Week 06 Report (19 Aug, 23 Aug) - Guangyao Xu

1、Work

1.1 Debug the algorithm of clustering

I use Affinity Propagation Clustering to cluster all of the correct programs. The advantage of Affinity Propagation Clustering is we don't need to decide the number of clusters we want and just focus on result of it. And you need to justify two parameters:

```
damping : float, optional, default: 0.5
    Damping factor (between 0.5 and 1) is the extent to
    which the current value is maintained relative to
    incoming values (weighted 1 - damping). This in order
    to avoid numerical oscillations when updating these
    values (messages).
```

```
preference : array-like, shape (n_samples,) or float, optional
    Preferences for each point - points with larger values of
    preferences are more likely to be chosen as exemplars. The number
    of exemplars, ie of clusters, is influenced by the input
    preferences value. If the preferences are not passed as arguments,
    they will be set to the median of the input similarities.
```

The default value is 0.5 and None separately.

1.2 Write programs which support both label or non-label

I write a program which support both label or non-label.

For example, I extract the operator code from the sentence like:

```
add nsw i32 %29, 32
```

And the corresponding operator code is `add`

And I cluster two results depending on the label and not.

You can find the result on the github:

https://github.com/BryceTsui/UIUCSummerResearch/tree/master/GED/cluster_result

```
classify - the matching result of incorrect programs to correct programs'clusters
correct - the clustering result of correct programs
incorrect - the clustering result of incorrect programs(it is not that important, you can ignore
this dir currently)
```

1.3 Match the incorrect programs to clusters which generated from correct programs

This part can be summerized by such following steps:

1. get cluster from the correct programs by the similarity
2. for each incorrect program, compare the similarity between it and the center points of all of the clusters, and find the most similar one
3. put the incorrect programs to the corresponding cluster

For example, there are two clusters of CFG, and the following is their center point program:

```
#include <stdio.h>
#include <math.h>
int main()
{
 int add, sum, final;
 sum = 0;
 printf("Enter an abitrarily long string, ending with carriage return > ");
 while(add != '\n')
 {
   add = getchar();
   if(add != '\n')
     sum = add + sum;
 }
 final = (sum % 64) + 32;
 printf("Check sum is ");
 putchar(final);
 printf("\n");
 return 0;
}
```

```
#include <stdio.h>
#include <math.h>
int main()
{
  char stuff;
  int sum=0;
  printf("Enter an abitrarily long string, ending with carriage return > ");
  scanf("%c", &stuff);//**/
  while(stuff!='\n') //**/
  {
  sum+=(int)stuff; //**/
  scanf("%c", &stuff);//**/
  }
  sum=sum%64+ (int)' '; //**/
  printf("Check sum is %c\n", (char) sum); //**/
  return 0;
}
```

They are different, because one contains if-else and another not.

And there are two incorrect programs:

Incorrect 1(it is incorrect, because it set the type of input to char, and it will overflow):

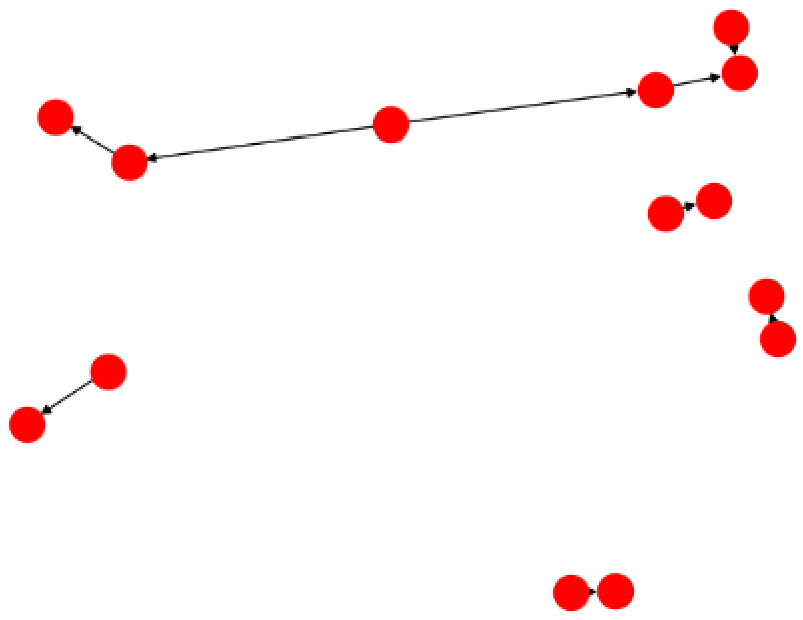
```
#include <stdio.h>
int
main(void)
{
   char input;
   int a, checksum;
   printf("Enter an abitrarily long string, ending with carriage return > ");
   scanf("%c", &input);
   a = 0;
   do {
       input = (int)input + a;
       scanf("%c", &a);
   } while (a != '\n');
   checksum = ((int)input % 64) + (int)' ';
   printf("Check sum is %c\n", checksum);
   return(0);
}
```

incorrect 2:

