Unsupervised Learning and Dimensionality Reduction

Backgrounds

With the increased popularity of online shopping (amazon, neweggs) and online review services (yelp, google reviews), reviews have become an important factor to affect customer's behavior. Fake reviews also emerged as a side effect. In many cases the reviews can make or break certain business. Consequently, it is import to distinguish fake reviews from the rest of them, no matter for customers, business side or review platforms. However, it is usually extremely difficult and tedious to pinpoint fake reviews manually. This is where machine learning algorithms show their power.

Hand written digits recognition is a classic computer vision and machine learning problem. A lot of applications are based on our ability to solve this problem. It also provides a good model to understand multiple machine learning algorithms.

Here, I used the same data sets on tagged amazon reviews and tagged hand written digits as I used in my first project. These two data sets are quite different regarding their feature space and labeling. Thus different data pre processing procedures were used. The features of amazon reviews dataset are text (strings), which requires natural language processing to convert the text feature to numeric values. In contrast, the features of hand written digits are pixel values, which only requires data normalization. The amazon reviews dataset only has two labels: fake and non-fake while the hand written digit dataset has 10 different categories.

Specifically, each entry of amazon review dataset contains not only the review text, but also some other information, including whether the user is verified user or not, the rating for the product. Each entry of the hand written dataset contains an array of pixel color value.

In this project, I applied two clustering algorithms: K means clustering (KM) and Expectation maximization (EM) to these two data sets. Then I tried to improve the clustering results by applying dimension reduction algorithms including PCA, ICA, Randomized reduction and Featured Agglomeration and then compared the clustering results. Finally, I applied the dimension reduction algorithm to the amazon review dataset and then repeated the fitting with neural networks. Then I performed the clustering algorithm (K means clustering) to the dataset and added the cluster information as a feature to do the fitting again. Please find the analysis in the results part.

Results

PART I Amazon reviews

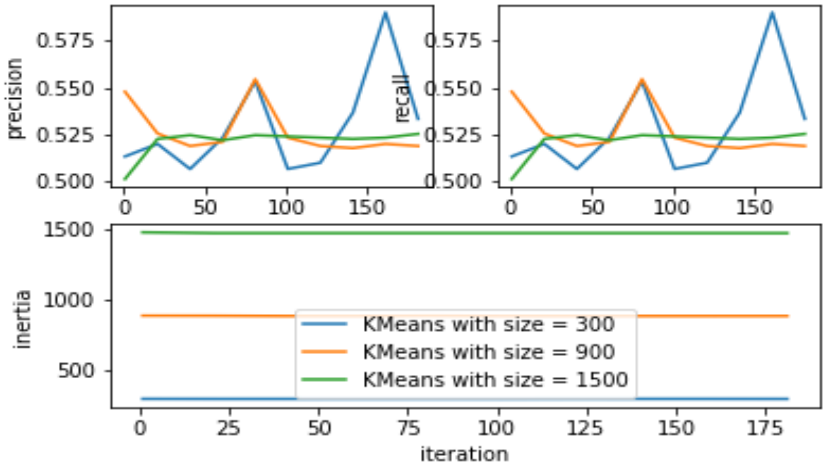
1. K means clustering.

Since the data is labeled with fake reviews and non-fake reviews, I directly chose k = 2 for the k clustering algorithm. In cases where the labels are unknown in advance, cross validation (and domain knowledge) can be used to determine the k value. The initial clustering was done with various training size and iteration numbers to gain insight into how these parameters might affect the clustering results. After clustering, I compared the predicted cluster label with the real label to determine the precision and recall.

At the beginning, only the review text was used as the feature space. To convert the text into numeric values, I simply using the bag of words strategy. Each field was a 0 or 1 value depending on whether the word exist in the review text. The fitting results was terrible with almost random precision and recall (close to 0.5 with the 1500 sample size and 200 iterations).

Considering that the feature space was extremely huge after vectorizing the text, the poor performance of k means clustering resulted from the high dimension curse: the sample sized needed grows exponentially with the feature increases. To reduce the feature space, I first tried to remove unimportant words from the text: the stop words. Stop words are commonly used words that are necessary for sentence structure but contain no information. For example, the words "the", "a/an", "and" are stop words. Symbols and punctuation can also be considered as stop words. After removing stops words, I was able to reduce the feature space slightly.

