Expert Systems with Applications 38 (2011) 11176–11181

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

# A semi-supervised tool for clustering accounting databases with applications to internal controls

Argyris Argyrou *, Andriy Andreev

*HANKEN School of Economics, Arkadiankatu 22, 00101 Helsinki, Finland*

ARTICLE INFO

*Keywords:*
Self-organizing map
Clustering accounting databases
Internal controls

ABSTRACT

A considerable body of literature attests to the significance of internal controls; however, little is known on how the clustering of accounting databases can function as an internal control procedure. To explore this issue further, this paper puts forward a semi-supervised tool that is based on self-organizing map and the IASB XBRL Taxonomy. The paper validates the proposed tool via a series of experiments on an accounting database provided by a shipping company. Empirical results suggest the tool can cluster accounting databases in homogeneous and well-separated clusters that can be interpreted within an accounting context. Further investigations reveal that the tool can compress a large number of similar transactions, and also provide information comparable to that of financial statements. The findings demonstrate that the tool can be applied to verify the processing of accounting transactions as well as to assess the accuracy of financial statements, and thus supplement internal controls.



## 1. Introduction

A number of statutory and professional pronouncements require a public company's management to implement and maintain appropriate internal controls in order to ensure the integrity and reliability of accounting transactions. In particular, the Sarbanes–Oxley Act of 2002 Section 404 (US Congress, 2002) mandates a public company's CFO and CEO not only to implement and maintain internal controls over financial reporting but also to assess and certify the effectiveness of such controls on an annual basis. The Act also requires an external auditor to attest management's certification of internal controls. Further, the Statement on Auditing Standards 94 (AICPA, 2001) suggests that an auditor should not rely exclusively on substantive testing when evidence of a company's recording and processing of accounting transactions exists only in an electronic form. Instead, an auditor should assess controls over a company's information technology (i.e. I.T) environment in which the recording and processing of accounting transactions occur. These I.T controls form an integral part of a company's internal control system as they underpin the completeness, accuracy, and timeliness of a company's financial reporting (Canadian Institute of Chartered Accountants, 2004). To discharge their duties, managers and auditors can seek advice and guidance from two widely adopted frameworks, COSO (COSO, 1992) and COBIT (IT Governance Institute, 2000). However comprehensive the two frameworks are, they provide little guidance on the design and application of specific tools.

Motivated by this unexplored issue, this paper designs and validates a semi-supervised tool for clustering accounting databases in order to supplement internal control procedures. To elaborate, the proposed tool provides a holistic snapshot of an accounting database, it supports a manager in checking whether an accounting database can process the right transactions accurately, and in assessing the accuracy of financial statements. The tool and its practical applications in the domain of accounting constitute the paper's novelty and contribution.

To design the tool, the paper uses self-organizing map (i.e. SOM) (Kohonen, 1997) for its ability to cluster and visualize data as well as to map the probability density function of a multi-dimensional input space to a two-dimensional output space (i.e. SOM grid) while preserving the original topology. SOM has been applied successfully in a multitude of research domains; for example, financial benchmarking (Eklund, Back, Vanharanta, & Visa, 2003), predicting corporate bankruptcy (Serrano-Cinca, 1996; Lee, Booth, & Alam, 2005), market segmentation (Kiang, Hu, & Fisher, 2006), evaluating the creditworthiness of loan applicants (Huysmans, Baesens, Vanthienen, & van Gestel, 2006), selecting an MBA program (Kiang & Fisher, 2008), improving the accuracy of a naive Bayes classifier to classify text documents (Isa, Kallimani, & Lee, 2009), and identifying peer schools for AACSB accreditation (Kiang, Fisher, Chen, Fisher, & Chi, 2009).

In addition to SOM, the paper incorporates into the proposed tool accounting knowledge in the form of the International Accounting Standard Board (i.e. IASB) XBRL Taxonomy (IASCF,

* Corresponding author. Tel.: +358 46 6598649.
*E-mail addresses:* argyris.argyrou@hanken.fi (A. Argyrou), andriy.andreev@hanken.fi (A. Andreev).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.02.163

2009) that describes the disclosure and presentation of financial statements pursuant to the International Financial Reporting Standards (i.e. IFRSs) (IASB, 2009). Briefly, the IASB XBRL Taxonomy is a hierarchical structure describing accounting concepts (e.g. Assets, Liabilities, Equity, etc.) and the semantic relationships between them as *child* $\prec$ *parent* links. The paper opted for the IFRSs rather than some national Generally Accepted Accounting Principles (i.e. GAAPs), because IFRSs are adopted by most of the leading economies. For instance, starting in 2005, all European Union companies listed on a regulated market must adopt IFRSs for preparing their consolidated financial statements (European Parliament, 2002); US Securities and Exchange Commission (2008) proposed a roadmap that, once completed, could lead to the mandatory adoption of IFRSs by U.S. publicly listed companies for fiscal periods ending on or after the 15th of December 2014; and companies that are publicly listed in Canada must adopt IFRSs for a fiscal period ending in 2011 (AcSB, 2009).

