Homework 1

Name: Dejun Qian
email: electronseu@gmail.com

Step 1

The results for the pruned (“simplified”) trees are shown in the following table.

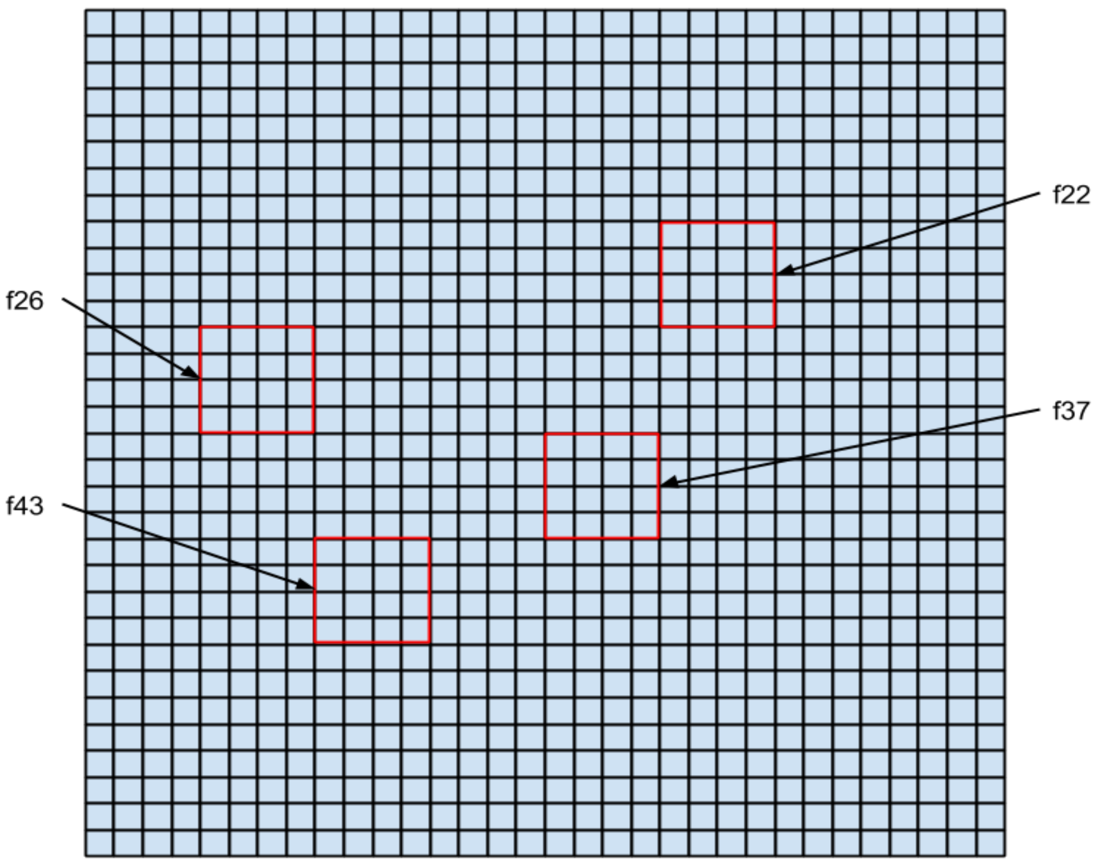
attribute at root of tree		f37
% accuracy on test set		1-21% = 79%
Number of type 1 errors (false positives)	0	2+1+1=4
	1	11+7+10+1+7+14+2=52
	2	1+5+2+5+2+8=23
	3	2+1+1+2=6
	4	2+5+1+7+4+3=22
	5	11+2+1+3=17
	6	4+1+1=6
	7	1+2+6+3+4=16
	8	1+8+3+3+2+2+1=20
	9	4+4+6+7+1+6+2=30
Number of type 2 errors (false negatives)	0	1+2+1=4
	1	2+5+5+11+4+1+8+4=40
	2	11+2+1+2+3+4=23
	3	2+2+6+3+6=19
	4	1+7+1+1+3+2+7=22
	5	10+5+1+1+2+1=20
	6	1+2+7=10
	7	7+1+6=14
	8	1+14+8+1+4+2=30
	9	2+2+3+3+4=14

