MACAU UNIVERSITY OF SCIENCE AND TECHNOLOGY

Faculty of Information Technology

Thesis for Degree of Bachelor of Science

Title: A Novel Feature Selection Method for Clustering

Student Name : Zhang Tian Liang

Student No. : 1509853M-I011-0011

Supervisor : Alex Leung

May, 2019

澳門科技大學

資訊科技學院

理學學士學位畢業論文

論文題目: 對聚類問題進行變量選擇的新方法

姓　　名 : 張天亮

學　　號 : 1509853M-I011-0011

指導老師 : 梁寶

2019 年 5 月

Abstract

This paper introduces a novel approach of combining clustering and feature selection to do feature selection for unsupervised learning datasets. The strategy of our method is to link features through correlations defined by feature subsets' partitions. On technical side, it calculates distance between pairwise features through the variation of information between its feature subsets' partitions (obtained by K-means clustering). With distance matrix of pairwise features, it is able to link similar features into the same group (or feature-partition) while features in different groups are highly discriminated from each other. It chooses only one feature from each partition to derive a subset. Thus, it ensures the features in derived subset are highly discriminated from each other and the efficiency of removing redundant features. The free parameter of this method is partition-size and the number of preserved features.

Experiments with real-world datasets are applied in this paper. Through hundreds of experiments, it demonstrates that our method is able to obtain high-discriminative subsets of features and remove redundant features. Experiments' results prove that our method does improve the accuracy and performance of K-means++ clustering when it is applied to a real-world dataset.

Keywords: Unsupervised Learning; K-means Clustering; Feature Selection; Real-World Dataset; Variation of Information

摘要

本文介紹主要介紹了一種結合聚類和特徵選擇，為無監督學習數據集做特徵選擇的新方法。我們方法的策略主要是通過特徵子集之間的距離來連接特徵。 從技術角度來看，它通過 K-Means++的聚類結果，計算 variation of information 來鏈接特徵實例。 通過這些鏈接，它能夠導出包括類似特徵的分區組，同時每個組與其他分區組高度區分。 該策略保證了通過特徵選取的特徵之間是高度不同的，使得通過特徵選擇獲得的子集中的特徵具有高辨別度，並減少了冗餘特徵。

本文使用了真實數據集的實驗。 通過數百次實驗，證明了我們的方法的確能夠獲得高辨別度的特徵子集並消除噪聲。 實驗結果證明，當 K-means ++聚類應用於實際數據時，我們的方法確實提高了其準確性和聚類性能。

關鍵詞： 非監督學習；K 均值算法；聚類；變量選擇；真實世界中的數據集；Variation of Information

Table of Contents

List of Figures

Chapter 1 Introduction

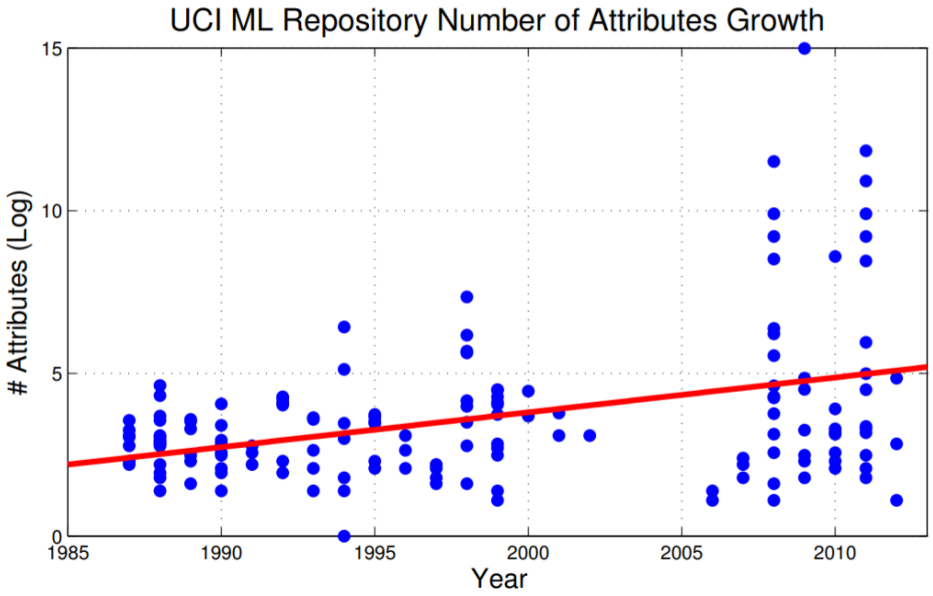
1.1 Background and motivation

As the information technology develops so fast, collecting data becomes easier than ever before, resulting in the exponential growth of data size and its dimensionality. Figure 1-1 shows the trend of this growth in UCI machine learning repository. In data era, processing data manually is impractical while recognizing the underlying correlation between features and its target through computer techniques is more dependable. However, data we crawled from internet mostly have a lot of noises due to the way it collects data is imperfect or the source of data itself, it is a general problem due to the unexpected facts in real world. That makes a lot of trouble for data mining or machine learning. It makes task like removing noises and selecting features more challenging. Thus, do the feature selection to remove noises before automating pattern recognition and knowledge discovery process is necessary, nor the result of recognition may be affected by noises and redundant data. Feature selection is categorized into four models, they are filter model, wrapper model, embedded model and hybrid model, it aims at selecting features to compose a subset of high discriminative features and reduce the dimensionality. Dimensionality reduction techniques are capable to improve learning performance and reducing computational complexity, subset of sample dataset that have been removed noises can help machine learning models do better prediction and recognition.

Currently, there already exits many ways of removing noisy features and redundant features (i.e. irrelevant features or AKA features) like Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA), those techniques achieve dimensionality reduction through extracting approach, aims at selecting a subset of attributes that maximize the relevance to the target and minimize irrelevance. Normally, feature selection selects features and classifies them into different classes, each class includes features are similar but are highly discriminated from features in other classes. In order to select relevant features, labeled samples are needed as training data, this kind of method called supervised learning, use the dataset has its own labels. With labeled training set data, supervised learning is capable to define the correlation between features and target

class. However, defining correlation is not that easy for unsupervised learning since unlabeled data poses a challenge that defining relevancy is unclear. In this paper, we aim to use an approach that define the correlation among features through k-means clustering, and then select some of features to derive a better dataset with less noises and redundant features.

