An Investigation into the Effect of Unlabeled Neurons on Self-Organizing Maps

Willem S. van Heerden
Department of Computer Science
University of Pretoria
Pretoria, Gauteng, South Africa
Email: wvheerden@cs.up.ac.za

Andries P. Engelbrecht
Department of Computer Science
University of Pretoria
Pretoria, Gauteng, South Africa
Email: engel@cs.up.ac.za

***Abstract*—Self-Organizing Maps (SOMs) are unsupervised neural networks that build data models. Neuron labeling attaches descriptive textual labels to the neurons making up a SOM, and is an important component of SOM-based exploratory data analysis (EDA) and data mining (DM). Several neuron labeling approaches tend to leave some neurons unlabeled. The interaction between unlabeled neurons and SOM model accuracy affect the choice of labeling algorithm for SOM-based EDA and DM, but has not been previously investigated. This paper applies the widely used example-centric neuron labeling algorithm to several classification problems, and empirically investigates the relationship between the percentage of neurons left unlabeled and classification accuracy. Practical recommendations are also presented, which address the treatment of unlabeled neurons and the selection of an appropriate neuron labeling algorithm.**

I. Introduction

The self-organizing map (SOM) is an unsupervised learning neural network [1] upon which a large body of research work has been conducted [2]–[4]. In particular, SOMs have been widely used for data analysis in domains as varied as the financial [5], industrial [6], and medical [7] sectors.

In the view of this research, data analysis of any kind falls within one of the following two broad categories [8]:

- *Exploratory data analysis (EDA)*, which entails the computational assistance of a human data analyst, and often relies on information visualization techniques [9]. Many SOM visualization methods exist [10], resulting in SOMs being predominantly used for EDA tasks.
- *Data mining (DM)*, which automatically extracts knowledge from a data set, usually in the form of a rule set. Very few SOM-oriented DM methods exist, including only the SIG* algorithm [11], a boundary-based rule extractor [12], and the HybridSOM framework [13].

Neuron labeling attaches characterizations, which are usually textual and describe the map structure, to the neurons making up a SOM. Labels are an integral part of all SOM-based DM, and many SOM-based EDA applications. The selection of an optimal neuron labeling method is thus important.

Several neuron labeling methods potentially leave neurons uncharacterized. However, to the authors' knowledge, the impact of such neurons has not been investigated. This paper empirically examines the relationship between unlabeled neurons and the accuracy of a SOM-based classifier. Practical insight is also offered on the handling of unlabeled neurons and the selection of appropriate labeling algorithms.

The remainder of this paper is organized as follows: Section II briefly discusses the SOM architecture and algorithm, while Section III outlines the various approaches to SOM neuron labeling. Section IV discusses the experimental work conducted during this investigation, and presents results. Section V provides practical recommendations for the use of labeling algorithms with SOMs, which follow from the results described in the previous section. Finally, Section VI presents conclusions and potential avenues for future research.

II. Self-Organizing Maps

The SOM was proposed in 1982 by Teuvo Kohonen [14]. The self-organizing behavior of associative memory and human cerebral cortices served as inspiration for the method. The algorithm is unsupervised, meaning that SOMs can be trained on unclassified data. Such data is very common in practice, and any possible class-based bias is also eliminated.

Fig. 1 (a) shows the SOM architecture. A training set of data examples, $\mathcal{D}_T = \{\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_{P_T}\}$, is required. A training data example is an I-dimensional vector, $\vec{z}_s = (z_{s1}, z_{s2}, \ldots, z_{sI})$, where each $z_{sv} \in \mathbb{R}$ is an attribute value. A map structure contains neurons, and is usually a $Y \times X$ grid. The number of grid rows and columns are Y and X, respectively. A neuron, n_{yx}, is at grid row x and column y, and has an I-dimensional weight vector, $\vec{w}_{yx} = (w_{yx1}, w_{yx2}, \ldots, w_{yxI})$. Each weight, $w_{yxv} \in \mathbb{R}$, corresponds to z_{sv} over the training set.

The training operation of a SOM models the I-dimensional data space using the weight vectors of a map space with fewer dimensions. The mapping has two important properties:

- The mapping models the probability density function of the data space. This means that each neuron represents a group of similar data examples, and that neuron weights drift towards dense areas in the data space.
- The mapping preserves the local topological structure of the data space. This means that examples that are close to one another in the data space, will also be represented by neurons that are close to one another in the map space.

Fig. 1 (b) illustrates the effect of SOM training on the weight vectors using a hypothetical map structure in which $I = 2$. Gray circles denote the original positions of neuron weight

