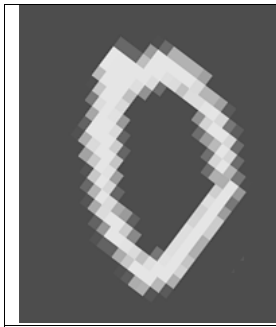
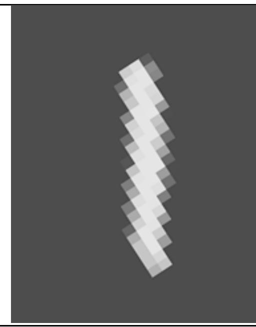
GTID: sjiang98

0. Data Preprocessing

4 sample images, one from each class, along with their class labels to demonstrate you've read the data correctly.

			
train_0_1 (class lable: 0)	train_3_5 (class lable: 3)	test_0_1 (class lable: 1)	test_3_5 (class lable: 5)

1. Theory

a. Write down the formula for computing the gradient of the loss function used in Logistic Regression. Specify what each variable represents in the equation.

sigmoid function

$$h_\theta(x) = \frac{1}{1+e^{-\langle\theta,x\rangle}}$$

apply log to loss function

$$\log(1+\exp(y^{(i)}\langle\theta,x^{(i)}\rangle))$$

gradient decent

J(θ) in the pseudocode

$$\theta_j = \theta_j - \alpha\left(\frac{\partial \sum_{i=1}^{n}\log(1+\exp(y^{(i)}\langle\theta,x^{(i)}\rangle))}{\partial\theta_j}\right)$$

$$= \theta_j - \alpha\left(\sum_{i=1}^{n}\frac{\exp(y^{(i)}\langle\theta,x^{(i)}\rangle)\cdot y^{(i)}\cdot x_j^{(i)}}{1+\exp(y^{(i)}\langle\theta,x^{(i)}\rangle)}\right)$$

$$= \theta_j - \alpha\left(\sum_{i=1}^{n}\frac{y^{(i)}x_j^{(i)}}{\exp(-y^{(i)}\langle\theta,x^{(i)}\rangle)+1}\right) \bigstar$$

α: learning rate or step size, if it's too small, then the speed will be slow; if it's too large, then we would miss the optimal solution; **θ**: current parameter estimate; **n**: number of training samples; **y**: output; **x**: feature

b. Write pseudocode for training a model using Logistic Regression.

Initialize θ to a random non-zero vector

Repeat until convergence

for all j = 0, .., n, do $\theta_i' = \theta_j - \alpha\,(\partial J(\theta) / \partial\theta_j)$

for all j = 0, .., n, do $\theta_j = \theta_j'$

Output θ

c. Calculate the number of operations per gradient descent iteration.

(Hint: Use variable n for number of examples and d for dimensionality.)

O(nd) per iteration

3. Training

For 3a, train and test accuracies for 0/1 and 3/ 5 classification.

train_0_1	train_3_5	test_0_1	test_3_5
99.25%	91.05%	99.76%	93.10%

For 3b, average test and train accuracies for 0/1 and 3/5 classification. (Since I used Batch, I sampled 80% data for 10 runs.)

Dataset	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
train_0_1	99.62%	99.64%	99.66%	99.65%	99.65%	99.68%	99.65%	99.65%	99.63%	99.62%	99.65%
train_3_5	91.32%	91.16%	91.18%	91.23%	91.36%	91.20%	91.17%	91.15%	90.95%	91.32%	91.20%
test_0_1	99.65%	99.59%	99.76%	99.65%	99.70%	99.70%	99.65%	99.76%	99.76%	99.65%	99.69%
test_3_5	93.62%	93.56%	93.95%	93.29%	93.89%	93.23%	93.03%	93.50%	93.56%	93.36%	93.50%

3c- the accuracy on "_3_5" datasets is lower than the accuracy on "_0_1" datasets. The reason could be that 3 and 5 have a more complicated pixel matrix than 0 and 1.
3d- if we have more than two classes to classify, we can make it into binary classification. Labeling one class as one class in binary classification and all other classes as the other class in binary classification.