To validate the proposed tool, the paper applies it to accounting transactions that were provided by a shipping company as a text-dump of its database covering fiscal year 2006. In the first step, the paper uses the graph-theoretical approach (Argyrou, 2009) to pre-process accounting-hierarchical data into a numerical representation for SOM-based clustering. Second, the paper uses bootstrap to select random samples with replacement; and for each bootstrapped sample, it uses SOM-Toolbox[1] for Matlab (Vesanto, Himberg, Alhoniemi, & Parhankangas, 2000) to train a SOM in batch mode with hexagonal topology and Gaussian neighborhood. Further, the paper evaluates each SOM by calculating the following three quality measures: (i) topographic error, T.E., (ii) quantization error, Q.E., and (iii) Davies-Bouldin Index, DBI. In addition, the paper uses the bootstrap bias-corrected with acceleration method to estimate the two-sided 95% confidence interval of the mean and standard deviation for the aforesaid quality measures. Finally, to ensure that the proposed tool satisfies its intended use, the paper benchmarks the tool's output against the case company's financial statements.

The rest of the paper proceeds as follows. The next section describes the research design and methodology, Section 3 presents and discusses the results, and Section 4 concludes as well as suggests future research perspectives.

## 2. Research design and methodology

### 2.1. Data description

The data were provided by a shipping company in the form of a text-dump of its accounting database. The data describe the economic activities of the case company for fiscal year 2006, and consist of 25,440 accounting transactions each of which is described by seven variables, as shown in Table 1. The accounting dataset is denoted as matrix $A = (a_{ij})_{nm}$, where $n = 25{,}440$ and $m = 7$ reflecting the number of transactions and variables, respectively. All variables but "Account Class" are specific to the case company, and hence they are unlikely to be used by any other company. In contrast, the variable "Account Class" conveys information on how to aggregate accounting transactions for a company to prepare financial statements. Because the presentation and disclosure of financial statements are dictated by IFRSs, "Account Class" is likely to be used by other companies that report under IFRSs, and thus it provides a link between an accounting database and IFRSs. On these grounds, the proposed tool employs "Account Class" as a lens through which a user can probe into an accounting database. To avoid duplication, we describe "Account Class" in the same table, Table 3, in which we describe the results from the benchmarking exercise.

**Table 1**
Description of variables.

| Name | Type |
|---|---|
| Account number | Alphanumeric |
| Account description | Text |
| Posting date | Date |
| Debit-credit indicator | Binary |
| USD amount | Numerical |
| Transaction details | Text |
| Account class | Hierarchical-categorical |

However, "Account Class" can not be processed directly by SOM, because it lacks intrinsic quantitative information for SOM to calculate the Euclidean distance. Although SOM can use non-metric similarity measures (e.g. Levenshtein distance) in lieu of the Euclidean distance (Kohonen & Somervuo, 1998, 2002); such measures can not capture the semantic relationships between hierarchical-categorical data, and hence they are not suitable for the present study. Instead, the paper observes that "Account Class" can be represented as a directed acyclic graph (i.e. DAG) that adheres to an a priori known hierarchy, the IASB XBRL Taxonomy.

### 2.2. Data pre-processing

Motivated by this observation, the paper uses the graph-theoretical approach[2] to pre-process "Account Class" into a numerical representation that takes the form of a distance matrix. In particular, the graph-theoretical approach operates in two steps. First, the paper encodes "Account Class" as a DAG, shown in Fig. 1, that in turn represents a graphical instantiation of the IASB XBRL Taxonomy. The root vertex represents the complete set of "Account Class", and all other vertices are ordered in such a way that each vertex represents a sub-set of its parent vertex. As a result, the *child* $\prec$ *parent* relationships specified in the Taxonomy are preserved. For example, "*Bank Account*" $\prec$ "*Cash in Bank*" $\prec$ "*Cash and Cash Equivalents*" $\prec$ "*Current Assets*" constitute *child* $\prec$ *parent* relationships. In the second step, the graph-theoretical approach quantifies the aforementioned relationships by using Dijkstra's algorithm (Dijkstra, 1959) to calculate the all-pairs distances between vertices. This calculation yields a symmetric distance matrix, $D = (d_{ij})_{NN}$, where $N = 29$ denoting the number of account classes, and $d_{ij}$ the distance between a pair of account classes. The distance is the sum of the weighed-edges that exist between the path of two account classes; for the experiments, the weight for each edge is set to 1. For example, the distances between "Accounts Payable" and: {itself, "Office Expenses", "Trade Creditors"} are 0, 8, and 2, respectively. The distance metric, $d_{ij}$, satisfies the conditions of a metric space, as follows (Jungnickel, 2002, p. 65): (i) $d_{ij} > 0$ for all $i \neq j$, (ii) $d_{ij} = 0$ if and only if $i = j$, (iii) $d_{ij} = d_{ji}$ for all $i$ and $j$, and (iv) $d_{iz} \leqslant d_{ij} + d_{jz}$ for all $i$, $j$, and $z$.

The distance matrix, $D$, thus derived forms the numerical representation of "Account Class". In essence, the distance matrix defines the semantic relationships between "Account Class", and by extension, between accounting transactions; and in doing so, it represents the accounting knowledge the paper incorporates into the tool.

