Comp560 Jiabing Song/Yingying Wu

HW4 Report

a.Computering Features

1) Tiny Image: First we convert our image size to 32x32 with 3 channels by calling python image library function thumbnail. Then we get the data for each pixel in the image by calling image.getdata() and we parse the data in to a list.

2) Color Histogram: First we create three arrays of size 8, each represeing a color bin. We divide the array into differnt range, for example arr[0] represents value ranging from 0-32. If the value of a pixel falls into certain range, we increment that slot by one. If the image mode isn't RGB, we simply copy the value from L into three color bins. In the end we concatenate three arrays into one and return it.

b. Training

For this part first we introduce LibSVM library into our codes. Since we are doing a binary classification, we maintain a list of label containg only 1 and -1. Our data list consistss of random images from the training set.

We set the parameter C for linear kernel SVMs. We learn that a larger value of C will result in a smaller hyperplane trying to classify everything right. However, this leads to the risk of losing the generalization while running test list. Balancing between the trade off, we decided to set C to 2 for the best performance.

For RBF SVMs, we tune the parameters C and g on our hold lists. When gamma is too large, the radius of the model will be too small for generailzation. If gamma is too small, the model becomes too constrained. By setting a larger C, we are able to receive a good regularizer.The result is particularly bad for overall accuracy of argmax. Because the model tends to classify everything into neither label we increase C to 10000, and set g to 0.00001.

We read data into SVM and build the model all by calling built-in function svm_problem and svm_training and run the kernel on test sets.

c. Testing

First we save four models for each class for future reference. The built-in function svm_predict returns a list of predicted label and a list of predict probability.

Results from testing are attached below.

Performance

Data below has 5 sections. The first four are the performance and result of each class prediction and the fifth onw is the overall prediction by using argmax to the result of the four classes ones.

TinyImage-Linear

predict label with None

```
Model supports probability estimates, but disabled in predicton.
Accuracy = 95.7035% (1136/1187) (classification)
hobo:  [1.0, -1.0, -1.0, 1.0, -1.0, 1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, 1.0, -1.0, 1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, -1.0
hobo_val:  [[0.09602404007714593], [-3.342594536097203], [-3.615957671051512], [4.228667227295353], [-0.6604818374420001], [3.800186563194621], [-2.935280187831574], [-2.6
[1, -1, -1, 1, -1, 1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1, 1, -1, 1, 1, -1,
==========
Model supports probability estimates, but disabled in predicton.
Accuracy = 91.0699% (1081/1187) (classification)
c:  [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0, -1.0, 1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0
c_val:  [[-1.0021795384613528], [-4.967441316992383], [-3.842726667308445], [-2.8648914668043277], [-0.9464691997260717], [-1.263362850122367], [2.9766259900584613], [2.3
[-1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1,
==========
Model supports probability estimates, but disabled in predicton.
Accuracy = 90.6487% (1076/1187) (classification)
f:  [-1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0, 1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, 1.0, -1.0, 1.0, -1.0, 1.0, -1.0,
f_val:  [[-5.353525875363928], [6.380115239814117], [-0.40478509898754567], [-6.365790198243545], [-3.718835705744212], [-9.8483451974666], [-2.6381124977022097], [-2.6646
[-1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1, -1, 1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, 1, -1, 1, -1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1,
==========
Model supports probability estimates, but disabled in predicton.
Accuracy = 91.6596% (1088/1187) (classification)
p:  [1.0, -1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0, -1.0, -1.0, 1.0, -1.0, -1.0, 1.0, 1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, -1.0, 1.0, -1
p_val:  [[1.2696038203657176], [-3.275665008802945], [1.5891677868410747], [-2.3594917459629974], [-2.0280938042955645], [-4.3032398650151995], [-2.8905548249450126], [-2.
[-1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1,
========== argmax ==========
predict result:  [4, 3, 4, 1, 0, 1, 2, 2, 1, 4, 4, 1, 2, 4, 3, 3, 3, 4, 4, 3, 1, 2, 0, 3, 1, 3, 4, 3, 4, 3, 3, 4, 1, 3, 2, 1, 3, 1, 4, 0, 3, 1, 3, 1, 1, 2, 4, 3, 2, 3, 1,
true value:      [1, 3, 3, 1, 3, 1, 2, 2, 1, 4, 4, 1, 2, 4, 3, 3, 3, 4, 1, 3, 1, 2, 2, 3, 1, 3, 4, 3, 4, 3, 3, 4, 1, 3, 3, 1, 3, 2, 4, 4, 3, 1, 3, 1, 1, 2, 4, 4, 2, 3, 1,
accuracy of argmax:  83.4035383319
```