4. Evaluation (only for "_3_5" datasets)

4a1- I changed alpha (learning rate) while kept other perimeters the same as the baseline
Dataset: train_3_5

alpha	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
0.1	80.12%	80.29%	80.24%	80.19%	80.41%	80.18%	80.46%	80.23%	80.23%	79.82%	80.22%
0.5	89.55%	89.63%	89.51%	89.28%	89.45%	89.31%	89.51%	89.27%	89.51%	89.44%	89.45%
0.9(baseline)	91.32%	91.16%	91.18%	91.23%	91.36%	91.20%	91.17%	91.15%	90.95%	91.32%	91.20%
1.5	93.00%	92.66%	92.87%	92.89%	92.80%	92.63%	92.52%	92.54%	92.63%	92.41%	92.70%
2	92.25%	93.33%	93.28%	93.41%	93.47%	93.12%	93.50%	93.24%	93.36%	93.42%	93.24%
20	94.82%	94.78%	95.02%	95.23%	94.77%	95.10%	94.90%	94.28%	95.14%	93.32%	94.74%

Dataset: test_3_5

alpha	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
0.1	81.13%	80.80%	81.92%	80.67%	80.87%	81.07%	81.20%	80.80%	80.74%	80.54%	80.97%
0.5	90.99%	91.45%	91.45%	90.60%	91.52%	91.72%	91.52%	91.26%	91.32%	90.93%	91.28%
0.9(baseline)	93.62%	93.56%	93.95%	93.29%	93.89%	93.23%	93.03%	93.50%	93.56%	93.36%	93.50%

1.5	95.13 %	93.95 %	93.75 %	95.13 %	94.87 %	94.28 %	94.67 %	94.81 %	95.73 %	94.67 %	94.70 %
2	95.60 %	95.40 %	95.40 %	95.92 %	93.95 %	95.27 %	94.67 %	94.94 %	95.79 %	96.25 %	95.32 %
20	94.81 %	97.17 %	94.61 %	95.00 %	97.11 %	97.17 %	97.70 %	95.07 %	97.50 %	97.50 %	96.36 %

As shown in the tables, smaller learning rate does not guarantee larger accuracy.

4a2- I changed initial theta while kept other perimeters the same as the baseline
Dataset: train_3_5

Initial theta	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
rep(0,ncol(train_3_5))	91.32%	91.16%	91.18%	91.23%	91.36%	91.20%	91.17%	91.15%	90.95%	91.32%	91.20%
rep(1,ncol(train_3_5))	92.53%	92.47%	92.65%	92.26%	92.62%	92.52%	92.81%	92.64%	92.47%	92.71%	92.57 %

Dataset: test_3_5

alpha	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
rep(0,ncol(train_3_5))	93.62%	93.56%	93.95%	93.29%	93.89%	93.23%	93.03%	93.50%	93.56%	93.36%	93.50%
rep(1,ncol(train_3_5))	94.08%	94.35%	94.21%	94.41%	94.41%	94.48%	94.41%	94.21%	94.28%	94.54%	94.34 %

As shown in the tables, the initialization of theta does not matter.

4b- I changed epsilon (convergence) while kept other perimeters same as the baseline

Dataset: train_3_5

epsilon	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
0	90.88 %	91.09 %	91.29 %	91.09 %	91.15 %	91.03 %	91.14 %	91.25 %	91.42 %	91.08 %	91.14 %
0.001(baseline)	91.32%	91.16%	91.18%	91.23%	91.36%	91.20%	91.17%	91.15%	90.95%	91.32%	91.20 %
0.1	89.45 %	89.14 %	89.08 %	88.58 %	88.71 %	88.69 %	88.96 %	88.94 %	89.32 %	89.28 %	89.02 %
1	54.80 %	55.15 %	55.17 %	55.06 %	55.05 %	55.12 %	55.05 %	55.16 %	54.99 %	54.96 %	55.05 %

Dataset: test_3_5

epsilon	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
0	93.49 %	92.97 %	93.36 %	93.36 %	93.95 %	92.64 %	93.95 %	93.56 %	93.36 %	92.97 %	93.36 %
0.001(baseline)	93.62%	93.56%	93.95%	93.29%	93.89%	93.23%	93.03%	93.50%	93.56%	93.36%	93.50 %
0.1	90.73	90.73	90.53	90.34	90.01	90.66	93.82	90.01	90.01	57.59	87.44

	%	%	%	%	%	%	%	%	%	%	%
1	58.65%	57.26%	57.66%	56.67%	58.78%	58.05%	58.58%	58.71%	56.02%	57.46%	57.78%

As shown in the tables, smaller epsilon would have larger frequency.

