Exploratory Multilevel Hot Spot Analysis: Australian Taxation Office Case Study

Denny[1,2] **Graham J. Williams**[3,1] **Peter Christen**[1]

[1] Department of Computer Science,
The Australian National University,
Canberra 0200, Australia,
Email: denny@cs.anu.edu.au, peter.christen@anu.edu.au

[2] Faculty of Computer Science,
University of Indonesia

[3] The Australian Taxation Office,
Email: graham.williams@ato.gov.au

Abstract

Population based real-life datasets often contain smaller clusters of unusual sub-populations. While these clusters, called 'hot spots', are small and sparse, they are usually of special interest to an analyst. In this paper we introduce a visual drill-down Self-Organizing Map (SOM)-based approach to explore such hot spots characteristics in real-life datasets. Iterative clustering algorithms (such as k-means) and SOM are not designed to show these small and sparse clusters in detail. The feasibility of our approach is demonstrated using a large real life dataset from the Australian Taxation Office.

Keywords: self-organizing maps, cluster analysis, neural network, imbalanced data, drill-down, visualization.

1 Introduction

Cluster analysis is often used to help in understanding and dealing with the complexities of large datasets. For example, it may be easier to devise marketing strategies based on groupings of customers sharing similar characteristics because the number of groupings/clusters can be small enough to make the task manageable.

Self-Organizing Map (SOM) (Kohonen 1982) is a popular tool for cluster analysis for several reasons. First, SOM performs topological mapping from high-dimensional data into a two-dimensional map where similar entities are placed nearby. Second, SOM performs vector quantization which produces a smaller representative dataset that follows the distribution of the original dataset. Third, SOM offers various visualizations which are relatively easy to interpret for non-technical users when exploring a dataset. Applications of SOM for cluster analysis can be found in many domains, such as health (Markey et al. 2003, Viveros et al. 1996) or marketing (Dolnicar 1997).

In real life, cluster sizes are normally not equal and clusters do not have the same interestingness. Distribution of clusters is often very skewed as captured by the Pareto distribution (Pareto 1972) also known as the "80:20 rule". Thus, the interesting clusters are usually only a small fraction of a dataset. Furthermore, the variance of items at the tail or margin of the normal distribution of a population is also larger compared to the center of the normal distribution. In other words, in real life it is common to find large dense clusters for common sub-populations and small sparse clusters for interesting sub-populations. In a taxation context this could be a group of tax entities who have a tax debt, while in an insurance context this could be a group of high claiming clients. Williams (1999) proposed the hot spots methodology that aims to identify important or interesting groups in a very large dataset. The methodology uses a combination of clustering and rule induction. As a result, business organizations can make improvements on their strategies, such as treatment strategies to improve tax compliance, by understanding these small and interesting clusters that are called hot spots. It can be interesting to analyze these hot spots in relation to the whole population.

However, iterative clustering algorithms (such as k-means) and SOM tend to merge these small sparse clusters, thus reducing the ability to analyze them in detail. The k-means algorithm tries to generate a relatively uniform distribution on the cluster sizes as shown by Xiong et al. (2006). As a result, k-means is unsuitable for highly skewed datasets.

When SOM is used for cluster analysis, it also has similar issues. Increasing the map size of a SOM only gives a better resolution map (in terms of lower quantization error and finer cluster borders) but with significant additional computational cost. However, an increased map size does not provide extra information about these small and sparse clusters. Small sparse clusters are represented as a few nodes in a SOM, which reduces the capability to characterize them.

Hierarchical clustering algorithms (Han & Kamber 2006), on the other hand, require high computational resources, thus making them impractical for very large datasets. Furthermore, different definitions of between cluster distances (such as minimum, maximum, or average distance) will often produce different clustering results. Moreover, the definition of the between cluster distance has to be determined beforehand.

Therefore, the approach presented in this paper is aimed to help analysts to identify and understand hot spots behaviour. The main contribution of our approach is drill-down hot spot exploration using SOM-based visualizations that capable in handling imbalanced data.

The rest of the paper is organized as follows. Section 2 briefly introduces SOMs and explain their limi-

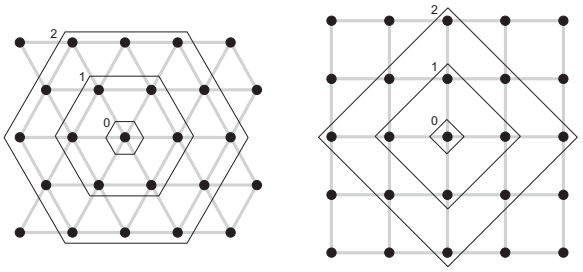

Figure 1: Local lattice structure: hexagonal topology (left) and rectangular topology (right) and its neighbourhood radius in the map space (adapted from Vesanto et al. (2000)).

tation for analyzing hot spots. Section 3 reviews current SOM-based clustering techniques. Our approach is discussed in Section 4 and Section 5 discusses the results of our experiments with a real life dataset from the taxation domain.