Step 2

A. The first six lines of the “simplified” decision tree is shown below.

```
f37 <= 0 :
|  f43 <= 4 :
|  |  f22 <= 7 : 5 (11.0/1.3)
|  |  f22 > 7 :
|  |  |  f26 <= 3 : 9 (17.0/2.5)
|  |  |  f26 > 3 : 5 (2.0/1.0)
```

Features f37, f43, f22 and f26 is shown on a 32x32 grid as following.



By comparing the shape of “5” and “9”, we can find that the biggest difference is that the upper-right side of the shape is closed for “9” and open for “5”. This means the value of feature f22 should be large for “9” and small for “5”. Another difference between “5” and “9” is that the uppper-left part is round for “9” and acute for “5”, this makes the value of feature f26 large for “5” and small for “9”. Because the lower side of the two number is similar, so the value of feature f37 and f43 have no difference, and they cann't be used to seperate “5” and “9”.

In general, this part of the decision tree works as follows. The features f37 and f43 is not used to seperate “5” and “9”. The feature f22 is used first. If f22 is small, then the number is “5”. Otherwise, we look at f26, if it is small, then the number is “9”, otherwise the number is “5”.

B. Most frequently confused digit for each digit is shown below,

0	1	2	3	4	5	6	7	8	9
1	8	8	2 and 9	1	1	1	3	1	4

No. This result is not what I expected. This method confuses 1 with most of the digits – “0”, “4”, “5”, “6”, and “8”. As “1” is slim, and the others are fat, I didn't expect the confusion of this kind. However, other confusions do make sense to me, as they look like each other at some level.

Step 3

To generate the small training set, I go through the original data, get the first 50 sample for each digit, and write them into the new file.