The other effort that I tried to improve the clustering algorithm was to include some other useful information (domain knowledge) from the dataset, especially rating of the product and whether the reviewer is a verified user. However, since the feature space was still huge even after removing stopped words (more than 5000), these two added features might not be averaged out during the fitting process.

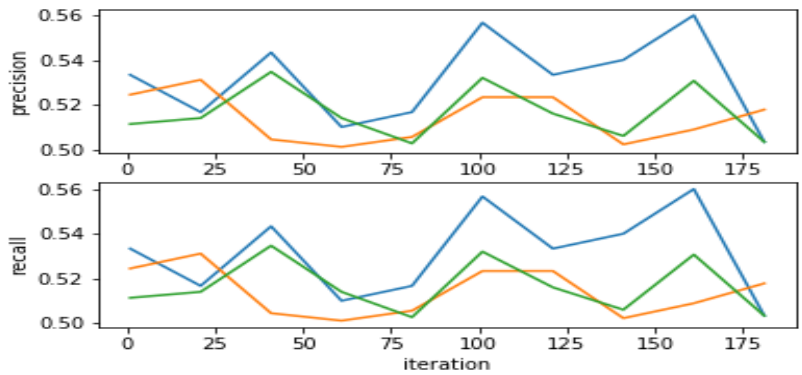


The above figure shows the clustering results after optimization. Generally larger sample size performed better (blue line) compared to smaller sample sizes simply because larger sample provides more information. Interestingly, the precision v.s. iteration or recall v.s. iteration didn't have a simple monotonmic increase shape. Instead, the top two figures showed kind of periodic pattern. My guess is that a lot of samples are lying on the boundary of two clusters. The bottom figure showed the inertia v.s. iterations. The inertia basically is the average distance of samples from cluster centers. Based on the results, the iteration doesn't have any effect on the inertia. Larger sample sizes have smaller values of inertia as expected. Larger sample size provides more information and results in more accurate clustering, as a result, the distance is also minimized compare to smaller sample sizes.

In conclusion, the fitting without dimension reduction has poor performance despite the effort of optimization. The high dimension curse is the main reason causing the poor performance.

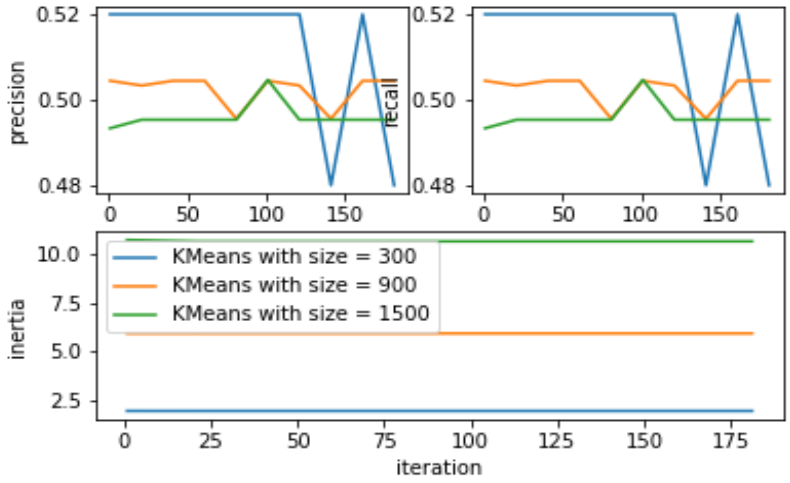
2. Expectation maximization

I also tried to use expectation maximization method to cluster the data with n_components = 2. Similar as the k means clustering, EM without dimension reduction also showed poor results with precision and recall in the range of [0.50, 0.56].



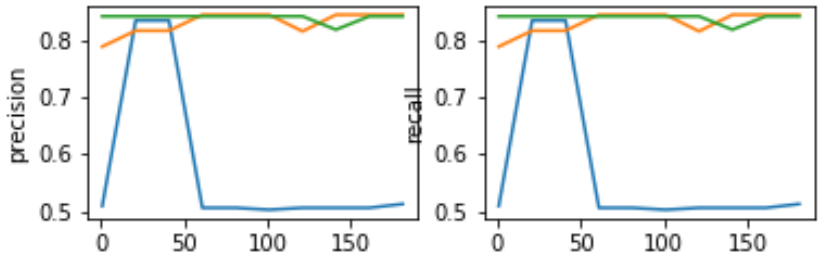
3. Dimension reduction using PCA and K means clustering after dimension reduction.

