Homework 3: Linear Regression and Logistic Regression

Student Name	Student ID
米家龙	18342075

Exercise One: Linear Regression

(a) 梯度下降法训练线性回归模型

> 迭代次数 1500000
> 学习率 0.00015
> 其他参数 0.0

线性回归模型为：

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_3 x_2$$

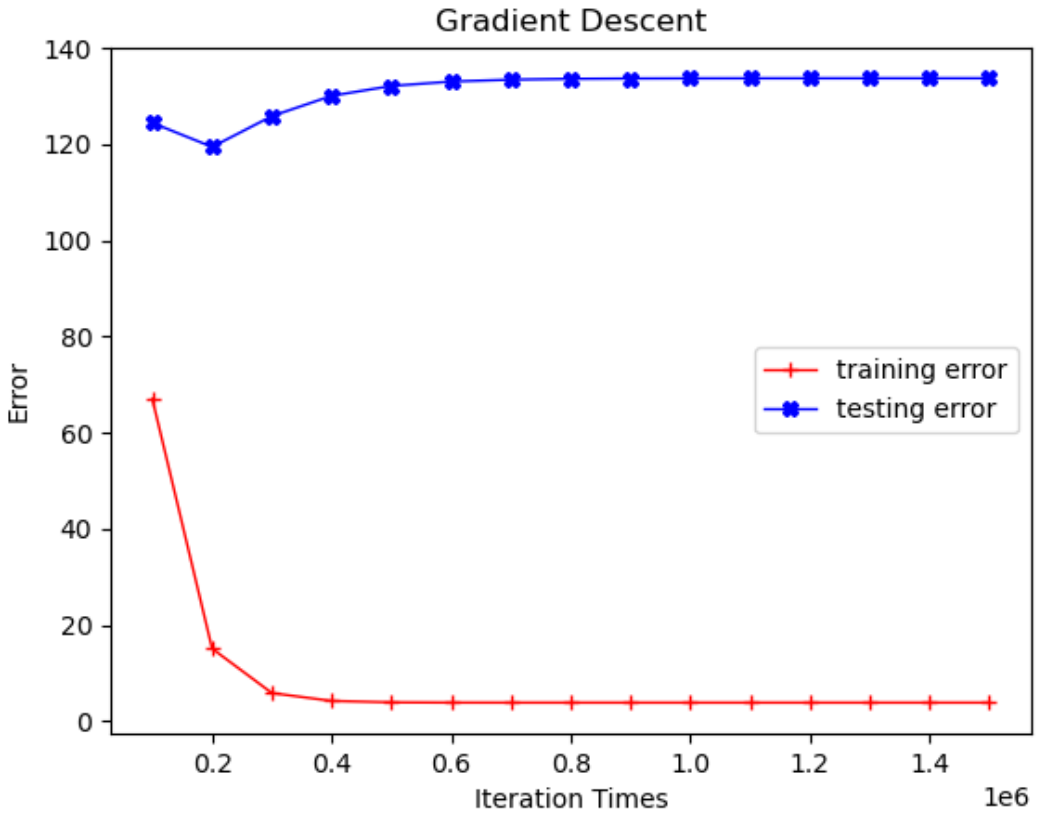
梯度下降迭代，单次迭代为：

$$\begin{aligned}
\theta_0 =& \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)}) \cdot x_0^{(i)}) \\
\theta_1 =& \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)}) \cdot x_1^{(i)}) \\
\theta_2 =& \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)}) \cdot x_2^{(i)})
\end{aligned}$$

计算误差：

$$E_\theta = \frac{1}{2m} \sum (h_\theta(x^{(i)}) - y^{(i)})^2$$

结果如图：



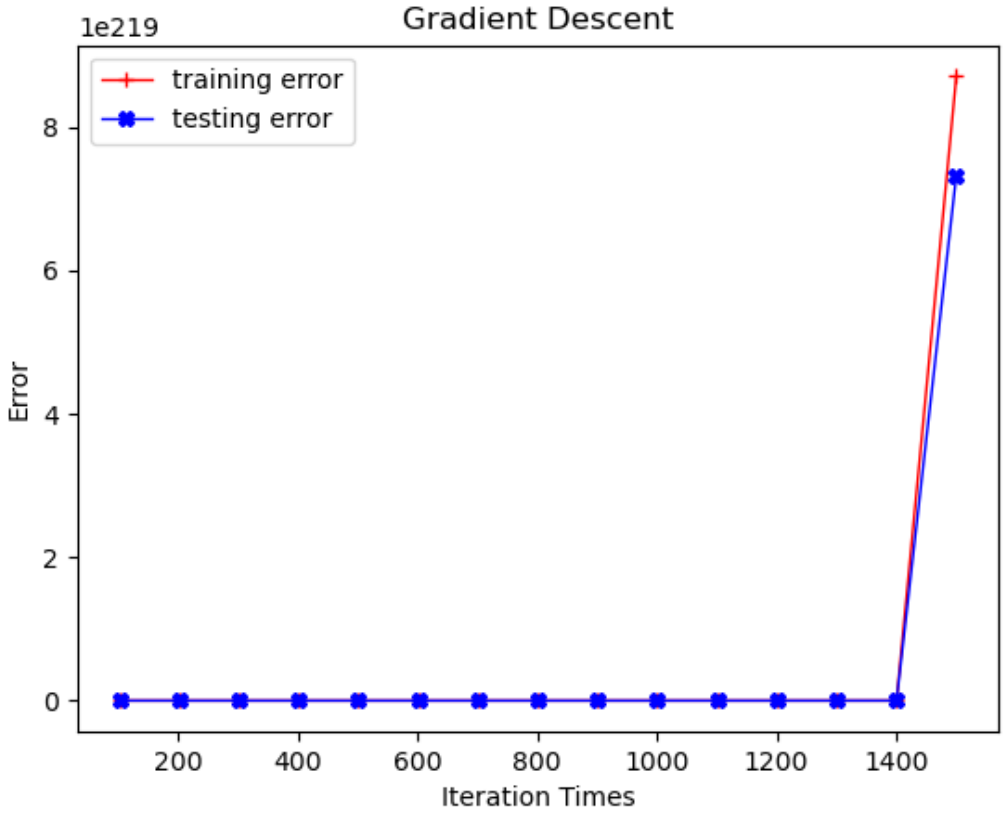

```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [12:39:28]
$ python ex1_1.py
迭代次数:  100000
迭代次数:  200000
迭代次数:  300000
迭代次数:  400000
迭代次数:  500000
迭代次数:  600000
迭代次数:  700000
迭代次数:  800000
迭代次数:  900000
迭代次数:  1000000
迭代次数:  1100000
迭代次数:  1200000
迭代次数:  1300000
迭代次数:  1400000
迭代次数:  1500000
```

iteration_times	theta_0	theta_1	theta_2	training_error	testing_error
100000	(46.3296090740901360+0j)	(7.089390249476696+0j)	(-72.75988620414542+0j)	(67.007704333427820+0j)	(124.463298589317320+0j)
200000	(65.487263774553910+0j)	(6.900063029404427+0j)	(-72.540751779944130+0j)	(15.059289289876860+0j)	(119.455291679237750+0j)
300000	(73.568301352866330+0j)	(6.820201459503370+0j)	(-72.448317004166030+0j)	(5.8160994701566920+0j)	(125.872118898378770+0j)
400000	(76.977025926052620+0j)	(6.786514436873318+0j)	(-72.409326380925220+0j)	(4.1714572231944040+0j)	(130.096468598810450+0j)
500000	(78.414886238325980+0j)	(6.772304654933331+0j)	(-72.392879446446940+0j)	(3.8788257644394490+0j)	(132.148404116384280+0j)
600000	(79.021401148292820+0j)	(6.766310717594361+0j)	(-72.385941838600900+0j)	(3.8267578020536805+0j)	(133.061993307844630+0j)
700000	(79.277239864247160+0j)	(6.763782368814935+0j)	(-72.383015432911550+0j)	(3.8174933411561920+0j)	(133.455910308005830+0j)
800000	(79.385157159446820+0j)	(6.762715866582936+0j)	(-72.381781023234820+0j)	(3.8158449141429080+0j)	(133.623592576504420+0j)
900000	(79.430678583204330+0j)	(6.762265997079153+0j)	(-72.381260327401200+0j)	(3.8155516092600570+0j)	(133.694595745257800+0j)
1000000	(79.449880324846260+0j)	(6.762076234165977+0j)	(-72.381040688689510+0j)	(3.8154994214752738+0j)	(133.724592630115750+0j)
1100000	(79.457979959076230+0j)	(6.761996188813019+0j)	(-72.380948041196350+0j)	(3.8154901356943514+0j)	(133.737254909110160+0j)
1200000	(79.461396528044120+0j)	(6.761962424267465+0j)	(-72.380908960845200+0j)	(3.8154884834738114+0j)	(133.742597606945280+0j)
1300000	(79.462837697262290+0j)	(6.761948181784970+0j)	(-72.380892476061890+0j)	(3.8154881894940003+0j)	(133.744851522661150+0j)
1400000	(79.463445607926230+0j)	(6.761942174053969+0j)	(-72.380885522488740+0j)	(3.8154881371860750+0j)	(133.745802312439960+0j)
1500000	(79.463702035398380+0j)	(6.761939639886750+0j)	(-72.380882589348670+0j)	(3.8154881278789357+0j)	(133.746203380980400+0j)