5. Learning Curves (for both "_0_1" and "_3_5" datasets, 5 times)

Assumption: the logistic loss/negative loglikelihood is the cost function, the formula is

```
#Cost Function
cost <- function(theta, X, X_label)
{
  m <- nrow(X)
  g <- sigmoid(X%*%theta)
  J <- (1/m)*sum((-X_label*log(g)) - ((1-X_label)*log(1-g)))
  return(J)
}
```

train_0_1: Accuracy

Dataset	#1	#2	#3	#4	#5	Average
5%	0.9984202	0.9984202	0.9984202	0.9952607	0.9968404	0.99747234
10%	0.9976303	0.9968404	0.9960506	0.9960506	0.9976303	0.99684044
15%	0.9968404	0.9963138	0.9968404	0.997367	0.9968404	0.9968404
20%	0.9980261	0.9968417	0.9960521	0.9956573	0.9968417	0.99668378
25%	0.997789	0.9965256	0.9958939	0.9958939	0.9971573	0.99665194
30%	0.9971045	0.996578	0.9960516	0.9968413	0.9963148	0.99657804
35%	0.996841155	0.997518051	0.99616426	0.995938628	0.996841155	0.99666065
40%	0.996644295	0.997039084	0.99684169	0.997039084	0.996446901	0.996802211
45%	0.995788735	0.996490612	0.996139674	0.996666082	0.995964204	0.996209861
50%	0.996367656	0.995893872	0.996999368	0.993682881	0.99684144	0.995957044
55%	0.99626705	0.996697775	0.99684135	0.992534099	0.996697775	0.995807609
60%	0.996183708	0.99684169	0.996578497	0.996052112	0.996446901	0.996420582
65%	0.995869776	0.996477162	0.996477162	0.996355685	0.996963071	0.996428571
70%	0.996503102	0.996615905	0.996164693	0.995826283	0.996164693	0.996254935
75%	0.996525584	0.99684144	0.995999158	0.996736155	0.996315014	0.99648347
80%	0.996545598	0.996742992	0.996545598	0.996644295	0.996446901	0.996585077
85%	0.996562935	0.996562935	0.996284255	0.996470042	0.992754296	0.995726893
90%	0.996753816	0.996315143	0.996578347	0.996315143	0.996666082	0.996525706
95%	0.996675256	0.996592137	0.996758374	0.996758374	0.996509018	0.996658632
100%	0.996604816	0.996604816	0.996604816	0.996604816	0.996604816	0.996604816

train_0_1: Cost

Dataset	#1	#2	#3	#4	#5	Average
5%	0.002953524	0.003159094	0.002880068	0.00352131	0.006900007	0.003882801
10%	0.003072759	0.004154343	0.009038108	0.004152358	0.00684468	0.00545245
15%	0.004253495	0.005279488	0.003329692	0.003655959	0.004705436	0.004244814
20%	0.005607296	0.00341844	0.009451962	0.00533151	0.004660397	0.005693921
25%	0.004646916	0.006437089	0.005653453	0.005374341	0.005060145	0.005434389
30%	0.005649699	0.006225945	0.004847174	0.005525012	0.006776634	0.005804893
35%	0.006691876	0.007216488	0.006675374	0.007721416	0.004383996	0.00653783
40%	0.006391613	0.00645345	0.005416824	0.006237241	0.005456822	0.00599119
45%	0.005993538	0.006738372	0.006096018	0.006537064	0.007934657	0.00665993
50%	0.005040758	0.007804071	0.00710274	0.024676108	0.006692531	0.010263241
55%	0.006279386	0.005914842	0.007024034	0.025283429	0.006320289	0.010164396
60%	0.007157143	0.006438815	0.007251651	0.007170963	0.006868452	0.006977405
65%	0.008036488	0.005847511	0.006639965	0.006916144	0.006386382	0.006765298