I firstly performed dimension reduction of PCA to the review text to two components and used these two components to do the k means clustering.



After clustering, I compared the clustering labels with the real labels. In this case, the inertia (the average distances of data points from clustering centers) are much smaller (< 10) than that without dimension reduction (> 500) as expected since the feature space is much smaller. However, the precision and recall is still around 0.5 even with the largest sample size.

Then I tried to incorporate other useful information in addition to the review texts to the feature space, including whether the reviewer is verified and product rating. After adding these new features, the precision and recall was increased (> 0.85) dramatically when sample size is large enough (900 or 1500). This result showed that 1) Domain knowledge is extremely important. Here whether a user is an verified custom or not can be seen as domain knowledge. 2) A small amount of transformed feature is enough to represent the original features. 3) Large amount of features may obscure the important ones.



After PCA, I printed out the words that are important in the first transformed component:

FIRST COMPONENT (weight >= 2%):

amazon 0.031148351211390073
amzn 0.02290367152794322
battery 0.02028303879172534
book 0.02861159280686399
br 0.9702592267073068
cons 0.02852521659050047
even 0.02025016922455429
first 0.022881600100401377
get 0.02744121280599627
headset 0.029576449949517268
hour 0.021473343668484284
http 0.02714435985284351
life 0.024640342852402387

never 0.02150218461157936
pros 0.02053799432501495
something 0.020138252813292276
time 0.02418122681875218
tv 0.041637346337224816

Interestingly, in the first transformed component, the word **br, which is even not a solid English word,** accounts for **97%** of the first transformed component! Here is my reasoning: A lot of fake reviews are copy-paste from other sources. The br is actually copied from the **
 tag** from the html. Here are some examples that are fake reviews containing br tag from my dataset:

*This item is splendid! It fit the way I needed it to and holds my Galaxy S5 and there are no worries about it dropping out. Used it for running and it held up great!<**br** /><**br** />I have bought ones that were definitely more expensive that didn't work half as well or did not go on right. Terrific item!*

*I was shocked at the awful smell and the fact that I could not feel anything good about this product. I used the whole bottle and it just seems to be a placebo that really does not work.<**br** /><**br** />I am switching back to my old resveratrol*

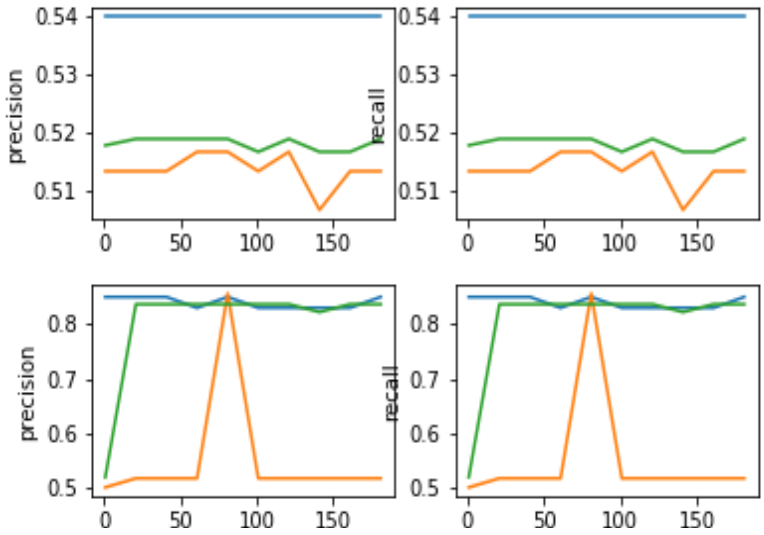
However, a large portion of fake reviews didn't contain br tags. That's why other useful information are important for the clustering here.

The other words selected, such as good, price, first, never, product, work, use, recommend and so on, are very common words in people's reviews, which makes a lot of sense.