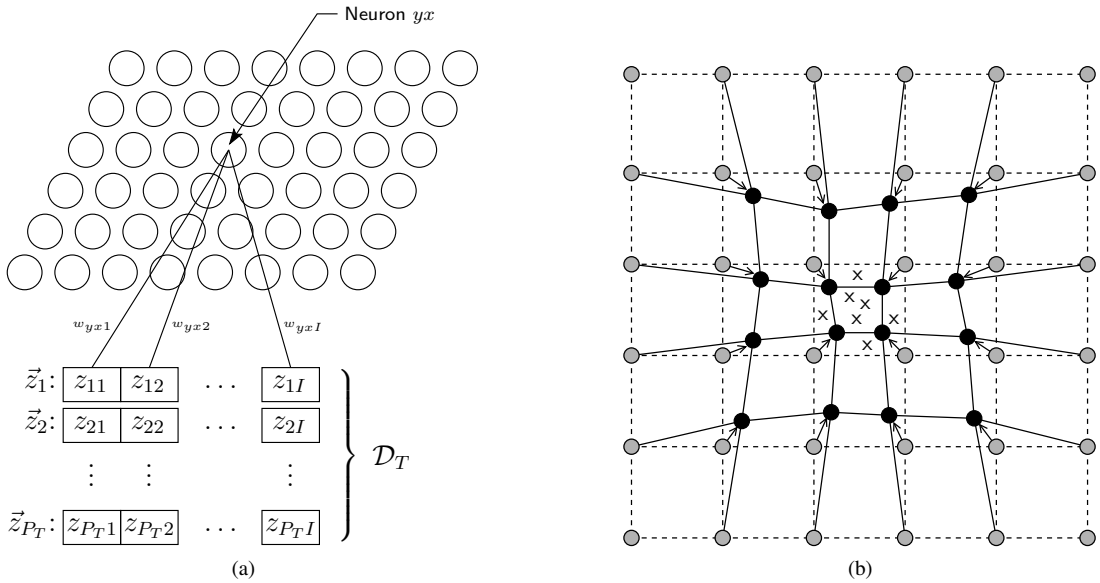


Fig. 1. The basic structure and operation of a SOM: (a) shows the architecture of a SOM, (b) shows the local effect of training in a two-dimensional case.

vectors in a grid, with dashed lines linking the weight vectors of adjacent neurons. Crosses represent the positions of training data examples. Black circles indicate the positions of the weight vectors after training, with solid lines linking the weight vectors of neighboring neurons. The first property of the mapping is illustrated by the post-training weight vectors' locations near high concentrations of training data examples. The fact that all the training examples are close to adjacent weight vectors highlights the second mapping property.

Several SOM training algorithms exist, but the experimental work reported here used the original and popular stochastic training approach [14]. This technique iteratively selects training examples, and then updates all weight vectors across the map in response to the chosen training example.

III. Neuron Labeling

Neuron labeling techniques apply textual descriptions to a subset of map neurons. When using a SOM, neuron labeling is often an important part of EDA, and is essential for DM. Neuron labeling requires a labeling data set, which is either the training set or separate data reserved for labeling.

Two broad categories of neuron labeling techniques exist [15], namely supervised neuron labeling and unsupervised neuron labeling. These categories differ according to the use of classifications associated with data examples in the labeling set. Each category is elaborated upon separately below, where Sec. III-A focuses on supervised neuron labeling, and Sec. III-B elaborates on the unsupervised methods.

A. Supervised Neuron Labeling

Supervised neuron labeling builds labels for neurons by relying upon a set of labeling data examples, where each labeling example has an associated class. Real-world data often lacks such classes, which limits the applicability of the supervised labeling methods. Supervised labeling is, however, popular because the labels are easy to analyze empirically.

The authors have previously identified three SOM-based supervised neuron labeling algorithms within the literature [8]:

- *Example-centric neuron labeling* [16] maps labeling examples to neurons. Each neuron is labeled with the most common class amongst its mapped data examples.
- *Example-centric cluster labeling* [17] uses a clustering method [18] on the weight vectors, and maps labeling examples to neurons. The most common class in a cluster's mapped examples labels all neurons in the cluster.
- *Weight-centric neuron labeling* [16] labels each neuron with the class of the labeling set data example that is closest to the weight vector of the neuron.

Pseudocode algorithms for example-centric neuron labeling, example-centric cluster labeling, and weight-centric neuron labeling are respectively illustrated in Figs. 2, 3, and 4.

The chief disadvantage of example-centric neuron labeling is that neurons remain unlabeled if no labeling examples map to the weight vector of the neuron. Example-centric cluster labeling tends to produce fewer unlabeled neurons, because the aggregating effect of the weight vector clusters ensures that larger groups of neurons are labeled. Weight-centric neuron labeling ensures that every neuron receives a label, but the label accuracy is questionable if the closest labeling example is too dissimilar to the weight vector of the neuron.

B. Unsupervised Neuron Labeling

Unsupervised neuron labeling techniques [19] assign neuron labels using labeling data sets that need not be classified. Because unclassified data is common in practical data analysis, unsupervised labeling is more widely applicable. Approaches in this category include exploratory labeling [20], unique cluster labeling [21], a method proposed by Serrano-

Create and train a SOM, map, with $Y \times X$ neurons
forall *neurons* n_{yx} *in* map **do**
  Define an empty mapped example set, M_{yx}
  Associate M_{yx} with n_{yx}
end
forall *labeling example vectors* $\vec{z}_s$ **do**
  Find neuron, n_{yx}, which has $\vec{w}_{yx}$ closest to $\vec{z}_s$
  Add labeling example $\vec{z}_s$ to M_{yx}
end
forall *neurons* n_{yx} *in* map **do**
  Find the most common class label, $\mathcal{A}_{cls}$, in M_{yx}
  Label n_{yx} with $\mathcal{A}_{cls}$
end

Fig. 2. Pseudocode for the example-centric neuron labeling algorithm.

Create and train a SOM, map, with $Y \times X$ neurons
Find clusters, $\mathcal{L} = \{S_1, S_2, \ldots, S_k\}$, of all $\vec{w}_{yx}$ in map
forall *clusters* $S_i \in \mathcal{L}$ **do**
  Define an empty mapped example set, N_i
  Associate N_i with S_i
end
forall *labeling example vectors* $\vec{z}_s$ **do**
  Find neuron, n_{yx}, which has $\vec{w}_{yx}$ closest to $\vec{z}_s$
  Find cluster S_i, such that n_{yx} is in S_i
  Add labeling example $\vec{z}_s$ to N_i
end
forall *clusters* $S_i \in \mathcal{L}$ **do**
  Find the most common class label, $\mathcal{A}_{cls}$, in N_i
  Label all $n_{yx} \in S_i$ with $\mathcal{A}_{cls}$
end

Fig. 3. Pseudocode for the example-centric cluster labeling algorithm.