(a)



(b)

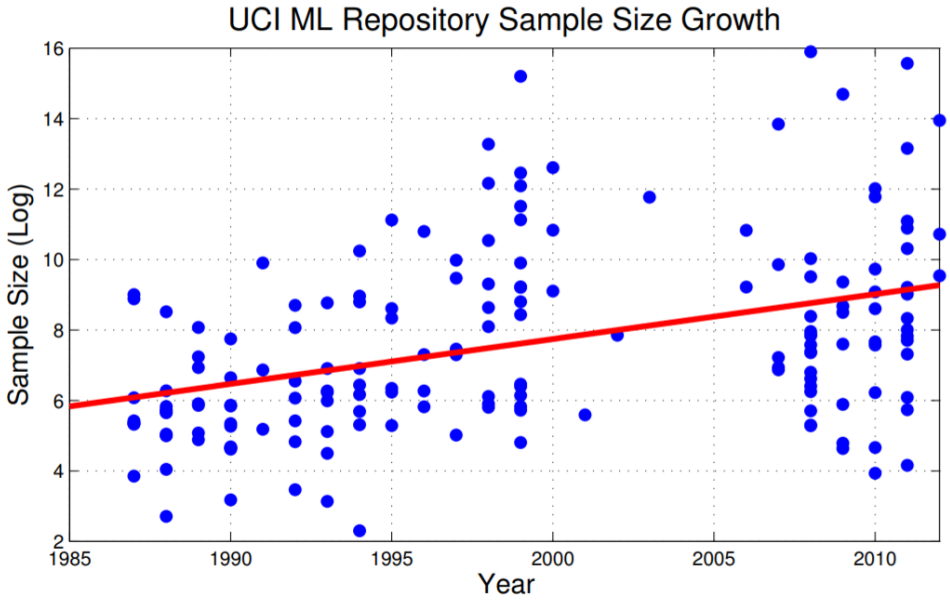


Figure 1-1: (a) shows the attributes growth and (b) shows the sample size growth from 1985 - 2010

1.2 Thesis organization

Data clustering is one of the most popular unsupervised learning techniques, it can classify samples into groups, and each group is so called cluster. Clustering is utilized in many machine learning and data mining tasks, such as information retrieval, pattern recognition, pattern classification, and so on. Today, the mostly utilized methods of clustering are partitioning method and hierarchical method, K-means is one of the mostly utilized partitioning methods. K-means is now the most popular clustering techniques, it has been extensively used in solving machine learning or data mining problems. However, in high-dimensional spaces, finding clusters is not an easy task, calculating distance between plots (sample vectors) requires mass computational works and noisy features may have bad effects on every calculation result. In the situation that dataset have too many features, it is a challenge for k-means to do well as it does normally.

In this paper, we use a method that treat each sample feature as one partition (the partition derived by K-means), every partition is corresponding to one feature subset. Then we run clustering method for all feature subsets one by one, and feature subsets that give very similar clustering result are picked into a bigger partition subset (i.e. a group of feature subsets that have similar clustering results), the size of partition is determined by its cluster number (how does partition size works will be explained later in this paper). We treat the feature subsets that are in one derived partition as redundant features, and randomly choose only one feature subset for each derived partition to derive a new sample dataset that have less features and lower dimensionality. We run the ***K-means++*** method both for derived subset and sample dataset that have full dimensionality. The two clustering results are measured by calculating the ***variation of information*** (a measure of the distance between two clusters) between themselves and ground truth. It turns out that in most of the cases, derived subset from our feature selection method can leads to a better clustering result.

Chapter 2 Clustering: K-Means++

2.1 K-means algorithm

Originally, K-means is a method used for signal processing, but now it is widely utilized in data mining and machine leaning clustering tasks. It is a method to automatically cluster similar data examples together. The intuition behind K-means is an iterative procedure that starts by initial random centroids and then converges quickly to a local optimum of centroids, that is, ***k*** clustering centroids found at the end of K-means. In the beginning of this chapter, it introduces K-means briefly, next several chapters illustrate how our approach improves the performance of K-means.

2.1.1 Steps of k-means

The inner-loop of K-means carries out two steps: (i) Calculating the distance between sample plots and every centroid to find the closest centroid. Set samples have the same closest centroid into the same group. (ii) Repeating the first step until all centroids converge. Algorithm 1 shows the detail of how k-means do the clustering.

Algorithm 1 K-means

Input:

Sample dataset D $= \{x_1 + x_2 + \cdots + x_m\}$

Cluster centroid k

Steps:

1: Randomly pick k samples as initial mean vectors $\{\mu_1,\ \mu_2, \ldots, \mu_k\}$

2: **repeat**

3: let $C_i\ =\ \emptyset\ (1 \leq i\ \leq k)$

4: **for** j $= 1,2, \ldots$, m **do**

5: calculate the distance d_{ji} between x_j and every mean vector $\mu_i\ (1 \leq i\ \leq k)$;

6: select the mean vector that is nearest from x_j: $d_{ji} = \arg min_{i\in\{1,2,\dots,k\}} d_{ji}$;

7: group x_j into the corresponding centroid: $C_{\lambda_j} \cup \{x_j\}$;

8: **end for**

9: **for** i = 1,2, … , k **do**

10: calculate the new mean vector: $\mu_i' = \frac{1}{|C_i|}\sum_{x\in C_i} x$;

11: **if** $\mu_i' \neq \mu_i$ **then**

12: change the current mean vector from μ_i to μ_i'

13: **else**

14: keep the current mean vector unchanged

15: **end if**

16: **end for**

17: **until** the current mean vectors do not update

Output: Centroid set $\mathrm{C} = \{C_1, C_2, \dots, C_k\}$

The output centroids are the result of this clustering, it groups sample data into k clusters. However, due to the initial centroids are randomly picked, it may lead to different local optimum, sometimes, K-means gives a bad local optimum that does not separate the sample plots very well. In order to solve this problem, K-means++ come out.

2.1.2 K-means++

Compared to K-means, K-means select centroids in the way that the distance between centroids is as big as possible. Although the initial selection in the algorithm takes extra time, the k-means part itself converges faster after this seeding and thus the algorithm actually lowers the total computation time and improves in the final error that may occur to k-means. The algorithm has five steps totally:

(i) Choose on center uniformly at random from among the data points.