4. Dimension reduction using ICA and K means clustering after dimension reduction.

Similarly as IPA, I also tried to reduce the feature dimension to 2 components using ICA and then used the reduced features to do k means clustering.

I first tried to include only the review text as feature space (top) , then tried to add additional information (verified user or not, product rating, bottom).



Again, after adding these new features, the precision and recall was increased (> 0.85) dramatically when sample size is large enough (900 or 1500 blue and green lines in the figure).

After ICA, I also analyzed the components of the first components after dimension reduction:

FIRST COMPONENT (weight >= 2%):

bag 0.02186146610507982
comfortable 0.023481324428439403
fit 0.04710962909164785
love 0.05690252637214438

shoe 0.03540325532737858
size 0.048106086265312355
wear 0.039881284909646236

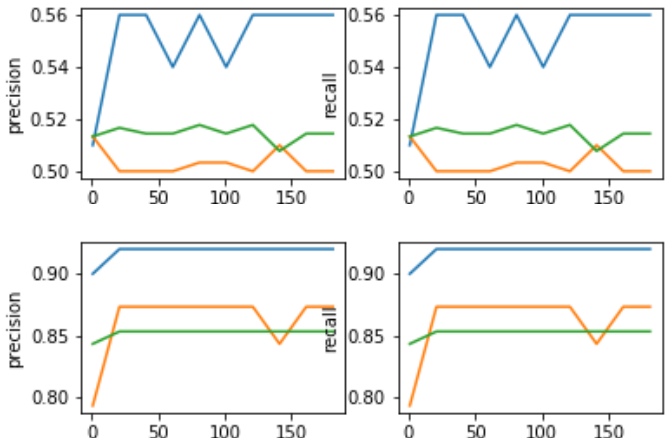
ICA actually aims to separate independent features. The components here are indeed different aspects of certain product.

5. Randomized reduction and K means clustering after dimension reduction.

Similarly, I then tried to reduce the feature dimension to 2 components using randomized reduction and then used the reduced features to do k means clustering. I tried to ran 3 times of the randomized reduction but it didn't change the k clustering precision and recall significantly. The figure showed below is only one of them.

I first tried to include only the review text as feature space (top two), then tried to add additional information (verified user or not, product rating, bottom two).

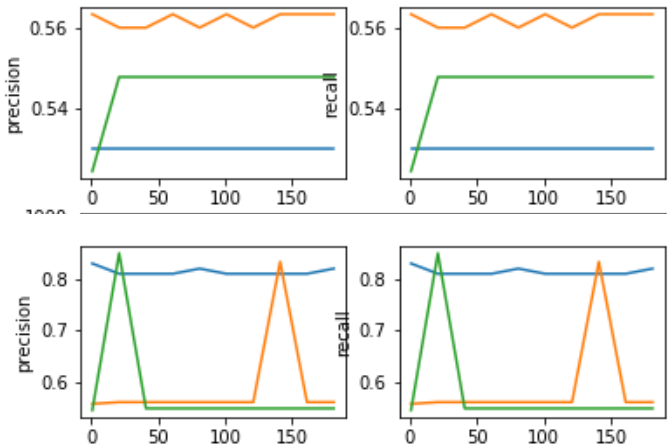
Again, after adding these new features, the precision and recall was increased (> 0.82) dramatically, even when the training sample size is small (300).



6. Dimension reduction using featured agglomeration and K means clustering.

Then I used featured agglomeration algorithm to reduce the feature space. Basically, the Feature Agglomeration uses agglomerative clustering to group together features that look very similar, thus decreasing the number of features. Here again I reduced the components to two and then did the k means clustering.

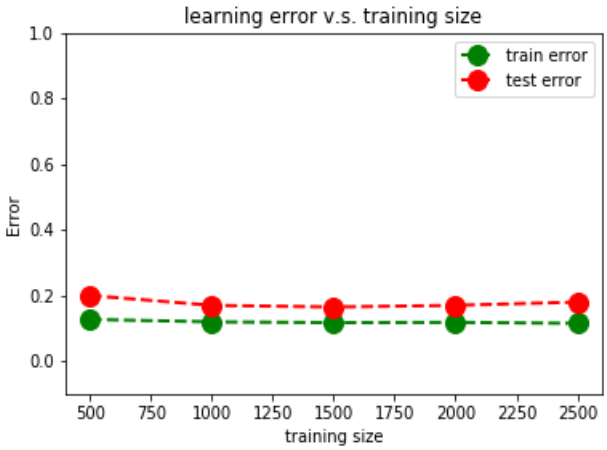
I first tried to include only the review text as feature space (top two subfigures), then tried to add additional information (verified user or not, product rating, bottom two subfigures).