2 Self-Organizing Maps

A SOM is an artificial neural network that performs unsupervised competitive learning (Kohonen 1982). Importantly, SOMs allow the visualization and exploration of a high-dimensional data space by non-linearly projecting it onto a lower-dimensional manifold, most commonly a 2-D plane (Kohonen 2001). Artificial neurons are arranged on a low-dimensional grid. Each neuron i has an n-dimensional prototype vector, m_i, also known as a weight or codebook vector, where n is the dimensionality of the input data. Each neuron is connected to neighbouring neurons, determining the topology of the map. In a hexagonal grid, each neuron is connected to six neighbours, while in a rectangular grid each neuron is connected to four neighbours, as shown in Figure 1. In the map space, neighbours are equidistant.

SOMs are trained by presenting data vectors to the map and adjusting the prototype vectors accordingly. These prototype vectors are initialized to different values. There are two approaches to training a SOM: sequential training and batch training. In sequential training, one data vector is presented to the map at a time and the prototype vectors are updated. On the other hand, in batch training, the whole dataset is presented to the map and all prototype vectors are updated at once.

In sequential training, the training vectors can be taken from the dataset in random order, or cyclically. At each training step t, the *Best Matching Unit* (BMU) b_i for training data vector x_i, i.e. the prototype vector m_j closest to the training data vector x_i, is selected from the map according to Equation 1:

$$\forall j, \qquad \|x_i - m_{b_i}(t)\| \leq \|x_i - m_j(t)\|, \qquad (1)$$

where only non-missing values are used in the distance calculation. Then, the prototype vectors of node b_i and its neighbours are moved closer to x_i:

$$m_j(t+1) = m_j(t) + \alpha(t) h_{b_i j}(t)[x_i - m_j(t)], \qquad (2)$$

where $\alpha(t)$ is the learning rate (a tuning parameter) and $h_{b_i j}(t)$ is the neighbourhood function (often Gaussian) centered on b_i. This process of updating the prototype vectors is repeated until a predefined number of iteration or epochs is completed. Both $\alpha(t)$ and the radius of $h_{b_i j}(t)$ are decreased after each iteration. Since the time complexity of SOMs is linear in the number of prototype vectors, number of data vectors, and number of iteration, SOMs are able to cope with large and high-dimensional datasets.

In the batch algorithm, the values of new prototype vectors are the weighted averages of the training data vectors that are mapped to m_j and its neighbours, where the weight is the neighbourhood kernel value $h_{b_i j}$ centered on unit b_i (Kohonen 2001). The new prototype vectors are calculated using Equation 3.

$$m_j(t+1) = \frac{\sum_{i=1}^{N} h_{b_i j}(t) x_i}{\sum_{i=1}^{N} h_{b_i j}(t)}, \qquad (3)$$

where N is the number of training data vectors. SOM is capable in handling missing values, as Equation 3 only performs summation and counting of the non-missing values.

The batch algorithm is similar to k-means. The difference is that the batch algorithm uses weights in calculating the new 'centroids' that are based on the chosen neighbourhood kernel function, while k-means assigns the same weight (weight of one for data vectors assigned to a cluster, weight of zero for the rest) when calculating the centroids.

The map is usually trained in two phases: rough training phase and fine tuning phase. The rough training phase usually has shorter training length and wider initial radius compared to fine tuning phase. In the rough phase, the learning rate $\alpha(t)$ and the radius of $h_{b_i j}(t)$ decrease in a faster rate compared to the fine tuning phase.

After a SOM is trained using a real life dataset, the common population is usually located in the center of the map and the remainder at the border, because of the topologically ordering property and the neighbourhood kernel function used in the training. In real life datasets, the remainder of a population usually has a few different characteristics compared to the common population. For example, in a taxation context, entities who rely mainly on salary and wages for income are mapped onto the center of the map since they are the common population. Other entities might have a few variations, such as having salary and wages and interest income; or having salary and wages, interest, and dividend income.

Since we are interested in the hot spots or 'uncommon but interesting clusters', these clusters are usually located at the border of the map. However, SOMs have a problem with an issue called the *border effect* (Kohonen 2001). The neighbourhood definition is not symmetric at the borders of the map. As shown in Figure 1, the number of neighbours per unit on the border and corner of the map is not equal to the number of neighbours in the middle of the map. Therefore, the density estimation for the border units is different to the units in the middle of the map (Kohonen 2001). As a result, the tails of the marginal distributions of variables (normally located at border units) are less well represented than their centers. As we are interested in hot spots, and these hot spots are usually located at the borders of the map, there is a need to address this problem.

Besides the single level SOM proposed originally by Kohonen (1982), there are SOMs with hierarchical structure, such as Hierarchical SOM (Koikkalainen & Oja 1990) and Growing-Hierarchical SOM (Dittenbach et al. 2000). In these approaches, only one node can be drilled down to the next level. The problem of drilling down only one node at a time is that the Voronoi border of the prototype vector in a sparse area might not be a good cut of the entities in a hot spot area. Furthermore, the goal of Hierarchical SOM is to achieve lower computational cost by using a Tree-Structured SOM to find a BMU faster.