(b) 学习率 0.0002 进行训练

> 迭代次数 2000
> 学习率 0.0002
> 其他参数 0.0

> 由于学习率增大，导致 numpy 中的 float64 无法储存数据，出现越界情况，因此需要降低迭代次数




```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [14:47:17]
$ python ex1_1.py
迭代次数:  200
迭代次数:  400
迭代次数:  600
迭代次数:  800
迭代次数:  1000
迭代次数:  1200
迭代次数:  1400
迭代次数:  1600
迭代次数:  1800
迭代次数:  2000
```

iteration_times	theta_0	theta_1	theta_2	training_error	testing_error
200	(-7419647681519.265+0j)	(-7971089923731 54.8+0j)	(-400274004001 91.13+0j)	(6.941092026272314e+33+0j)	(5.843277999585407e+33+0j)
400	(-1.5334309487509745e+27+0j)	(-1.6473984357466904e+29+0j)	(-8.2725295332492 48e+27+0j)	(2.964760964451936e+62+0j)	(2.4958497095327123e+62+0j)
600	(-3.169167291385024e+41+0j)	(-3.404705795653743e+43+0j)	(-1.709697462096011 6e+42+0j)	(1.2663436161151593e+91+0j)	(1.0660567190225037e+91+0j)
800	(-6.549770844892271e+55+0j)	(-7.036562196141691e+57+0j)	(-3.533460231418971e+56+0j)	(5.408955977575895e+119+0j)	(4.5534669969602 81e+119+0j)
1000	(-1.3536520535605036e+70+0j)	(-1.454258033201477 2e+72+0j)	(-7.302661133808845e+70+0j)	(2.3103369729226776e+148+0j)	(1.9449304452973656e+148+0j)
1200	(-2.7976152532687 98e+84+0j)	(-3.005539307661744 4e+86+0j)	(-1.509253144015754 7e+85+0j)	(9.868183343665849e+176+0j)	(8.3074159526574 05e+176+0j)
1400	(-5.781878057020486e+98+0j)	(-6.211598164607341e+100+0j)	(-3.119198619494886e+99+0j)	(4.215014677318 33e+205+0j)	(3.5483613296985 66e+205+0j)
1600	(-1.194950371649372e+113+0j)	(-1.283761342272082e+115+0j)	(-6.446499758132 87e+113+0j)	(1.880366704922734e+234+0j)	(1.5156178765880466e+234+0j)
1800	(-2.469623843019604 3e+127+0j)	(-2.653170955749332 2e+129+0j)	(-1.332308845992517 6e+128+0j)	(7.689938280965801e+262+0j)	(6.473685553405475e+262+0j)
2000	(-5.104012744556454 3e+141+0j)	(-5.483352620646181e+143+0j)	(-2.753504890573424 4e+142+0j)	(3.284616995158333e+291+0j)	(2.765116807589 78e+291+0j)

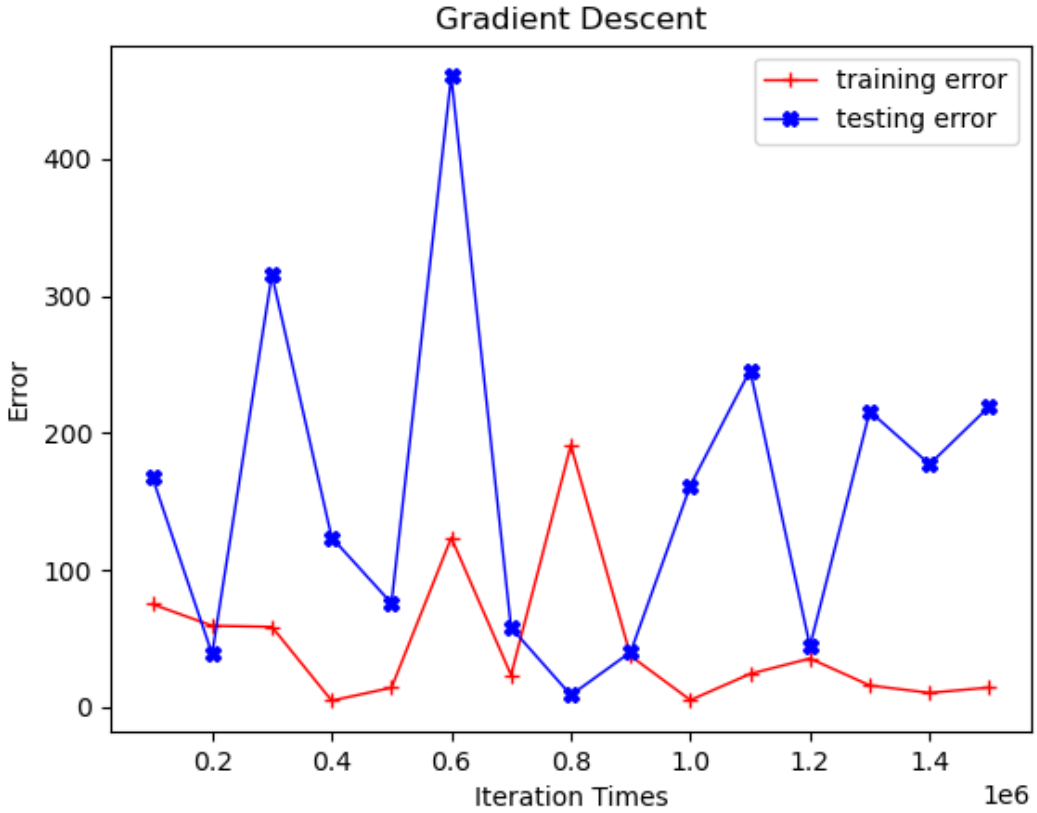
发现结果会在最后一轮迭代时误差变得极大，结果难以正确收敛，因此需要适当降低学习率，从而加快模型收敛

(c) 随机梯度下降迭代

> 迭代次数 1500000
> 学习率 0.00015
> 其他参数 0.0

对于每次迭代，用随机选择 k 个数据取代原本的使用全部数据，从而降低训练时间

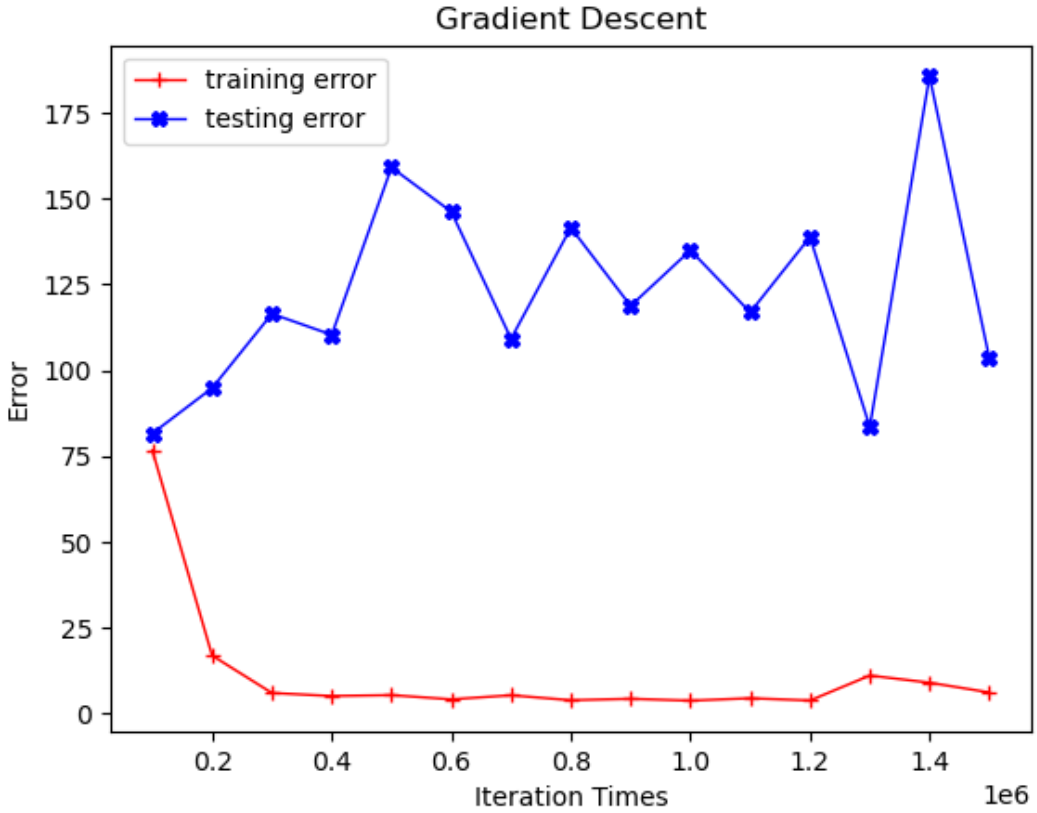
每次迭代随机选择1个点：




```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [14:46:23]
$ python ex1_1.py
迭代次数:  100000
迭代次数:  200000
迭代次数:  300000
迭代次数:  400000
迭代次数:  500000
迭代次数:  600000
迭代次数:  700000
迭代次数:  800000
迭代次数:  900000
迭代次数:  1000000
迭代次数:  1100000
迭代次数:  1200000
迭代次数:  1300000
迭代次数:  1400000
迭代次数:  1500000
```

