CSCI 720 Big Data Analytics HW07 Results

Student: Guo, Zizhun & Qian, Martin
Submission: Apr/1st/2020
Due Date: Apr/1st/2020 11:59 PM

Part 1 (Guo, Zizhun): Using Cross-Correlation for Feature Rejection and Selection

Codes:

```
import pandas as pd

# rule out ID column
column_names = df.columns[1:column_names_size]
df_new = df.iloc[:, 1:column_names_size]

# use panda package to get correlation matrix
corr = df_new.corr()
```

The cc matrix:

	Beans	Bread	Cerel	ChdBby	Chips	Corn	Eggs	Fish	Fruit	Meat	Milk	Pepper	Rice	Salza	Sauce	Soda	Tomato	Tortya	Vegges	YogChs
Beans	1.000000	-0.119332	-0.180597	0.025656	-0.062487	0.430382	-0.511366	-0.370691	-0.008510	-0.632335	-0.082067	0.277554	0.411067	0.519287	0.520403	-0.403328	0.221605	0.416137	0.263199	0.555409
Bread	-0.119332	1.000000	0.444009	0.005108	0.118074	-0.133571	0.002145	-0.065627	0.042469	0.010868	0.439962	0.202720	-0.080480	0.020554	-0.076847	-0.056733	-0.274576	-0.205630	-0.025703	-0.070575
Cerel	-0.180597	0.444009	1.000000	0.013300	0.343644	-0.307890	-0.014874	-0.035770	-0.011541	-0.020136	0.287522	0.014198	-0.363120	0.076334	-0.042312	0.270626	-0.474217	-0.347393	-0.396666	-0.409933
ChdBby	0.025656	0.005108	0.013300	1.000000	-0.043383	-0.038147	-0.015607	0.034767	-0.016260	-0.043050	-0.035475	-0.051206	0.017592	-0.015251	0.035463	-0.030395	-0.023454	-0.048586	0.000585	-0.027245
Chips	-0.062487	0.118074	0.343644	-0.043383	1.000000	0.283476	0.189594	-0.313534	-0.006616	0.296488	0.046260	0.395141	-0.508947	0.489282	-0.458284	0.536912	0.207363	0.365681	-0.541429	-0.294565
Corn	0.430382	-0.133571	-0.307890	-0.038147	0.283476	1.000000	0.108008	-0.251822	0.021371	0.021903	0.226857	0.718210	0.348095	0.630477	-0.290959	-0.322733	0.692017	0.832840	0.320751	0.529213
Eggs	-0.511366	0.002145	-0.014874	-0.015607	0.189594	0.108008	1.000000	0.449056	-0.043175	0.537484	0.241591	0.110031	-0.041513	-0.041092	-0.686711	0.069131	0.252594	0.126446	-0.032883	-0.273479
Fish	-0.370691	-0.065627	-0.035770	0.034767	-0.313534	-0.251822	0.449056	1.000000	-0.044383	0.077293	0.155617	-0.337757	0.221456	-0.269813	-0.143706	-0.202991	-0.100718	-0.285389	0.127811	-0.346818
Fruit	-0.008510	0.042469	-0.011541	-0.016260	-0.006616	0.021371	-0.043175	-0.044383	1.000000	-0.002114	0.001482	0.000416	-0.020624	-0.010956	0.009706	-0.001972	-0.039547	-0.031387	0.004984	0.019034
Meat	-0.632335	0.010868	-0.020136	-0.043050	0.296488	0.021903	0.537484	0.077293	-0.002114	1.000000	0.090413	0.104444	-0.358994	-0.222987	-0.729068	0.389524	0.196344	0.059180	-0.210369	-0.268943
Milk	-0.082067	0.439962	0.287522	-0.035475	0.046260	0.226857	0.241591	0.155617	0.001482	0.090413	1.000000	0.487064	0.247173	0.222134	-0.334728	-0.417346	0.013328	0.108244	0.300422	0.142018
Pepper	0.277554	0.202720	0.014198	-0.051206	0.395141	0.718210	0.110031	-0.337757	0.000416	0.104444	0.487064	1.000000	0.190006	0.625752	-0.396704	-0.269767	0.507050	0.678290	0.232638	0.434547
Rice	0.411067	-0.080480	-0.363120	0.017592	-0.508947	0.348095	-0.041513	0.221456	-0.020624	-0.358994	0.247173	0.190006	1.000000	0.171912	0.212541	-0.802872	0.255934	0.261671	0.690419	0.564325
Salza	0.519287	0.020554	0.076334	-0.015251	0.489282	0.630477	-0.041092	-0.269813	-0.010956	-0.222987	0.222134	0.625752	0.171912	1.000000	-0.097773	-0.169133	0.403100	0.643361	0.016341	0.254567
Sauce	0.520403	-0.076847	-0.042312	0.035463	-0.458284	-0.290959	-0.686711	-0.143706	0.009706	-0.729068	-0.334728	-0.396704	0.212541	-0.097773	1.000000	-0.209635	-0.373380	-0.313939	0.123691	0.198444
Soda	-0.403328	-0.056733	0.270626	-0.030395	0.536912	-0.322733	0.069131	-0.202991	-0.001972	0.389524	-0.417346	-0.269767	-0.802872	-0.169133	-0.209635	1.000000	-0.166914	-0.204660	-0.756236	-0.591682
Tomato	0.221605	-0.274576	-0.474217	-0.023454	0.207363	0.692017	0.252594	-0.100718	-0.039547	0.196344	0.013328	0.507050	0.255934	0.403100	-0.373380	-0.166914	1.000000	0.737151	0.227902	0.382737
Tortya	0.416137	-0.205630	-0.347393	-0.048586	0.365681	0.832840	0.126446	-0.285389	-0.031387	0.059180	0.108244	0.678290	0.261671	0.643361	-0.313939	-0.204660	0.737151	1.000000	0.217949	0.469915
Vegges	0.263199	-0.025703	-0.396666	0.000585	-0.541429	0.320751	-0.032883	0.127811	0.004984	-0.210369	0.300422	0.232638	0.690419	0.016341	0.123691	-0.756236	0.227902	0.217949	1.000000	0.605484
YogChs	0.555409	-0.070575	-0.409933	-0.027245	-0.294565	0.529213	-0.273479	-0.346818	0.019034	-0.268943	0.142018	0.434547	0.564325	0.254567	0.198444	-0.591682	0.382737	0.469915	0.605484	1.000000