Create and train a SOM, map, with $Y \times X$ neurons
forall *neurons* n_{yx} *in* map **do**
  Find labeling example, $\vec{z}_e$, which is closest to $\vec{w}_{yx}$
  Find the class label $\mathcal{A}_{cls}$ associated with $\vec{z}_e$
  Label n_{yx} with $\mathcal{A}_{cls}$
end

Fig. 4. Pseudocode for the weight-centric neuron labeling algorithm.

Cinca [22], unsupervised weight-based cluster labeling [19], LabelSOM [23], and a labeling approach developed by Azcarraga *et al* [15]. Unsupervised labels are subjective and difficult to verify empirically, and are thus not focused on in this paper. It is worth noting, however, that LabelSOM is also susceptible to unlabeled neurons, which warrants future investigation.

IV. Experimental Work

This section describes the experimental work, which investigated the effect of unlabeled neurons. Sec. IV-A outlines the experimental procedure, while Sec. IV-B describes the measures that were used to assess algorithm performance, and Sec. IV-C presents and analyzes the results of the experiments.

A. Experimental Procedure

To assess the effect of unlabeled neurons on the accuracy of the SOM model, a simple classification problem was constructed. To perform a classification, a SOM was trained and labeled using the example-centric neuron labeling approach. This labeling algorithm was focused on due to the high chance the algorithm has to leave neurons unlabeled, in contrast to example-centric cluster labeling. A data example was classified by first matching it to the nearest weight vector, which was in turn associated with the neuron that best represented the data example. The data example then received the classification denoted by the label of the matched neuron.

Table I summarizes the SOM algorithm parameters, where the decay constants are based on the work of Samarasinghe [17], and ensure algorithm convergence. One parameter setting is likely to produce a roughly equivalent percentage of unlabeled neurons on every run. Consequently, a population of different parameter setting combinations were used to produce varying levels of label coverage on the maps.

A set of candidate parameter settings were chosen in such a way as to adequately sample the ranges of valid parameter values shown in Table I. The sampling method was inspired by the work of Franken [24], and relied on Sobol' sequences [25]. Sobol' sequences consist of quasi-random points with characteristics that are similar to randomly sampled points, while filling a unit hypercube as uniformly as is possible. For these experiments, 512 five-dimensional Sobol' points were generated. Each dimension value corresponded to one of the SOM parameters, and the dimensions were scaled to the appropriate parameter ranges. For the map dimension parameter, the scaled dimension values were rounded to the nearest integer.

For each parameter configuration, a 30-fold cross-validation was performed to estimate algorithm performance. For each cross-validation fold, a test set of unique examples was unused during training, while the remaining data examples were used to train and label a SOM. The test set for the cross-validation fold was then classified using the trained SOM.

B. Performance Measures

The first measure recorded for each of the 30 simulations was the percentage of unlabeled neurons generated by the example-centric neuron labeling algorithm. The mean of this measure, which is denoted $\mathcal{E}_U$, was calculated over the 30 simulations performed in each cross-validation.

The overall percentage of erroneous classifications over each test set was also used as a performance measure. The mean, $\mathcal{E}_G$, and standard deviation, $\mathcal{S}_G$, over the cross-validation were once again recorded for analysis.

To provide further detail on classification performance, the overall test set misclassifications were further differentiated into the percentage of errors due to unclassified test set examples (those examples that mapped to unlabeled neurons, and thus received no classification), and the percentage of

TABLE I
PARAMETER CHARACTERISTICS AND RANGES USED FOR EXPERIMENTAL ALGORITHMIC SETTINGS

Parameter	Symbol	Data type	Data set	Range
Map dimensions	Y and X	Ordinal	Iris plants	[2, 12]
			Ionosphere	[2, 18]
			Monk's problems	[2, 20]
			Pima Indians diabetes	[2, 27]
Initial learning rate	$\eta(0)$	Continuous	All	[0.0, 10.0]
Learning rate decay constant	τ_1	Continuous	All	(0.0, 1 500.0]
Kernel width	$\sigma(0)$	Continuous	All	(0.000, Y]
Kernel width decay constant	τ_2	Continuous	All	(0.0, 100.0]

errors due to misclassified test set examples (the examples that mapped to a labeled neuron, where the neuron label and actual example classification did not match). These two measures were also taken over the 30 cross-validation folds, to produce the mean and standard deviation of the error due to unclassified test set examples (respectively denoted $\mathcal{E}_{GU}$ and $\mathcal{S}_{GU}$), and the mean and standard deviation of the test set error due to misclassified examples ($\mathcal{E}_{GM}$ and $\mathcal{S}_{GM}$, respectively).

The objective of the analysis was to determine whether the percentage of unlabeled neurons had an effect on any of the measures that indicated classification performance accuracy. To this end, testing ascertained whether correlations existed between $\mathcal{E}_U$ and any of the other performance measures. The non-parametric Spearman's rank correlation coefficient [26] was used to assess the presence of correlations. The test statistic, ρ, indicates the degree of correlation, and has a value in the range $[-1, 1]$. A negative ρ value indicates a negative correlation, and a positive value denotes a positive correlation. Values of ρ closer to 1 or -1 indicate stronger correlation, while a value of 0 is indicative of uncorrelated data.

