练损失随迭代次数的变化



只需证明最优解中，必有 $\xi_i \geq 0$。假设最优解的某个 $\xi_i < 0$。现在令 $\xi_i' = 0$，有 $y_i(w^\top x_i + b) \geq 1 - \xi_i \geq 1 - \xi_i'$。也就是说将 ξ_i 替换成 $\xi_i' = 0$，仍然能满足约束条件，但目标函数的值减少了。这与最优解的假设矛盾。

解答 3.3.7.

Lagrange Function:

$$L(w, b, \xi, \alpha) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i^2 + \sum_{i=1}^{m}\alpha_i(1 - y_i(w^\top x_i + b) - \xi_i)$$

$$\text{where} \quad \alpha_i > 0, i = 1, 2, \ldots, n$$

Dual Problem:

$$\Pi(\alpha) = \sum_{i=1}^{m}(\alpha_i - \frac{m}{4}\alpha_i^2) - \frac{1}{2\lambda}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j \langle x_i, x_j \rangle$$

解答 3.4.2.

首先列出训练损失随着迭代次数的变化：以下列出不同学习率对验证集准确率的影响：于是选择学习率为 0.1。接下来列出不同 λ 对验证集准确率的影响：故选择 $\lambda = 0.0005$，作为超参数。

解答 3.4.3.

表 1: 不同学习率对验证集准确率

学习率	**验证集上准确率**
0.001	0.66
0.005	0.83
0.01	0.85
0.05	0.85
0.1	0.86
0.5	0.83

表 2: 不同 λ 对验证集准确率

λ	**验证集上准确率**
0.0001	0.85
0.0005	0.86
0.001	0.85
0.005	0.68
0.1	0.67
0.5	0.51

可以看出几乎不需要 10000 次迭代，模型就会收敛。在 UML 一书上，作者给出了以下结论：

$$\sum_{t=1}^{T}(\mathbb{E}[f(\mathbf{w}^{(t)})]-f(\mathbf{w}^{\star}))\leq\frac{\rho^2}{2\lambda}\sum_{t=1}^{T}\frac{1}{t}\leq\frac{\rho^2}{2\lambda}(1+\log(T))$$

故如此计算的收敛速度是 $O(\frac{\log T}{T})$ 的。这远小于原先的 $O(\frac{1}{\sqrt{T}})$。

解答 3.4.4.

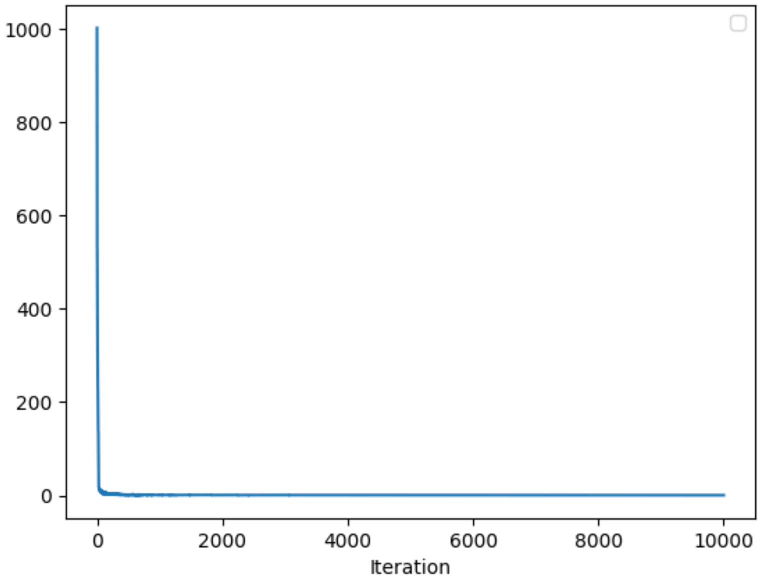
在实现中，曾尝试了**高斯核、双曲正切核以及线性核**。不知是因为我实现有误的原因，还是超参数调整不当，前两者的效果都非常差，故最后选择了线性核。即：

$$k(x,y)=x^\top y$$

学习率设置为 0.01，正则化系数 $\lambda=0$，训练 100000 轮，最终在验证集上的准确率约为 0.7。

解答 3.4.5.

图 4: λ-强突问题 SGD 训练损失随迭代次数的变化



选取不带核函数的线性 SVM 作为最终的结果。学习率为 0.01，正则化系数为 0.005。验证集上准确率：0.86，F1-Score: 0.83，混淆矩阵为：

$$\begin{bmatrix} 607 & 69 \\ 158 & 549 \end{bmatrix}$$