(ii) For each data point x, compute D(x), the distance between x and the nearest center that has already been chosen.

(iii) Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to $D(x)^2$.

(iv) Repeat Steps 2 and 3 until ***k*** centers have been chosen.

(v) Now that the initial centers have been chosen, proceed using standard k-means clustering.

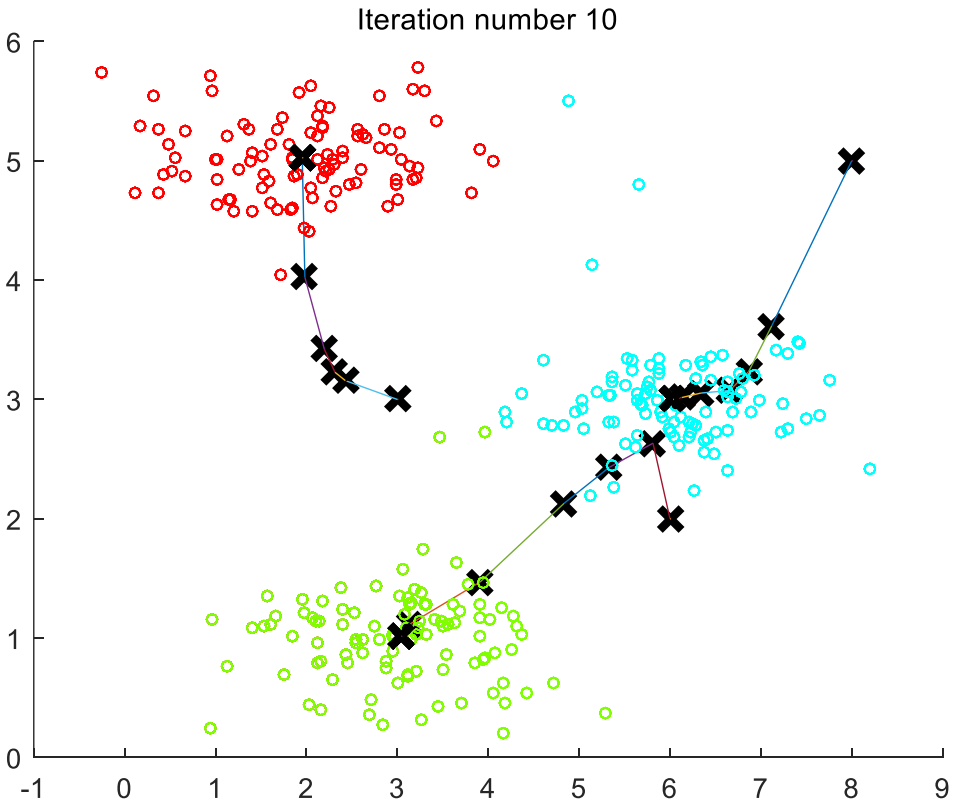


Figure 2-1: Simple K-means implementation

K-means++ improve the problem of a bad local optimum that k-means may converge to. In next chapter, we talk about a method that measures the distance between two cluster, and this method is used to measuring the result of our approach to improve the K-means when sample data have some noises.

Chapter 3 Comparing Two Clusters

3.1 Variation of information

Variation of information (also called shared information distance) is a measure of calculating the distance between two partitions of elements. It does not concern with the relationship between pairs of plots but is based on the relation between a point and its cluster in each of the two clusters that are compared. Mutual information is closely related to variation of information, however, unlike the mutual information, the variation of information is a true metric, in that it obeys the triangle inequality.

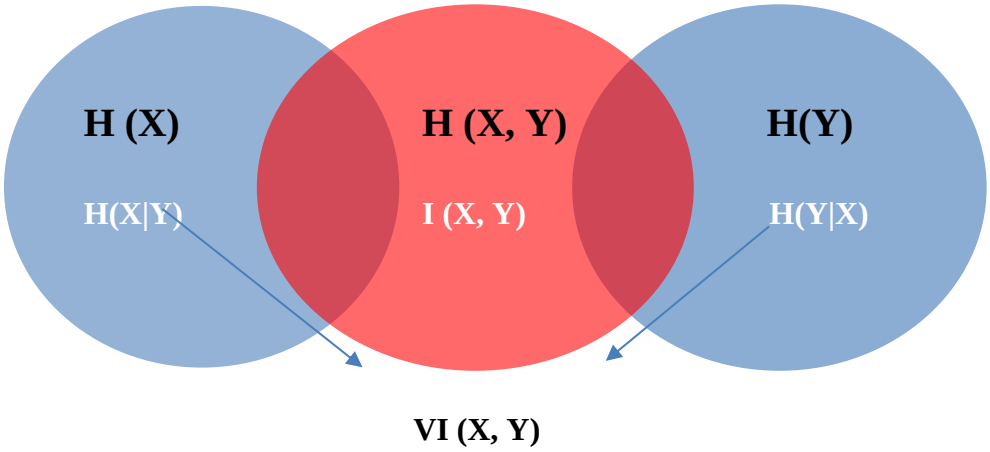


Figure 3-1: shared information distance

In next section, variation of information is used to measure the distance between clustering result obtained by K-means++ and the ground truth of the data.

3.2 Apply K-means++ to our data

3.2.1 Introduction of our data

Data we used in this paper is a real-world data for unsupervised learning, from UCI machine learning repository: Diagnostic Wisconsin Breast Cancer Database. The dataset has 569 samples and 32 attributes, the first attribute is ID number and the second is ground truth, where 0 means benign and 1 means malignant (this dataset have two clusters). Thus, in order to apply the data into K-means++, we remove the first feature and the second (i.e. our data size is 569×30). Most of the other features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass which describe characteristics of the cell nuclei present in the image. The data of second feature is preserved as ground truth to compared with the result derived by K-means++ clustering. However, we come up with an interesting strategy, that is, select some features randomly and do the K-means++ clustering to see whether the result of random-select-subset would lead to a better clustering result than full-dimension data do.

3.2.2 Comparing results of clustering

We pick 4 features randomly from 30 features and do the K-mean++ clustering. To ensure that the stability of our experiment result, steps of calculating the distance of variation of information are in an iteration. Whenever we finish calculating distance of the current 4-feature-subset, we refresh the subset to get a new 4-feature-subset and do the iteration until it is up to the iteration time we set (since there are 27405 cases of selection 4 features from 30, add a iteration and calculate the mean distance can reduce the influence of special cases. In this time, we iterate for 500 times).

