k-Means Clustering

Machine Learning (CS 545), Portland State University, Winter 2016

Bruce Marron

08 March 2016

Introduction

The k-means algorithm is an algorithm for placing N data points $\mathbf{X} = \{\mathbf{x}^{(n)}\}$ from an I-dimensional space into k clusters where each cluster is parameterized by a vector called its mean. Once the number of cluster has been defined and an initial parameterization of the means accomplished, the k-means algorithm proceeds as an iterative, two-step process. In the *assignment step* each data point is assigned to the nearest mean (centroid). In the *update step* means are adjusted to match the sample means of the data points for which they are responsible. The algorithm is run until the change in centroids is below some user-determined threshold (convergence).

This report documents the use of k-means clustering in two experiments to classify handwritten digits that have been extracted to normalized bitmaps given a feature vector in $\mathbb{R}^{64}$. The "OptDigits" dataset was the source of raw data and was provided by the instructor. All data processing and evaluation tasks were done in Python 2.7.11 (Anaconda 2.4.1 (32-bit)) using the integrated development environment (IDE), "Spyder". The open source, "scikit-learn" machine learning package was used as a reference for code construction.

Data Processing

The raw data in OptDigits contains a training set of 3823 cases and a test set of 1797 cases. Each case is a vector of 64 features plus one class attribute. Feature values range from 0 - 16 and class attributes correspond to the natural number set, N = 0,1, 2,...,9. The data were imported and class attribute data were separated from case data for both the training dataset and the test dataset. This resulted in four, datasets: `tr_d_X, tr_d_y, te_d_X, te_d_y`. The procedures and methods used for all data processing tasks are detailed in the script `DataProcessing01.py`.

Implementation of the k-means Algorithm

The implementation and actualization of the k-means algorithm is detailed in the following scripts,

`DataProcessing02.py, DataProcessing03.py, DataProcessing04.py, DataProcessing05.py`
`DataProcessing06.py, DataProcessing07.py, DataProcessing08.py, DataProcessing09.py`
`DataProcessing10.py, DataProcessing11.py, DataProcessing12.py`

Experiment 1: Results

The results of Experiment 1 (k=10 clusters) are presented in Table 1 through Table 12. Accuracy was determined as 20.14% (=(176+3+2+1+173+2+5)/1797). As shown in Table 1, the fourth

Table 1: The SSE, sum-squared separation, and entropy for k-means classification (k=10) on the OptDigits training dataset.

Initialization (random cluster centers)	SSE	Sum-squared separation	Entropy
rcc1	2869491	166318	2.51909
rcc2	2716914	148726	2.94148
rcc3	2953240	161250	2.62772
rcc4	2596054	128680	3.09108
rcc5	2874411	159626	2.70705

random center cluster ('rcc4') had the minimum SSE and maximum entropy and was selected to generate the 'best centers' for Experiment 1. A confusion matrix was generated for each class to better detail classifier performance.

Experiment 2: Results

The results of Experiment 2 (k=30 clusters) are presented in Table 13 through Table 24. The accuracy was determined as 51.53% (= 84+100+59+112+76+100+92+108+116+79)/1797). As shown in Table 13, the fourth random center cluster ('rcc4') again had the minimum SSE and maximum entropy and was selected to generate the 'best centers' for Experiment 2. Again, a confusion matrix was generated for each class to better detail classifier performance.

Table 2: The assignment of cluster center labels from the best run of Experiment 1 (using rcc4) to the most frequent class they contain.

Cluster Center	Most Freq. Class
0	0
7	1
7	2
8	3
5	4
9	5
6	6
4	7
2	8
8	9

Table 3: The confusion matrix for Class = 0 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		0	!0
Actual	0	176	10
	!0	2	1609

Table 4: The confusion matrix for Class = 1 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		1	!1
Actual	1	0	0
	!1	182	1615

Table 5: The confusion matrix for Class = 2 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		2	!2
Actual	2	3	128
	!2	174	1492

Table 6: The confusion matrix for Class = 3 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		3	!3
Actual	3	2	131
	!3	181	1483

Table 7: The confusion matrix for Class = 4 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		4	!4
Actual	4	1	152
	!4	180	1464

Table 8: The confusion matrix for Class = 5 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		5	!5
Actual	5	0	187
	!5	182	1428

Table 9: The confusion matrix for Class = 6 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		6	!6
Actual	6	173	8
	!6	8	1608

Table 10: The confusion matrix for Class = 7 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		7	!7
Actual	7	2	336
	!7	177	1282

Table 11: The confusion matrix for Class = 8 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		8	!8
Actual	8	5	382
	!8	169	1295

Table 12: The confusion matrix for Class = 9 using the 'best centers' k-means classifier from Experiment 1 as applied to the OptDigits test dataset.

		Predicted	
		9	!9
Actual	9	0	155
	!9	180	1462

Table 13: The SSE, sum-squared separation, and entropy for k-means classification (k=30) on the OptDigits training dataset.