A p-value was also computed for the statistic, which was used to perform hypothesis testing on the reported results. A confidence level of 95% was used throughout the reported experiments. Additionally, to account for the family-wise error rate, a Bonferroni correction [27] was applied for each set of comparisons between $\mathcal{E}_U$ and one of the other measures.

The experimental data sets used during this investigation all originated from the UCI Machine Learning Repository [28]. Specifically, this research used the Iris plants, Ionosphere, monk's problems 1 to 3, and Pima Indians diabetes data sets.

C. Experimental Results

Tables II to IV show the statistical results for the comparisons between $\mathcal{E}_U$ and the means of the three test set misclassification error measures. The results of the comparisons between $\mathcal{E}_U$ and the standard deviations of the test set error measures are presented within Tables V to VII. Each table lists Spearman's rank correlation coefficients and p-values for all data sets. The strength of each correlation was also judged and is presented alongside the correlation coefficient. Asterisks marked p-values that denoted significant correlations.

TABLE II
CORRELATION STATISTICS FOR $\mathcal{E}_U$ AND $\mathcal{E}_G$

Data set	Correlation		p-value
Iris	0.717	Strong	$* 4.486 \times 10^{-82}$
Ionosphere	0.431	Moderate	$* 1.266 \times 10^{-24}$
Monk 1	-0.170	Very weak	$* 1.097 \times 10^{-4}$
Monk 2	-0.419	Moderate	$* 3.955 \times 10^{-23}$
Monk 3	-0.019	Very weak	0.671
Diabetes	0.708	Strong	$* 4.433 \times 10^{-79}$

TABLE III
CORRELATION STATISTICS FOR $\mathcal{E}_U$ AND $\mathcal{E}_{GU}$

Data set	Correlation		p-value
Iris	0.642	Strong	$* 6.960 \times 10^{-61}$
Ionosphere	0.923	Very strong	$* 1.310 \times 10^{-213}$
Monk 1	0.028	Very weak	0.530
Monk 2	0.015	Very weak	0.735
Monk 3	0.021	Very weak	0.635
Diabetes	0.854	Very strong	$* 9.945 \times 10^{-147}$

TABLE IV
CORRELATION STATISTICS FOR $\mathcal{E}_U$ AND $\mathcal{E}_{GM}$

Data set	Correlation		p-value
Iris	-0.315	Weak	$* 3.114 \times 10^{-13}$
Ionosphere	-0.821	Very strong	$* 1.883 \times 10^{-126}$
Monk 1	-0.220	Weak	$* 4.874 \times 10^{-07}$
Monk 2	-0.394	Moderate	$* 1.930 \times 10^{-20}$
Monk 3	-0.096	Very weak	0.030
Diabetes	-0.589	Moderate	$* 4.255 \times 10^{-49}$

The results in Table II indicate that all of the data sets except for the third monk's problem showed significant correlations between $\mathcal{E}_U$ and $\mathcal{E}_G$. The Iris and diabetes data sets showed strong positive correlations, while a moderate positive correlation was observed for the ionosphere set. The remaining two data sets showed negative correlations, where monk's problem 3 exhibited a moderate correlation, while monk's

problem 1 showed a very weak correlation. The results indicate that the correlations observed amongst the investigated data sets exhibited relatively mixed results, although the positive correlations were stronger than the negative ones.

Half the data sets showed no correlations between $\mathcal{E}_U$ and $\mathcal{E}_{GU}$, as illustrated in Table VI. However, the Iris, ionosphere, and Pima Indians diabetes data sets showed positive correlations. A strong correlation was seen in the Iris data set results, while the ionosphere and diabetes data sets both exhibited very strong correlations. A positive correlation was expected in this case, because an increased percentage of unlabeled neurons should result in more unclassified test set examples.

The final comparison to a mean test error measure was between $\mathcal{E}_U$ and $\mathcal{E}_{GM}$, and is shown in Table IV. Only the third monk's problem showed no correlation, while a negative correlation was visible in all other cases. The correlations were weak in the cases of the Iris and monk's problem 1 data sets, moderate for the second monk's problem and diabetes data sets, and very strong in the instance of the ionosphere set. These correlations are surprising, because lower percentages of unlabeled neurons are associated with a higher number of incorrectly classified test set examples. This paper hypothesizes that the correlation is due to a high number of labeled neurons resulting in poor quality labels. A higher number of labeled neurons requires a sparser mapping of labeling examples over the map. Labels are thus based on less representative samples of labeling data, reducing label accuracy.

The focus of the investigation then shifted to the correlation between the percentage of unlabeled neurons and the standard deviation of the three test set error measures. Correlations of this type were interesting because they are indicative of the effect on the variability of observed errors.

Table V depicts the correlations between $\mathcal{E}_U$ and $\mathcal{S}_G$. Every data set produced a statistically significant correlation between the measures. As for the $\mathcal{E}_U$ and $\mathcal{E}_G$ comparison, a mix of positive and negative correlations were present. Positive correlations between $\mathcal{E}_U$ and $\mathcal{E}_G$ coincided with positive correlations between $\mathcal{E}_U$ and $\mathcal{S}_G$. The same is true for negative correlations. The third monk's problem, which saw no statistically significant correlation between $\mathcal{E}_U$ and $\mathcal{E}_G$, produced a negative correlation when comparing $\mathcal{E}_U$ and $\mathcal{S}_G$. In comparison to the correlations between $\mathcal{E}_U$ and $\mathcal{S}_G$, the Iris data set produced a similar correlation, while the first two monk's problems generated stronger correlations, and the remaining three data sets all exhibited weaker correlations.