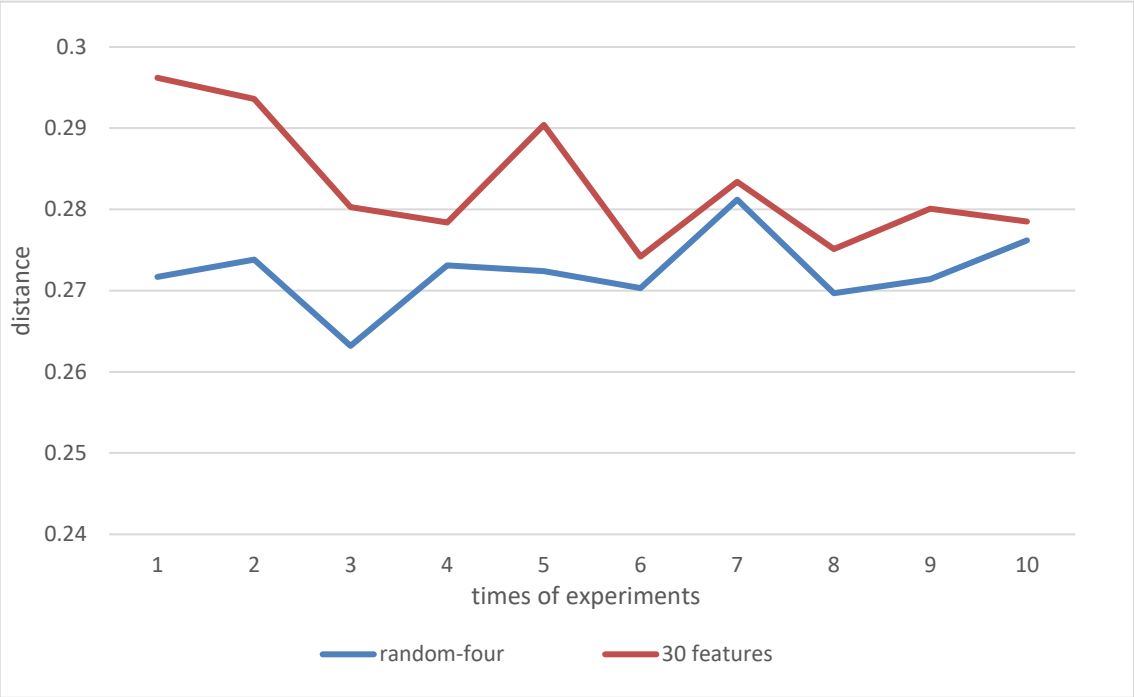


Figure 3-2: Comparing random-pick-subset and full dataset clustering

It turns out that subset randomly picked even works better than full dataset, as the mean distance derived by subset randomly picked is near 0.275 and the mean distance derived by full dataset is near 0.285. The experiments prove the statement that, for high dimensional clustering tasks in real world, there are noisy features in data that have bad effects on clustering results. This case, without feature selection, full dataset does not only have high computational complexity, but also get a worse accuracy performance than random-picked-subset.

In next chapter, we apply our feature selection method to our data to see whether this approach could improve the performance of K-means++ clustering. If the subset obtained by our method lead s to a best clustering results, it shall be able to remove the noises.

Chapter 4 Our Feature Selection Method

4.1 Defining correlation

In our method, we extract data in the way that setting the similar features into the same group, then combine the features chosen from every group to obtain a subset. The correlation of features is defined by clustering results of each feature subset. Generally, Euclidean distance is used for defining correlation between plots (depends on how long the distance is). However, Euclidean distance is used to define distance in space, it does not define distance between clusters (or partitions). For our method, it uses partitions to define the correlation between feature subsets, that is, defining correlation between features by the shared information distance of their partitions derived by K-means++. Figure 4-1 shows how it defines correlations in space. Due to the distance in space is intuitive, it is easy for us to find two clusters in this figure. However, defining correlation by partitions of feature subsets may sound more abstract, it will be illustrated later in this paper.

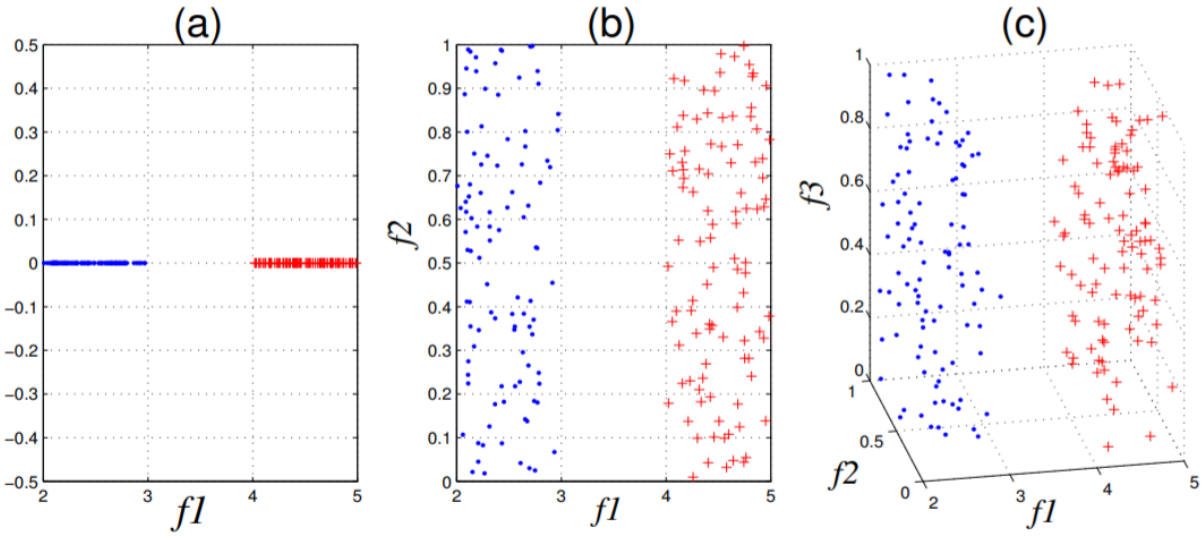


Figure 4-1: correlation samples – feature f_1 is relevant while f_2 and f_3 are irrelevant. We are able to distinguish the two clusters from f_1 only. Thus, removing f_2 and f_3 will not effect the accuracy of clustering.