predict label without None

```
Model supports probability estimates, but disabled in predicton.
Accuracy = 95.7035% (1136/1187) (classification)
hobo:  [1.0, -1.0, -1.0, 1.0, -1.0, 1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, 1.0, -1.0, 1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, -1.0
hobo_val:  [[0.09602404007714593], [-3.342594536097203], [-3.615957671051512], [4.228667227295353], [-0.6604818374420001], [3.800186563194621], [-2.935280187831574], [-2.6
[1, -1, -1, 1, -1, 1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1, 1, -1, 1, 1, -1,
==========
Model supports probability estimates, but disabled in predicton.
Accuracy = 91.0699% (1081/1187) (classification)
c:  [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0, -1.0, 1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0
c_val:  [[-1.0021795384613528], [-4.967441316992383], [-3.842726667308445], [-2.8648914668043277], [-0.9464691997260717], [-1.263362850122367], [2.9766259900584613], [2.3
[-1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1,
==========
Model supports probability estimates, but disabled in predicton.
Accuracy = 90.6487% (1076/1187) (classification)
f:  [-1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0, 1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, 1.0, -1.0, 1.0, -1.0, 1.0, -1.0,
f_val:  [[-5.353525875363928], [6.380115239814117], [-0.40478509898754567], [-6.365790198243545], [-3.718835705744212], [-9.8483451974666], [-2.6381124977022097], [-2.6646
[-1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1, -1, 1, -1, 1, -1, 1, -1, 1, 1, -1, -1, 1, 1, -1, 1, -1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1,
==========
Model supports probability estimates, but disabled in predicton.
Accuracy = 91.6596% (1088/1187) (classification)
p:  [1.0, -1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0, -1.0, -1.0, 1.0, -1.0, -1.0, 1.0, 1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, -1.0, 1.0, -1
p_val:  [[1.2696038203657176], [-3.275665008802945], [1.5891677868410747], [-2.3594917459629974], [-2.0280938042955645], [-4.3032398650151995], [-2.8905548249450126], [-2.
[-1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1,
========== argmax ==========
predict result:  [4, 3, 4, 1, 1, 1, 2, 2, 1, 4, 4, 1, 2, 4, 3, 3, 3, 4, 4, 3, 1, 2, 2, 3, 1, 3, 4, 3, 4, 3, 3, 4, 1, 3, 2, 1, 3, 1, 4, 4, 3, 1, 3, 1, 1, 2, 4, 3, 2, 3, 1,
true value:      [1, 3, 3, 1, 3, 1, 2, 2, 1, 4, 4, 1, 2, 4, 3, 3, 3, 4, 1, 3, 1, 2, 2, 3, 1, 3, 4, 3, 4, 3, 3, 4, 1, 3, 3, 1, 3, 2, 4, 4, 3, 1, 3, 1, 1, 2, 4, 4, 2, 3, 1,
accuracy of argmax:  86.352148273
```

confusion matrix

predict label with None

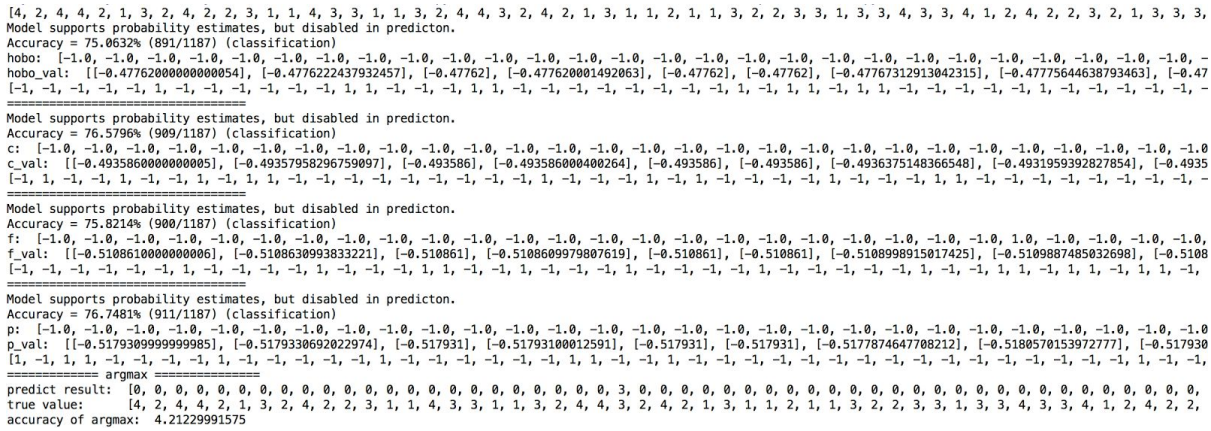
true->	hobo	clutch	flats	pumps
hobo	**288**	16	5	9
clutch	7	**254**	16	9
flats	6	24	**247**	34
pumps	9	10	17	**236**