70%	0.006934512	0.006577893	0.006560383	0.007102559	0.007522806	0.006939631
75%	0.006120903	0.006948268	0.007233643	0.007148843	0.007710366	0.007032405
80%	0.007220158	0.006480309	0.007010102	0.007149712	0.006670569	0.00690617
85%	0.007058487	0.006864242	0.007036721	0.006891354	0.024288146	0.01042779
90%	0.006979236	0.006834624	0.006669986	0.006898785	0.006951483	0.006866823
95%	0.006829718	0.007087644	0.006813261	0.007022934	0.007113146	0.006973341
100%	0.006992821	0.006992821	0.006992821	0.006992821	0.006992821	0.006992821

train_3_5: Accuracy

Dataset	#1	#2	#3	#4	#5	Average
5%	0.9254766	0.9237435	0.9116118	0.9306759	0.9306759	0.92443674
10%	0.9238095	0.9186147	0.9255411	0.9272727	0.9116883	0.92138526
15%	0.9076212	0.9099307	0.9128176	0.9110855	0.915127	0.9113164
20%	0.9177489	0.9181818	0.9164502	0.9199134	0.9155844	0.91757574
25%	0.9182825	0.9144737	0.916205	0.9106648	0.9106648	0.91405816
30%	0.9177489	0.9142857	0.9171717	0.9139971	0.9099567	0.91463202
35%	0.919119466	0.910462528	0.907989117	0.910462528	0.907494435	0.911105615
40%	0.917748918	0.906926407	0.916666667	0.911038961	0.911038961	0.912683983
45%	0.914582532	0.913620623	0.91208157	0.905540593	0.91439015	0.912043093
50%	0.910145429	0.916724377	0.914300554	0.914819945	0.915339335	0.914265928
55%	0.911695262	0.914528569	0.914528569	0.915315599	0.914213757	0.914056351
60%	0.914730919	0.912999567	0.913432405	0.910691098	0.913143846	0.912999567
65%	0.914890783	0.911694193	0.911960575	0.913425679	0.911161428	0.912626532
70%	0.911451892	0.91095721	0.911080881	0.915038338	0.912688598	0.912243384
75%	0.911588181	0.914358264	0.91089566	0.912857802	0.91089566	0.912119114
80%	0.910507521	0.911156801	0.910940374	0.913537496	0.910940374	0.911416513
85%	0.91282208	0.913025766	0.910887056	0.914146043	0.91496079	0.913168347
90%	0.911023471	0.911985379	0.910831089	0.911312043	0.912273952	0.911485187
95%	0.911609258	0.912793876	0.910971387	0.912702752	0.912156005	0.912046656
100%	0.911963296	0.911963296	0.911963296	0.911963296	0.911963296	0.911963296

train_3_5: Cost

Dataset	#1	#2	#3	#4	#5	Average
5%	0.06215226	0.08567888	0.07849483	0.07544053	0.07026688	0.074406676
10%	0.09349295	0.08290632	0.08345026	0.09248236	0.1029647	0.091059318
15%	0.1134574	0.1055544	0.1054554	0.1088755	0.1061035	0.10788924
20%	0.09824488	0.1122668	0.1038377	0.1072915	0.1112562	0.106579416
25%	0.1114228	0.1058948	0.1077876	0.1146198	0.1139088	0.11072676
30%	0.1064192	0.1131509	0.1123825	0.1160002	0.1159931	0.11278918
35%	0.109356363	0.123728666	0.125726378	0.116714464	0.121362361	0.119377647
40%	0.109895722	0.124053146	0.110825852	0.11792287	0.112374434	0.115014405
45%	0.111075341	0.113265658	0.1195911	0.125100538	0.118028761	0.11741228
50%	0.116950384	0.119072121	0.118925631	0.117011213	0.115277627	0.117447395
55%	0.12067449	0.116831194	0.116063469	0.118171471	0.118943959	0.118136917
60%	0.117851595	0.120242563	0.11977424	0.120983125	0.116640783	0.119098461
65%	0.11430137	0.117975131	0.119009676	0.120706944	0.124567604	0.119312145
70%	0.122534245	0.1234659	0.122120993	0.116444109	0.120281632	0.120969376
75%	0.120433683	0.12019883	0.121847053	0.12061527	0.12593468	0.121805903
80%	0.123854598	0.123869303	0.122213059	0.119875838	0.122341021	0.122430764
85%	0.120563679	0.120492066	0.122501558	0.120154152	0.119391366	0.120620564
90%	0.122697343	0.121103777	0.122418084	0.123546807	0.12273137	0.122499476
95%	0.12291516	0.122540633	0.123587927	0.122224186	0.121969331	0.122647447
100%	0.122826937	0.122826937	0.122826937	0.122826937	0.122826937	0.122826937