Questions:

a. Which two attributes are most strongly cross-correlated with each other? (¼)

> Corn & Tortya

b. Which attribute is fish most strongly cross-correlated with? (¼)

> Eggs

c. Which attribute is meat most strongly cross-correlated with? (¼)

> Sauce

d. Which attribute is beans most strongly cross-correlated with? (¼)

> Meat

e. Which one attribute is least correlated with all other attributes? (¼)

> Milk

f. Which second attribute is least correlated with all other attributes? (¼)

> Fruit

g. If you were to delete two attributes, which would you guess were irrelevant? (¼)

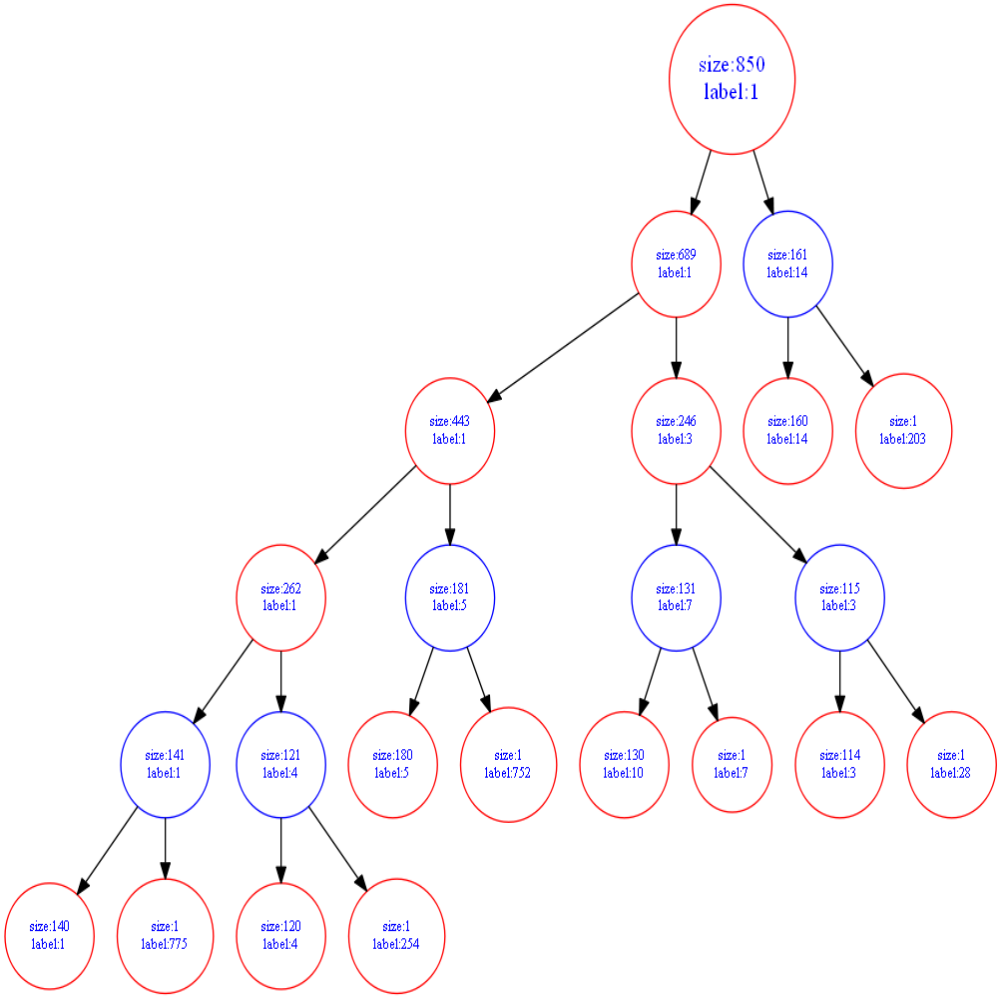
> Milk & Fruit

h. If buying fish is strongly cross-correlated with buying cereal, and buying cereal is strongly crosscorrelated with buying baby products, is buying fish strongly cross-correlated with buying baby products? Can you explain this? (¼)

> Yes. Buying fish is also strongly cross-correlated with buying baby products.
> Suppose:
> $cov(buying fish) = x = cos\alpha$
> $cov(buying cereal) = y = cos\beta$
> $cov(buying baby products) = z = cos\gamma$
> Since the value of x, y and z are in range between -1 to 1
> $\alpha, \beta, \gamma \in [0, \pi]$
> so $cos\alpha cos\beta - sin\alpha sin\beta <= cos\gamma <= cos\alpha cos\beta + sin\alpha sin\beta$
> so $cos(\alpha + \beta) <= cos\gamma <= cos(\alpha - \beta)$
> so $|\alpha - \beta| <= \gamma <= min(\alpha + \beta, 2\pi - \alpha - \beta)$
> This proves if the other sides from α and β are at the same side of the common side which they both have (x and y are both either positive or negative), γ has the minimal value of $|\alpha - \beta|$. $cos\gamma$ would be closer to 1.0 or -1.0 than α or β.
> If α and β are on the wrong sides, γ can still get smaller value, which makes the correlation coefficients keep strongly the same.

Part 3 (Qian Martin): Agglomeration

Final agglomeration clustering result:



The nodes show two things: their size and their labels. According to the requirement of letting small labels be main label while merging, the root label is definetaly 1. As we can see, six clusters(blue nodes) has been clearly generated, Due to the tree structure, no more nodes are needed to be shown.

The model training part is very time consuming, about 3 hours. So it is not recommended to run to code. Alternatively, we saved the model into a text file and rebuild it with information we preserved. This is a rough display of the model, and if you want to see all the label for trained dataset, you can read the HW07_MQ_Trained_Classifier.txt for details.