### 2.3. Self-organizing map

To set up the input dataset for SOM, the paper concatenates accounting dataset, $A$, with distance matrix, $D$, by using "Account

[1] SOM-Toolbox for Matlab and its documentation are available in the public domain at http://www.cis.hut.fi/somtoolbox/.

[2] We implement the graph-theoretical approach by writing the required code in Mathematica (Wolfram Research Inc., 2007).

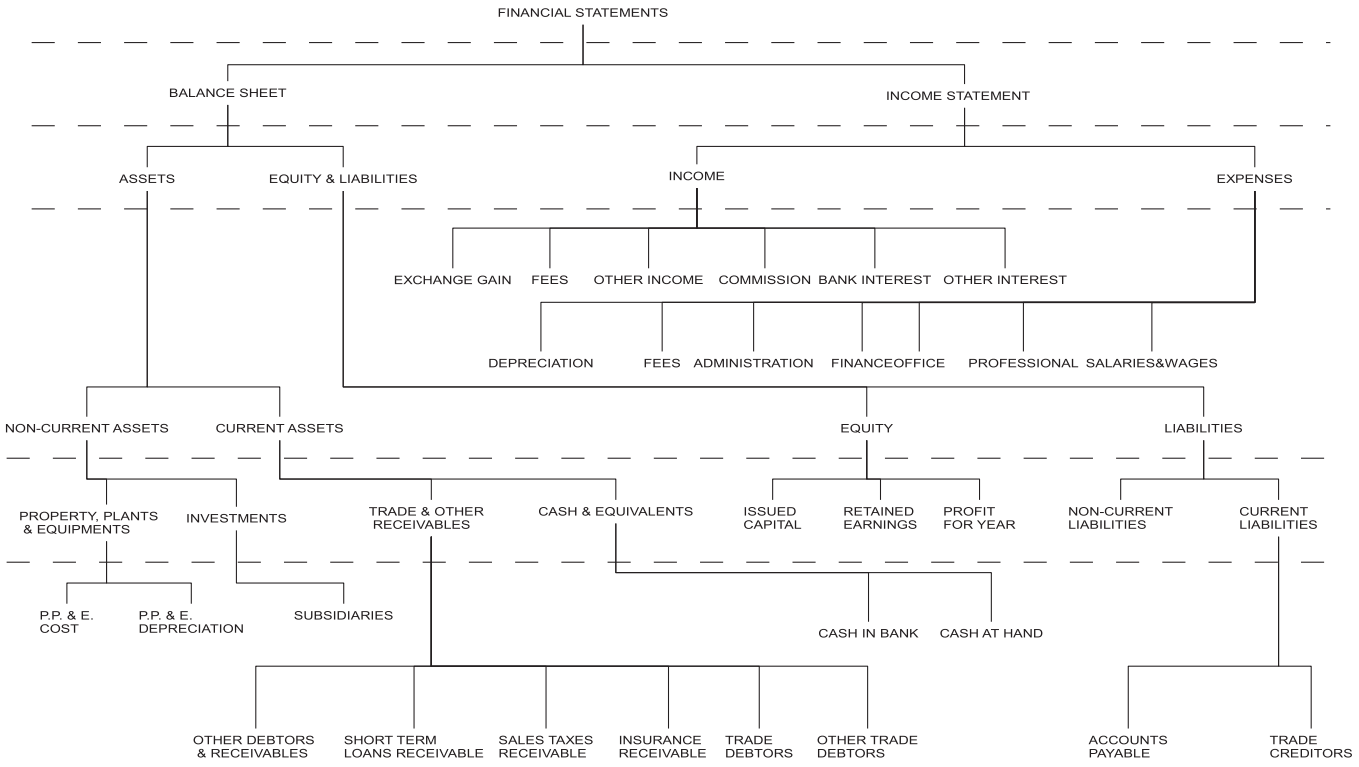


**Fig. 1.** The Accounting Tree, a graphical instantiation of the IASB XBRL Taxonomy.

Class" as the key variable. This operation produces matrix $X = (x_{ik})_{n(N+m)}$ that makes up the input dataset to SOM, where $n = 25{,}440$, $N = 29$, and $m = 7$. In the context of this study, SOM maps the probability density function of the input dataset to a two-dimensional hexagonal grid of neurons, as shown in Fig. 2. The number of neurons is set to 820 corresponding approximately