iteration_times	theta_0	theta_1	theta_2	training_error	testing_error
100000	(45.56427166518742+0j)	(7.1208199740053 24+0j)	(-72.816973572695 52+0j)	(74.97398453731172+0j)	(167.75076768938254+0j)
200000	(64.90846655522468+0j)	(6.8568170793583 46+0j)	(-72.821351658232 83+0j)	(59.225796794407 05+0j)	(38.549673517654 27+0j)
300000	(73.17368037088248+0j)	(6.8949893510949 43+0j)	(-72.483043150060 82+0j)	(58.398587893439 53+0j)	(315.50989907402186+0j)
400000	(76.48365318043486+0j)	(6.7902737131022 82+0j)	(-72.471380060433 22+0j)	(4.5160393359687 98+0j)	(123.60297326669 93+0j)
500000	(78.10656345247246+0j)	(6.745915001714 1+0j)	(-72.416233263174 28+0j)	(13.988274197590 213+0j)	(75.585807220302 75+0j)
600000	(78.90298641266008+0j)	(6.872655963265 24+0j)	(-72.392800772082 03+0j)	(123.417302146646 95+0j)	(460.35036462246 2+0j)
700000	(79.3541023118299+0j)	(6.7196882465166 45+0j)	(-72.361589393483 06+0j)	(23.2670051812108+0j)	(57.844367360587 846+0j)
800000	(79.4735999304176+0j)	(6.6416196315011 08+0j)	(-72.598255386978 35+0j)	(190.997528612172 94+0j)	(8.5483823714741 17+0j)
900000	(79.4025303477398 4+0j)	(6.7034249576387 46+0j)	(-72.306102497841 69+0j)	(37.122312832188 81+0j)	(39.988656114588 816+0j)
1000000	(79.5021892679335 5+0j)	(6.7777845140717 69+0j)	(-72.513133140188 27+0j)	(4.9193224754465 91+0j)	(161.727728841019 14+0j)
1100000	(79.5699114130747 8+0j)	(6.8050906504390 95+0j)	(-72.399406043801 96+0j)	(24.1093591948625 7+0j)	(245.750254725355 03+0j)
1200000	(79.5938975198529 7+0j)	(6.7075715054892 32+0j)	(-72.393759991574 97+0j)	(35.276040615721 16+0j)	(44.333757484945 316+0j)
1300000	(79.4965411082491 5+0j)	(6.7951315894837 96+0j)	(-72.391349820293 2+0j)	(15.631156150106 316+0j)	(215.730561042922 9+0j)
1400000	(79.3686425616194+0j)	(6.7739824806303 69+0j)	(-72.134951740047 06+0j)	(10.196702542149 337+0j)	(177.045282188473 58+0j)
1500000	(79.4347748477340 2+0j)	(6.8031270831041 48+0j)	(-72.591072385622 2+0j)	(14.112072178456 312+0j)	(219.387162718380 8+0j)

每次迭代随机选择10个点:




```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [14:03:32]
$ python ex1_1.py
迭代次数:  100000
迭代次数:  200000
迭代次数:  300000
迭代次数:  400000
迭代次数:  500000
迭代次数:  600000
迭代次数:  700000
迭代次数:  800000
迭代次数:  900000
迭代次数:  1000000
迭代次数:  1100000
迭代次数:  1200000
迭代次数:  1300000
迭代次数:  1400000
迭代次数:  1500000
```

iteration_times	theta_0	theta_1	theta_2	training_error	testing_error
100000	(46.34809186052259+0j)	(7.059019801405366+0j)	(-72.75534094173833+0j)	(76.697418606638312+0j)	(81.625239089393363+0j)
200000	(65.493483570144394+0j)	(6.886562328709425+0j)	(-72.537102976434384+0j)	(16.959802218144054+0j)	(94.839881846209984+0j)
300000	(73.56643226873334+0j)	(6.815790462377278 5+0j)	(-72.454435283566244+0j)	(6.0613156379987474+0j)	(116.471644460452814+0j)
400000	(76.983615272689864+0j)	(6.776668718314834+0j)	(-72.405893220405674+0j)	(5.1736777574063174+0j)	(110.346526901997404+0j)
500000	(78.419417309644534+0j)	(6.784078012956168+0j)	(-72.389580044389864+0j)	(5.4383581478466744+0j)	(159.070375383708674+0j)
600000	(79.017304704292214+0j)	(6.772296508453784+0j)	(-72.386177374972974+0j)	(4.2098193838280854+0j)	(146.196001321130354+0j)
700000	(79.280313759520844+0j)	(6.752262050109669+0j)	(-72.39388029992564+0j)	(5.3927491951657884+0j)	(109.210832044538354+0j)
800000	(79.392301772606344+0j)	(6.766509917915975+0j)	(-72.39075382262934+0j)	(3.9426795314875056+0j)	(141.53027458158114+0j)
900000	(79.428732976255824+0j)	(6.755194120222312+0j)	(-72.38199524303974+0j)	(4.3675483854246364+0j)	(118.765897020681 94+0j)
1000000	(79.447094971960554+0j)	(6.762396217233191+0j)	(-72.37152635441066+0j)	(3.8225393386450194+0j)	(134.970026534679334+0j)
1100000	(79.466509461359074+0j)	(6.753930106545324+0j)	(-72.3838768697084+0j)	(4.5345212400896574+0j)	(116.879345271700744+0j)
1200000	(79.464918253996334+0j)	(6.764311682829974+0j)	(-72.38315575548212+0j)	(3.8714844025031536+0j)	(138.809853512295834+0j)
1300000	(79.463885609524 3+0j)	(6.735732068879663+0j)	(-72.37606299956574+0j)	(11.141734917805847+0j)	(83.555683824769524+0j)
1400000	(79.46774282308194+0j)	(6.783774268168681+0j)	(-72.37898253005474+0j)	(9.0599088879231024+0j)	(185.625645920685 74+0j)
1500000	(79.464675908010114+0j)	(6.747018955084334+0j)	(-72.38004724914009+0j)	(6.2183865169305355+0j)	(103.558585786424294+0j)

由于随机性的存在，即使随着迭代次数的增加导致测试误差整体下降，但是波动依然很大，但是训练时间较短

Exercise Two: Logistic Regression

(a) 建立梯度下降模型

使用 Sigmoid 函数作为逻辑函数

$$g(z) = \frac{1}{1+e^{-z}}$$

逻辑回归模型为：

$$P(y=1|x;\theta)=g(\theta^T x)=\frac{1}{1+e^{-\theta^T x}},\qquad \theta=[\,w_0\quad w_1\quad w_2\quad w_3\quad w_4\quad w_5\quad w_6\,]$$

因为 y 符合二项分布，对应条件的对数似然函数为：

$$LCL=\sum\log p^{(i)}+\sum\log(1-p^{(i)})$$

(b) 模型公式

对上述对数似然函数求偏导

$$\begin{aligned}&\text{对}\,w_0\,\text{求偏导}\qquad &&\frac{\partial}{\partial w_0}LCL=\sum(y^{(i)}-p^{(i)})\\&\text{对}\,w_j\,\text{求偏导}\qquad &&\frac{\partial}{\partial w_j}LCL=\sum(y^{(i)}-p^{(i)})x_j^{(i)},\qquad j\in[1,6],\ j\text{是整数}\end{aligned}$$