Again, the fitting result was still poor when only using the review text as features. After adding these new features, the precision and recall was increased (> 0.8) dramatically, but only when the sample size is large (1500).

7. Dimension reduction using PCA and then apply neural networks for fitting.

Based on the above four dimension reduction algorithms, I decided to reduce the text feature to 2 components, and then add additional information (verified user or not, product rating) to the feature space. Then I used the new data set to run a neural network fitting. I splite the data into training class and test class and record the training error and test error with different sample size.

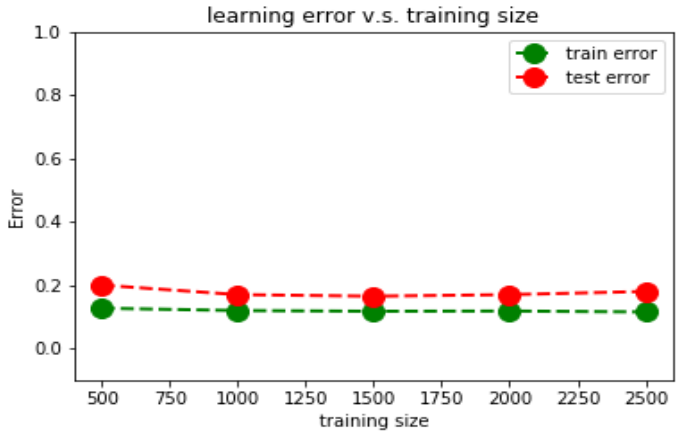


Basically, the fitting results is improved (train error = 0. 11 and test error = 0.15) compared to not doing reduction (test error = 0.20). In addition, the fitting process was much faster (0.81 s)than previous fitting (2 minutes), which can be explained by fewer feature used.

8 Adding cluster info to feature space.

After doing dimension reduction as in step 7, I did k means clustering to the reviews data. Then I added the predicted cluster to the feature lists for neural network fitting.

The results of fitting was almost the same as from step 7. Since the clustering information can be seen as a combination of all other features, it didn't add more information to the feature space. Consequently, addition of clustering info to the feature didn't improve the fitting.



9. Running time

The running time for different clustering algorithms and dimension reduction algorithms are listed as following:

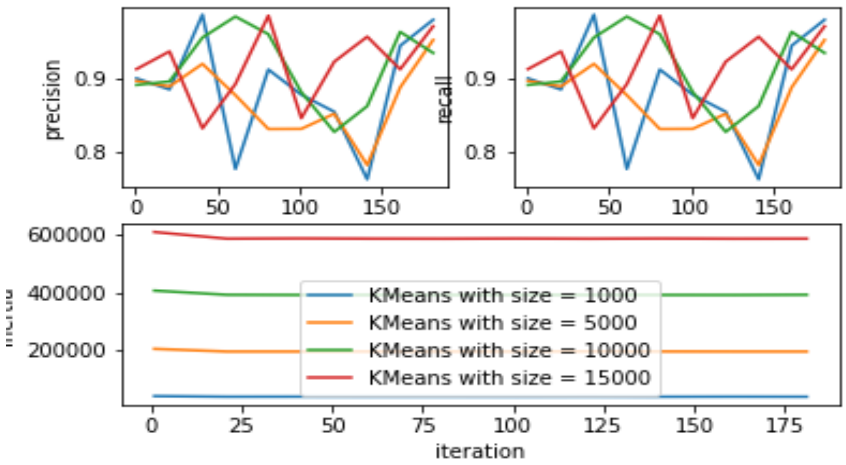
algorithm	Dimension reduction time	Fitting time
K means clustering	None	96.7 s
Expectation Maximization	None	48 s
K means clustering with PCA	2.63 s (n_components = 2)	0.43 s
K means clustering with ICA	5.5 s (n_components = 2)	1.1 s
K means clustering with Randomized Projections	0.08 s (n_components = 2)	0.6 s
K means clustering with Feature Agglomeration	238 s (n_components = 2)	0.36 s

Apparently, the fitting time after dimension reduction decreased significantly: Fitting time before reduction is in 1-2 minutes while after reduction is in seconds. This is mainly because the feature space is much smaller after dimension reduction. The time that dimension reduction algorithms take is generally short (also in seconds) except the Kmeans clustering with Feature Agglomeration.