The correlations between $\mathcal{E}_U$ and $\mathcal{S}_{GU}$ are shown in Table VI. Once again, as was observed in Table III, statistically significant positive correlations were observed for the Iris, ionosphere, and diabetes sets. The remaining data sets again showed no statistically significant correlation. These results illustrate that an increased percentage of unlabeled neurons was not only associated with a higher number of unclassified data examples in the test set, but also higher variability in the number of unclassified test set data examples.

Finally, Table VII summarizes the correlations observed between $\mathcal{E}_U$ and $\mathcal{S}_{GM}$. The correlations observed here were

TABLE V
CORRELATION STATISTICS FOR $\mathcal{E}_U$ AND $\mathcal{S}_G$

Data set	Correlation		p-value
Iris	0.714	Strong	$* 5.325 \times 10^{-81}$
Ionosphere	0.145	Very weak	$* 9.861 \times 10^{-4}$
Monk 1	−0.281	Weak	$* 1.008 \times 10^{-10}$
Monk 2	−0.123	Very weak	$* 5.323 \times 10^{-3}$
Monk 3	−0.203	Weak	$* 3.548 \times 10^{-6}$
Diabetes	0.404	Moderate	$* 1.418 \times 10^{-21}$

TABLE VI
CORRELATION STATISTICS FOR $\mathcal{E}_U$ AND $\mathcal{S}_{GU}$

Data set	Correlation		p-value
Iris	0.621	Strong	$* 5.095 \times 10^{-56}$
Ionosphere	0.906	Very strong	$* 3.876 \times 10^{-192}$
Monk 1	0.037	Very weak	0.400
Monk 2	0.023	Very weak	0.608
Monk 3	0.031	Very weak	0.485
Diabetes	0.846	Very strong	$* 4.087 \times 10^{-141}$

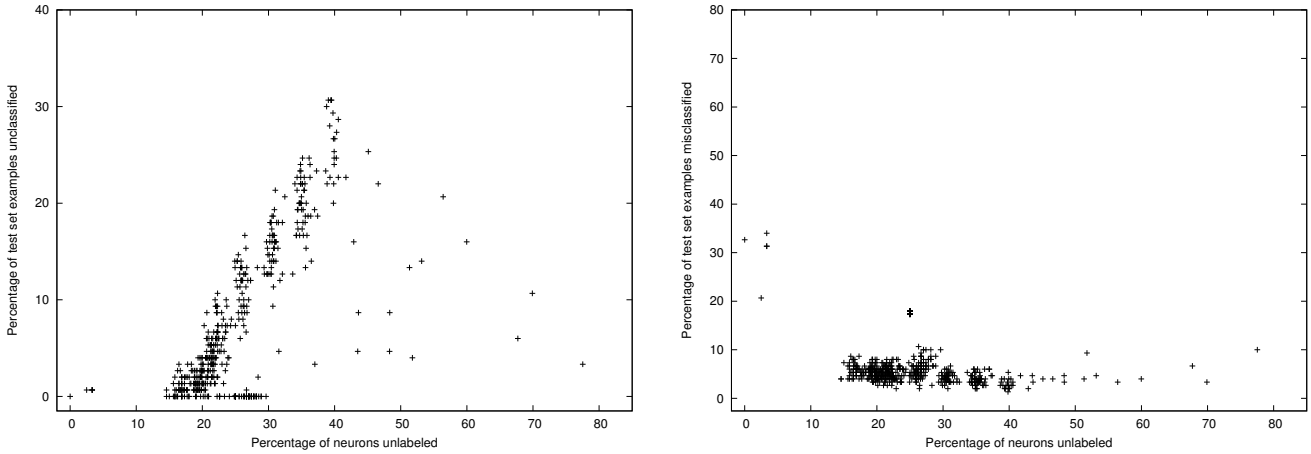
TABLE VII
CORRELATION STATISTICS FOR $\mathcal{E}_U$ AND $\mathcal{S}_{GM}$

Data set	Correlation		p-value
Iris	−0.290	Weak	$* 2.129 \times 10^{-11}$
Ionosphere	−0.660	Strong	$* 2.754 \times 10^{-65}$
Monk 1	−0.326	Weak	$* 4.169 \times 10^{-14}$
Monk 2	−0.143	Very weak	$* 1.223 \times 10^{-3}$
Monk 3	−0.197	Weak	$* 7.045 \times 10^{-6}$
Diabetes	0.020	Very weak	0.656

negative in most cases, in much the same way as Table IV illustrated for $\mathcal{E}_U$ and $\mathcal{E}_{GM}$. The only exception was for the diabetes data set, which showed no significant correlation. The Iris and monk's problem 1 data sets showed the same level of correlation strength as in Table IV. In only one case, for the final monk's problem, did the strength of the correlation increase, while the correlation was weaker in the case of ionosphere and the second monk's problem. These observations indicate that lower percentages of unlabeled neurons were associated with a higher degree of uncertainty in the test set error due to misclassification, and vice versa.

