Robust Vector Quantization

Taylor Brown

December 9, 2010

Abstract

When working with large data sets, there is often a need to perform clustering for exploratory data analysis. Often, some of clustering algorithms, such as spectral clustering, would be usefull, but are too slow to be used effectively. A new method of reducing samples in a data set, Robust Vector Quantization (RVQ) is proposed. Using RVQ, it is shown that for noisy data sets, the clustering ability is improved compared to similar methods using k-means.

1 Introduction

With the recent explosion in massive datasets, the need to cull intelligence on a large scale has risen. Unfortunately some of the more flexible methods of clustering, specifically spectral clustering techniques [8, 6, 12] are relatively slow, with a computational complexity of $O(n^3)$. While various methods relating to creating sparse matrices [4] have been developed, one recently developed technique involves reducing the sample space using various methods, such that the amount of data input to the spectral clustering algorithm is drastically reduced [11]. The process of taking a large data set, reducing its size, and then expanding it again is a common problem in signal and image processing. In fact, the common k-means algorithm comes directly from signal processing, originally known as Lloyd's algorithm. The various techniques used to reduce multidimensional data are commonly referred to as vector quantization in the signal processing literature. With this in mind, some of the terms and ideas from signal processing will be used. First, the problems of noise and aliasing will be addressed. Then, our algorithm, Robust Vector Quantization, RVQ, will be presented. Finally, the performance of the algorithm will be assessed on a toy problem as well as real world data.

2 Problem

From a machine learning perspective, we are particularly interested in the KASP algorithm presented by Yan et al. The algorithm works by reducing the number of points in a given data set to a set that hopefully still represents the original structure, with some of the redundancy removed. This is done with the use of k-means, or Lloyd's algorithm, to reduce the data to a given number of points. These representative points are then used to construct the affinity matrix and input to the spectral clustering algorithm. The original data is then placed in the same cluster as its representative point from the k-means step. The crux of the problem is of course the choice of appropriate points in such a manner as to allow spectral clustering to perform well. Vector quantization via Lloyd's algorithm is an iterative algorithm which seeks to minimize the error associated within each centroid. The basic assumption is that the data is capable of being represented by several overlapping Gaussians. The algorithm will then try to assign each Gaussian a vector. There are two main problems associated with this approach which we will address. The first problem, choosing at least k points to represent the data accurately, may be described

by using an example. Take a single dimensional space, where we generate points according to a Gaussian about two points, -5 and 5. From a quantization perspective, assuming the variances are equal, we would cluster the final data by splitting in two at zero. Using vector quantization to reduce the data size, a run of k=1 will produce a single point, at roughly zero, depending on the run. Clearly, this does not allow for an accurate split of the data by the spectral algorithms. However, a run of vector quantization with k=2 might find a fairly accurate split, such that further processing would split the data rather well. A run of k=4 could also find an accurate enough split such that future algorithms are able to accurately split the data. In fact, the greater the k, the more like the original data the representation will become. In fact, it is clear that there exists some distribution of points that if we knew the actual distribution, we could place the k points in such a way as to allow an effective split. For example, we could just split the points evenly and place half at -5, and half at 5. For the case of k=1, this is not possible there is no way to convey the presence of more than one mean using a single point. Expanding this idea further, if we had three Gaussians, one at -5, 0, and 5, we could not adequately explain these points with k=2 centers; we need at least 3, or at least two of the actual means will be represented by a single point. In signal processing, the effect of capturing too few features for an accurate representation during quantization is known as aliasing. Essentially, the number of points of the representation is inadequate to represent the underlying signal, or in this case, distribution. In exploratory data analysis, we do not know the appropriate number of centers, k, to adequately represent an arbitrary data set. The other issue faced in choosing representative points is the possibility of overfitting. As described in [1], k-means has a tendency to degenerate into singularities. In the above example, with two Gaussians, say we chose k=3. It is possible that one center might be at zero, such that it overlaps both actual means. This would cause a future clustering algorithm to be unable to adequately split the data represented by that point. However, if we knew that such a center represented a single point, we could discard it in this example. Simply eliminating all centers of size one is not adequate though. Suppose there are two points of random noise relatively close to each other. Under a condition of sufficient k, it is reasonable to assume that a center may focus on these two points, thus assigning more weight than is truly warranted during construction of the Affinity matrix. A way is needed to maximize $max(x_i \in c)$ while minimizing $min(\sum_{i=1}^{n} (\|x_i - c_k\|)$. From the perspective of input to spectral clustering, it can be shown that spectral clustering is analogous to graph partitioning problem [3]. The graph partitioning problem is then given a graph with edge weights, minimize the cut over the ratio of links to degree. The inputs to the spectral clustering algorithm then should provide as clear a definition between partitions as possible, while minimizing the distance between truly associated points.

3 Proposal

In order to satisfy the competing requirements of maximizing the number of points, x in each center c $max(x_i \in c)$ while minimizing the distance between each point x and the center c $min(\sum_{i=1}^{n} (\|x_i - c_k\|)$ simultaneously, we shall fix the maximization such that each center contains the same number of points. This allows us to minimize the ability to center on arbitrary points, and thus is designed to perform better on noisy data. A simple modification to Lloyd's algorithm accomplishes this, as seen in Algorithm 1, Robust Vector Quantization.