PART II Hand written digits

1. K means clustering on hand written digits.

Since the data is labeled with fake reviews and non-fake reviews, I directly chose k = 10 for the k clustering algorithm. The initial clustering was done with various training size and iteration numbers to

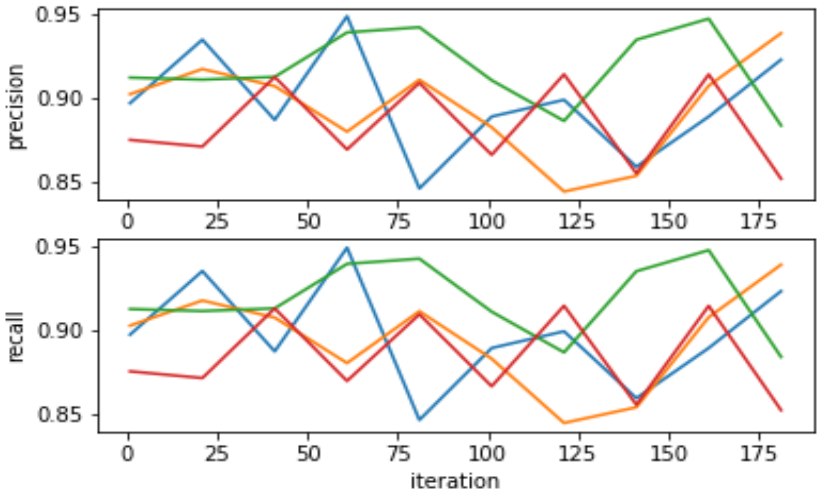


gain insight into how these parameters might affect the clustering results. The result is shown in the following figure.

The predicted label based on clustering is already impressive ([0.85, 0.95]) without dimension reduction. This can be explained by the small feature space (784) and large sample size (15 000). Larger training size also has smaller inertia value (average distance from the center of cluster).