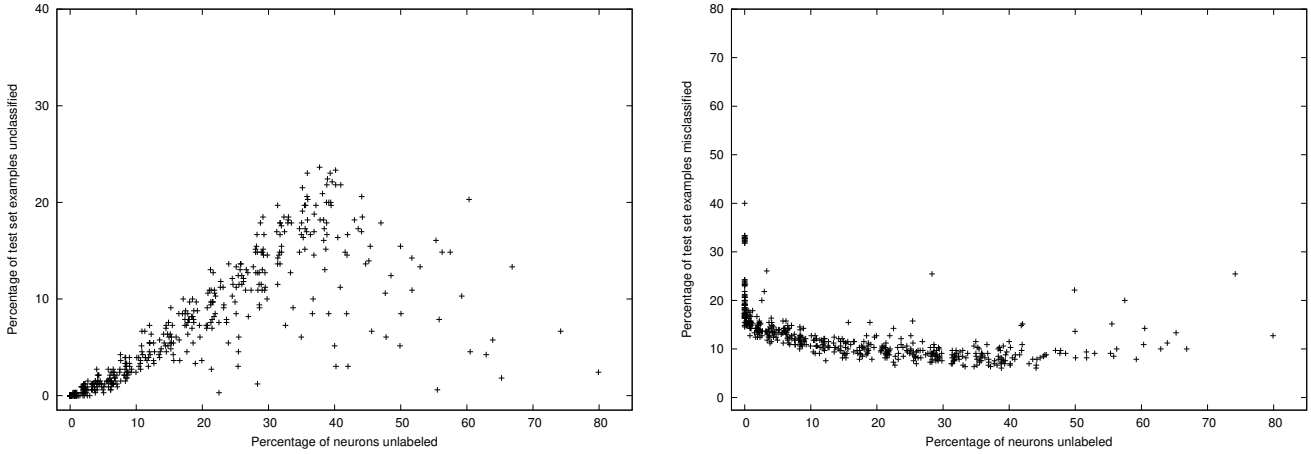
In order to graphically illustrate the correlations between $\mathcal{E}_U$ and the three test set classification error measures, two scatter plots are presented for each data set. In each case, the first scatter plot illustrates the correlation between $\mathcal{E}_U$ versus $\mathcal{E}_{GU}$, while $\mathcal{E}_U$ versus $\mathcal{E}_{GM}$ are compared in the second. The Iris, ionosphere, monk's problem, and Pima Indians diabetes data sets are respectively illustrated in Figs. 5 to 10.

These scatter plots are shown to illustrate the ranges over which the error measures fluctuated. It is apparent that the error due to data example misclassifications tends to be



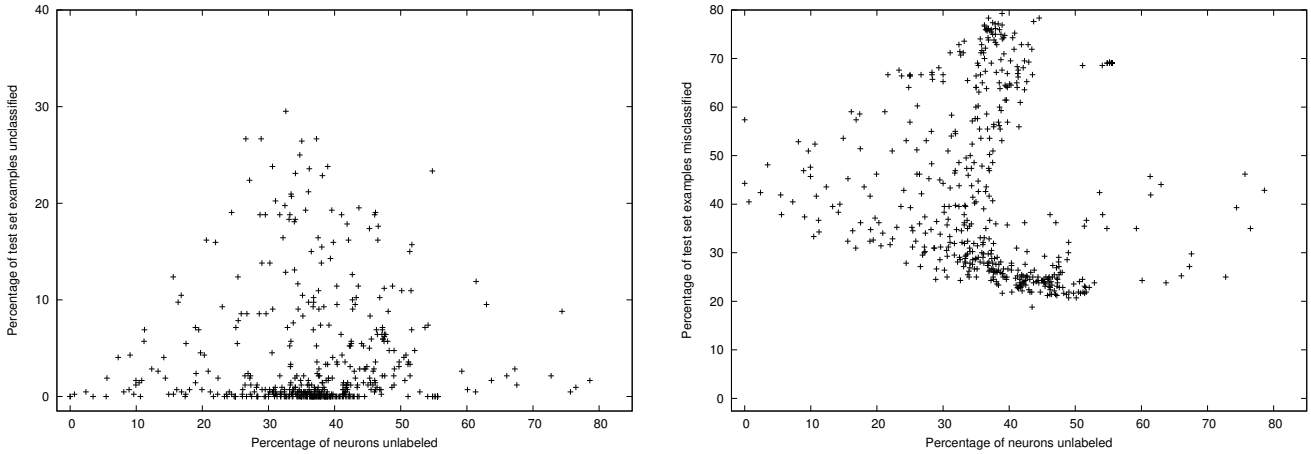
(a) (b)

Fig. 5. Performance measure comparisons for the Iris plants data set: (a) $\mathcal{E}_U$ versus $\mathcal{E}_{GU}$ comparison, (b) $\mathcal{E}_U$ versus $\mathcal{E}_{GM}$ comparison.



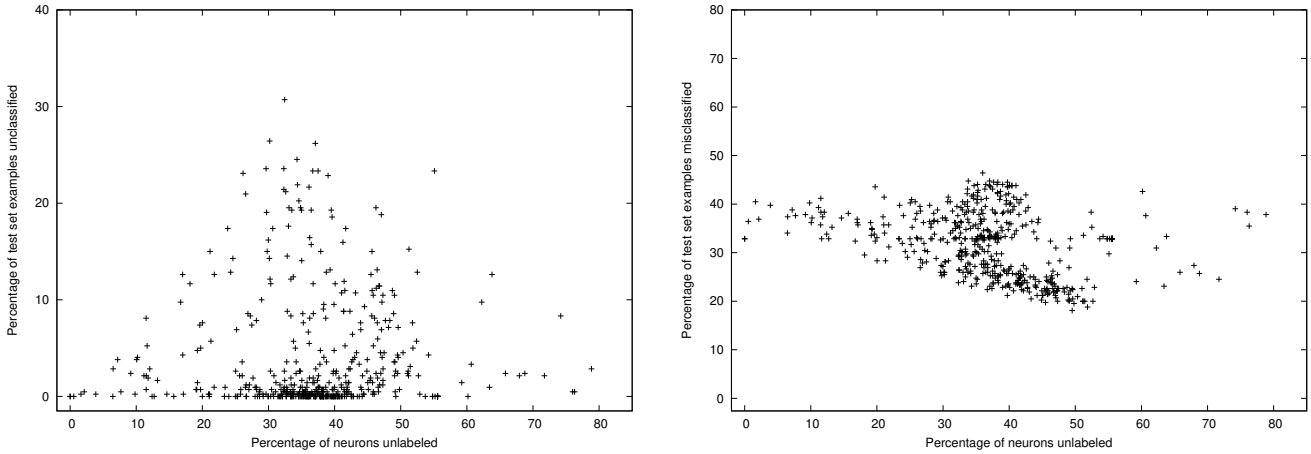
(a) (b)