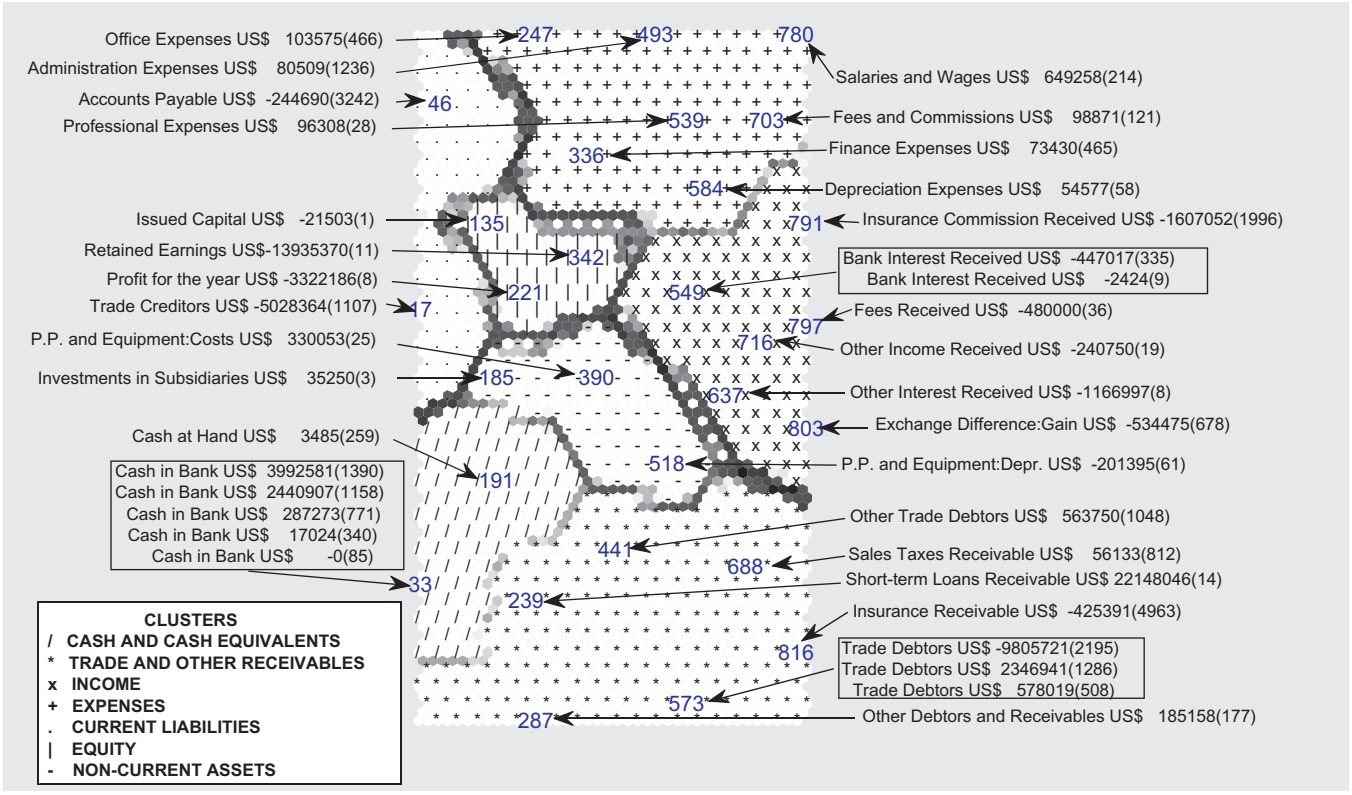


**Fig. 2.** *The tool's output, SOM grid.* The 29 account classes are represented on the SOM grid by as many neurons identified by their corresponding numbers. For each account class, the total number of transactions is shown in brackets; positive and negative US$ amounts denote debit and credit balances respectively. The cluster boundaries are defined by the U-matrix.

to $5 \times \sqrt{n}$, where $n = 25{,}440$ reflecting the number of accounting transactions. The size of the SOM grid is set to 41 rows and 20 columns so that the ratio of the grid's sides (i.e. 2 : 1) matches the ratio of the two biggest eigenvalues of the covariance matrix of the input data (Vesanto et al., 2000, p. 30). Each neuron has two representations, as follows: (i) a coordinate on the SOM grid, and (ii) a codevector, $\vec{m}_j \in \mathbb{R}^{36}$, in the input space, where $j = 1, 2, \ldots, 820$.

The formation of SOM involves three iterative processes (Haykin, 1999, p. 447). First, in the competition process, each input vector, $\vec{x}_i \in \mathbb{R}^{36}$, is compared with all codevectors, $\vec{m}_j \in \mathbb{R}^{36}$, and the best match in terms of the smallest Euclidean distance, $\|\vec{x}_i - \vec{m}_j\|$, is mapped onto neuron $j$ termed the best-matching unit (i.e. BMU) and denoted by the subscript $c : \|\vec{x}_i - \vec{m}_c\| = \min_j\{\|\vec{x}_i - \vec{m}_j\|\}$ (Kohonen, 1997, p. 86). Second, in the co-operation process, the BMU locates the center of the Gaussian neighborhood: $h_{cj} = exp\left[-\frac{\|r_c - r_j\|^2}{2\sigma^2(t)}\right]$, where $r_c, r_j \in \mathbb{R}^2$ are the radius of BMU and neuron $j$, respectively, $t$ denotes discrete time, and $\sigma(t)$ defines the width of the kernel (Kohonen, 1997, p.87). Third, in the adaptive process, the batch-training SOM updates the BMU codevector as follows (Vesanto et al., 2000, p. 9): $\vec{m}_j(t+1) = \frac{\sum_{i=1}^{n} h_{cj}(t)\vec{x}_i}{\sum_{i=1}^{n} h_{cj}(t)}$.