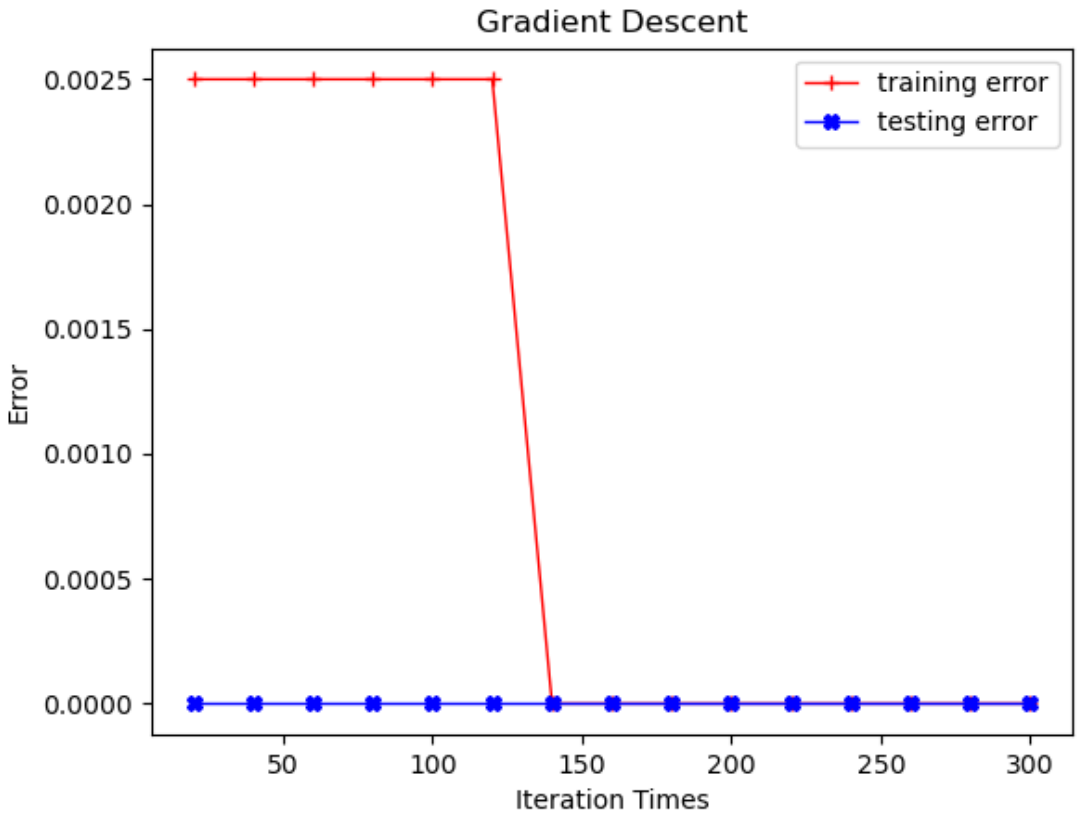
梯度下降为

$$\begin{aligned}w_0&:=w_0+\alpha\frac{\partial}{\partial w_0}LCL=w_0+\alpha\sum(y^{(i)}-p^{(i)})\\w_j&:=w_j+\alpha\frac{\partial}{\partial w_j}LCL=w_j+\alpha\sum(y^{(i)}-p^{(i)})x_j^{(i)},\qquad j\in[1,6],\ j\text{是整数}\end{aligned}$$

(c) 训练模型

> 学习率 0.00015
> 初始参数 $\theta=[\,w_0\quad w_1\quad w_2\quad w_3\quad w_4\quad w_5\quad w_6\,]=[\,0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\,]$

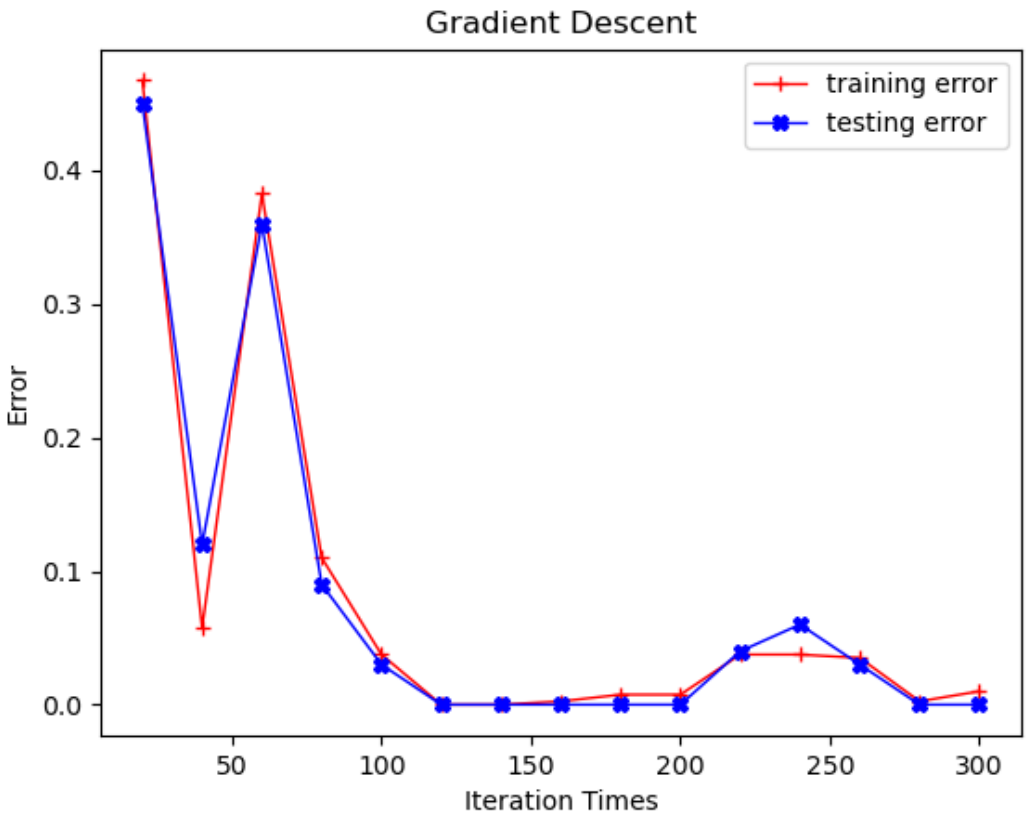
在不使用随机抽样的梯度下降的情况下，发现在迭代次数较小的时候便会收敛，并且在测试集上的效果也好



```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [21:41:28]
$ python ex2_1.py
```

iteration_times	theta_0	theta_1	theta_2	theta_3	theta_4	theta_5	theta_6	train_error_rate	test_error_rate
20	0.0182	-0.1252	0.1548	-0.1261	0.1531	-0.1093	-0.0034	0.0024999999999999467	0.0
40	0.0394	-0.2368	0.2984	-0.2390	0.2953	-0.2058	-0.0042	0.0024999999999999467	0.0
60	0.0609	-0.3378	0.4309	-0.3412	0.4266	-0.2925	-0.0040	0.0024999999999999467	0.0
80	0.0813	-0.4301	0.5530	-0.4346	0.5474	-0.3714	-0.0034	0.0024999999999999467	0.0
100	0.1004	-0.5148	0.6658	-0.5205	0.6588	-0.4437	-0.0027	0.0024999999999999467	0.0
120	0.1182	-0.5932	0.7702	-0.5997	0.7619	-0.5104	-0.0020	0.0024999999999999467	0.0
140	0.1347	-0.6658	0.8673	-0.6733	0.8575	-0.5721	-0.0013	0.0	0.0
160	0.1501	-0.7335	0.9579	-0.7417	0.9466	-0.6295	-0.0006	0.0	0.0
180	0.1645	-0.7968	1.0426	-0.8057	1.0299	-0.6830	0.0001	0.0	0.0
200	0.1779	-0.8561	1.1222	-0.8658	1.1081	-0.7332	0.0008	0.0	0.0
220	0.1906	-0.9120	1.1972	-0.9222	1.1816	-0.7804	0.0015	0.0	0.0
240	0.2025	-0.9648	1.2680	-0.9756	1.2510	-0.8249	0.0021	0.0	0.0
260	0.2138	-1.0147	1.3352	-1.0260	1.3167	-0.8669	0.0027	0.0	0.0
280	0.2244	-1.0621	1.3989	-1.0739	1.3791	-0.9067	0.0034	0.0	0.0
300	0.2346	-1.1072	1.4596	-1.1194	1.4384	-0.9446	0.0040	0.0	0.0