4.1.1 Steps of our method

In our method, it does K-means clustering for feature subset data into ***p*** cluster (***p*** partitions), and then computes the shared information (variation of information) distance between every pairwise feature partitions in order to find the minimal distance between pairwise features and merge them. For each feature partitions, it selects one feature subset at random for every iteration until it obtains enough features needed. Followed algorithm shows the pseudo code of our method.

Algorithm 2 Our method

Input:

Sample dataset: $D = \{x_1, x_2, \dots, x_m\}$

Data size: {M times D} (D is number of features)

Number of clusters: ***k***

Partition size (over clustering for feature selection): p ($p \geq k$)

Number of features desired as output: f

Collection of feature subsets: F

Naïve distance: naïve distance

Subset distance: subset distance

Steps:

1: **while** |F|> f do:

2: **foreach** Fi in F (where Fi is a feature subset)

3: do k-means clustering of the data into p clusters

4: based on the features in Fi only (call this partition Pi);

5: **for all (i,j):**

6: compute pairwise distances between (Pi, Pj) using shared

7: information distance;

8: merge Fi, Fj corresponding to minimal sharedinfo(Pi,Pj);

9: **end while;**

10: select at random a feature f_i from each Fi;

11: cluster the data into k clusters based on the f_i using K-means;

Output: the clusters

Note that in for the above code, line 8 is a simplified description. When it merges Fi and Fj, the partition is recomputed at each step only for Fij resulting from the union of the Fi and Fj that have been merged, call this Pij, only the distances between Pij and all the other partitions are updated. For the naïve method, it does not recompute the Pij, it just updates the distance matrix between Pi and Pj according to the minimum linkage rule.

Algorithm 3 The "naive" baseline

Input:

Sample dataset: $D = \{x_1, x_2, \dots, x_m\}$

Data size: {M times D} (D is number of features)

Number of clusters: ***k***

Partition size (over clustering for feature selection): p ($p \geq k$)

Number of features desired as output: f

Collection of feature subsets: F

Steps:

1: **foreach** Fi in F (where Fi is a feature subset)

2: do k-means clustering of the data into p clusters

3: based on the features in Fi only (call this partition Pi);

4: **for all (i,j):**

5: compute pairwise distances between (Pi, Pj) using shared

6: information distance;

7: **while** |F|> f **do**:

8: merge Fi, Fj corresponding to minimal sharedinfo(Pi,Pj);

9: update the distance matrix between the Pi according to the minimum linkage

10: rule (**do not recompute** the Pi);

11: select at random a feature f_i from each Fi;

12: cluster the data into k clusters based on the f_i using K-means;

Output: the clusters

In the aim of making comparison for full dimensional data clustering, clustering results derived by naïve method of algorithm 3, and clustering results derived by method of algorithm, we calculate the shared information distance between ground truth and those three clustering results. It adds a iteration to calculate the average shared information distance, thus, the influence of some special selections will be as less as possible (the aim of our test protocol is not to find the best choice of feature selection, but to find the best method, so we choose to calculate the average distance after iterations).

Algorithm 4 Test protocol

Input:

Sample dataset: $D = \{x_1, x_2, \dots, x_m\}$

Data size: {M times D} (D is number of features)

Number of clusters: ***k***

Partition size (over clustering for feature selection): p ($p \geq k$)

Number of features desired as output: f

Number of iterations: n

Steps:

1: **for** iter = 1 to n:

2: cluster data according to Algorithms 1, 2, 3;

3: compute shared information distance between the output of the three algorithms

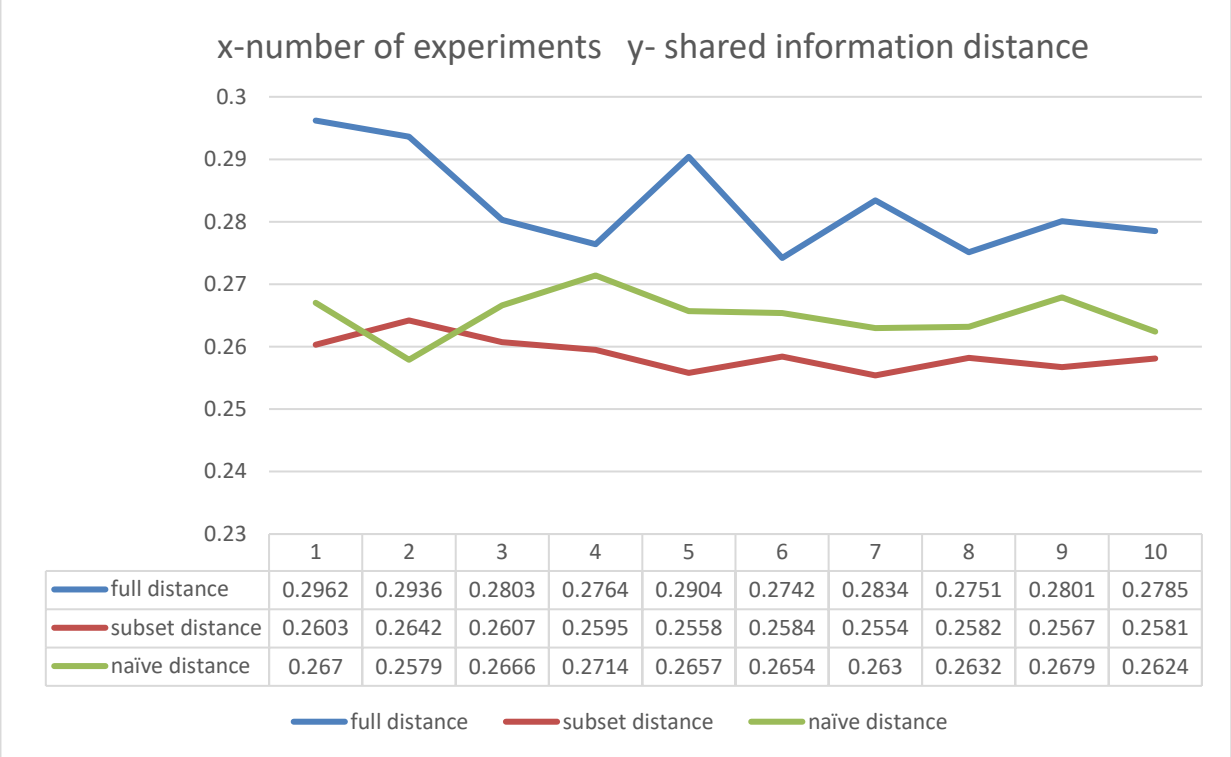
4: and the ground truth;