test_0_1: Accuracy

Dataset	#1	#2	#3	#4	#5	Average
5%	1	1	1	1	1	1
10%	1	1	1	1	1	1
15%	1	0.9968454	1	1	1	0.99936908
20%	0.9976359	1	1	0.9976359	0.9952719	0.99810874
25%	0.9981061	0.9981061	0.9962121	1	0.9981061	0.99810608
30%	0.9984227	0.9984227	0.9984227	0.9984227	0.9968454	0.99810724
35%	1	0.995945946	0.995945946	0.997297297	0.998648649	0.997567568
40%	0.997635934	0.998817967	0.997635934	0.998817967	0.997635934	0.998108747
45%	0.998948475	0.995793901	0.995793901	0.997896951	0.995793901	0.996845426
50%	0.996215705	0.994323557	0.998107852	0.997161779	0.997161779	0.996594134
55%	0.997420464	0.995700774	0.996560619	0.997420464	0.997420464	0.996904557
60%	0.996847912	0.998423956	0.996847912	0.997635934	0.99605989	0.997163121
65%	0.997088792	0.997088792	0.99636099	0.995633188	0.997088792	0.996652111
70%	0.996621622	0.997972973	0.995945946	0.997297297	0.99527027	0.996621622
75%	0.996847415	0.997477932	0.996216898	0.996216898	0.995586381	0.996469105
80%	0.996453901	0.996453901	0.997044917	0.996453901	0.997635934	0.996808511
85%	0.997217585	0.996661102	0.997774068	0.996104619	0.996104619	0.996772398
90%	0.99737257	0.996321597	0.996321597	0.996321597	0.996321597	0.996531792
95%	0.996515679	0.996515679	0.996515679	0.996515679	0.99701344	0.996615231
100%	0.996690307	0.996690307	0.996690307	0.996690307	0.996690307	0.996690307

test_0_1: Cost

Dataset	#1	#2	#3	#4	#5	Average
5%	0.001222669	0.00120591	0.001304057	0.001309075	0.001410811	0.001290504
10%	0.001950616	0.001845721	0.001415418	0.001825657	0.002257925	0.001859067
15%	0.001466156	0.003071264	0.001700693	0.001741428	0.001727604	0.001941429
20%	0.00218349	0.001918796	0.00180967	0.002201772	0.002951584	0.002213062
25%	0.00200367	0.00267617	0.002644379	0.002128613	0.002124168	0.0023154
30%	0.002763722	0.001791226	0.002610413	0.002091882	0.002828736	0.002417196
35%	0.002066793	0.00330508	0.003127491	0.003024685	0.002191368	0.002743084
40%	0.002990931	0.00188153	0.002685361	0.002263307	0.002979452	0.002560116
45%	0.00277392	0.003225397	0.002642278	0.002560002	0.003536467	0.002947613
50%	0.003210208	0.003527674	0.002958707	0.002552662	0.003196634	0.003089177
55%	0.003037487	0.003524424	0.00308895	0.002777159	0.002857689	0.003057142
60%	0.003304958	0.002956798	0.003289443	0.002786213	0.003543305	0.003176143
65%	0.003303611	0.002897599	0.003394986	0.003453067	0.003248772	0.003259607
70%	0.003396414	0.002954482	0.022212088	0.003071459	0.003598839	0.007046656
75%	0.002948586	0.00312866	0.003410213	0.003478277	0.003465062	0.00328616
80%	0.003314363	0.003325678	0.003065703	0.003635831	0.002932413	0.003254797
85%	0.003448197	0.003306714	0.00322329	0.003346168	0.003457873	0.003356448
90%	0.003121808	0.003466114	0.003483098	0.003526711	0.003538474	0.003427241
95%	0.003409476	0.003467345	0.00334035	0.003410721	0.003306346	0.003386848
100%	0.003414463	0.003414463	0.003414463	0.003414463	0.003414463	0.003414463