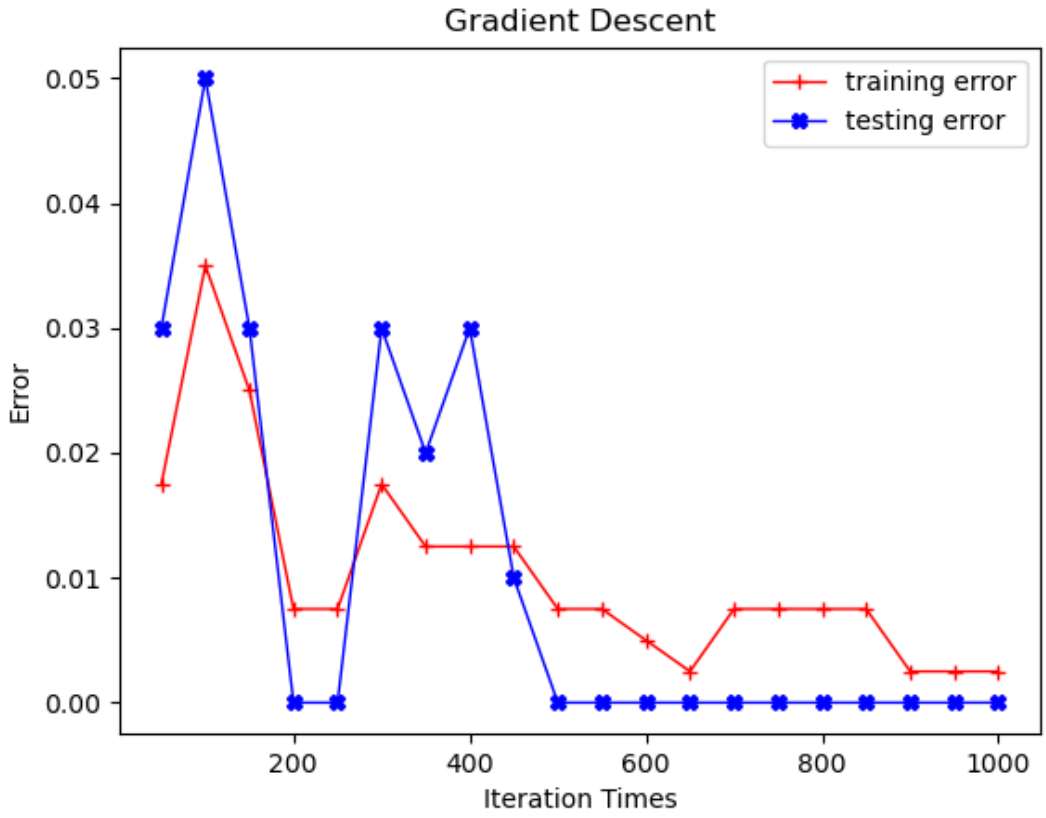
使用随机抽样进行随机梯度下降后，每次迭代选择一个点，效果如下：




```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [22:18:14]
$ python ex2_1.py
```

iteration_times	theta_0	theta_1	theta_2	theta_3	theta_4	theta_5	theta_6	train_error_rate	test_error_rate
20	0.0003	-0.0002	0.0006	-0.0002	0.0005	-0.0000	-0.0001	0.4675	0.449999999999999996
40	-0.0000	-0.0007	0.0008	-0.0007	0.0006	-0.0005	-0.0003	0.057499999999999996	0.12
60	0.0007	-0.0006	0.0016	-0.0008	0.0014	-0.0002	0.0001	0.38249999999999995	0.36
80	0.0007	-0.0010	0.0020	-0.0011	0.0017	-0.0005	-0.0001	0.10999999999999999	0.089999999999999997
100	0.0007	-0.0014	0.0025	-0.0015	0.0021	-0.0008	-0.0002	0.03749999999999998	0.030000000000000027
120	0.0004	-0.0019	0.0027	-0.0020	0.0023	-0.0012	-0.0004	0.0	0.0
140	0.0006	-0.0021	0.0033	-0.0023	0.0026	-0.0014	-0.0004	0.0	0.0
160	0.0004	-0.0026	0.0036	-0.0027	0.0028	-0.0019	-0.0003	0.0024999999999999467	0.0
180	0.0004	-0.0029	0.0038	-0.0030	0.0031	-0.0022	-0.0003	0.0074999999999999951	0.0
200	0.0004	-0.0032	0.0042	-0.0033	0.0035	-0.0024	-0.0004	0.0074999999999999951	0.0
220	0.0000	-0.0037	0.0043	-0.0039	0.0036	-0.0031	-0.0005	0.03749999999999998	0.040000000000000036
240	0.0000	-0.0041	0.0048	-0.0042	0.0040	-0.0035	-0.0005	0.03749999999999998	0.059999999999999994
260	0.0002	-0.0043	0.0052	-0.0044	0.0044	-0.0038	-0.0005	0.03500000000000003	0.030000000000000027
280	0.0009	-0.0044	0.0060	-0.0046	0.0052	-0.0036	-0.0002	0.0024999999999999467	0.0
300	0.0006	-0.0049	0.0062	-0.0050	0.0055	-0.0041	-0.0003	0.010000000000000009	0.0

每次迭代选择10个点，效果如下：




```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [23:18:35]
$ python ex2_1.py
```

iteration_times	theta_0	theta_1	theta_2	theta_3	theta_4	theta_5	theta_6	train_error_rate	test_error_rate
50	0.0006	-0.0082	0.0098	-0.0089	0.0096	-0.0077	-0.0000	0.01749999999999996	0.030000000000000027
100	-0.0003	-0.0168	0.0194	-0.0180	0.0185	-0.0164	-0.0009	0.03500000000000003	0.049999999999999993
150	0.0006	-0.0251	0.0289	-0.0265	0.0286	-0.0239	-0.0008	0.025000000000000022	0.030000000000000027
200	0.0037	-0.0313	0.0397	-0.0335	0.0394	-0.0289	0.0004	0.007499999999999951	0.0
250	0.0049	-0.0394	0.0498	-0.0418	0.0490	-0.0357	0.0014	0.007499999999999951	0.0
300	0.0028	-0.0492	0.0579	-0.0509	0.0571	-0.0450	-0.0004	0.01749999999999996	0.030000000000000027
350	0.0041	-0.0570	0.0679	-0.0582	0.0671	-0.0518	-0.0007	0.012499999999999956	0.020000000000000018
400	0.0045	-0.0654	0.0773	-0.0661	0.0766	-0.0588	-0.0015	0.012499999999999956	0.030000000000000027
450	0.0065	-0.0727	0.0876	-0.0736	0.0866	-0.0650	-0.0006	0.012499999999999956	0.010000000000000009
500	0.0098	-0.0794	0.0981	-0.0802	0.0969	-0.0701	0.0002	0.007499999999999951	0.0
550	0.0104	-0.0875	0.1075	-0.0881	0.1062	-0.0771	-0.0007	0.007499999999999951	0.0
600	0.0127	-0.0947	0.1177	-0.0956	0.1160	-0.0829	-0.0008	0.0050000000000000044	0.0
650	0.0148	-0.1020	0.1278	-0.1028	0.1259	-0.0887	-0.0012	0.0024999999999999467	0.0
700	0.0132	-0.1113	0.1363	-0.1113	0.1337	-0.0971	-0.0020	0.007499999999999951	0.0
750	0.0128	-0.1192	0.1451	-0.1196	0.1421	-0.1046	-0.0033	0.007499999999999951	0.0
800	0.0149	-0.1260	0.1549	-0.1265	0.1518	-0.1102	-0.0029	0.007499999999999951	0.0
850	0.0164	-0.1335	0.1642	-0.1340	0.1612	-0.1164	-0.0033	0.007499999999999951	0.0
900	0.0197	-0.1401	0.1743	-0.1406	0.1713	-0.1212	-0.0022	0.0024999999999999467	0.0
950	0.0219	-0.1473	0.1841	-0.1474	0.1804	-0.1278	-0.0010	0.0024999999999999467	0.0
1000	0.0246	-0.1536	0.1939	-0.1540	0.1904	-0.1335	-0.0008	0.0024999999999999467	0.0