Fig. 6. Performance measure comparisons for the Ionosphere data set: (a) $\mathcal{E}_U$ versus $\mathcal{E}_{GU}$ comparison, (b) $\mathcal{E}_U$ versus $\mathcal{E}_{GM}$ comparison.



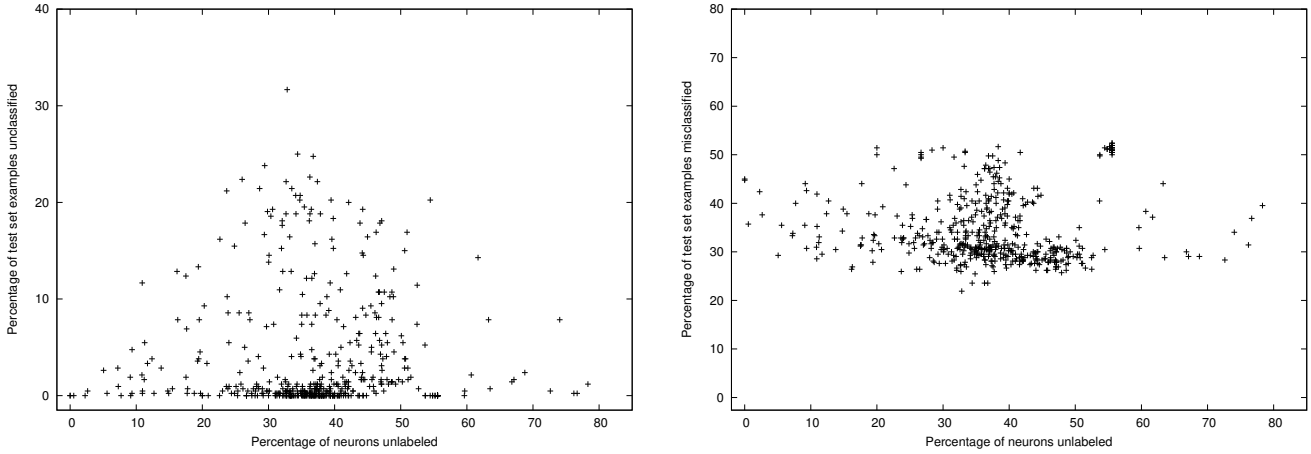
(a) (b)

Fig. 7. Performance measure comparisons for the Monk's problem 1 data set: (a) $\mathcal{E}_U$ versus $\mathcal{E}_{GU}$ comparison, (b) $\mathcal{E}_U$ versus $\mathcal{E}_{GM}$ comparison.



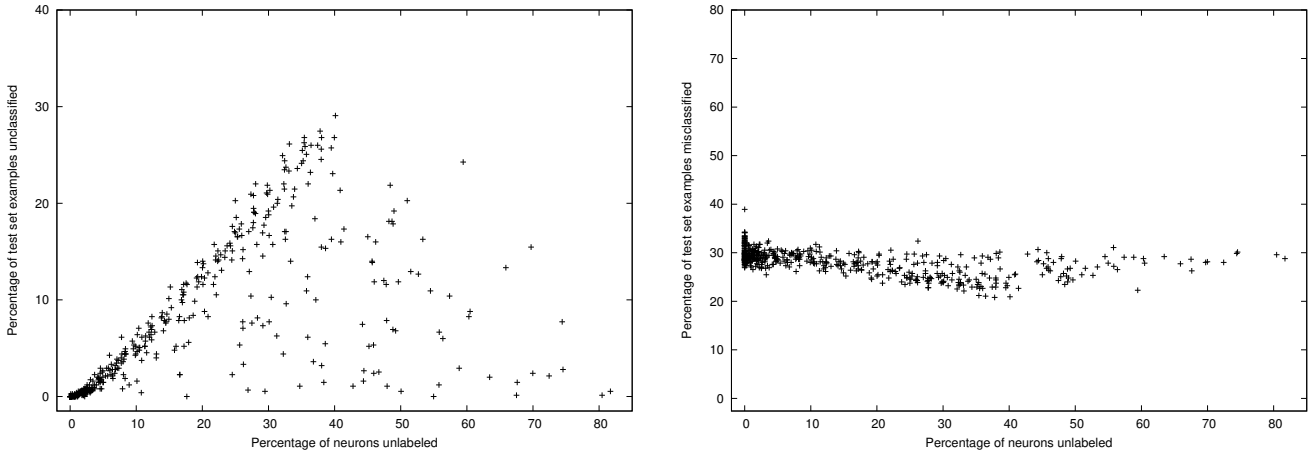
(a) (b)

Fig. 8. Performance measure comparisons for the Monk's problem 2 data set: (a) $\mathcal{E}_U$ versus $\mathcal{E}_{GU}$ comparison, (b) $\mathcal{E}_U$ versus $\mathcal{E}_{GM}$ comparison.



(a) (b)

Fig. 9. Performance measure comparisons for the Monk's problem 3 data set: (a) $\mathcal{E}_U$ versus $\mathcal{E}_{GU}$ comparison, (b) $\mathcal{E}_U$ versus $\mathcal{E}_{GM}$ comparison.