test_3_5: Accuracy

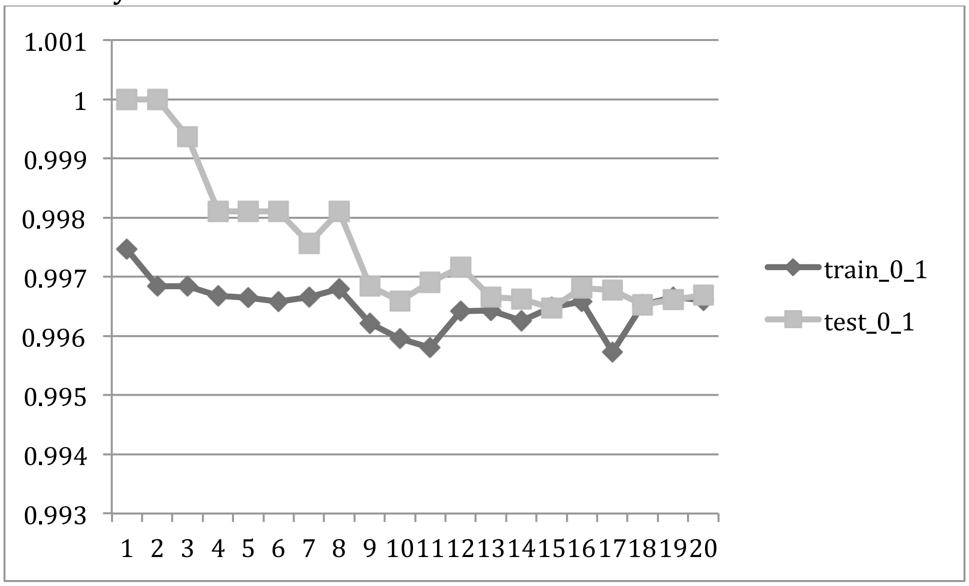
Dataset	#1	#2	#3	#4	#5	Average
5%	1	1	1	0.9894737	1	0.99789474
10%	0.9894737	0.9736842	0.9894737	0.9789474	0.9578947	0.97789474
15%	0.9438596	0.9508772	0.9578947	0.9578947	0.954386	0.95298244