### 2.4. Quality measures

The quality of the SOM grid, Fig. 2, can be evaluated with respect to its topology preservation, and resolution. The SOM grid preserves faithfully the original topology if vectors that are close in the input space are mapped onto adjacent neurons. Topographic error, T.E., quantifies topology preservation as the proportion of all input vectors whose first and second BMUs are not mapped onto adjacent neurons (Kiviluoto, 1996). Analytically, $T.E. = \frac{1}{n}\sum_{i=1}^{n} \varepsilon(\vec{x}_i)$; if the first and second BMUs of $\vec{x}_i$ are adjacent, then $\varepsilon(\vec{x}_i) = 0$, otherwise $\varepsilon(\vec{x}_i) = 1$. Further, the SOM grid exhibits high resolution when vectors that are distant in the input space are not mapped onto neighboring neurons. Quantization error, Q.E., measures resolution as the average distance between each input vector and its BMU: $Q.E. = \frac{1}{n}\sum_{i=1}^{n} \|\vec{x}_i - \vec{m}_c\|$.

The internal validity of clustering can be assessed via the Davies–Bouldin Index (Davies & Bouldin, 1979), defined as: $DBI = \frac{1}{C}\sum_{i=1}^{C} \max_{i \neq j}\left\{\frac{\Delta(C_i)+\Delta(C_j)}{\delta(C_i,C_j)}\right\}$; where $C = 7$, the number of clusters identified by SOM, $\delta(C_i, C_j)$, and $\{\Delta(C_i), \Delta(C_j)\}$ the intercluster and intracluster distances, respectively. A small value indicates highly-compact clusters whose centroids are well-separated.

### 2.5. Experiments

The paper conducts the experiments in six steps. First, the paper uses bootstrap to draw one hundred random samples with replacement from the empirical distribution of the input dataset, $X = (x_{ik})_{n(N+m)}$. Second, for each bootstrapped sample, the paper uses SOM-Toolbox for Matlab (Vesanto et al., 2000) to train a SOM in batch mode with hexagonal topology and Gaussian neighborhood. Third, to cluster the neurons and their associated codevectors, the paper uses the Unified-distance matrix (i.e. U-matrix) (Ultsch & Siemon, 1990). The U-matrix calculates the average distance between a neuron's codevector and that of its neighboring neurons and then superimposes this distance as a height, Z coordinate, on the SOM grid. Because Unified-distance matrix superimposes the height between rather than at neighboring neurons, there are more than 820 neurons on the grid. The dark and light areas on the SOM grid indicate long and short distances between neurons respectively; dark areas denote cluster boundaries whereas light areas denote clusters, as shown in Fig. 2. Fourth, for each SOM, the paper calculates the following three quality measures: (i) topographic error, T.E., (ii) quantization error, Q.E., and (iii) Davies–Bouldin Index, DBI.

Fifth, the paper uses the bootstrap bias-corrected with acceleration method, B = 1,000, (DiCiccio & Efron, 1996) in order to estimate the two-sided 95% confidence interval of the mean and standard deviation for the foregoing quality measures. The rationale behind this statistical analysis is threefold. First, the paper must ensure that SOM converges to a stable and adequate state rather than to a local minimum; second, the SOM grid, shown in Fig. 2, preserves faithfully the topology of the accounting dataset; and third, the clusters identified on the SOM grid possess good statistical properties in terms of compactness and separation, and are also relevant and meaningful within an accounting context. This analysis could provide evidence on the internal consistency of the tool, and also on the ability of the results to be generalized beyond the particular dataset. Finally, for the proposed tool to be able to benefit its intended users, it must correspond accurately with the economic reality of the case company. To this end, the paper benchmarks the tool's output against the financial statements prepared by the case company.

## 3. Results presentation and discussion

The proposed tool generates a two-dimensional SOM grid as depicted in Fig. 2, presenting the non-linear projection of the 25,440 accounting transactions as well as identifying seven clusters. As shown in Fig. 2, each of the 29 account classes conveys two pieces of information, as follows: (i) total number of its respective accounting transactions, shown in brackets, and (ii) total USD amount of these transactions. Positive and negative amounts denote debit and credit balances respectively. The 29 account classes are represented on the SOM grid by as many neurons identified by their corresponding numbers. For example, "Administration Expenses" consists of 1,236 transactions having a total debit balance of USD 80,509 and being represented by neuron 493.

The quality of SOM grid is evaluated by the topographic and quantization errors as described in Table 2. Both quality measures exhibit small values and narrow 95% confidence intervals for their respective means and standard deviations. These results provide evidence that the grid can map the input data accurately as well as preserve faithfully the original topology; they also suggest that the tool can be generalized to cases beyond the particular accounting database.