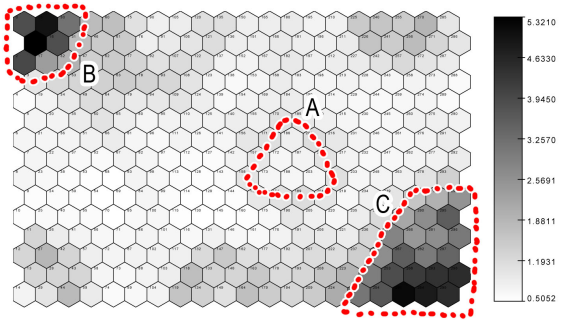

Figure 2: The distance matrix visualization of the whole population dataset, where distance is the median of distances a node to its neighbours.

In our approach, several nodes can be selected to be drilled down interactively by feedback from the user.

3 SOM-based clustering

As mentioned earlier, SOMs perform vector quantization and projection to a 2-D map, and have a topology-retaining property. This makes SOMs suitable for clustering data based on their spatial relationships on the map using visualizations. Existing SOM-based clustering methods can be categorized into visualization based clustering, direct clustering, and two-level clustering (hybrid) as discussed below.

A rough cluster structure can be observed using a distance-matrix based visualization. The distance-matrix based visualization, such as u-matrix visualization (Iivarinen et al. 1994), shows distances between neighbouring nodes using a colour scale representation on a map grid, as shown in Figure 2[1]. As shown in the colour bar, white indicates a short distance between a node and its neighbouring nodes, while black indicates a long distance between the node and its neighbours. The distance matrix visualization methods can be used to show borders between clusters. Long distances that show highly dissimilar features between neighbouring nodes divide clusters, i.e. the dense parts of the maps with similar features (white regions) (Iivarinen et al. 1994). In other words, the distances of the neighbouring units in the data space are represented using shades of colour in the map space.

By using this visualization, users can see the cluster structure of the dense part of the map, for example the center of the map (region marked 'A') in Figure 2. However, it is difficult to see the cluster structure of the sparse parts at the lower-right and the upper-left corners of the map (regions marked 'B' and 'C').

Another method to analyze a hierarchical cluster structure is by using a variant of the data hit histogram that shows how many data vectors are mapped to each node. This is called "Smoothed Data Histogram" (SDH) and proposed by Pampalk et al. (2002). In this visualization technique, each data vector is mapped to its s closest units (BMU) with a linearly decreasing membership degree. The first BMU has a s/c_s degree of membership, the second BMU has a $(s-1)/c_s$, and so forth for the s closest units. The remainder units have zero degree of membership. Pampalk et al. (2002) define $c_s = \sum_{i=0}^{s-1}(s-i)$ to ensure the total membership of each data item adds up to 1. They argue that a hierarchical cluster structure in the data can be observed by changing the value of s. The drawback of this visualization technique is sensitive to the parameter s. The authors did not give any heuristics to choose a suitable value of s. They argued that the optimal value of the smoothing parameter depends on an application. Furthermore, large values of s will give more value to the units at the center of the map due to the topological ordering property of a SOM.

This technique might be able to visualize cluster structure of the dense parts of the map. However, this approach cannot show the hierarchical structure of a sparse part (hot spot) of a map due to the limitation of SOM as described in Section 2.

In direct clustering, each map unit is treated as a cluster, its members being the data vectors for which it is the BMU. This approach has been applied to a breast cancer database (Markey et al. 2003), to a health insurance industry (Viveros et al. 1996) and for market segmentation (Dolnicar 1997).

A disadvantage is that the map resolution must match the desired number of clusters, which must be determined in advance. Furthermore, taking each map unit as a cluster centroid does not guarantee that the clustering result will minimize within-cluster distances and maximize between-cluster distances since SOMs will produce more units for large clusters. Again, this technique cannot show the cluster structure of the sparse part of a map due to the limitation of SOM.

In contrast to direct clustering, in two-level clustering, the units of a trained SOM are treated as 'proto-clusters' serving as an abstraction of the dataset (Vesanto & Alhoniemi 2000). Their prototype vectors are clustered using a traditional clustering technique, such as k-means or agglomerative hierarchical clustering, to form the final clusters. Each data vector belongs to the same cluster as its BMU.

When a SOM is used in the first level of the procedure, it leads to two advantages. Firstly, the original data vectors are characterized by a considerably smaller-sized set of prototype vectors, allowing efficient use of clustering algorithms to divide the prototypes into groups, as shown by Vesanto & Alhoniemi (2000). As a result, this approach is suitable for large or high-dimensional datasets, such as genome data, and for obtaining an initial understanding of possible clusters. For example, after the optimal number of clusters is decided, based on data exploration of the clustering of the maps, clustering with that number of clusters can be performed directly on the data vectors instead of on the prototype vectors, if desired. Furthermore, it allows a visual presentation and interpretation of the clusters via the 2-D grid.