5: **end for**

Output: the average distances

For the aim of making comparison, some experiments results are collected as followed figure 4-2. Three method are to be compared: (1) K-means++ clustering based on full dimensionality samples (all features are involved, i.e. full distance) (2) K-means++

clustering based on features selected by naïve method (i.e. naïve distance) (3) K-means++ clustering based on features selected by our method (i.e. subset distance). After clustering, all clustering results are to be compared with ground truth using shared information distance. The lower the distance is, the better clustering results we get.



	1	2	3	4	5	6	7	8	9	10
full distance	0.2962	0.2936	0.2803	0.2764	0.2904	0.2742	0.2834	0.2751	0.2801	0.2785
subset distance	0.2603	0.2642	0.2607	0.2595	0.2558	0.2584	0.2554	0.2582	0.2567	0.2581
naïve distance	0.267	0.2579	0.2666	0.2714	0.2657	0.2654	0.263	0.2632	0.2679	0.2624

Figure 4-2: comparison of three distance

4.1.2 Correlations among features

To find the correlation between features, we derive a clustering tree visualization (based on the minimal distance between partitions) to observe the process of naïve method and our method. It records how the method merge two partitions together based on the minimal shared information distance. As figure 4-3 and figure 4-4 show, it merges pairwise feature subsets that have low shared information distance between each other (Y-axis is the shared information distance of pairwise feature subsets, X-axis is the number of corresponding feature subset). Thus, at the end of this process, it poses feature groups that are highly discriminated from each other. Choose one feature from each group,

then we able to find a subset that have high discriminative features, what means that it has less noises normally.

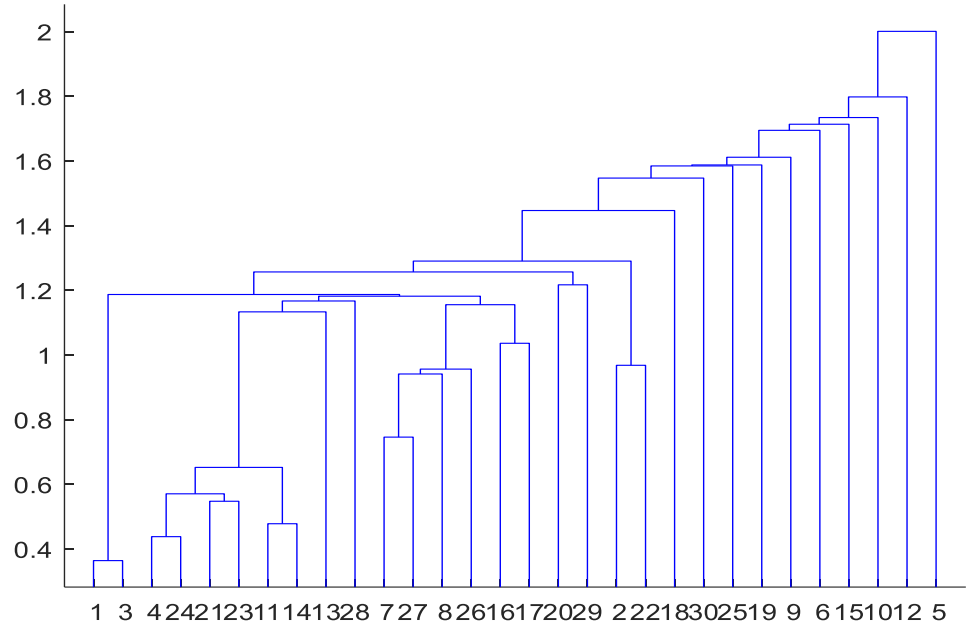


Figure 4-3: naïve method merge process

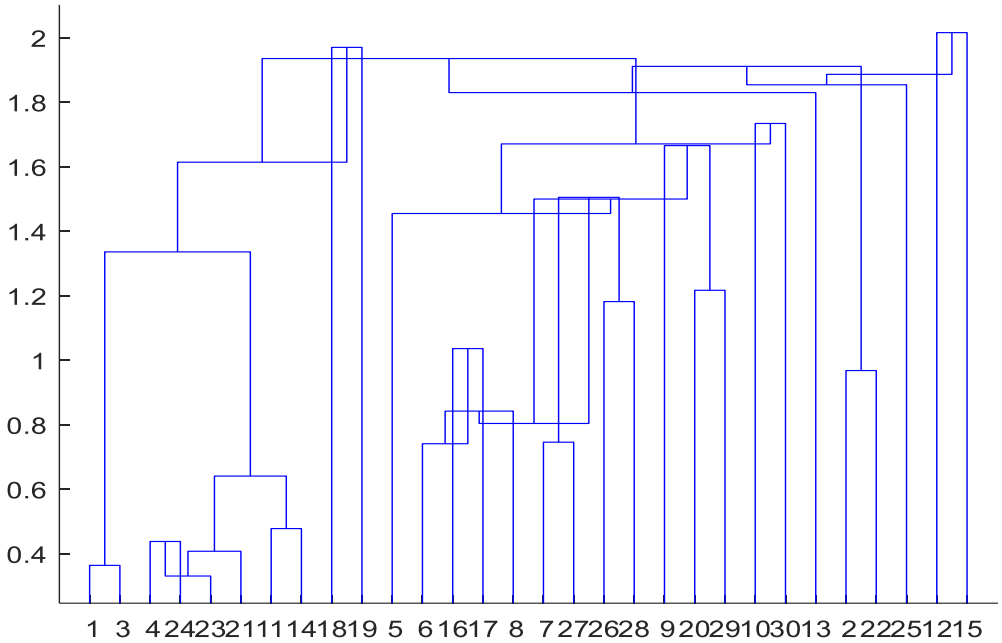


Figure 4-4: subset method merge process

Figure 4-5 shows the correlations between features in a visual way, that is, coloring the pixel point of x-y (x and y are the number of attributes) at coordinate based on their pairwise distance. Like RGB coloring, the lighter the color is, the lower the distance is. On the contrary, the darker the color is, the larger the distance is. Since then, the correlation between features becomes intuitive (dark pixel means that two features are high discriminated from each other while light pixel means that two features are correlated, or close to each other). It gives an intuitive view of the distance matrix generated by pairwise partitions. Not like other measuring ways, it does not focus on the distance between plots, but focus on the shared information distance between partitions (i.e. the clustering result).

