Building Geodemographics on Parallel Graphics Processing Unit Architecture

1. Introduction

Geodemographic classification categorise small geographic areas into a series of discrete categories that aim to represent the multidimensional characteristics of individuals living within these neighbourhoods. Real-time geodemographic classification is the vision for an online and automated web based system that enables users to build, visualise and test a bespoke classification within a short time period (probably in minutes). There have been a number of technological advances which are enabling us to develop online systems for the creation of real-time classifications. This paper presents a summary of our research to date in this area, and cumulates in a pilot real-time geodemographic information system for specification, estimation and testing of classifications on the fly.

There are numerous methodologies for creating geodemographic classifications which differ based on the datasets used, the normalisation technique applied, the method of aggregation and finally, the visualisation techniques used. Geodemographic classifications are created by a clustering algorithm searching the attribute space of a matrix of standardised input data comprising a row for each small area (however defined) and a column for each attribute measure. For example, Vickers and Rees (2007) used *k*-means clustering for the creation of the National Statistics Output Area Classification (OAC) with data derived entirely from the 2001 Census of the Population. The *k*-means algorithm is a commonly used method for the geocomputation of geodemographic classification (Harris et al, 2005), however, in its original form, *k*-means is unstable and relatively sensitive to outlier values within the input data matrix. Because of this instability the algorithm requires multiple runs in order to ensure a robust result. For example, Singleton and Longley (2008) created a geodemographic classification using *k*-means with approximately 10,000 runs.

The geodemographic classification system described in this paper uses a parallel implementation of *k*-means (see Adnan et al, 2010) build upon NVIDIA's Computer Unified Device Architecture (CUDA)[i]. CUDA allows different processes to run in parallel on the Graphical Processing Units (GPUs) of NVIDIA's graphics cards enabling greater computational power than standard non parallel *k*-means clustering.

2. Clustering by parallel *k*-means

The *K*-means clustering algorithm has remained the core algorithm used in the creation of geodemographic classifications. *K*-means seeks to find a set of cluster centroids that minimises expression (3) below.

$$V = \sum_{x=1}^{n} \sum_{y=1}^{n} (z_x - \mu_y)^2 \qquad (1)$$

Where n is the number of clusters, μ_y is the mean centroid of all the points z_x in cluster y. The *k*-means algorithm assigns a set of n seeds within the data set and then proceeds by assigning each data point to its nearest seed. Cluster centroids are then created for each cluster, and the data points are

assigned to the nearest centroid. The algorithm, then, re-calculates the cluster centroids and repeats these steps until a convergence criterion is met (usually when the switching of data points no longer takes place between the clusters).

This paper presents a parallel implementation of the *k-means* algorithm using CUDA. CUDA is a general-purpose parallel computing architecture that uses the GPUs of NVIDIA graphics cards to solve complex computational problems. A typical CUDA enabled NVIDIA graphics card has a number of GPUs and a set of memory capable of storing a reasonably large amounts of data. For example, "GeForce 8400M GT" graphics card has 16 GPUs and 512MB of internal memory. CUDA requires that the computational problem to be programmed in the C language for parallel processing.

Our proposed parallel *k*-means algorithm via CUDA works as follows:

Total number of runs is specified by *N*.

a) Central Processing Unit (CPU) prepares the data points and counts the number of GPUs available on the NVIDIA graphics card. Afterwards the CPU uploads the data points and code instructing one *k*-means run to each GPU.

b) GPU performs *k*-means clustering on the data points by minimizing expression (1). When an optimal solution is achieved, GPU returns the result to CPU and claims the next *k*-means run from CPU if there are any.

c) CPU stores the results returned by GPUs in a local data structure contained in Random Access Memory (RAM). CPU keeps on delegating requests to GPUs until number of runs are less than *N*.

d) If number of runs is equal to *N*, CPU compares the "within sum of squares distance" optimisation criteria of all the runs.

e) The optimal solution is the one that has minimum "within sum of squares distance".

In order to compare the "computational time" of *k*-means and parallel *k*-means, we ran *k*-means and parallel *k*-means for (k=2-30) cluster solutions at Output Area level using the London datasets, and then compared the time taken for each algorithm to converge on a specified number of clusters. For each value of *k*, each algorithm was run 100 times and the results are shown in Figure 1. "Computational time" represents the time an algorithm takes to complete 100 iterations for each value of *k*. The hardware used for this evaluation comprised an "Intel Core2 Duo 2.10GHz" CPU, 4GB RAM, and "GeForce 8600M GS" NVIDIA graphics card. The graphics card has 16 GPUs and 512 MB of RAM.