The two-level clustering method also has the same drawback as the previously mentioned methods, as it also uses SOM as the abstraction layer. It is not possible to see the cluster structure of the sparse part of the map, even when using an agglomerative hierarchical clustering on top of the map.

In detecting changes in cluster structure using SOM, Denny & Squire (2005) used two level clustering as described previously and multiple visualization linking to show how clusters change over time, such as emerging clusters, missing clusters, enlarging clusters, and shrinking clusters. Their method were tested using synthetic and real-life datasets using the World Development Indicator data published by the World Bank (World Bank 2003). The results verify that the methods are capable of revealing changes in cluster structure, corresponding to known changes in economic fortunes of countries.

[1]All the SOM figures were originally in colour. For printing purposes, they were converted into gray scale and therefore some details are lost. In the original version, for example, low values are represented as shades of blue and high values are represented as shades of reds.

4 Our Visual Drill-Down Approach

Our visual SOM drill-down approach is applied to the task of exploring taxpayer compliance, in the context of a project with the Australian Taxation Office (ATO) and using a de-identified client dataset. In this section, we discuss data pre-processing, map training, identifying hot spots, and drilling-down the hot spots.

4.1 Dataset

Due to data confidentiality, the complete data description and results cannot be shown in this paper. However, we do provide aggregate indicative results that demonstrate the effectiveness of our approach.

The motivation of the analysis is to understand the logic and structures that drive tax payers' compliance behaviour (behavioural archetypes). The idea is to construct 'psychographic groups' (Wells 1975) by using data mining. Understanding the difference between low and high risk tax payers will be valuable for the ATO.

The archetype dataset consists of about 6.5 million entities with 89 numerical attributes which reflect tax payers behaviour. In general, these attributes can be categorized into: income profile (amount and proportion of each income source), propensity to lodge correctly and on time (lodgement profile), propensity/capacity to pay (debt profile), market segments, demographics, Socio-Economic Indicators for Areas (SEIFA) (Trewin 2003), and involvement in tax avoidance schemes. These attributes were manually selected by the ATO's analysts.

4.2 Data Preprocessing

In distance-based clustering methods, it is important to perform normalization prior to clustering since attributes might have different scale/range (Han & Kamber 2006). Without normalization, attributes with larger ranges will have more influence on the distance measurement. Common normalization technique are: z-score normalization, min-max normalization, and decimal scaling.

In the dataset, we found that some attributes have a large range to variance ratio. When all of the attributes in the dataset are normalized using z-score, the normalized values of these attributes will still have larger ranges.

The range of the z-score normalized value ($range_{A'}$) can be calculated as the range in the original dataset ($range_A$) divided by the standard deviation of the original dataset (σ_A) as shown below. The normalized value v' of attribute A can be calculated by: $v' = \frac{v-\overline{A}}{\sigma_A}$.

$$\begin{aligned} range_{A'} &= max_{A'} - min_{A'} \\ &= \frac{max_A - \overline{A}}{\sigma_A} - \frac{min - \overline{A}}{\sigma_A} \\ &= \frac{max_A - min_A}{\sigma_A} = \frac{range_A}{\sigma_A} \end{aligned}$$

where $\overline{A}$ is the mean, min_A and max_A are the minimum and maximum value of the original attribute values, and $min_{A'}$ and $max_{A'}$ are the minimum and maximum of the normalized values. Therefore, when an attribute has a large range to variance ratio, the range of the normalized value would be high, outweighing other attributes in the distance calculation. Therefore, it is suggested to use a mixed normalization method, such as z-score and min-max normalization, or use weight coefficients in the distance calculation.

As SOMs can only handle numerical attributes, all non-numerical attributes have to be transformed into numerical attributes. Categorical attributes, such as market segmentation and lodgement channel, are converted into numerical attributes by encoding each categorical value into a binary attribute. Furthermore, some numerical attributes that can have negative and positive values are split into two new variables that only contain the positive values or only the negative values to make it easier to interpret the result.

4.3 Map Training

The map is initialized using linear initialization (Kohonen 2001), and trained in two phases using batch training. In linear initialization, the prototype vectors are initialized based on the two largest principal components. Linear initialization is chosen over random initialization because it speeds up the learning process by an order of magnitude by having shorter training lengths (Kaski & Kohonen 1998). Furthermore, linear initialization combined with batch training will produce the same map if the learning process were redone. Random initialization might produce different orientations of the map.

Batch training is chosen because it produces more stable asymptotic values for the prototype vectors and it does not have the convergence problem of sequential training (Kohonen 2001). Furthermore, with a batch training algorithm, it is possible to utilize multi-processor environments to speed up the training process.

The map size, training length, initial and final radius are chosen by considering the best practice, as suggested by Vesanto et al. (2000).