The entry for quantization error, Q.E., in Table 2 merits additional discussion, because it points towards the convergence of SOM to a stable state. The values reported for Q.E. are truncated to the third decimal place, because the corresponding original values are in the order of $10^{-5}$. A mathematical analysis of SOM convergence lies beyond the scope of this paper. Suffice it to say that in a stable state we expect $E\left\{h_{cj}\left(\vec{x}_i - \vec{m}_j^*\right)\right\} = 0$, where $E\{.\}$ denotes the expectation function, and $\vec{m}_j^* = \lim_{t\to\infty}\vec{m}_j(t)$ (Kohonen, 1997, p. 113); the rest of the notations are defined in Section 2.3. The minuscule Q.E. values confirm that SOM converges to a stable state producing an optimal grid, Fig. 2, of the input dataset.

Furthermore, a visual inspection of the SOM grid, Fig. 2, indicates that the proposed tool can cluster the accounting transactions in

**Table 2**
Two-sided 95% confidence intervals using the bootstrap bias-corrected with acceleration method, $B = 1000$.

| Quality measures | Mean | | | Std. deviation | | |
|---|---|---|---|---|---|---|
| | Lower | Upper | Std. err. | Lower | Upper | Std. err. |
| T.E. | 0.269 | 0.336 | 0.049 | 0.150 | 0.192 | 0.029 |
| Q.E. | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.001 |
| DBI | 1.037 | 1.107 | 0.047 | 0.153 | 0.207 | 0.036 |

seven homogeneous and well-separated clusters. This result is corroborated by statistical analysis, Table 2, that shows both the mean and standard deviation of Davies–Bouldin Index to have small values and narrow 95% confidence intervals. More importantly, as we demonstrate in Table 3, the seven clusters are meaningful and interpretable from an accounting perspective. In particular, clusters 3 and 4 represent income and expense items respectively, and their combination makes up the Income Statement. Clusters 7, 2, and 1 correspond to "Non-Current Assets", "Trade and Other Receivables", and "Cash and Cash Equivalents" respectively. These three clusters compose the "Assets" side of the Balance Sheet Statement. Clusters 6 and 5 comprise "Equity" and "Current Liabilities" respectively, and collectively form the "Equity and Liabilities" side of the Balance Sheet Statement.

A closer investigation reveals that the proposed tool enjoys two properties that can enhance its potential applications. First, it can preserve the semantic relationship between one account class and another; this relationship can be quantified in terms of the Euclidean distance between neurons in Fig. 2. For example, "Professional Expenses" is closer to "Finance Expenses" than either is to "Sales Taxes Receivable"; a proximity that is proportional to their relationship according to accounting theory: the first two are part of "Expenses", whereas the third is part of "Trade and Other Receivables". Second, it can compress a large number of similar transactions onto a single neuron. For example, neuron 46 represents all the 3,242 transactions concerning "Accounts Payable".

**Table 3**
Results from benchmarking the tool's output against company's financial statements.

| Company's financial statements | | SOM grid (Fig. 2) | |
|---|---|---|---|
| Account class | USD | Neuron | Cluster |
| *Income statement for the year ending 31 December 2006* | | | |
| *Income* | | | |
| Exchange gain | 534,475 | 803 | 3 |
| Fees received | 480,000 | 797 | 3 |
| Other income received | 240,750 | 716 | 3 |
| Insurance commission received | 1,607,052 | 791 | 3 |
| Bank interest received | 449,440 | 549 | 3 |
| Other interest received | 1,166,997 | 637 | 3 |
| Total income | 4,478,714 | | |
| *Expenses* | | | |
| Depreciation expenses | 54,577 | 584 | 4 |
| Fees and commissions | 98,871 | 703 | 4 |
| Administration expenses | 80,509 | 493 | 4 |
| Finance expenses | 73,430 | 336 | 4 |
| Office expenses | 103,575 | 247 | 4 |
| Professional expenses | 96,308 | 539 | 4 |
| Salaries and wages | 649,258 | 780 | 4 |
| Total expenses | 1,156,528 | | |
| Profit for the year | 3,322,186 | | |
| *Balance sheet as at 31 December 2006* | | | |
| *Assets* | | | |
| *Non-current assets* | | | |
| Property, plants and equipment_costs | 330,053 | 390 | 7 |
| Property, plants and equipment_depr. | −201,395 | 518 | 7 |
| Investment in subsidiaries | 35,250 | 185 | 7 |
| *Current assets* | | | |
| *Trade and other receivables* | | | |
| Short-term loans receivable | 22,148,046 | 239 | 2 |
| Other debtors and receivables | 185,158 | 287 | 2 |
| Sales taxes receivable | 56,133 | 688 | 2 |
| Other trade debtors | 563,750 | 441 | 2 |
| Insurance receivable | −425,391 | 816 | 2 |
| Trade debtors | −6,880,761 | 573 | 2 |
| Cash and cash equivalents | | | |
| Cash in bank | 6,737,785 | 33 | 1 |
| Cash at hand | 3,485 | 191 | 1 |
| Total assets | 22,552,113 | | |
| *Equity and liabilities* | | | |
| *Equity* | | | |
| Issued capital | 21,503 | 135 | 6 |
| Retained earnings | 13,935,370 | 342 | 6 |
| Profit for the year | 3,322,186 | 221 | 6 |
| Total equity | 17,279,059 | | |
| *Liabilities* | | | |
| *Current liabilities* | | | |
| Accounts payable | 244,690 | 46 | 5 |
| Trade creditors | 5,028,364 | 17 | 5 |
| Total equity and liabilities | 22,552,113 | | |

Given these properties, a manager can apply the proposed tool to check whether a database can process the right transactions accurately. For example, Fig. 2 indicates that although an entry in "Cash in Bank" consists of 85 transactions, it has a nil balance, thereby signalling the occurrence of suspicious transactions that need to be followed up.