2. EM on hand written digits.

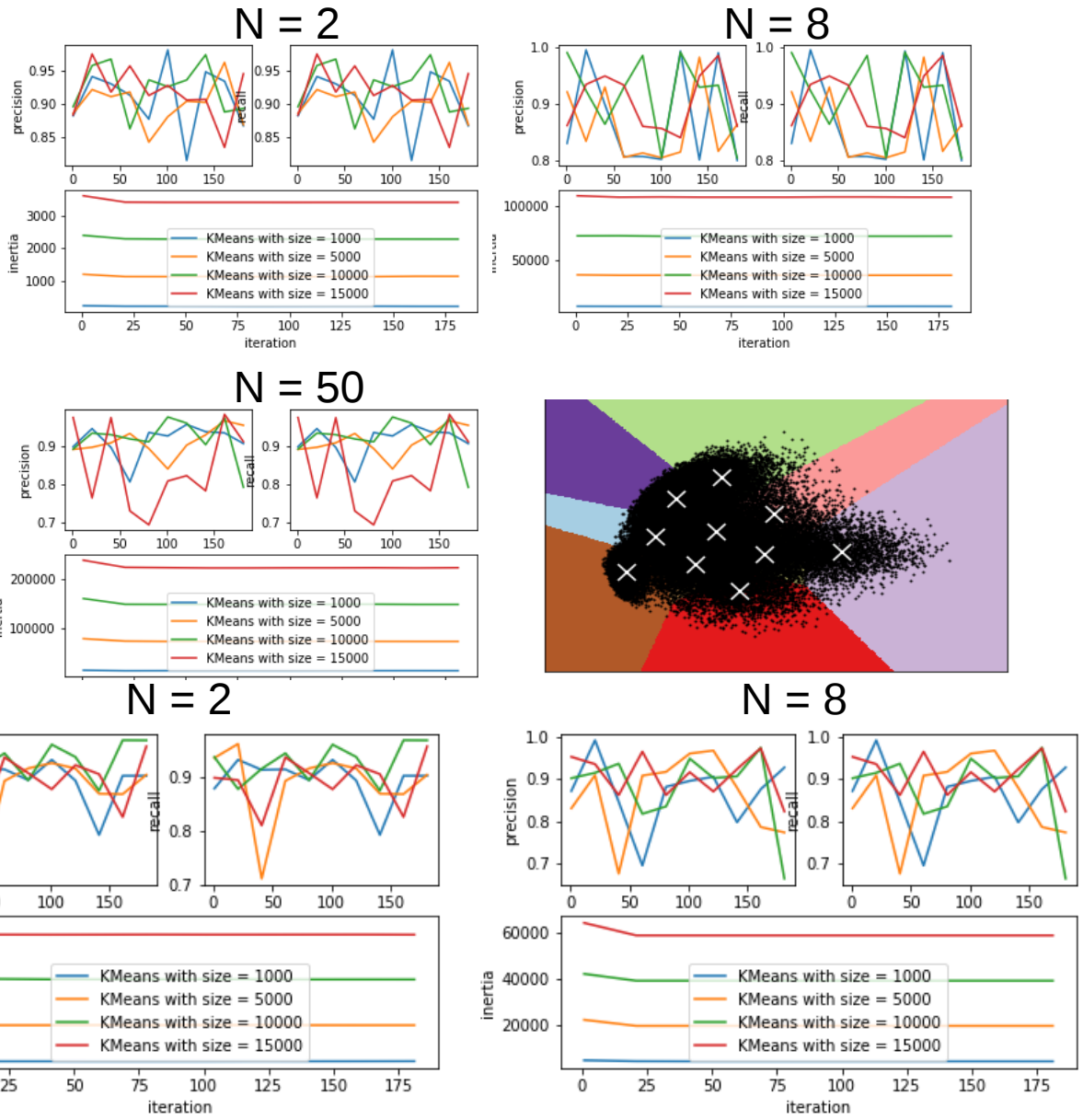
Similar as the k means clustering, clustering with EM also had good performance regardless of sample size (different colors in the figure). The precision and recall also lies within the range of [0.85, 0.95].



3. Dimension reduction using PCA and K means clustering after dimension reduction.

I performed dimension reduction of PCA to the hand written digits dataset and then used these two components to do the k means clustering.

Here I tried three different component numbers: 2, 8 and 50. The precision and recall are similar for N = 2 and N = 8, both between 0.85 and 0.97, regardless of sample size (1000, 5000, 10000 and 15000).

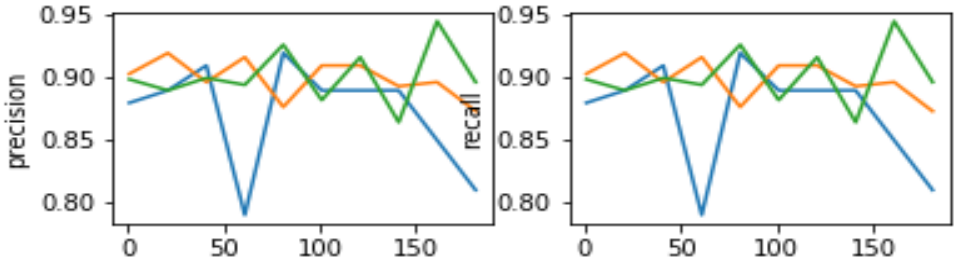


Here I tried three different component numbers: 2, 8. The precision and recall are similar for N = 2 and N = 8, both between 0.8 and 0.95, regardless of sample size (1000, 5000, 10000 and 15000) when the iteration is between 60 and 150.

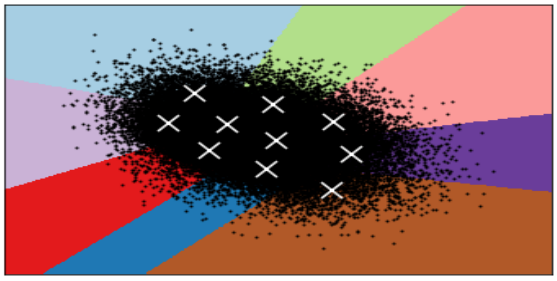
The results here showed that two transformed features are enough to capture the main feature of the hand written data. To view these two transformed features, I also plot the feature space and the clusters.