For each iteration the points are reassigned to the centroid in a stepped fashion. Instead of taking all points closer to a center than any other center as in Lloyd's, each center takes the one point closest to it that is not already assigned to a different point. This process is iter-

```
Input: Set of n points with k initialized centers
Output: k centers

Choose an initial set of k cluster centers
While the change in centers < some delta
    Calculate the distance from each center to each point
    While some point is unassigned
         Choose a center with the least number of points assigned to it
             Assign the closest unassigned point to this cluster
   End
   For each center
       Reassign the center according to the mean of all points assigned to it
   end
end
```

Program 1: Robust Vector Quantization

ated until all points are taken by centers. The centers are then recalculated using the member points. Unlike Lloyd's, the resultant clusters are not a Voronoi diagram, but a set of arbitrary polygons with potentially non-contiguous noise points. For purposes of analysis, the algorithm may be seen as Expectation Maximization. The update step is $n/k * \sum_{x_i \in c_j} x_i$ and the assignment step is $\max x_i \in c_j s.t. x_i \notin c_{k \neq j}$. As described in the experimental section, the algorithm performs similarly to k-means. Also note that the computational complexity is slightly increased compared to k-means to $O(ntklog(n)$ if we sort the distances. Similar to KASP, the overall complexity is $O(k^3) + O(ntklog(n))$. The effect of restricting the minimum size of a cluster may be seen as similar to structured risk minimization, SRM, where the ability to model arbitrarily small clusters of data is inversely similar to the VC dimension [9]. Upon limiting this ability, only choice of k may affect the ability of RVQ to model arbitrarily small clusters. Again, this is similar to the choice of the power of the classifier in SRM. The analogy breaks down in analysis of precisely what minimum number k is required, as there are no known classes to perform the analysis with.

4 Experiments

Robust Vector Quantization was applied to two problems to test the viability of the proposal. The first is a termed toy problem involving circles, which will demonstrate visually how RVQ is able to handle noise better than the standard Lloyd's algorithm. We will then review the results on an actual data set, the USPS handwritten digits. Initially, experiments were performed in R as in Yan et. al. However, for large datasets R proved to be too slow, as Yan et.al. also noted after several of their experiments took days to run. Thus, Matlab was chosen as the appropriate tool. To implement KASP, the built-in Matlab k-means was used, with parameters set such that a 10

The spectral library used was SpectraLib [10]. Specifically, the mcuts algorithm [7] with k-means used to determine the final clusters. For the affinity matrix, a complete graph was used, with the Radial Basis Function as the distance measure. For σ in the radial basis function, a simple search from .1 to 200 by .1 was performed, with .1 returning the optimal results on the 2D circles. For the handwritten digits, a method due to [2] was used to determine a range for σ automatically. The middle of the range was used,

	σ=1.5	σ=1.5	σ=2.0	σ=2.0	σ=2.5	σ=2.5
α	KASP	RVQ	KASP	RVQ	KASP	RVQ
4	74.04%	99.89%	61.47%	63.90%	58.65%	61.55%
8	66.50%	99.92%	60.60%	65.98%	57.51%	61.97%
25	86.18%	99.93%	62.14%	88.39%	56.92%	61.17%
50	98.50%	99.93%	69.04%	96.86%	59.11%	62.44%

Table 1: Accuracy of 2D Gaussian circles

to optimal effect. Since all data was spatial in nature, no normalization was performed.

The first problem was to separate two noisy 2D circles. The points were generated as Gaussian noise around the perimeter of both circles. The radii, 5 and 15, as well as the number of points, 1000 and 2000, were held constant while the standard deviation of the Gaussian for both circles was varied from .5 to 2.5. At each standard deviation, the number of points was varied with reduction parameters α of 4,8,25 and 50 for 750,375,120 and 60 points. Each parameter combination was run 30 times, and the averages used. The original Yan paper only attempted reduction levels of 4 and 8, but according to our theory, lower numbers of points should represent the original function better, to a point. The fraction of points which were clustered appropriately between the circles is listed in 1. For brevity, σ levels at 0.5 and 1 were left out, as the accurracy was 100% for all parameters.

As can be seen from the table, RVQ consistently outperforms standard k-means reductions. As expected, k-means performs better at higher noise levels when the number of points is reduced, while RVQ performs better across a wider range of representative points. Note that in all cases, RVQ provides equivalent or better clustering's than standard k-means, especially for larger numbers of representative points. For visual comparison, Figure 1 contains the original points and two comparative graphs using each method.

The experiment on real world data was performed using the USPS handwritten digits data set [5]. The data set consists of images of handwritten digits, where each image is a 16x16 grid of pixel intensities. For simplicity, a subset of the data containing only ones and zeros was used, 1100 points for each class. Reduction was then performed using k-means and RVQ with the same reduction parameters used for the previous experiment: 4, 8, 25, and 50. For comparison, K-means was run directly on the data, with no reduction level, and is included in the results for comparison. Again, all of the parameters were tested 30 times, and the averages used. The results are seen in Figure 2.