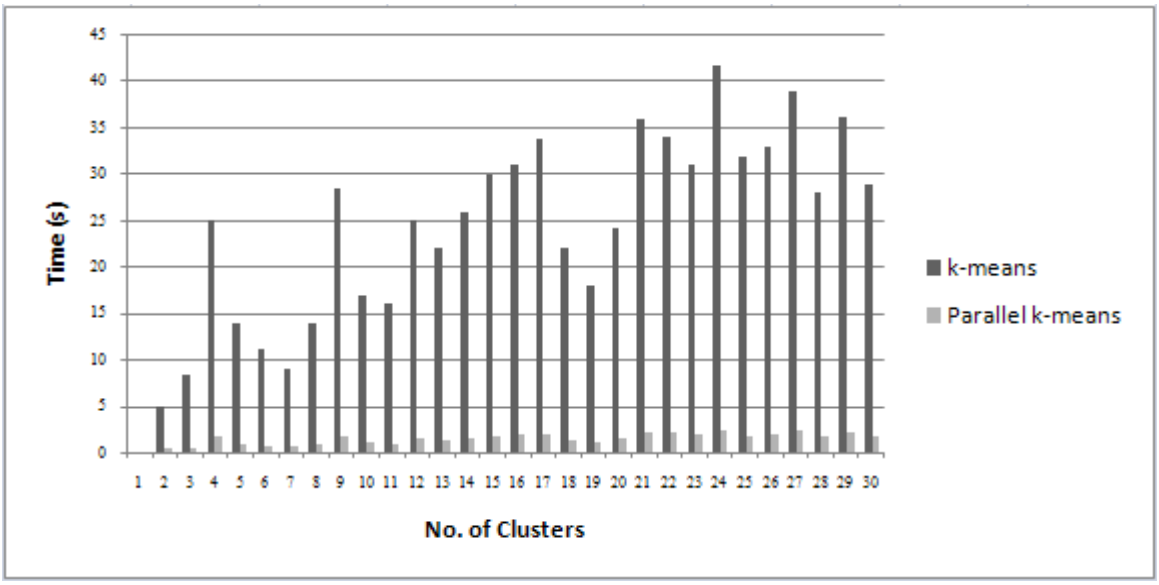

Figure 1: Output Area (OA) level results for the two clustering algorithms

Figure 1 indicates that parallel *k*-means is a lot faster than *k*-means clustering algorithm, and thus is the best choice for an online geodemographic system.

3. Creating a bespoke real-time geodemographic classification

A real-time geodemographic information system produces a classification in four steps which are Specification, Normalisation, clustering by Parallel *k*-means, and Visualisation. In the first step, user selects variables and their weightings. Weighting describes the importance of variables in the classification. User also specifies the number of Geodemographic Classes. In the second step, information system normalises the data using one of the normalisation techniques e.g. Z-scores, Range Standardisation, or Principal Component Analysis. In the third step, the system clusters the data using Parallel *k*-means clustering algorithm. In the final step, the information system shows the result in the form of maps and statistics.

We can represent the real-time geodemographic information system as a block diagram with different components communicating with each other. Following Figure 2 illustrates this.

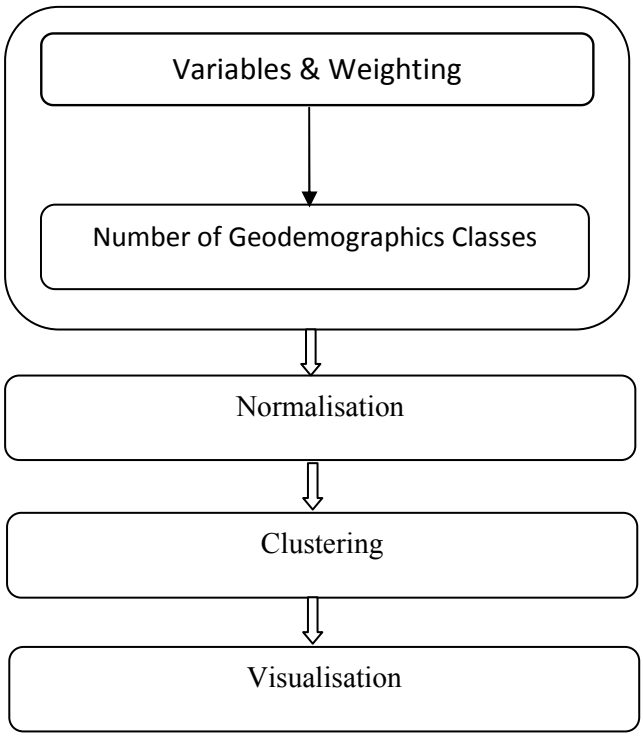

Figure 2: Block diagram of a real-time Geodemographics information System

The remainder of this paper outlines our beta real-time geodemographics information system. This uses the 2001 Census inputs to the National Statistics Output Area Classification (Vickers & Rees, 2007) aggregated at Output Area (OA). The normalisation technique incorporated into the system is z-score, and it uses parallel *k*-means to cluster the data.