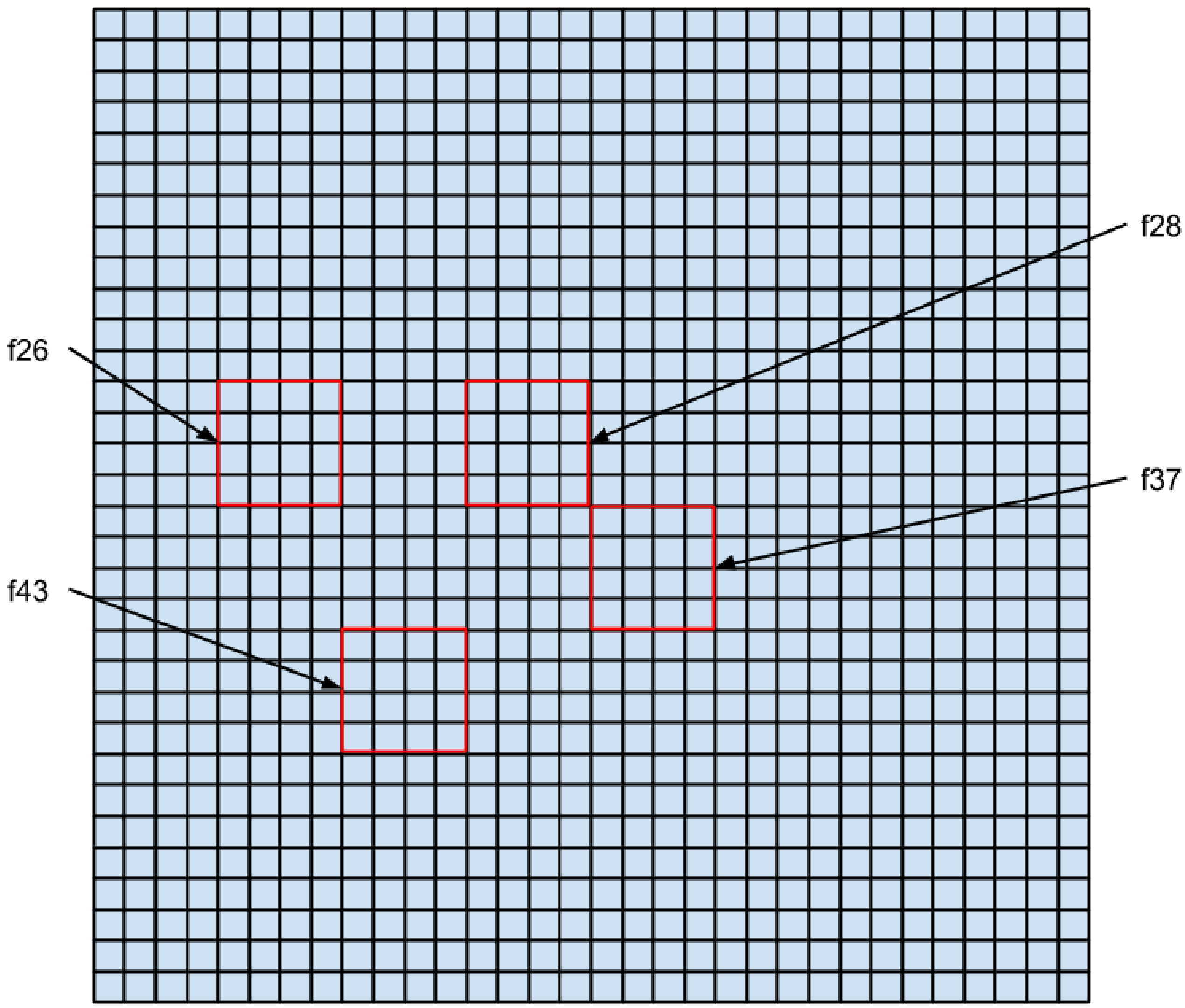
The results for the pruned (“simplified”) trees are shown in the following table.

<table>
<tr><td colspan="2">attribute at root of tree</td><td>f37</td></tr>
<tr><td colspan="2">% accuracy on test set</td><td>1-25.4% = 74.6%</td></tr>
<tr><td rowspan="10">Number of type 1 errors (false positives)</td><td>0</td><td>2+5+1+2+2=12</td></tr>
<tr><td>1</td><td>1+1+1+5+9+3+3+5=28</td></tr>
<tr><td>2</td><td>2+1+6+4+2=15</td></tr>
<tr><td>3</td><td>9+2+1+9+1+2+4=28</td></tr>
<tr><td>4</td><td>5+7+1+1+4+1+2=21</td></tr>
<tr><td>5</td><td>1+5=6</td></tr>
<tr><td>6</td><td>0</td></tr>
<tr><td>7</td><td>1+6+5+2=14</td></tr>
<tr><td>8</td><td>3+11+18+4+3+2+4+9+3=57</td></tr>
<tr><td>9</td><td>3+3+4+10+8+10+12+6=56</td></tr>
<tr><td rowspan="10">Number of type 2 errors (false negatives)</td><td>0</td><td>1+5+3+3=12</td></tr>
<tr><td>1</td><td>2+9+7+1+11+3=33</td></tr>
<tr><td>2</td><td>1+2+1+18+4=26</td></tr>
<tr><td>3</td><td>1+6+4+10=21</td></tr>
<tr><td>4</td><td>2+5+1+1+5+3+8=25</td></tr>
<tr><td>5</td><td>5+9+9+1+2+10=36</td></tr>
<tr><td>6</td><td>1+3+1+4+4=13</td></tr>
<tr><td>7</td><td>6+1+9+12=28</td></tr>
<tr><td>8</td><td>2+3+4+2+1+6=18</td></tr>
<tr><td>9</td><td>2+5+2+4+2+5+2+3=25</td></tr>
</table>

A. The first six lines of the "simplified" decision tree is shown below.

```
f37 <= 0 :
|   f43 <= 4 :
|   |   f26 <= 2 : 9 (5.0)
|   |   f26 > 2 : 5 (6.0)
|   f43 > 4 :
|   |   f28 <= 10 : 0 (48.0/1.0)
```

Features f37, f43, f28 and f26 is shown on a 32x32 grid as following.