4.4 Identifying Hot Spots in Self-Organizing Maps

Generally, in business, users are more interested in "abnormal clusters" or hot spots (e.g. clusters of entities who have debts) than "normal clusters". Hot spots in SOMs can be identified by two approaches, by using the distance matrix visualizations as well as analysts' feedback based on component plane visualizations.

With the idea that entities in hot spots are usually less homogenous because they are often located at the tail of distributions compared to the common/regular entities, these regions can be identified by using the distance matrix. Using distance matrix visualizations, homogenous groups (low variation) will have shorter neighbour distances (the white regions) compared to high variation groups (the dark regions) as shown in Figure 2. Then, regions that have longer distances should be investigated further by using component plane visualizations.

Component planes show the spread of values of a certain component of all prototype vectors in a SOM (Tryba et al. 1989). The value of a component in a node is the 'average' value of entities in the node and its neighbours according to the neighbourhood function and the final radius used in the final training (Equations 2 and 3). The colour coding of the map is created based on the maximum and the minimum values of the component of the map. In this paper, we use the 'gray' colour map where the maximum value is assigned black and the minimum value is assigned white. Component planes can be used to see interesting cluster patterns and correlations between variables (Himberg 1998, Vesanto 1999)

In Figure 2, there are two hot spots according to the aforementioned criteria, one in the top-left corner (region marked 'B') and another one in the bottom-right corner (region marked 'C'). According the component planes, such as the component plane of the

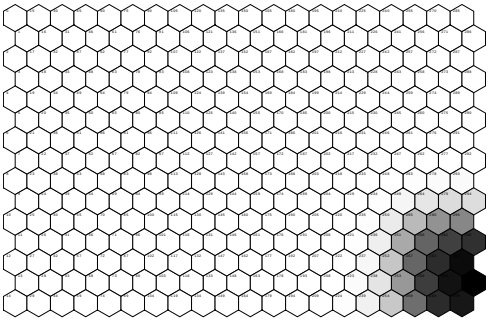

Figure 3: Component plane of 'number of debt cases' of the whole population.

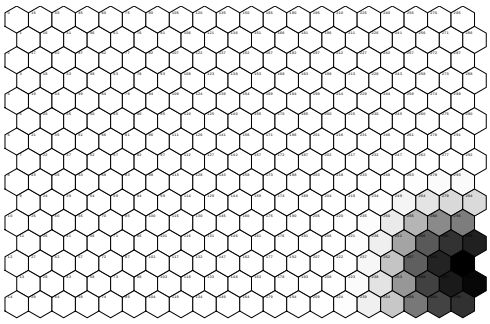

Figure 4: Component plane of 'number of debt cases paid' of the whole population.

number of debt cases as shown in Figure 3, and domain expertise, the hot spot in the bottom-right corner is more interesting than the one in the top-left corner. The bottom-right corner region consists of entities who have debt, have high taxable income, are involved in tax avoidance schemes, and have high risk scores. The top-left corner, on the other hand, consists of entities who received allowances and have more amendments.

The entities in the bottom-right region have highly dissimilar characteristics. However, at this level, it is difficult to differentiate the debt behaviour as shown in Figures 3 and 4. Therefore, it is a good idea to drill down into this region as discussed in the next section.

In identifying hot spots, the domain knowledge of analysts is invaluable because some attributes are more interesting compared to others. In this case, for example: involvement in tax avoidance schemes, lodgement behaviours, number of debt cases, and taxable income, are more interesting in identifying hot spots compared to market segmentation.

4.5 Drill Down and Visualizing Hot Spots

After analysts choose a part of the top level map (distinguish this group as a hot spot) that is interesting to be explored, a sub-map of the region is trained using entities that are mapped to the chosen region. Some issues that need to be taken care of in training the sub-map are: consistency of interpretation of the visualization of the sub-map, and maintaining the sub-map quality with respect to the sub-population.

In order to make interpretation of the visualization of the sub-map consistent to the analysts, the orientation of the map should be preserved and the colour coding should be consistent. The drawback of using linear initialization for the sub-map based on the entities in the sub-map is that the orientation of the sub-map might be different to the orientation of the top level map. For example, the debt entities were located at the bottom-right corner of the top level map but they might be located at the top-left corner as we drill down. This might confuse the user. This could happen when the two largest principal components of the whole population and the sub-population are different.

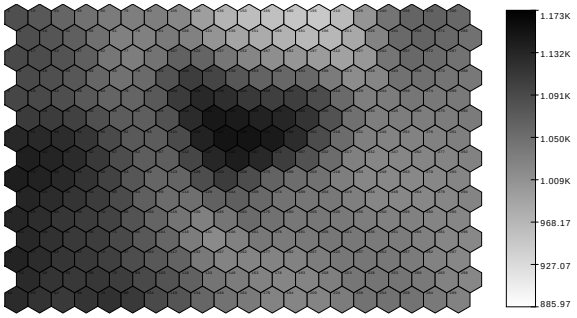

Figure 5: Component plane of SEIFA of the sub-map of region marked 'C' in Figure 2.