Initialization (random cluster centers)	SSE	Sum-squared separation	Entropy
rcc1	2209306	1698804	4.04929
rcc2	2438089	1272426	3.21328
rcc3	2121108	1713960	3.97819
rcc4	2012952	1692684	4.42242
rcc5	2585148	1267474	3.12321

Table 14: The assignment of cluster center labels from the best run of Experiment 2 (using rcc4) to the most frequent class they contain.

Cluster Center	Most Freq. Class
13	0
9	1
7	2
27	3
19	4
0	5
26	6
23	7
10	8
20	9

Table 15: The confusion matrix for Class = 0 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		0	!0
Actual	0	84	0
	!0	94	1619

Table 16: The confusion matrix for Class = 1 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		1	!1
Actual	1	100	23
	!1	82	1592

Table 17: The confusion matrix for Class = 2 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		2	!2
Actual	2	59	20
	!2	118	1600

Table 18: The confusion matrix for Class = 3 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		3	!3
Actual	3	112	3
	!3	71	1611

Table 19: The confusion matrix for Class = 4 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		4	!4
Actual	4	76	2
	!4	105	1614

Table 20: The confusion matrix for Class = 5 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		5	!5
Actual	5	100	3
	!5	82	1612

Table 21: The confusion matrix for Class = 6 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		6	!6
Actual	6	92	0
	!6	89	1616

Table 22: The confusion matrix for Class = 7 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		7	!7
Actual	7	108	12
	!7	71	1606

Table 23: The confusion matrix for Class = 8 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		8	!8
Actual	8	116	14
	!8	58	1609

Table 24: The confusion matrix for Class = 9 using the 'best centers' k-means classifier from Experiment 2 as applied to the OptDigits test dataset.

		Predicted	
		9	!9
Actual	9	79	21
	!9	101	1596

Discussion

The accuracy certainly improved with k=30 clusters although it is still far short of the results reported by the authors of the OptDigit dataset (90%). This is most likely to the random selection for all starting clusters rather than a pre-selected set of initial clusters. Interestingly, the top two digits correctly classified differed between Experiment 1 and Experiment 2. In the first experiment, digits "0"and "6" were recognized with very high frequency (176/178 and 173/181, respectively) while in the second experiment, "8" and "3" were recognized with medium frequency (116/174 and 112/183, respectively).

All of the 'best center' clusters were translated to 8x8 bitmaps (.pmg files; Exp1Graphics and Exp2Graphics folders). Visualizing these images shows that many do resemble the digits being classified, as for example, 'best center' cluster 0 from Experiment 1 (Figure 1) and "best center" cluster 8 from Experiment 2 (Figure 2). Others appear strange, as for example, "best center" cluster 5 from Experiment 1 (Figure 3)and "best center" cluster 7 from Experiment 2 (Figure 4).

Conclusions

Despite its limitations (sensitivity to initial cluster selection; sensitivity to outliers; lack of rigorous methods for selection of the number of clusters; local optimization) this algorithm appears to be very useful for general sorting purposes, especially on new datasets that lack well-defined priors.

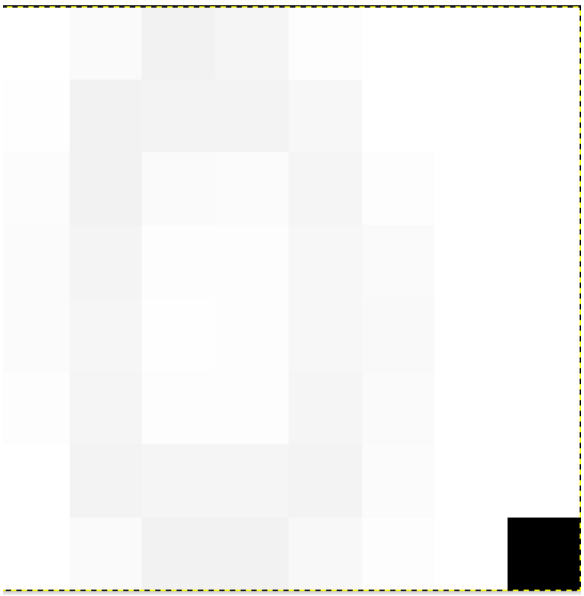

Figure 1: 'Best center' cluster 0 from Experiment 1. This centroid had good success in correctly identifying the digit, "0" from the OptDigits training dataset.

Figure 2: 'Best center' cluster 10 from Experiment 2. This centroid had good success in correctly identifying the digit, "8" from the OptDigits training dataset.

Figure 3: 'Best center' cluster 5 from Experiment 1. This centroid had poor success in correctly identifying the digit, "4" from the OptDigits training dataset.

Figure 4: 'Best center' cluster 7 from Experiment 2. This centroid had poor success in correctly identifying the digit, "2" from the OptDigits training dataset.