5. Randomized reduction and K means clustering after dimension reduction.

Similarly, I performed randomized reduction to the hand written digits dataset with n_components = 2 and then used these two components to do the k means clustering.

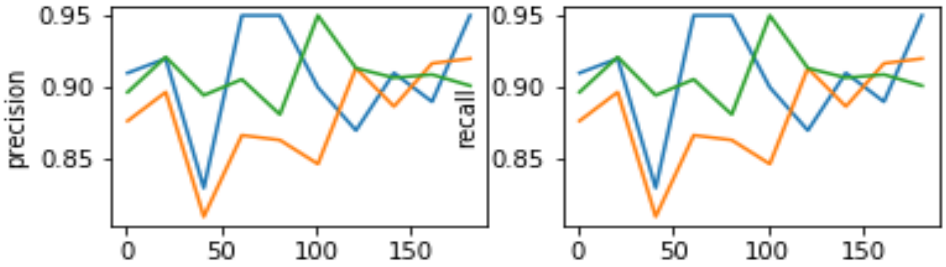


Based on the above figure, the precision and recall of k means clustering lyes between 0.85 and 0.95 when the sample size is large. To view these two transformed features, I also plot the feature space and the clusters.

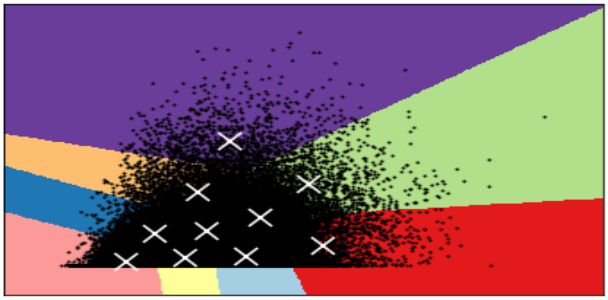


6. Dimension reduction using featured agglomeration and K means clustering.

Similarly, I performed feature agglomeration to the hand written digits dataset with n_components = 2 and then used these two components to do the k means clustering.



Based on the above figure, the precision and recall of k means clustering lyes between 0.80 and 0.95 when the sample size is large. To view these two transformed features, I also plot the feature space and the clusters.



7 Running time

algorithm	Dimension reduction time	Fitting time
K means clustering	None	161 s
Expectation Maximization	None	115 s
K means clustering with PCA	2.44 s (n_components = 2)	9.08 s
K means clustering with ICA	17.2 s (n_components = 2)	1.51 s
K means clustering with Randomized Projections	0.12 s (n_components = 2)	0.71 s
K means clustering with FeatureAgglomeration	10.44 s (n_components = 2)	0.74 s

Apparently, the fitting time after dimension reduction decreased significantly: Fitting time before reduction is in 1-2 minutes while after reduction is in seconds. This is mainly because the feature space is much smaller after dimension reduction. The time that dimension reduction algorithms take is generally short (also in seconds) except the Kmeans clustering with Feature Agglomeration.

Conclusion

The results of this project showed that feature dimension has great impact on clustering algorithms. Actually, high dimension curse exists in almost all machine learning algorithms. Dimension reduction algorithms can not only reduce the feature space dimension while capture the main features, but also decrease the running time significantly. Specifically, PCA tends to find the components that have the highest variance from the original data. ICA tends to find the independent features and capture certain sub feature. On the other hand, increase the training sample size is another way to solve the dimension issue, but it is less realistic compare to dimension reduction algorithms.

The four dimension reduction algorithms PCA, ICA, randomized reduction and feature agglomeration tested here were able to transform the data into much smaller feature space while still capture the main features of the datasets, although the underlying mechanisms for the dimension reduction are quite different from each other.

Domain knowledge will also significantly enhance the clustering/fitting. For example, the amazon review datasets has two other important fields: verified user and product rating. Whether a user is verified or not is a great indicator about whether the review is fake or not. Adding this feature to the feature space will greatly improve the clustering results.