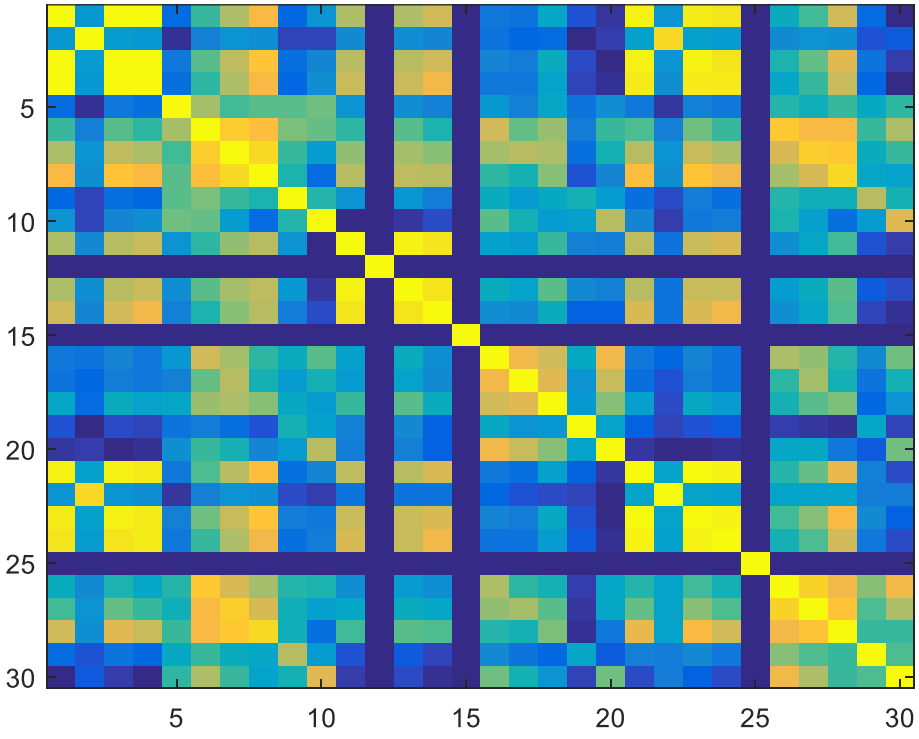


Figure 4-5: visual image of correlation between features

Some experiments are done to see if our dimensionality reduction method (or feature selection method) works well. The strategy of our experiment is to change the number of preserved attributes from small number to big number. If our method does works, when the number of preserved attributes get big enough, the curve of distance shall converge to the full distance baseline, that is, around 0.280 (the distance derived

by clustering data without and dimensionality reduction). It proves that more features or big sample size are to have noises which may have bad effects on clustering results.