3.1 Specification of Inputs

First step in creating a classification is the specification of input variables and an assignment of a weight of relative importance. This is shown in Figure 3 where the 'Born outside the UK' variable will have highest weight in the output classification.

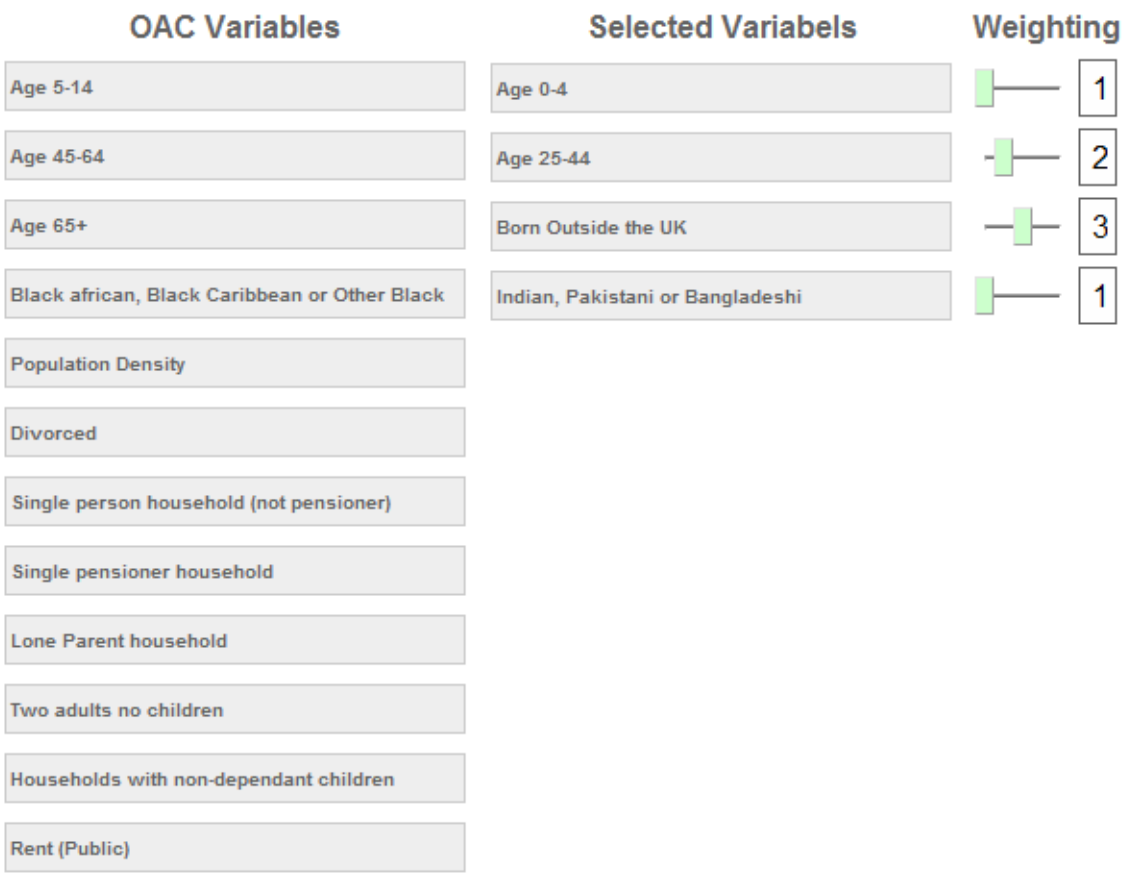

Figure 3: Specification of variables and their weight

After variables have been selected, the number of classes in the output classification can be specified. This is shown in Figure 4.

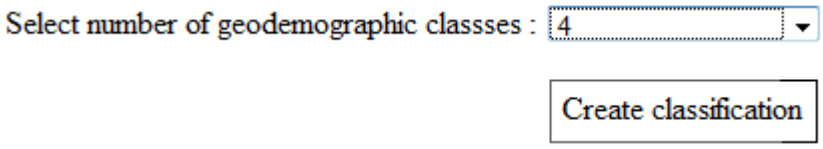

Figure 4: Specification of the number of geodemographic classes

3.2 Results

Based on the previous selected inputs, the system produced a classification for London. This is shown in Figure 5.