The feature f43 is used to differenciate "0" with "5" and "9". The reason why this work is because the left-bottom side is closed for "0", while open for "5" and "9". The feature f43 with larger value will suggest "0", while small value will suggest "5" and "9".

The feature f26 is further used to differenciate "5" and "9". The reason for this strategy is that the left middle is acute for "5", and round for "9". The feature f26 with a large value suggests "5", while small value suggest "9".

B. Most frequently confused digit for each digit is shown below,

0	1	2	3	4	5	6	7	8	9
5	5	7	1 and 5	1	9	0	3	2	7

The result makes more sense to me than the previous one.

The most confused digits do has some in common with the output. For example, “7” is confused with “9”, “9” is confused with “5”. These digits have similar shape.

Most importantly, no digit is get confused with “6”, which means no digit other than “6” is recoganized as “6”.

Step 4

To generate the small training set, I use optdigits.data as input, randomly add some noise by change 5% of the data.

The results for the pruned (“simplified”) trees are shown in the following table.

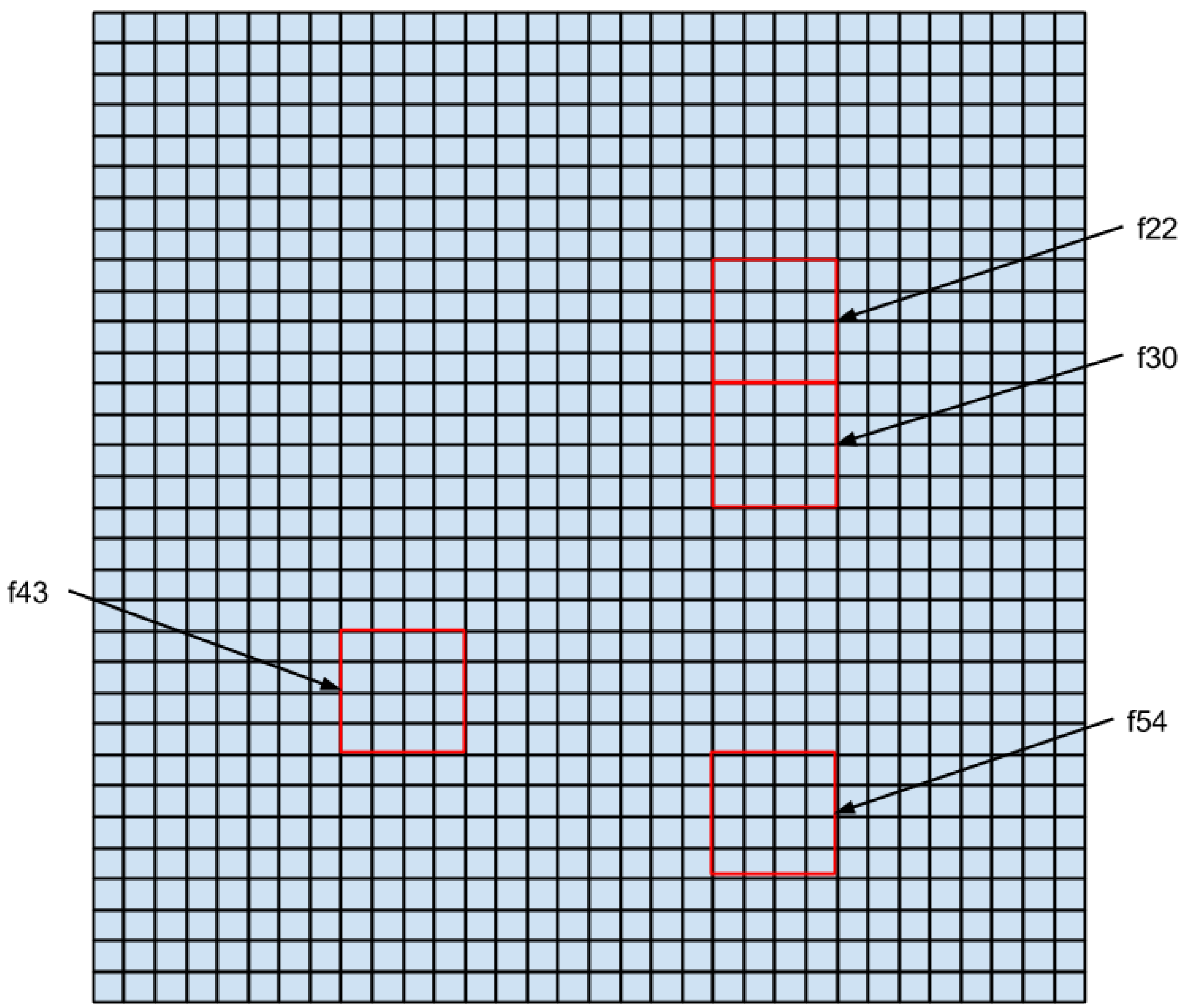
attribute at root of tree		f43
% accuracy on test set		1-24.2% = 75.8%
Number of type 1 errors (false positives)	0	3+2+1+5=11
	1	6+7+7+6+2+4+16+10=58
	2	4+2+1+1+6=14
	3	1+1+2+1+7=12
	4	5+3+3+3+4=18
	5	1+11+8+5+2+1+4+1=33
	6	2+2+2+1+1=8
	7	2+5+7+11=25
	8	2+8+4+4+1+3+1=23
	9	3+6+4+8+1+2=24
Number of type 2 errors (false negatives)	0	5+1+2+2=10