As seen in the graph, accuracy improves as the numbers of points are reduced, such that the noise is smoothed out. Again, we see that accuracy improves as the number of points is reduced for both KASP and RVQ. The jump in accuracy for levels above 25 in RVQ suggests that the actual minimum number of samples to perform well is below what was tested, and that further reductions might bring KASP within the same accuracy range as RVQ. However, as this is clustering not classification, further analysis would be required to determine this smaller range of optimal k's.

5 Conclusions

This paper has shown that while KASP performs well in general, under certain assumptions about noise and the nature of the data, it may be improved upon. Using RVQ, a more effective representation of the underlying points may be discovered under a wider range of representa-

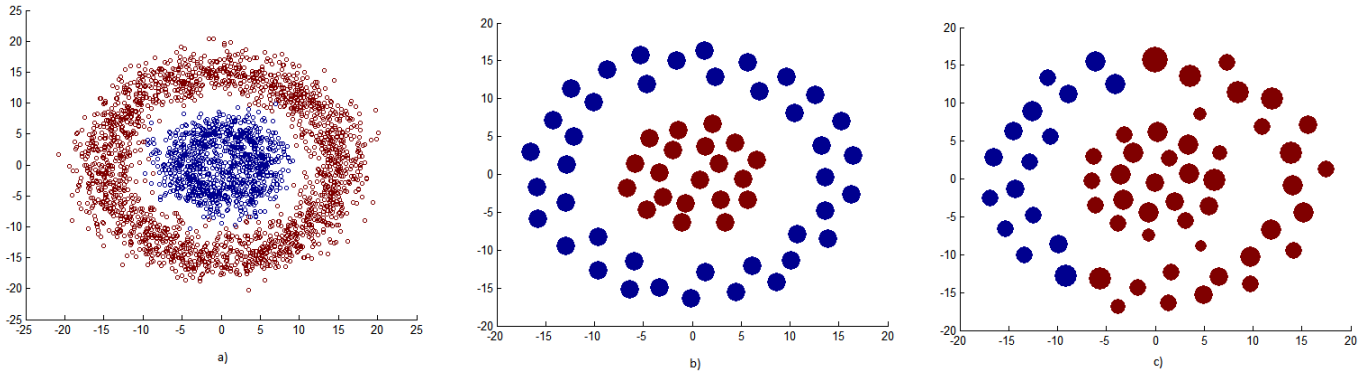

Figure 1: a) is the original circles, with colors representing the assignment given by the generating function as red and blue. b) is the RVQ reduction, with the final spectral classification. The size of the points is proportional to the number of representative points for both b) and c). c) is the k-means reduction, with the final spectral classification. Here too the size is proportional to number of representative points. Note the smaller points which lie between the two circles, and appear to have skewed the final clustering result.

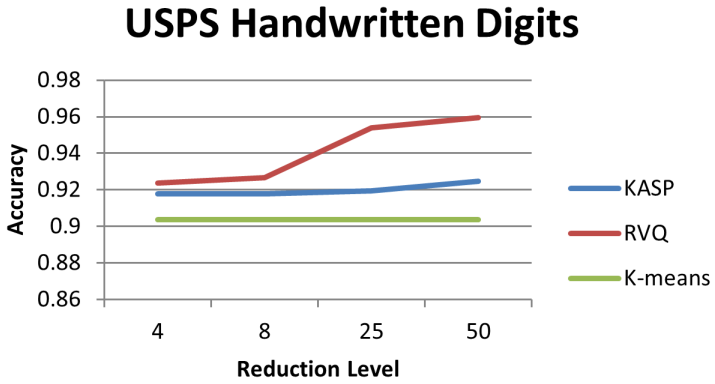

Figure 2: Accuracy of the three methods.

tive point choices. Future work includes a better understanding and formalization of the theoretical nature of over and under fitting in choosing representative points, as well as other potential algorithms for application of these ideas.

References

[1] C. M Bishop and SpringerLink (Online service). *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

[2] B Caputo, K Sim, F Furesjo, and A. Smola. Appearance-based object recognition using SVMs: which kernel should i use?, 2002.

[3] I. Dhillon, Y. Guan, and B. Kulis. *A unified view of kernel k-means, spectral clustering and graph cuts*. Citeseer, 2004.

[4] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystr\m method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 214225, 2004.

[5] J. J Hull. A database for handwritten text recognition research. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(5):550554, 2002.

[6] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395416, 2007.

[7] M. Meila and J. Shi. A random walks view of spectral segmentation. 2001.

[8] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14: Proceeding of the 2001 Conference*, page 849856, 2001.

[9] V. N. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264280, 1971.

[10] Deepak Verma. SpectraLib, 2003.

[11] D. Yan, L. Huang, and M. I Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 907916, 2009.

[12] S. X Yu and J. Shi. Multiclass spectral clustering. 2003.