Therefore, it is suggested that the top level map is used as the initial map of the sub-map. The radius of the rough phase training should be wide enough, otherwise parts of the map might be empty (no entities mapped to particular nodes). Therefore, as a guide, the initial radius of the rough phase can be half of the longest side and the initial radius of the fine tune phase can be a quarter of the longest side.

The sub-map can be visualized using distance matrix visualization and component plane visualization. In order to show the distribution of values of the sub-map with respect to the whole population, it is suggested that when showing the component planes of the sub-map, the colour map used for the whole population, as described in Section 4.4, is used to visualize the component planes of the sub-map. In other words, black colour in the sub-map visualizations is used for the maximum value of the component of the top level map, not the maximum value of the component of the sub map. For example, Figure 5 shows the distribution of Socio-Economic Indicator for Areas of the bottom-right corner of the whole map. As the sub-map has better quality in terms of quantization error (more homogenous/less variation of the entities mapped to a node), the component value in the sub-map might exceed the maximum value of the whole map. The colour for values more than the maximum value of the whole map would be black as well. Therefore, when a cluster of black nodes appears in the visualization, it is possible that the values are actually exceeding the values of black in the colour bar.

The training of the sub-map will be considerably faster than training of the whole population as the number of data vectors mapped to the region are considerably smaller. Therefore, it is possible for users to explore hot spots interactively.

5 Results and Discussion

To interpret multiple visualizations, analysts need to understand that these visualizations are linked by position or by colour. Visualization of the same map is linked by position, which means that the position of each entity remains the same in each visualization. For example, Figures 2, 3, and 4 are linked by position. Visualization of the whole map and the sub map is linked by colour as described previously. The colour map of the top level map is used as the colour map in the sub-map.

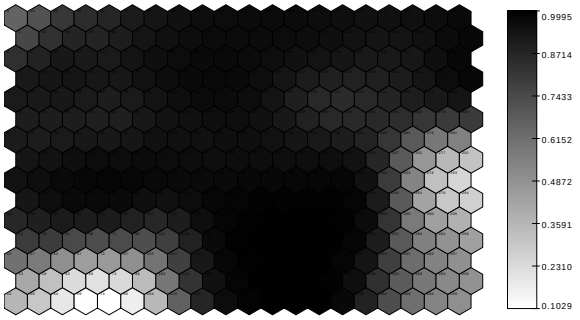

Figure 6: Component plane of 'employee market' of the whole population. Value of 1.0 means that the node consists of 100% employees.

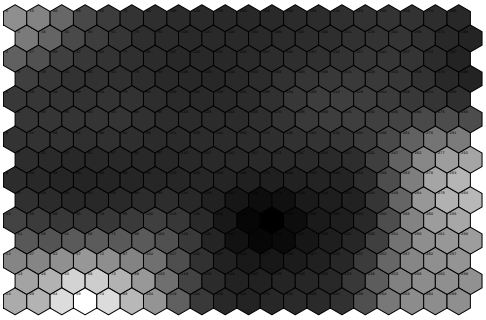

Figure 7: Component plane of percentage of salary and wages to total income of the whole population.

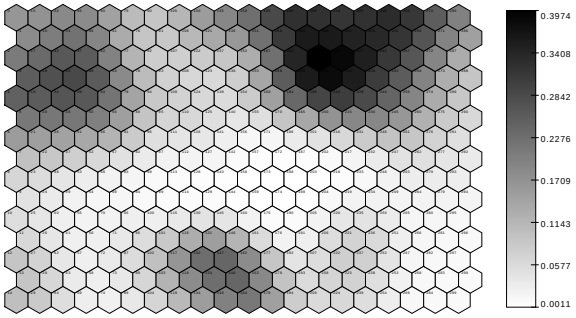

Figure 8: Component plane of 'usage of e-tax lodgement channel' of the whole population.

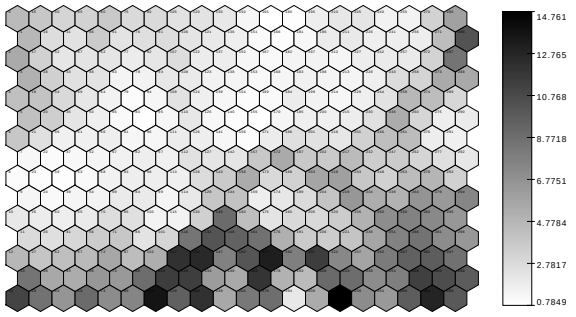

Figure 9: Distance matrix visualization of the sub-map of region marked 'C' in Figure 2.

In our experiments, the map size is 15x30, with hexagonal lattice structure. The initial radius of the rough phase and the fine tune phase are 8 and 4 respectively. The training length for the rough phase and the fine tuning phase are 6 and 10 epochs, respectively. The training processes took about 5 hours on a Debian GNU/Linux machine with two 64-bit AMD dual-core 3 GHz processors and 16 GB memory using our Java SOM Toolbox[2].