	1	4+1+3+11+2+2+8+3=34
	2	6+1+8+5+4+6=30
	3	3+7+2+5+4+4=25
	4	7+2+2+7+1+8=27
	5	6+1+2+1+3+1=14
	6	2+2+1+3+1=9

	7	4+1+2=7
	8	1+16+6+1+3+4+1=32
	9	5+10+7+4+1+11=38

A.The first six lines of the “simplified” decision tree is shown below.

f43 <= 6 :
| f54 <= 2 :
| | f30 <= 3 :
| | | f22 <= 1 : 5 (16.0/2.5)
| | | f22 > 1 : 1 (2.0/1.8)
| | f30 > 3 :

Features f54, f43, f30 and f22 is shown on a 32x32 grid as following.



Basically it says that if f43, f54 and f30 are small, then the digit is “5” when f22 is small, and “1” when f22 is large. I can't figure why this works. Probably because the noise makes the decision tree bad.

B. Most frequently confused digit for each digit is shown below,

0	1	2	3	4	5	6	7	8	9
9	8	8	9	0	1	0, 1 and 4	9	1	4

Some of the results make sense to me, while some not. For example, “9” is confused with “0”, “9” is confused with “3”. These are easy to understand as they look similar. However, others are not the case, like “1” is confused with “5”, “8” is confused with “1”.

Step 5

Summary: Generally speaking, decision tree work very good for recognizing handwritting digits. The size of the training data do have some effect on the result. As shown in this experiment, reduce training data leads to lower accuracy. However, the effect is not obvious, maybe it's because the training set still have a reasonable size even after the size is reduced.

Noise also makes the accuracy lower as shown in the experiment result. However, we didn't see obvious effect. This means decision tree can work with reasonable noise.