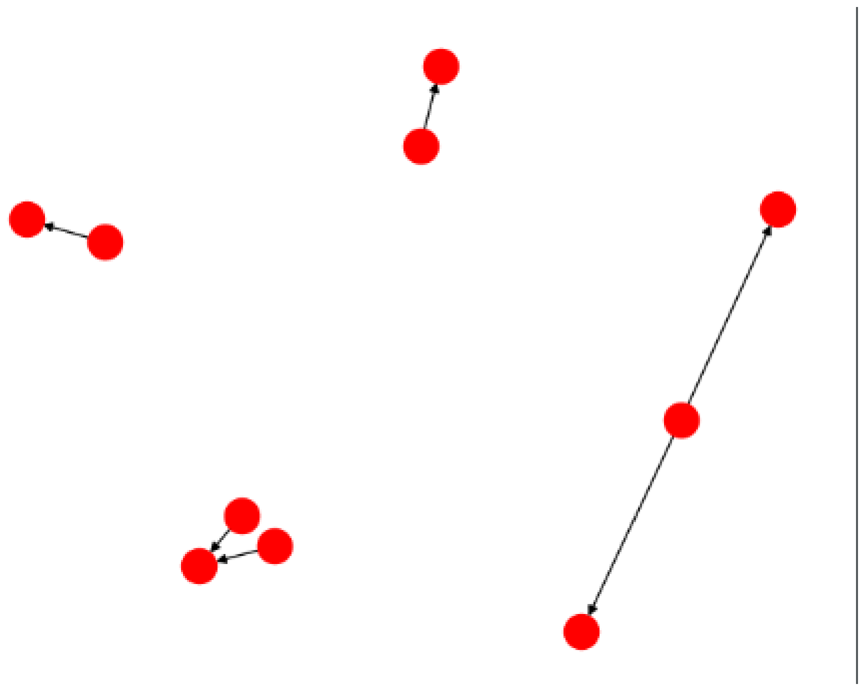
```
#include <stdio.h>
#include <math.h>
int main()
{
 int add, sum;
 sum = 0;
 printf("Enter an abitrarily long string, ending with carriage return > ");
 while(add != '\n')
 {
   add = getchar();
   if(add != '\n')
     sum = add + sum;
 }
 sum = sum - 32;
 printf("Check sum is ");
 putchar(sum);
 printf("\n");
 return 0;
}
```

1.4 Visualize the graphs like CFG,DD and CD

To make it more visualable, I write a program to visualize the graphs including CFG,CD and DD. And the graphs picture looks like this:



And

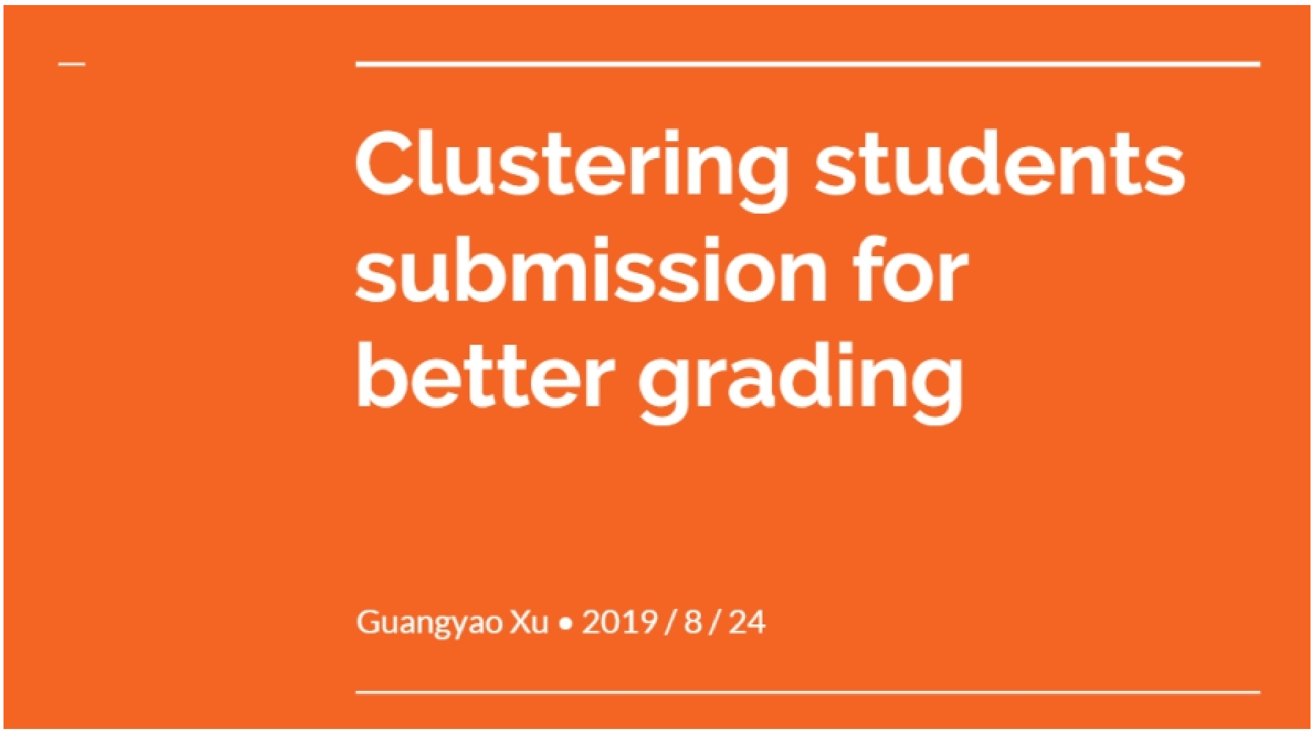


1.5 Prepare Slides and presentation

I prepare slides and presentation to report the work I have done till now, and then I meet with Angello at Friday.

And you can find the whole slides on my google doc:

(https://drive.google.com/drive/folders/1r5a9vlcOXQFLolX1s1EGg2SzsZTGhcD3?usp=sharing)



Clustering students submission for better grading

Guangyao Xu • 2019 / 8 / 24