As discussed in Section 2, the common population in a real life dataset are usually located in the center of the map. The entities in the center of the map of the whole population are relatively homogenous as shown in Figure 2. Based on the component plane visualizations, this common population mainly consists of employees (Figure 6) with salary and wages as the main source of income (Figure 7).

At this level, we can see that e-tax[3] is an income tax return lodgement channel that is commonly used by employees, as shown in Figure 8. This is as expected since their tax returns generally tend to be simpler. The usage of the e-tax lodgement channel can be further optimized since, as a group, only 40% of the entities mapped to the darkest nodes of the map were using this channel. The information can be useful, for example, in deciding whether to promote e-tax directly to groups of other (similar) tax payers who may benefit from using this lodgement channel.

At the whole population level, it is not possible to differentiate debt behaviours because these entities are mapped to a small number of units at the lower-right corner of the map, as shown in Figures 3 and 4. Debt behaviour can be differentiated by observing debt-related attributes of this sub-population, such as total payment arrangements made, total default payment arrangements, total finalized payment arrangements, and age of debt.

In order to see the debt behaviour in detail, we drill down the lower-right corner of the top level map as explained in the previous section. At this level, we can also use a distance matrix (Figure 9) visualization to highlight the hot spot at this sub-map. In Figure 9, they are located at the bottom of the map.

In the sub-map, we are able to identify a group with characteristics of nearly all of the debt cases paid (Figures 10 and 11) but with a higher stage of compliance enforcement taken by the ATO. It is interesting to note that these entities also live in areas with slightly above average Social-Economic Indicator for Areas (Figure 5) which could mean that they might have the capacity to pay. This kind of analysis is not possible at the whole population level, as these entities are squeezed into a few nodes over the whole map which makes it difficult to differentiate.

It is also interesting to note that the hot spot of the sub-map consists of entities that are involved in

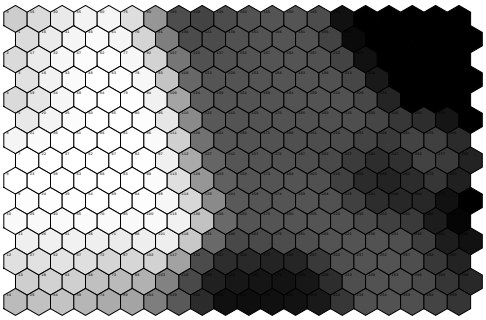

Figure 10: Component plane of 'number of debt cases' of the sub-map of region marked 'C' in Figure 2.

[2]Contact the author if you are interested in using the JavaSOM-Toolbox.

[3]http://www.ato.gov.au/etax

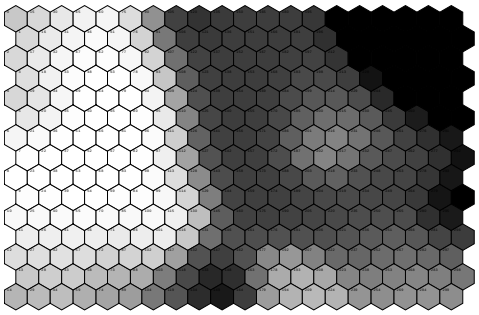

Figure 11: Component plane of 'number of debt cases paid' of the sub-map of region marked 'C' in Figure 2.

tax avoidance activities. Furthermore, this group has characteristics of longer debt age, higher stage of compliance enforcement taken by the ATO, and lower percentage of cases paid.

6 Conclusion and Future Work

We have highlighted the use of SOMs in exploring hot spots in a large real world dataset from the taxation domain. Based on our experiments, our approach is an effective tool for hot spots exploration since it offers visualizations that are easy to understand for non-technical users. Moreover, SOMs are able to handle missing values, are computationally feasible for large datasets, and are able to exploit multi-processor environments. Furthermore, in using our approach, users do not have to determine the number of clusters nor the between-cluster distance definition beforehand.

With our approach, users are able to select regions to drill down, whereas in agglomerative clustering algorithms, the between-cluster distance formula dictate how the population is split. Therefore, the user would be able to select regions/clusters based on their business drivers/needs. This is particularly useful as some attributes have higher importance compared to others.

This work is part of a larger research project where we are interested in observing the dynamics of hot spots over time such as to find entities who are moving in or out of hot spots. Such knowledge would be valuable as the analysts can derive strategies to encourage or to deter people to move in or out of the hot spots; or to evaluate effectiveness of their implemented strategies.

Acknowledgement

This research has been supported by the Australian Taxation Office and the authors express their gratitude to Grant Brodie, Georgina Breen, Nicole Wade, and Warwick Graco for providing key data and domain expertise.

References

Denny & Squire, D. M. (2005), Visualization of cluster changes by comparing Self-Organizing Maps, *in* T. B. Ho, D. Cheung & H. Liu, eds, 'PAKDD'05', Vol. 3518 of *Lecture Notes in Computer Science*, Springer, pp. 410–419.