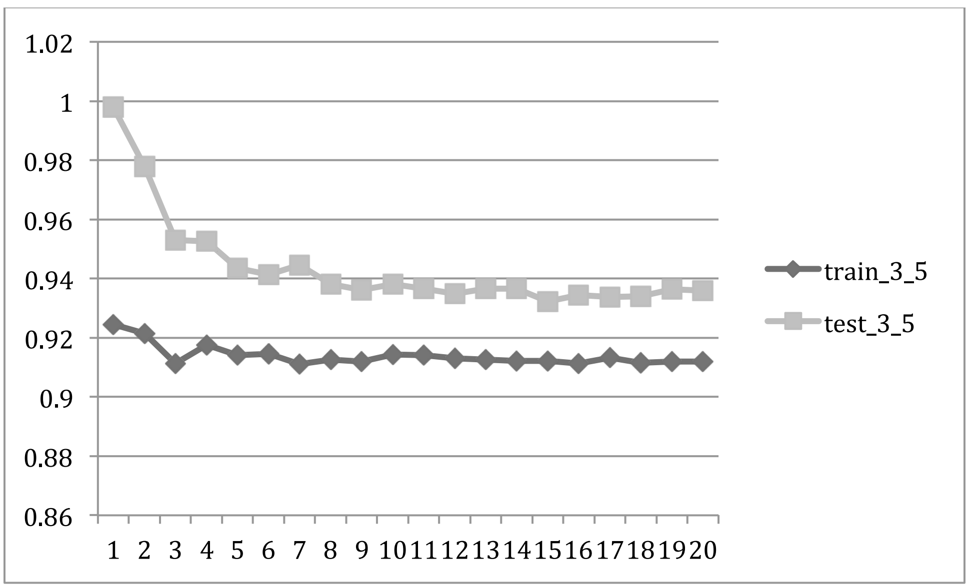
20%	0.95	0.95	0.9552632	0.9552632	0.9526316	0.9526316
25%	0.9473684	0.9347368	0.9494737	0.9452632	0.9410526	0.94357894
30%	0.9350877	0.9473684	0.9403509	0.9403509	0.9438596	0.9414035
35%	0.957894737	0.942857143	0.951879699	0.942857143	0.927819549	0.944661654
40%	0.931578947	0.940789474	0.928947368	0.942105263	0.947368421	0.938157895
45%	0.939181287	0.943859649	0.952046784	0.939181287	0.906432749	0.936140351
50%	0.939011567	0.936908517	0.936908517	0.940063091	0.937960042	0.938170347
55%	0.939770554	0.934034417	0.926386233	0.940726577	0.9416826	0.936520076
60%	0.941279579	0.933391762	0.930762489	0.93514461	0.933391762	0.93479404
65%	0.944174757	0.931229773	0.938511327	0.932038835	0.936893204	0.936569579
70%	0.938392186	0.938392186	0.937640872	0.934635612	0.933884298	0.936589031
75%	0.936886396	0.93828892	0.927068724	0.929172511	0.929873773	0.932258065
80%	0.933596318	0.933596318	0.93425378	0.934911243	0.935568705	0.934385273
85%	0.932549505	0.932549505	0.934405941	0.931311881	0.938118812	0.933787129
90%	0.931618936	0.93395675	0.934541204	0.936294565	0.933372297	0.93395675
95%	0.936323367	0.933554817	0.936877076	0.937430786	0.937430786	0.936323367
100%	0.935856993	0.935856993	0.935856993	0.935856993	0.935856993	0.935856993

test_3_5: Cost

Dataset	#1	#2	#3	#4	#5	Average
5%	0.009828374	0.01181289	0.01270638	0.01508019	0.01299931	0.012485429
10%	0.01522311	0.02557712	0.01955739	0.01639597	0.02858083	0.021066884
15%	0.04012489	0.02766972	0.0315901	0.02806267	0.04182451	0.033854378
20%	0.03882108	0.04171219	0.0378538	0.03425551	0.03211645	0.036951806
25%	0.03718196	0.06201977	0.04486811	0.05832433	0.06487328	0.05345349
30%	0.05718785	0.06249021	0.05002985	0.05304324	0.05152434	0.054855098
35%	0.049360881	0.058123183	0.053448585	0.05603713	0.065348555	0.056463667
40%	0.073777583	0.063419345	0.075184555	0.058166885	0.053530048	0.064815683
45%	0.061292508	0.071899191	0.060089282	0.07078321	0.126666671	0.078146172
50%	0.063058674	0.072097608	0.072594717	0.072726436	0.080438242	0.072183135
55%	0.074626771	0.077971194	0.074633158	0.078389094	0.067745748	0.074673193
60%	0.073961236	0.075949375	0.080314729	0.073735448	0.079778634	0.076747884
65%	0.077117899	0.073061506	0.080079913	0.082132603	0.076917372	0.077861859
70%	0.080258755	0.074769875	0.079599998	0.076260362	0.08453348	0.079084494
75%	0.07915951	0.081718571	0.083366253	0.084166234	0.080599776	0.081802069
80%	0.080578371	0.080158361	0.081750889	0.076496402	0.083979454	0.080592695
85%	0.081400265	0.08163804	0.086626605	0.084120318	0.076537925	0.082064631
90%	0.085020539	0.083754799	0.08585587	0.086124619	0.08720838	0.085592841
95%	0.085964883	0.0872525	0.085160297	0.083246229	0.08135366	0.084595514
100%	0.086086201	0.086086201	0.086086201	0.086086201	0.086086201	0.086086201