predict label without None

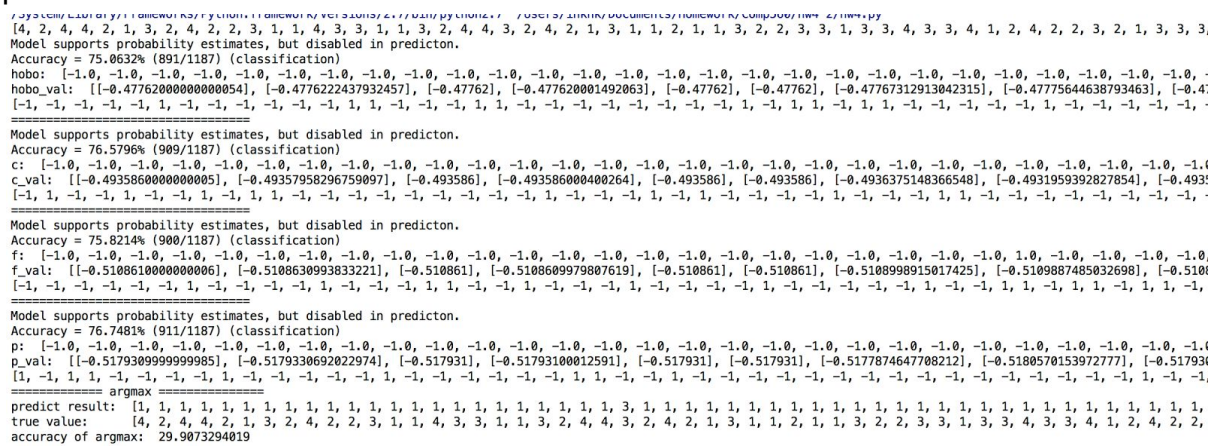
true->	hobo	clutch	flats	pumps
hobo	**282**	10	3	6
clutch	6	**242**	15	9
flats	4	21	**238**	29
pumps	6	9	15	**228**
None	12	22	14	16

TinyImage-RBF

predict label with None



predict label without None



confusion matrix

predict label with None

true->	hobo	clutch	flats	pumps
hobo	**7**	0	0	0
clutch	0	**16**	0	0
flats	0	0	**13**	0
pumps	296	278	287	**290**

predict label with None

true->	hobo	clutch	flats	pumps
hobo	**302**	270	286	275
clutch	0	**24**	0	0
flats	0	0	**14**	0
pumps	1	0	0	**15**

Color Histogram-Linear

Failed cause receive WARNING: reaching max number of iterations

Color Histogram-RBF

```
Model supports probability estimates, but disabled in predicton.
Accuracy = 76.2426% (905/1187) (classification)
hobo:  [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -
hobo_val:  [[-0.48724572305271474], [-0.48724572305271474], [-0.48724572305271474], [-0.48724572305271474], [-0.48724572305271474], [-0.48724572305273467], [-0.48724572305
[-1, -1, -1, -1, 1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -
==================================
Model supports probability estimates, but disabled in predicton.
Accuracy = 74.3892% (883/1187) (classification)
c:  [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0
c_val:  [[-0.5060781384373819], [-0.5060781384373819], [-0.5060781384373819], [-0.5060781384373819], [-0.5060781384373819], [-0.5060781384374011], [-0.5060781384373819], [
[-1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, 1, 1, -1, 1,
==================================
Model supports probability estimates, but disabled in predicton.
Accuracy = 76.4111% (907/1187) (classification)
f:  [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0,
f_val:  [[-0.5094736065018657], [-0.5094736065018657], [-0.5094736065018657], [-0.5094736065018657], [-0.5094736065018657], [-0.5094736065018848], [-0.5094736065018657], [
[1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, 1, 1, 1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, 1, -1, -
==================================
Model supports probability estimates, but disabled in predicton.
Accuracy = 75.7372% (899/1187) (classification)
p:  [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0
p_val:  [[-0.4972021120621171], [-0.4972021120621171], [-0.4972021120621171], [-0.4972021120621171], [-0.4972021120621171], [-0.4972021120620589], [-0.4972021120621171], [
[-1, 1, 1, 1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -
============= argmax ===============
predict result:  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
true value:      [3, 4, 4, 4, 1, 4, 3, 3, 1, 2, 1, 3, 3, 3, 4, 2, 2, 3, 1, 2, 1, 3, 4, 4, 3, 1, 2, 2, 4, 3, 3, 2, 3, 2, 3, 4, 1, 3, 4, 2, 2, 4, 2, 2, 3, 2, 4, 2, 4, 3, 3,
accuracy of argmax:  27.2114574558
```

confusion matrix

true->	hobo	clutch	flats	pumps
hobo	**290**	302	277	284
clutch	0	**8**	0	0
flats	0	0	**8**	1
pumps	0	0	0	**17**

Overall we find that when the predict label don’t include None (test image belongs to neither of the class) class, the overall accuracy is higher.