发现由于随机抽样的存在，会使得上述两个数据变化较大，但是都能在较小迭代次数时使得错误率收敛到0

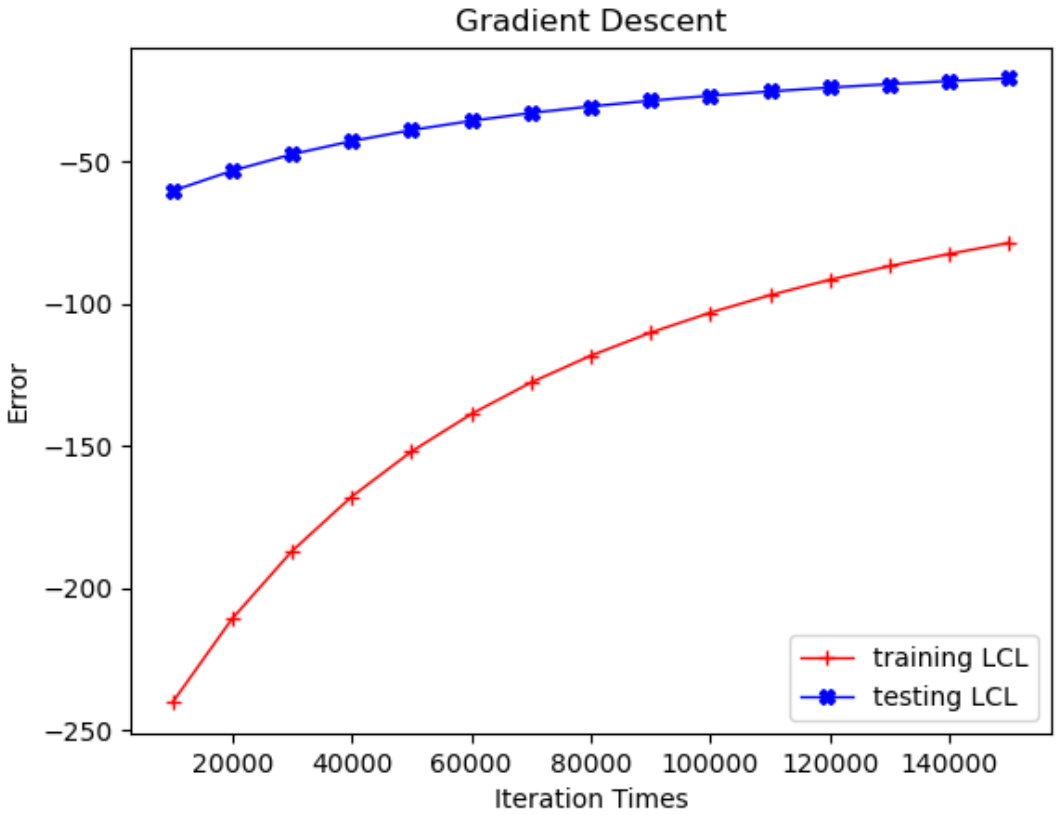
(d) 何时错误率为0

当迭代次数足够多时，错误率会降低到0

(e)

> 学习率为 0.00015
> 迭代次数 150000，每10000次进行一次对比
> 每次随机抽样一个点
> 初始参数 $\theta = [w_0 \quad w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6] = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

效果如图：

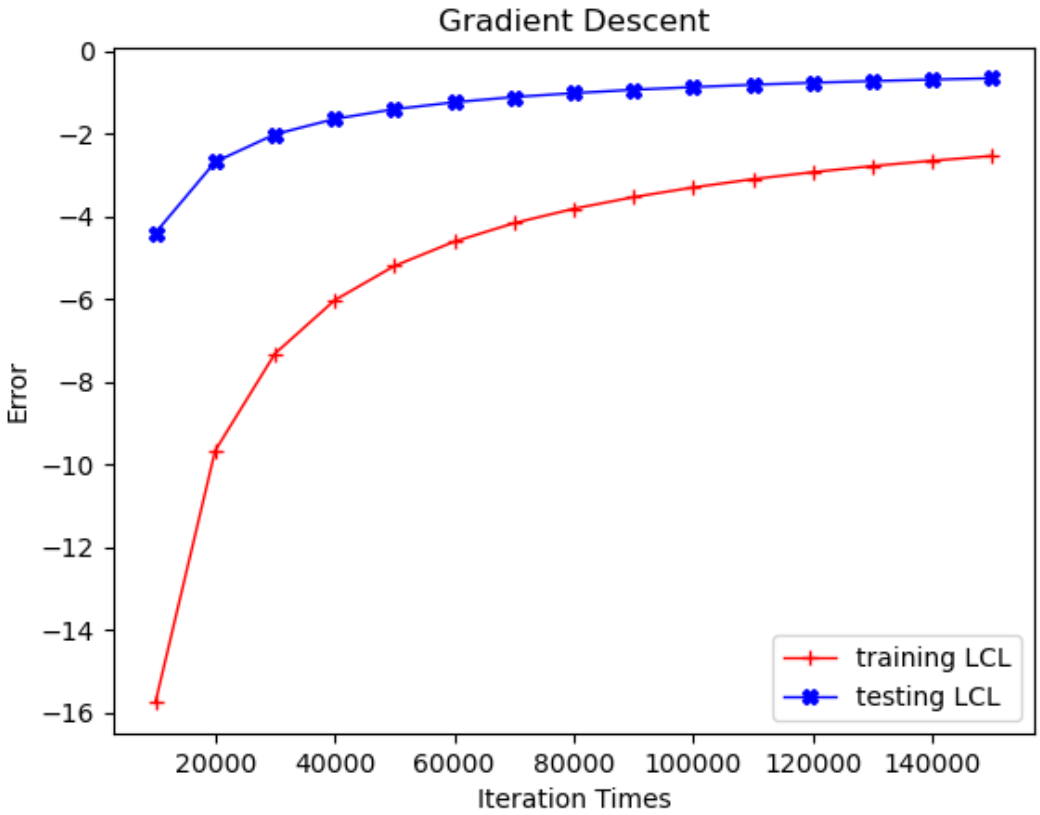



```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [23:32:17]
$ python ex2_1.py
```

iteration_times	theta_0	theta_1	theta_2	theta_3	theta_4	theta_5	theta_6	train_LCL	test_LCL
10000	0.0188	-0.1584	0.1903	-0.1557	0.1871	-0.1362	-0.0045	-240.146876770736 58	-60.471035063306 41
20000	0.0457	-0.2916	0.3654	-0.2917	0.3583	-0.2530	-0.0039	-210.674921222132 78	-53.378081684296 276
30000	0.0797	-0.4060	0.5249	-0.4098	0.5200	-0.3501	0.0019	-186.967987141329 36	-47.5883511537697
40000	0.1002	-0.5140	0.6651	-0.5195	0.6594	-0.4424	-0.0005	-167.7849146307326	-42.947230145643 07
50000	0.1204	-0.6127	0.7930	-0.6183	0.7862	-0.5270	-0.0001	-151.951359977701 26	-39.088188007471 25
60000	0.1426	-0.7001	0.9116	-0.7074	0.9039	-0.6006	0.0019	-138.7919585465978	-35.846399057498 85
70000	0.1679	-0.7772	1.0238	-0.7870	1.0141	-0.6652	0.0065	-127.712613182453 82	-33.072818470027 82
80000	0.1825	-0.8532	1.1223	-0.8643	1.1113	-0.7302	0.0062	-118.286072500093 03	-30.757302443046 438
90000	0.1964	-0.9233	1.2148	-0.9341	1.2022	-0.7895	0.0051	-110.199053167037 46	-28.762068277824 365
100000	0.2102	-0.9881	1.3013	-0.9987	1.2873	-0.8445	0.0053	-103.176921984446 2	-27.016930649066 573
110000	0.2192	-1.0512	1.3804	-1.0617	1.3639	-0.8983	0.0035	-97.038034072331 39	-25.513591572400 98
120000	0.2347	-1.1066	1.4573	-1.1180	1.4404	-0.9435	0.0053	-91.628020549849 51	-24.142266523860 14
130000	0.2460	-1.1603	1.5285	-1.1723	1.5104	-0.9889	0.0051	-86.822154854968 64	-22.941020269806 746
140000	0.2581	-1.2101	1.5976	-1.2224	1.5776	-1.0298	0.0061	-82.530540122070 86	-21.852745930776 162
150000	0.2688	-1.2587	1.6615	-1.2710	1.6405	-1.0695	0.0049	-78.671369010795 54	-20.884950748601 288

发现 LCL 虽然在收敛，但是收敛速率较慢，因此决定调整参数

> 学习率为 0.002
> 迭代次数 150000，每10000次进行一次对比
> 每次随机抽样10个点
> 初始参数 $\theta = [w_0 \quad w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6] = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$