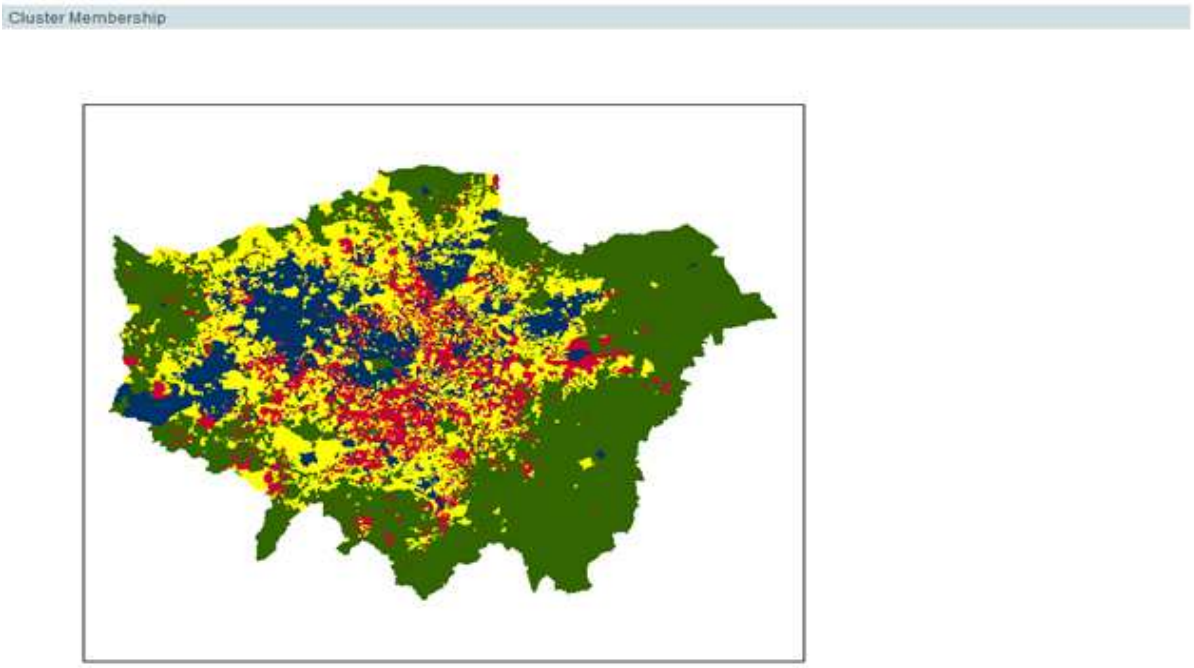

Figure 5: Classification produced for London based on selected variables

The system also gives considerable information about the sizes of clusters, which is important when the objective of building a classification is to produce clusters of reasonably equal sizes.

Within Sum of Squares

Within sum of squarues : 30311.46216817008

Clusters

Cluster No.	Cluster Size
1	4659.0
2	5285.0
3	6934.0
4	7262.0

Figure 6: Cluster Membership and Within Sum of Squares

4. Conclusion and Future Research

This paper has presented our pilot real-time geodemographic classification system based on CUDA parallel infrastructure. The system enables users to compile geodemographic classifications quickly (possibly within minutes) utilising the multiple processor architecture of graphics cards. Given that these technologies are now available as part of typical data centre and cloud architectures (e.g. Amazon EC2) we see this as a very scalable solution which could compile classifications based on inputs for more extensive geographies.

Future research aims to evolve the testing procedures used to produce the classifications. Also, alternate clustering algorithms could be incorporated into the system to allow users more flexibility when creating geodemographic classifications.

5. References

Adnan, M., Longley, P.A., Singleton, A.D., Brunsdon, C. (2010) Towards Real-Time Geodemographics: Clustering Algorithm Performance for Large Multidimensional Spatial Databases. Transactions in GIS, 14(3), 283 – 297.

Harris, R., Sleight, P., Webber, R. (2005). Geodemographics, GIS and Neighbourhood Targeting. Wiley, London.

Singleton, A.D., Longley, P.A (2008). Creating open source geodemographic classifications for Higher Education applications. Papers in Regional Science, 88(3), 643-666.

Vickers, D.W. and Rees, P.H. (2007). Creating the National Statistics 2001 Output Area Classification. Journal of the Royal Statistical Society, Series A. 170(2), 379-403.

[i] For more information on CUDA see the Nvidia website: http://www.nvidia.com/object/cuda_home.html