Shown in Table 3, the results from the benchmarking exercise demonstrate that the proposed tool corresponds accurately with the financial statements prepared by the case company; a correspondence that lends much credence to the tool. Indeed, the tool's output, shown in Fig. 2, conveys all the necessary and relevant information to function as financial statements. Further, during the benchmarking exercise two unexpected issues emerged. First, the tool may reduce the time and effort involved in preparing financial statements; an issue that could be the focus of future research. Second, to prepare financial statements, the case company relies on SQL queries and spreadsheets. As both procedures are prone to human error, the tool provided a much-needed method for the case company to assess the accuracy of their reporting function. Specifically, the tool clusters correctly "Insurance Receivable" and "Trade Debtors" in cluster 2, "Trade and Other Receivables", as shown in Fig. 2 and Table 3. However, both items have credit balances, whereas their normal balances are debit; a discrepancy that points towards erroneous transactions warranting further investigation.

Although coding is still in progress, the paper deems it necessary to estimate the algorithmic complexity of the tool in order to provide a machine-independent measure for evaluating the tool. As discussed in Sections 2.2 and 2.3, the tool performs two main operations, as follows: (i) it uses Dijkstra's algorithm to derive the distance matrix, and (ii) it uses SOM to process the input dataset. Dijkstra's algorithm has a $O(N^3)$ complexity, and $O(N^2)$ memory consumption for storing the distance matrix, $D$, where $N$ denotes the number of account classes. SOM has $O(nld)$ complexity for searching BMUs and updating codevectors, where $n$ and $l$ denote the number of input vectors and neurons respectively, and $d$ is the dimensionality of input space. If the number of neurons is set to be proportional to $\sqrt{n}$, then the complexity becomes $O(n^{1.5}d)$. The memory consumption of SOM is $O((n+l)d)$ for storing the input and codevector matrices, and $O(l^2)$ for storing the inter-neuron distance matrix.

Nonetheless, the proposed tool suffers from certain limitations that may restrict its potential applications. First, it is deterministic, and as such it does not address uncertainty; a limitation that prevents a user from applying it to budgeting and forecasting. Second, once coding has been completed, applying the tool ought to be straightforward; interpreting its output, however, requires a user to have some familiarity with SOM. Finally, the quality measures, Table 2, should be interpreted with caution, because topographic and quantization errors are data-dependent, and small values may be the result of SOM overfitting the input data rather than projecting them accurately.

## 4. Conclusions and future research

Based on the IASB XBRL Taxonomy and SOM, this paper presents a semi-supervised tool for clustering accounting databases in order to supplement internal control procedures. While a wealth

of pronouncements require a company's management to implement and maintain adequate internal controls, these pronouncements provide little guidance on the design and application of specific tools. Further, in contrast to published financial statements, existing literature has paid insufficient attention to accounting databases. These issues suggest that the clustering of accounting databases as an internal control procedure has not been fully explored.

Empirical analyses revealed that the proposed tool can cluster accounting databases in homogeneous and well-separated clusters that can be interpreted from an accounting perspective. Additional investigations indicated that the tool can preserve the semantic relationships between account classes, and also compress a large number of similar accounting transactions. Further, benchmarking results suggested that the tool's output corresponds accurately with the financial statements prepared by the case company. The paper's findings demonstrate that the tool can have practical applications in the domain of accounting, such as: (i) providing a holistic picture of an accounting database at a given point in time, (ii) assisting a manager to supervise the processing of accounting transactions, and (iii) enabling a manager to assess the accuracy of financial statements.

Ongoing research is aiming at coding the proposed tool as a toolbox for Matlab, and subsequently releasing it in the public domain. Additional experiments could investigate the tool's potential to produce financial statements. A further development could be to apply the proposed tool at consecutive time-intervals (e.g. monthly, quarterly) and use the results thus obtained for financial benchmarking or for moving-window comparisons. A promising line of future research could be the validation of the paper's results against independent domain-experts as well as the evaluation of the proposed tool within the end-user computing satisfaction framework (i.e. EUCS) (Doll, Doll, & Torkzadeh, 1988).

## Acknowledgments

We are especially indebted to the company's Finance Director, who wishes to remain anonymous, for providing us with data without which we could not have pursued this paper. We are grateful to the HANKEN Foundation for their financial support, and we also thank Mr. Tom Linström and Prof. Anders Tallberg for their insightful suggestions.

## References

AcSB, March 2009. Exposure Draft: Adopting IFRSs in Canada, II. Canadian Accounting Standard Board (AcSB), Toronto, Canada.