---

1. When you have clustered to six clusters, report the size of each cluster, from lowest to highest.
   115 121 131 141 161 181
2. When you have clustered to six clusters, report the average prototype of these six clusters.
   So for every element, according to the model we get, we compute its distance to the center of all the clusters, then the shortest one is this element's cluster. Center is recorded as a member of the nodes.
3. What typifies each of the six clusters? What name should we give each of these prototypes?
   As far as I am concerned, I think it is hard to say, but we can at first using their labels to represent them, like cluster 1,3,4,5,7,14(generally, this label is the smallest id inside the

cluster). They mean 6 different kinds of shoppers.
I am not sure what kind of name I should assign to each cluster. However, by looking into the dataset, for example, cluster labeled 1, we can see most of them are likely to buy a lot eggs, meat, vegges and rice. So I think they might be normal family member as they buy daily things a lot and drink less. Similar to the rest 5 clusters, they do have different features, and stand for some people.

4. Write a conclusion about what you learned overall. If each of you learned different things, tell me what each of you learned.

Martin Qian:

> I learned a lot of clustering algorithms, not only agglomeration, I also learned something about k-means, DBScan and PCA. Apart from this, I learned the importance of effeciency and code optimization as this time training the model is really time consuming. The time complecity is O(n^3) and as a matter of fact can be reduced to O(n^2) if we can backup all these distance rather than compute them again.
> As for data exploration itself, I have to say the data is exactly divided into 6 clusters, very clearly and explicit. When all beanches of one node is in big size, this indicating this two nodes are all complete and pure clusters.

Zizhun Guo:

> What did I learn?
> CC can be used to determine what attributes can be discarded or not. For example, some attributes like milk and fruit attributes, they are the two attributes that are least correlated with all other attributes. The reason may come from that all customers conducted consumption would highly possible purchase these two products, which makes them less considerably matter to the clustering. So the two can be pruned before conducting the clustering algorithm in order to reduce the dimensions of the data set so that to increase the efficiencies.