Dittenbach, M., Merkl, D. & Rauber, A. (2000), Growing hierarchical Self-Organizing Map, *in* 'Proceedings of the International Joint Conference on Neural Networks', Vol. 6, Technische Universität Wien, IEEE, Piscataway, NJ, pp. 15–19.

Dolnicar, S. (1997), The use of neural networks in marketing: market segmentation with self organising feature maps, *in* 'Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4–6', Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, pp. 38–43.

Han, J. & Kamber, M. (2006), *Data Mining: Concepts and Techniques (second edition)*, Morgan Kaufmann, San Francisco, CA.

Himberg, J. (1998), Enhancing the SOM-based data visualization by linking different data projections, *in* 'Proceedings of 1st International Symposium Intelligent Data Engineering and Learning (IDEAL'98)—Perspectives on Financial Engineering and Data Mining', Springer, Hong Kong, pp. 427–434.

Iivarinen, J., Kohonen, T., Kangas, J. & Kaski, S. (1994), Visualizing the clusters on the Self-Organizing Map, *in* C. Carlsson, T. Järvi & T. Reponen, eds, 'Proceedings of the Conference on Artificial Intelligence Research in Finland', Vol. 12, Finnish Artificial Intelligence Society, Helsinki, Finland, pp. 122–126.

Kaski, S. & Kohonen, T. (1998), Tips for processing and color-coding of Self-Organizing Maps, *in* G. Deboeck & T. Kohonen, eds, 'Visual Explorations in Finance with Self-Organizing Maps', Springer, London, pp. 195–202.

Kohonen, T. (1982), 'Self-organized formation of topologically correct feature maps', *Biological Cybernetics* **43**, 59–69.

Kohonen, T. (2001), *Self-Organizing Maps (Third Edition)*, Vol. 30 of *Springer Series in Information Sciences*, Springer, Berlin, Heidelberg.

Koikkalainen, P. & Oja, E. (1990), Self-organizing hierarchical feature maps, *in* 'Proceedings IJCNN-90, International Joint Conference on Neural Networks, Washington, DC', Vol. 2, IEEE Service Center, Piscataway, NJ, pp. 279–285.

Markey, M. K., Lo, J. Y., Tourassi, G. D. & Floyd Jr., C. E. (2003), 'Self-organizing map for cluster analysis of a breast cancer database.', *Artificial Intelligence in Medicine* **27**(2), 113–127.

Pampalk, E., Rauber, A. & Merkl, D. (2002), Using smoothed data histograms for cluster visualization in self-organizing maps, *in* 'Artificial Neural Networks - ICANN 2002: International Conference, Madrid, Spain, August 28-30, 2002. Proceedings', Vol. 2415/2002, Springer Berlin / Heidelberg, pp. 871–876.

Pareto, V. (1972), *Manual of Political Economy*, Macmillan, London. Translated by Ann S. Schwier. Edited by Ann S.Schwier and Alfred N.Page.

Trewin, D. (2003), Socio-economic indexes for areas: Australia 2001, Technical Report 2039, Australian Bureau of Statistics.

Tryba, V., Metzen, S. & Goser, K. (1989), Designing basic integrated circuits by self-organizing feature maps, *in* 'Neuro-Nîmes '89. International Workshop on Neural Networks and their Applications', ARC; SEE, EC2, Nanterre, France, pp. 225–235.

Vesanto, J. (1999), 'SOM-based data visualization methods', *Intelligent Data Analysis* **3**(2), 111–126.

Vesanto, J. & Alhoniemi, E. (2000), 'Clustering of the Self-Organizing Map', *IEEE Transactions on Neural Networks* **11**(3), 586–600.

Vesanto, J., Himberg, J., Alhoniemi, E. & Parhankangas, J. (2000), SOM toolbox for Matlab 5, Report A57, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

Viveros, M. S., Nearhos, J. P. & Rothman, M. J. (1996), Applying data mining techniques to a health insurance information system, *in* T. M. Vijayaraman, A. P. Buchmann, C. Mohan & N. L. Sarda, eds, 'Proceedings of 22th International Conference on Very Large Data Bases (VLDB'96), September 3-6, 1996, Mumbai (Bombay), India', Morgan Kaufmann, pp. 286–294.

Wells, W. D. (1975), 'Psychographics: A critical review', *Journal of Marketing Research (JMR)* **12**(2), 196–213.

Williams, G. J. (1999), Evolutionary hot spots data mining - an architecture for exploring for interesting discoveries, *in* 'PAKDD '99: Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining', Springer-Verlag, London, UK, pp. 184–193.

World Bank (2003), *World Development Indicators 2003*, The World Bank, Washington DC.

Xiong, H., Wu, J. & Chen, J. (2006), K-means clustering versus validation measures: a data distribution perspective, *in* 'KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM Press, New York, NY, USA, pp. 779–784.