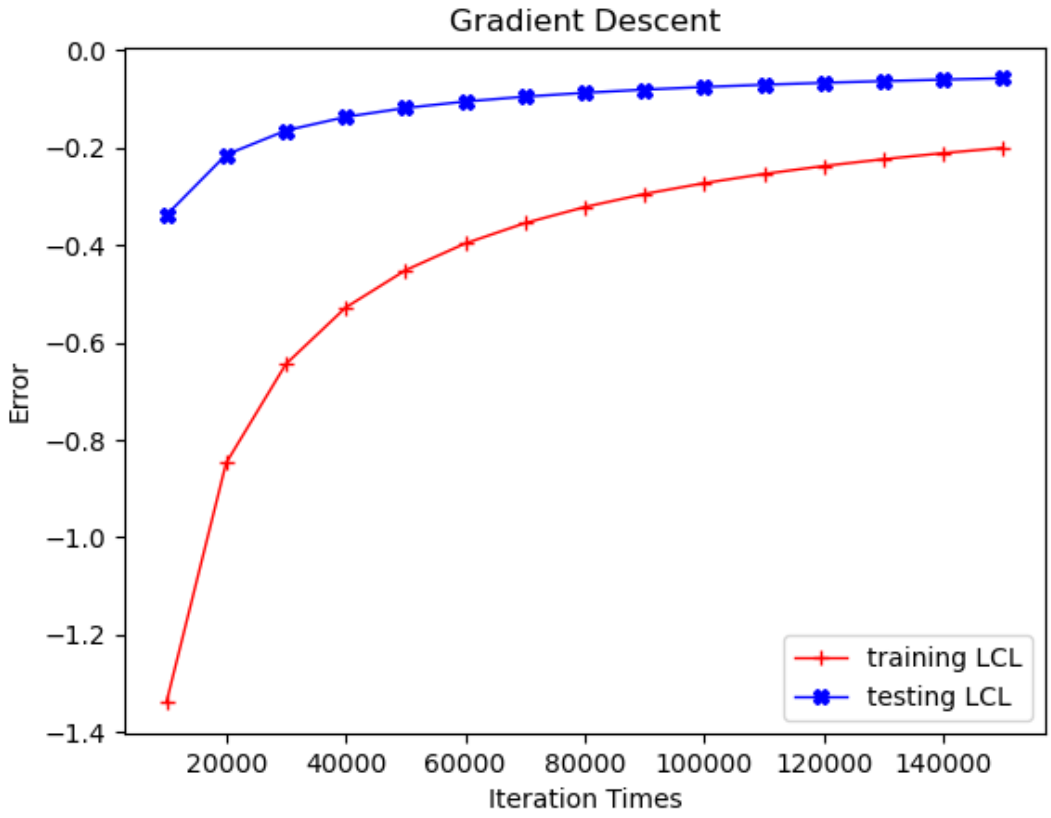

```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [23:35:36]
$ python ex2_1.py
```

iteration_times	theta_0	theta_1	theta_2	theta_3	theta_4	theta_5	theta_6	train_LCL	test_LCL
10000	0.6637	-3.0826	4.1441	-3.1128	4.0387	-2.5185	0.0736	-15.749085903369812	-4.396161109002259
20000	0.8048	-3.7593	5.1096	-3.8013	4.9317	-3.0378	0.1118	-9.659963910622967	-2.679273613127637
30000	0.8833	-4.1854	5.7315	-4.2388	5.4847	-3.3414	0.1331	-7.313771246468274	-2.0114347501704284
40000	0.9415	-4.4969	6.2008	-4.5664	5.8962	-3.5635	0.1526	-6.021470819408092	-1.6395830356292556
50000	0.9799	-4.7476	6.5793	-4.8263	6.2185	-3.7436	0.1614	-5.192702797680419	-1.40437594977767765
60000	1.0200	-4.9577	6.9057	-5.0432	6.4921	-3.8801	0.1611	-4.605252845105316	-1.2364596426214909
70000	1.0481	-5.1454	7.1931	-5.2347	6.7262	-4.0014	0.1600	-4.158115375316776	-1.1106031144082096
80000	1.0688	-5.3086	7.4473	-5.4067	6.9287	-4.1090	0.1454	-3.8100438607340195	-1.015339775413815
90000	1.0952	-5.4485	7.6777	-5.5603	7.1154	-4.1991	0.1487	-3.5291206893500577	-0.9349252442673722
100000	1.1104	-5.5829	7.8881	-5.7017	7.2822	-4.2837	0.1449	-3.294530239246505	-0.8697731249254488
110000	1.1271	-5.7045	8.0850	-5.8306	7.4382	-4.3635	0.1465	-3.0944970241130805	-0.8131268170643279
120000	1.1492	-5.8128	8.2663	-5.9488	7.5841	-4.4258	0.1550	-2.9257655862491245	-0.7646994707646675
130000	1.1606	-5.9162	8.4361	-6.0650	7.7118	-4.4905	0.1523	-2.777917159360264	-0.7241264887360227
140000	1.1674	-6.0166	8.5907	-6.1742	7.8310	-4.5552	0.1503	-2.648621667816832	-0.6885670283806418
150000	1.1840	-6.1058	8.7418	-6.2688	7.9477	-4.6072	0.1485	-2.534285684669245	-0.6567597221765253

可以看到，在至少90000次后，测试 LCL 低于 -1

> 学习率为 0.002
> 迭代次数 150000，每10000次进行一次对比
> 每次使用全部数据
> 初始参数 $\theta = [w_0 \quad w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6] = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

效果如下：



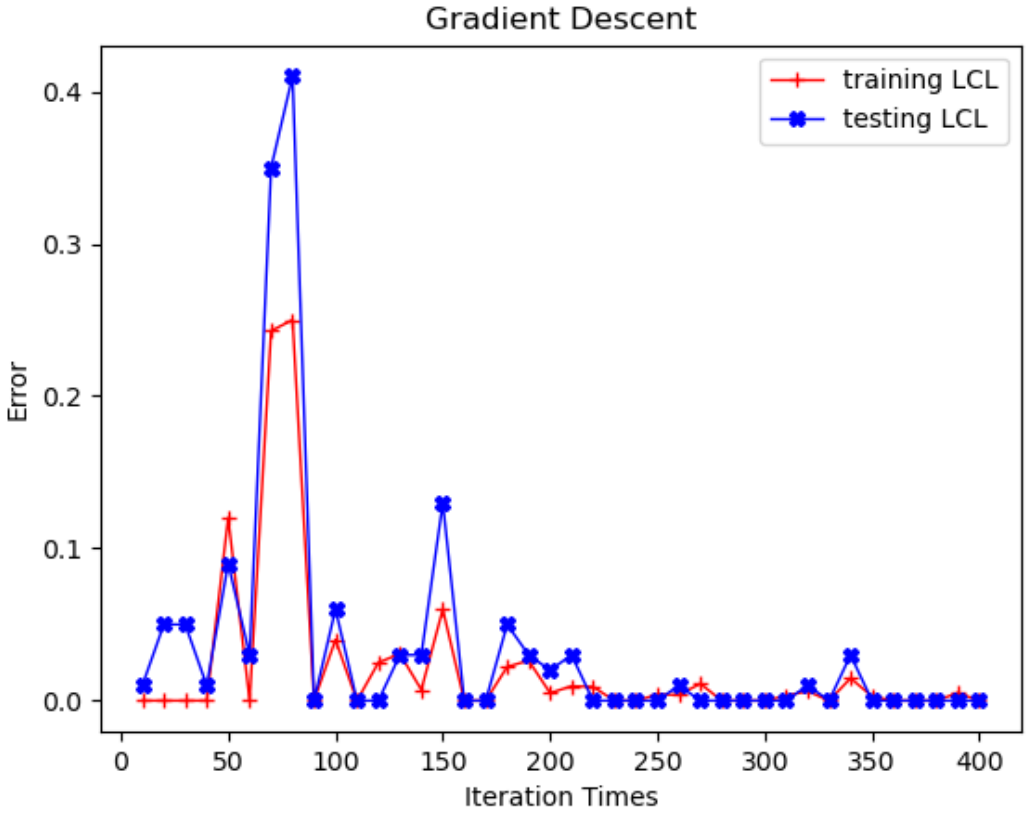

```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [23:40:34]
$ python ex2_1.py
迭代次数:  10000
迭代次数:  20000
迭代次数:  30000
迭代次数:  40000
迭代次数:  50000
迭代次数:  60000
迭代次数:  70000
迭代次数:  80000
迭代次数:  90000
迭代次数:  100000
迭代次数:  110000
迭代次数:  120000
迭代次数:  130000
迭代次数:  140000
迭代次数:  150000
+------------------+----------+----------+----------+----------+----------+----------+----------+---------------------+---------------------+
| iteration_times | theta_0 | theta_1  | theta_2 | theta_3  | theta_4 | theta_5 | theta_6 |      train_LCL      |      test_LCL       |
+------------------+----------+----------+----------+----------+----------+----------+----------+---------------------+---------------------+
|      10000       |  1.3529 | -7.5270  | 11.1754 | -7.8734  |  9.6892 | -5.3665 |  0.0729 |  -1.3395806064968 76 | -0.33821020863814194 |
|      20000       |  1.4504 | -8.6913  | 13.2218 | -9.2662  | 11.0464 | -5.8667 |  0.0145 | -0.84725487569186 11 | -0.21404030642625943 |
|      30000       |  1.5021 | -9.4524  | 14.5567 | -10.2057 | 11.8893 | -6.1344 | -0.0157 | -0.64395007939320 34 | -0.16433834883185244 |
|      40000       |  1.5377 | -10.0341 | 15.5671 | -10.9355 | 12.5102 | -6.3099 | -0.0340 | -0.52792040294851 83 | -0.13634100814741174 |
|      50000       |  1.5653 | -10.5114 | 16.3874 | -11.5399 | 13.0055 | -6.4371 | -0.0462 | -0.45129014997864 15 | -0.11794506285468936 |
|      60000       |  1.5885 | -10.9193 | 17.0813 | -12.0595 | 13.4192 | -6.5350 | -0.0546 | -0.39622709842316 73 | -0.10473789339652366 |
|      70000       |  1.6087 | -11.2773 | 17.6844 | -12.5172 | 13.7757 | -6.6137 | -0.0607 | -0.35441646337753 74 | -0.09469298093925343 |
|      80000       |  1.6268 | -11.5973 | 18.2190 | -12.9274 | 14.0895 | -6.6790 | -0.0651 | -0.32140583779449 59 | -0.086737234335853 63 |
|      90000       |  1.6434 | -11.8874 | 18.6998 | -13.2997 | 14.3702 | -6.7345 | -0.0683 |  -0.2945741612284 79 | -0.08024418452910695 |
|      100000      |  1.6589 | -12.1532 | 19.1373 | -13.6412 | 14.6246 | -6.7826 | -0.0707 | -0.27226788429696 89 | -0.07482104616634 03 |
|      110000      |  1.6735 | -12.3988 | 19.5388 | -13.9569 | 14.8574 | -6.8250 | -0.0723 | -0.25338716280333 28 | -0.07020773618841293 |
|      120000      |  1.6873 | -12.6273 | 19.9103 | -14.2507 | 15.0721 | -6.8628 | -0.0735 | -0.23716921856866607 | -0.0662244509239065  |
|      130000      |  1.7005 | -12.8412 | 20.2560 | -14.5256 | 15.2716 | -6.8970 | -0.0743 | -0.22306680569838 97 | -0.06274246473386888 |
|      140000      |  1.7132 | -13.0423 | 20.5794 | -14.7840 | 15.4580 | -6.9281 | -0.0747 | -0.21067614349300862 | -0.05966691719953799 |
|      150000      |  1.7255 | -13.2322 | 20.8835 | -15.0280 | 15.6329 | -6.9568 | -0.0749 | -0.19969219148435 76 | -0.05692618424958884 |
+------------------+----------+----------+----------+----------+----------+----------+----------+---------------------+---------------------+
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [0:17:14]
```