AICPA. (2001). Statement On Auditing Standards 94 (SAS 94): Effect of information technology on the auditor's consideration of internal control in a financial statement audit. American Institute of Certified Public Accountant.

Argyrou, A. (2009). Clustering hierarchical data using Self-Organizing map: A Graph-Theoretical approach. In *WSOM 2009: proceedings of the 7th International workshop on advances in self-organizing maps. LNCS* (Vol. 5629, pp. 19–27). Springer-Verlag.

Canadian Institute of Chartered Accountants. (2004). IT Control Assessments in the context of CEO/CFO Certification. Toronto, Canada.

COSO. (1992). Internal Control – An Integrated Framework. The Committee of Sponsoring Organizations of the Treadway Commission (COSO). AICPA, New Jersey, USA.

Davies, D., & Bouldin, D. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 1*(2), 224–227.

DiCiccio, T. J., & Efron, B. (1996). Bootstrap confidence intervals. *Statistical Science, 11*(3), 189–228.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik, 1*, 269–271.

Doll, W. J., Doll, W. J., & Torkzadeh, G. (1988). The measurement of End-User computing satisfaction. *MIS Quarterly, 12*(2), 259–274.

Eklund, T., Back, B., Vanharanta, H., & Visa, A. (2003). Financial benchmarking using self-organizing maps studying the international pulp and paper industry. In *Data mining: opportunities and challenges* (pp. 323–349). IGI Publishing.

European Parliament. (2002). Regulation (EC) No. 1606/2002 of the European Parliament and of the Council of 19 July 2002 on the application of international accounting standards. Official Journal L 243/1.

Haykin, S. (1999). *Neural Networks* (2nd ed.). *A comprehensive foundation*. Upper Saddle River, New Jersey, USA: Prentice Hall International.

Huysmans, J., Baesens, B., Vanthienen, J., & van Gestel, T. (2006). Failure prediction with self organizing maps. *Expert Systems with Applications, 30*(3), 479–487.

IASB. (2009). International Financial Reporting Standards (IFRS) 2009. International Accounting Standards Committee Foundation (IASCF), London, United Kingdom.

IASCF. (2009). IFRS Taxonomy Guide 2009 (XBRL). International Accounting Standards Committee Foundation (IASCF), London, United Kingdom.

Isa, D., Kallimani, V., & Lee, L. H. (2009). Using the self organizing map for clustering of text documents. *Expert Systems with Applications, 36*(5), 9584–9591.

IT Governance Institute. (2000). Governance, Control and audit for information technology, COBIT 3rd Edition. IT Governance Institute, Rolling Meadows, IL, USA.

Jungnickel, D. (2002). Graphs. *Networks and algorithms* (English ed.). *Algorithms and computation in mathematics* (Vol. 5). Berlin, Germany: Springer.

Kiang, M. Y., & Fisher, D. M. (2008). Selecting the right MBA schools – an application of self-organizing map networks. *Expert Systems with Applications, 35*(3), 946–955.

Kiang, M. Y., Fisher, D. M., Chen, J. V., Fisher, S. A., & Chi, R. T. (2009). The application of SOM as a decision support tool to identify AACSB peer schools. *Decision Support Systems, 47*(1), 51–59.

Kiang, M. Y., Hu, M. Y., & Fisher, D. M. (2006). An extended self-organizing map network for market segmentation–a telecommunication example. *Decision Support Systems, 42*(1), 36–47.

Kiviluoto, K. (1996). Topology preservation in Self-Organizing maps. In *Proceeding of international conference on neural networks (ICNN'96)* (pp. 294–299).

Kohonen, T. (1997). *Self-organizing maps* (2nd ed.). *Springer series in information sciences* (Vol. 30). Heidelberg, Germany: Springer-Verlag.

Kohonen, T., & Somervuo, P. (1998). Self-organizing maps of symbol strings. *Neurocomputing, 21*(1–3), 19–30.

Kohonen, T., & Somervuo, P. (2002). How to make large self-organizing maps for nonvectorial data. *Neural Networks, 15*(8–9), 945–952.

Lee, K., Booth, D., & Alam, P. (2005). A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms. *Expert Systems with Applications, 29*(1), 1–16.

Serrano-Cinca, C. (1996). Self organizing neural networks for financial diagnosis. *Decision Support Systems, 17*(3), 227–238.

Ultsch, A., & Siemon, H. (1990). Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings international neural network conference* (pp. 305–308). Dordrecht, Netherlands: Kluwer Academic Press.

US Congress. (2002). Sarbanes-Oxley Act of 2002, H.R.3763.

US Securities and Exchange Commission. (2008). Roadmap for the Potential Use of Financial Statements Prepared in Accordance With International Financial Reporting Standards by US Issuers. Federal Register 73:226 (November 21, 2008) (pp. 70816–70856).

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (2000). SOM Toolbox for Matlab 5. Tech. Rep. A57, SOM Toolbox Team, Helsinki University of Technology, Espoo, Finland.

Wolfram Research Inc. (2007). Mathematica, Version 6.0. Wolfram Research Inc., Champaign, IL, USA.