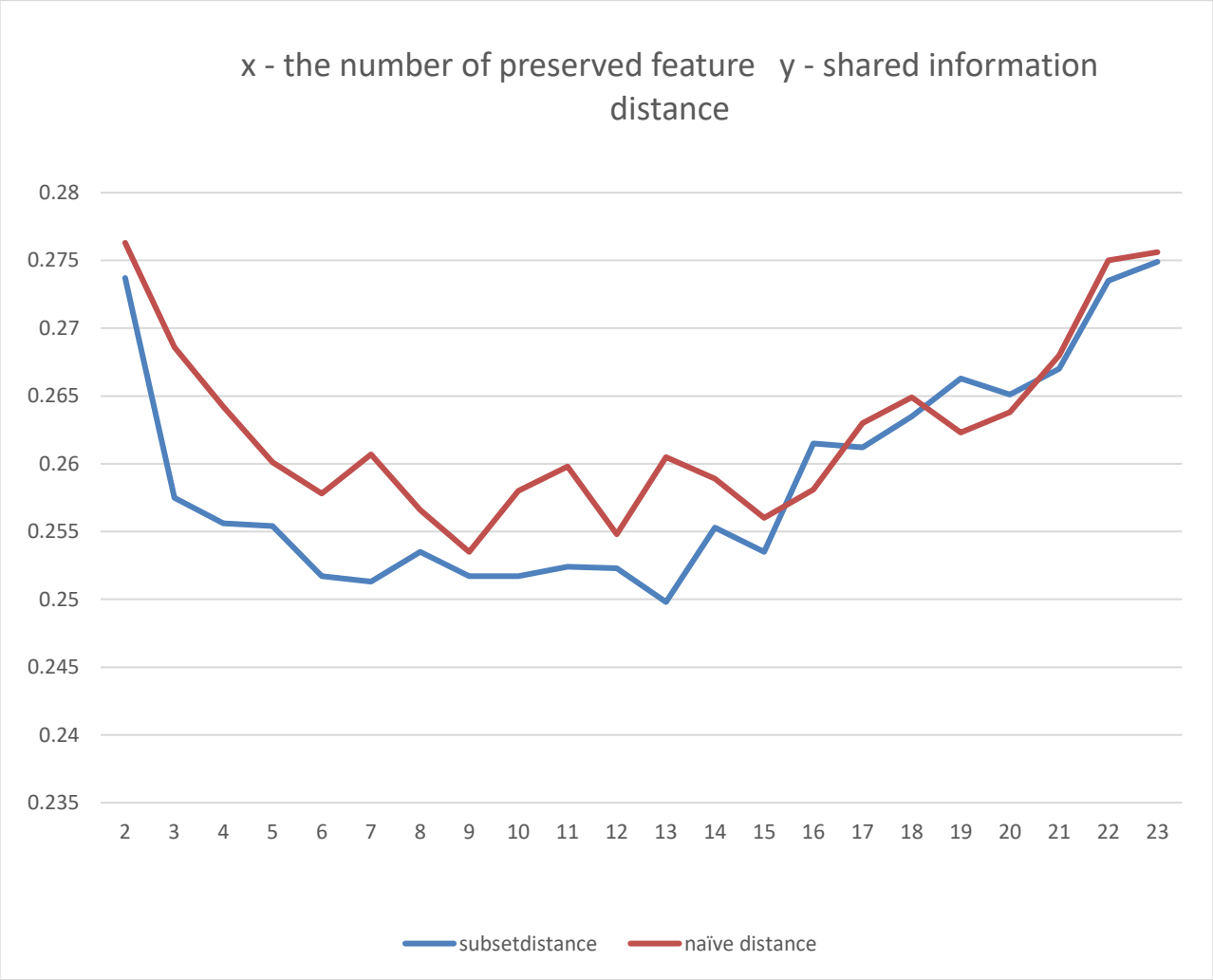


Figure 4-6: Curves of our methods converge when more features to be preserved

Chapter 5 Conclusion

5.1 Conclusion

Nowadays, as the size of data samples grows so fast, the noises among real world data becomes hard to remove and remains challenges for dimensionality reduction. This paper is mainly about a method we provide to do feature selection for clustering tasks. Clustering is an unsupervised learning problem, unlike supervised learning, it does not have labeled data to defining correlation between the attributes and target. Thus, feature selection for unsupervised learning becomes particularly difficult. It meets problems like: (1) how to define correlations among feature instances and link them. (2) how to take advantage of these relations to do feature selection.

The strategy of our method is to define correlations between pairwise feature instance through its K-means++ clustering result, that is, its partition. For each pairwise feature subset partitions, we calculate the shared information distance between them and obtained a distance matrix. The partitions that have the minimal distance are to be merged into a new big-partitions (group of features), then treat this merge-step as a loop and do the iteration until it obtains enough big-partitions. The final version of subset is composed by features that randomly selected from these big-partitions (one feature each big-partition). In the way of grouping similar feature subset into one group, choosing only one feature subset from each group ensures the high discriminative validity of features in subset we obtained. It provides two method on how to do feature selections: (1) naïve method (2) subset method (our method). Both methods use K-means++ partitions to define correlations among feature subsets, but their ways of merge partitions are different. Thus, it compares three ways of clustering: (1) K-means++ clustering based on full dimensionality data. (2) K-means++ clustering based on naïve-method subset (i.e. the subset obtained by naïve feature selection method) (3) K-means++ clustering based on subset-method (i.e. the subset obtained by our subset feature selection method). After all those clustering, their results are to be compared with the ground truth of data through variation of information (shared information distance), a way of measuring distance between clusters or partitions. Let three methods' comparison results are respected to: (1) full distance (2) naïve distance (3) subset distance. We do plenty of experiments on those

three methods to do feature selection on reducing the number of sample data's features from 30 to 4, it turns out that the full distance is around 0.2800, the naïve distance is around 0.2625 and the subset distance is around 0.255. Thus, these ways of feature selection do have the ability to remove noises and improve the performance of clustering.

5.2 Future work

The data we used in this paper is a real-world data (569×30) from UCI machine learning repository. Generally, a real-world data can have many noises and challenges of achieving high accuracy performance. Through the experiments we did so far, our method does improve the accuracy and clustering performance for this data. However, it may meet challenges on other dataset (we apply a 1024×256 Gaussian dataset to our method in the way of preserving 16 or 32 features and it does not improve the accuracy. Maybe it is because of the dataset is Gaussian-generated so every attribute is highly contributed to target. It reminds us to do more experiments to test the method). Moreover, most of the dataset in real world have plenty of data types, especially the text data. The datasets we used currently are all numerical data, so doing feature selections for text data is still a challenge for our method.

Currently, our method successfully implements feature selection to clustering sample data into two clusters, the number of samples is 569. In real world, the problem can be more complex and challenging. As the number of samples grows, our method may meet problems of choosing partition size (since our method defines correlations by each feature subset's partition). Thus, do more experiments on larger-size datasets to apply our method of selecting high discriminative features can serve to know more about the method.

Reference & Bibliography

[1] Wolberg, W. H., Street, W. N., Heisey, D. M., & Mangasarian, O. L. (1995). Computer-derived nuclear features distinguish malignant from benign breast cytology. *Human Pathology*, *26*(7), 792-796.

[2] Alelyani, S., Tang, J., & Liu, H. (2013). Feature selection for clustering: A review. In *Data Clustering* (pp. 29-60). Chapman and Hall/CRC.

[3] Meilă, M. (2003). Comparing clusterings by the variation of information. In *Learning theory and kernel machines* (pp. 173-187). Springer, Berlin, Heidelberg.

[4] Boutsidis, C., & Magdon-Ismail, M. (2013). Deterministic feature selection for k-means clustering. *IEEE Transactions on Information Theory*, *59*(9), 6099-6110.

[5] Roth, V., & Lange, T. (2004). Feature selection in clustering problems. In *Advances in neural information processing systems* (pp. 473-480).

[6] Model, F., Adorjan, P., Olek, A., & Piepenbrock, C. (2001). Feature selection for DNA methylation based cancer classification. *Bioinformatics*, *17*(suppl_1), S157-S164.

[7] Boutsidis, C., Drineas, P., & Mahoney, M. W. (2009). Unsupervised feature selection for the $ k $-means clustering problem. In *Advances in Neural Information Processing Systems* (pp. 153-161).

[8] Li, Y., Luo, C., & Chung, S. M. (2008). Text clustering with feature selection by using statistical data. *IEEE Transactions on knowledge and Data Engineering*, *20*(5), 641-652.

Appendix

Dataset we used in this paper can be find at:

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

http://cs.joensuu.fi/sipu/datasets/

Resume

1. Resume

Name: Zhang Tian Liang

Gender: Male

Email: 1069613929@qq.com

Education:

2012 ~ 2015, High School, Zhou Nan High School.

2015 ~ 2019, Bachelor of Science, Macau University of Science and Technology.

Awards:

Work experience:

2016 ~ 2017, Intern of iMacu Company

2017, Intern of ZhuHai Zhao Bang ZhiNeng Company

Acknowledgement

Thanks for the suggestion and guidance provided by professors Alex and Fabrizio.

Thanks for the algorithm, method, and code provide by Alex’s friend Fabrizio, that is a great help.