可以发现，使用全部数据收敛的更快，但是训练的时间更久

(f) k 类

> 学习率 0.00015
> 每个测试集迭代次数150次，迭代完成后进行对比
> 初始参数 $\theta = [w_0 \quad w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6] = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

输出如下：




```
# m7811 at LAPTOP-QTCGESHO in D:\blog\work\数据挖掘\003\src on git:master o +1 ~4 -1 ! [17:24:24]
$ python ex2_1.py
训练集数量为 100
训练集数量为 200
训练集数量为 300
训练集数量为 400
```

set	theta_0	theta_1	theta_2	theta_3	theta_4	theta_5	theta_6	train_error_rate	test_error_rate
10	0.0002	-0.0133	0.0280	-0.0289	0.0325	-0.0188	-0.0099	0.0	0.010000000000000009
20	0.0201	-0.0364	0.0818	-0.0273	0.0684	-0.0255	-0.0385	0.0	0.04999999999999993
30	0.0037	-0.0756	0.0758	-0.0858	0.0959	-0.0828	-0.0156	0.0	0.04999999999999993
40	0.0386	-0.0820	0.1290	-0.0910	0.1414	-0.0703	-0.0211	0.0	0.010000000000000009
50	0.0688	-0.0973	0.1763	-0.0843	0.1719	-0.0573	-0.0192	0.12	0.08999999999999997
60	-0.0072	-0.1507	0.1539	-0.1781	0.1653	-0.1132	0.0084	0.0	0.030000000000000027
70	-0.0516	-0.2095	0.1584	-0.2052	0.1435	-0.1888	-0.0160	0.24285714285714288	0.35
80	-0.0869	-0.2250	0.1942	-0.2336	0.1677	-0.2254	-0.0866	0.25	0.41000000000000003
90	0.0418	-0.1796	0.2553	-0.2124	0.2965	-0.1935	0.0210	0.0	0.0
100	-0.0171	-0.2065	0.2715	-0.2252	0.2249	-0.2459	0.0008	0.040000000000000036	0.05999999999999994
110	0.0431	-0.2173	0.3099	-0.2518	0.3325	-0.1931	-0.0186	0.0	0.0
120	0.0362	-0.2517	0.3221	-0.2789	0.3046	-0.2584	0.0221	0.025000000000000022	0.0
130	0.0102	-0.2768	0.3425	-0.3159	0.3411	-0.2642	-0.0452	0.03076923076923066	0.030000000000000027
140	0.0091	-0.3202	0.3746	-0.3230	0.3502	-0.2414	-0.0395	0.0071428571428572285	0.030000000000000027
150	-0.0427	-0.3469	0.3575	-0.3690	0.3945	-0.3218	-0.0845	0.05999999999999994	0.13
160	0.0526	-0.3427	0.4173	-0.3384	0.4206	-0.2786	-0.0193	0.0	0.0
170	0.0794	-0.3314	0.4776	-0.3670	0.4526	-0.2910	0.0192	0.0	0.0
180	-0.0107	-0.4105	0.4449	-0.3783	0.4541	-0.3922	-0.0773	0.022222222222222143	0.04999999999999993
190	0.0249	-0.4310	0.4978	-0.4001	0.4540	-0.3455	-0.0836	0.026315789473684218	0.030000000000000027
200	0.0511	-0.4239	0.4827	-0.4605	0.5269	-0.4038	-0.0268	0.005000000000000044	0.020000000000000018
210	0.0519	-0.4560	0.5348	-0.4696	0.5105	-0.4039	-0.0749	0.00952380952380949	0.030000000000000027
220	0.0540	-0.4188	0.6041	-0.4266	0.5112	-0.4197	-0.0050	0.00909090909090915	0.0
230	0.1425	-0.4155	0.6446	-0.4612	0.5939	-0.4065	0.0234	0.0	0.0
240	0.1387	-0.4913	0.6216	-0.4692	0.6180	-0.3982	0.0433	0.0	0.0
250	0.0799	-0.5124	0.6089	-0.4980	0.6033	-0.4475	0.0183	0.004000000000000036	0.0
260	0.1733	-0.4383	0.6563	-0.4692	0.7208	-0.4090	0.0777	0.0038461538461538325	0.010000000000000009
270	0.0709	-0.5164	0.6692	-0.5415	0.6390	-0.5072	0.0334	0.011111111111111072	0.0
280	0.1722	-0.5176	0.6679	-0.5330	0.7319	-0.4036	0.0448	0.0	0.0
290	0.1125	-0.5634	0.6556	-0.5889	0.7687	-0.4843	0.0195	0.0	0.0
300	0.1313	-0.5663	0.7326	-0.5829	0.7239	-0.4734	0.0217	0.0	0.0
310	0.0829	-0.6044	0.7527	-0.6161	0.7283	-0.4986	0.0207	0.0032258064516129856	0.0
320	0.0787	-0.6206	0.7447	-0.6029	0.7321	-0.5217	-0.0654	0.006249999999999978	0.010000000000000009
330	0.1247	-0.6081	0.8033	-0.6419	0.7893	-0.5105	0.0244	0.0	0.0
340	0.0617	-0.6647	0.8058	-0.6560	0.6999	-0.5308	-0.0453	0.014705882352941235	0.030000000000000027
350	0.1055	-0.5849	0.8428	-0.6310	0.8075	-0.5879	-0.0629	0.0028571428571428914	0.0
360	0.1400	-0.6375	0.8659	-0.6660	0.8536	-0.5439	0.0400	0.0	0.0
370	0.1502	-0.6622	0.8609	-0.6694	0.8885	-0.5608	-0.0187	0.0	0.0
380	0.1862	-0.6456	0.8977	-0.6707	0.8869	-0.5417	-0.0537	0.0	0.0
390	0.1280	-0.7036	0.8742	-0.6894	0.8657	-0.6090	0.0510	0.00512820512820511	0.0
400	0.1425	-0.6985	0.9070	-0.7307	0.8810	-0.5520	-0.0395	0.0	0.0

可以发现，在统一迭代次数和学习率的情况下，随着测试样本的增加，错误率整体逐渐降低；但是由于是随机抽样的样本选取，如果只进行一次完整的过程的话，可能导致错误率的波动