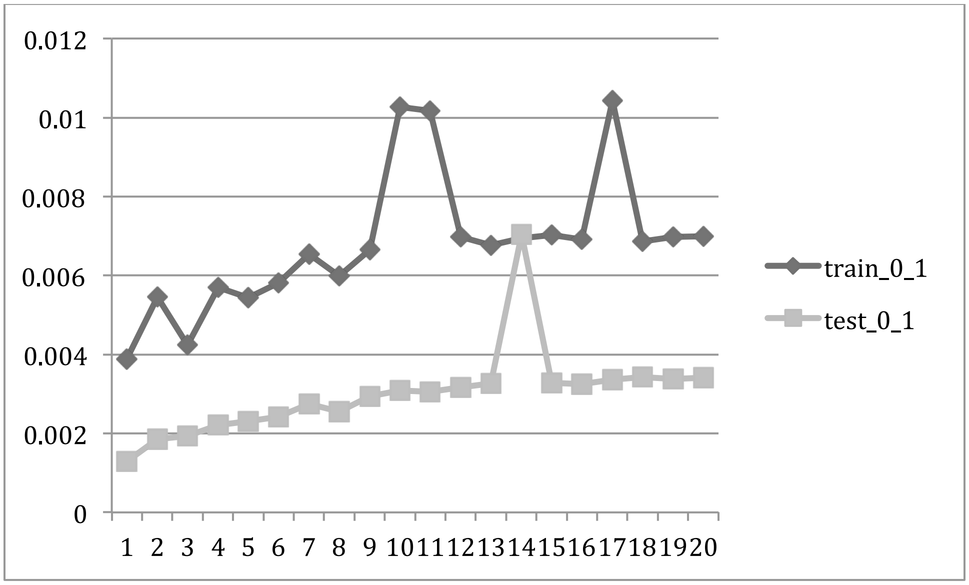
Accuracy:

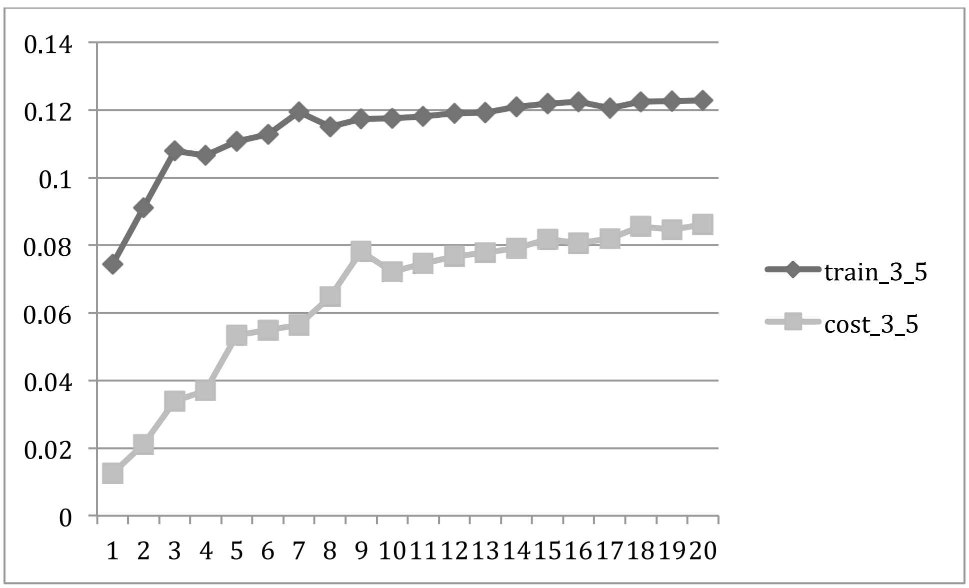




As shown in the two charts, the accuracy decreases as the sample size increases.

Cost:





As shown in the charts, the negative log-likelihood increase as the sample size increases.