(a) (b)

Fig. 10. Performance measure comparisons for the Pima Indians diabetes data set: (a) $\mathcal{E}_U$ versus $\mathcal{E}_{GU}$ comparison, (b) $\mathcal{E}_U$ versus $\mathcal{E}_{GM}$ comparison.

concentrated around fairly high levels, in comparison to the error due to unclassified data. This suggests that the error due to misclassification is a much more important consideration when a neuron labeling algorithm has to be selected for a SOM-based data analysis task, be it EDA or DM focused.

V. Practical Recommendations

This paper bases the practical suggestions outlined in this section on the different focuses of EDA and DM tasks. EDA and DM each have different focuses, due to the differing involvement of a human analyst in each domain.

In the case of EDA, the fine-grained accuracy of the SOM model is often less important than the ability of the map to provide a broad and interpretable overview of the analyzed data set. In such a situation, the number of unlabeled neurons is a more important factor, because uncharacterized neurons create a fragmented model that is difficult to interpret. This means that a potential increase in classification errors is less significant if it means a more completely labeled map.

On the other hand, if SOMs are to be used in a DM context, the accuracy of the model is of paramount importance because the map is not analyzed by human experts. In such a case the percentage of unlabeled neurons should not be focused on as an aspect of SOM performance that needs to be optimized. Instead, only the accuracy of the model is important, as embodied in the test set classification accuracy.

VI. Conclusions and Future Work

This paper investigated the effect that unlabeled neurons have on a SOM data model when a space of valid SOM parameter ranges is considered. Brief overviews of the SOM and several algorithms for labeling SOM neurons were presented. The experimental investigation analyzed a population of maps with different configurations, and focused on the correlations between the percentage of unlabeled neurons and the means and standard deviations of three test set classification error measures. The interaction between these measures has not been the focus of prior research, and is important because the adequate use of labeling algorithms relies on such knowledge. Finally, the importance of unlabeled neurons in the contexts of EDA and DM was considered. It was recommended that EDA should avoid unlabeled neurons, but that DM should not focus on optimizing the percentage of label coverage because model accuracy becomes the primary concern.

Future work will investigate the relationship between unlabeled neurons and the performance of example-centric cluster labeling, which was ignored in this study. The reported experiments focused on the effect of unlabeled neurons across an area of the valid SOM parameter space, while the influences that the parameters have on the percentage of unlabeled neurons and classification performance were ignored. Future work will thus investigate the effects of these parameters on the interaction between unlabeled neuron percentages and classification error. The result of unlabeled neurons produced by unsupervised labeling methods, such as the LabelSOM algorithm, is also a potential topic for future exploration.

References

[1] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Springer-Verlag, 2001.

[2] S. Kaski, J. Kangas, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) papers: 1981–1997," *Neural Computing Surveys*, vol. 1, pp. 102–350, 1998.

[3] M. Oja, S. Kaski, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) papers: 1998–2001 addendum," *Neural Computing Surveys*, vol. 3, pp. 1–156, 2003.

[4] M. Pöllä, T. Honkela, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) papers: 2002-2005 addendum," Helsinki Univ. Technology, Dept. Inform. and Comput. Sci., Tech. Rep. TKK-ICS-R23, 2009.

[5] G. Deboeck and T. Kohonen, Eds., *Visual Explorations in Finance with Self-Organizing Maps*. Springer-Verlag, 1998.

[6] E. Alhoniemi, "Analysis of pulping data using the self-organizing map," *TAPPI J.*, vol. 83, no. 7, pp. 66–75, 2000.

[7] J. Tuckova, "The possibility of Kohonen self-organizing map applications in medicine," in *Proc. ECMSM*, 2013, pp. 1–6.

[8] W. S. van Heerden and A. P. Engelbrecht, "A comparison of map neuron labeling approaches for unsupervised self-organizing feature maps," in *Proc. IJCNN*, 2008, pp. 2139–2146.

[9] S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds., *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[10] J. Vesanto, "Data exploration process based on the self-organizing map," Ph.D. dissertation, Helsinki Univ. Technology, Dept. Comput. Sci. Eng., 2002.

[11] A. Ultsch and D. Korus, "Automatic acquisition of symbolic knowledge from subsymbolic neural networks," in *Proc. EUFIT*, vol. 1, 1995, pp. 326–331.

[12] J. Malone, K. McGarry, W. Stefan, and C. Bowerman, "Data mining using rule extraction from Kohonen self-organising maps," *Neural Computing and Applicat.*, vol. 15, no. 1, pp. 9–17, 2006.

[13] W. S. Van Heerden and A. P. Engelbrecht, "HybridSOM: A generic rule extraction framework for self-organizing feature maps," in *Proc. CIDM*, 2009, pp. 17–24.

[14] T. Kohonen, "Self-organizing formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.

[15] A. Azcarraga, M.-H. Hsieh, S.-L. Pan, and R. Setiono, "Improved SOM labeling methodology for data mining applications," in *Soft Computing for Knowledge Discovery and Data Mining*, O. Maimon and L. Rokach, Eds. Springer, 2008, pp. 45–75.

[16] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed., ser. Springer Series in Information Sciences. Springer, 1989.

[17] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*. Auerbach Publications, 2007.

[18] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[19] W. S. Van Heerden and A. P. Engelbrecht, "Unsupervised weight-based cluster labeling for self-organizing maps," in *Proc. WSOM*, 2012, pp. 45–54.

[20] A. Corradini and H.-M. Gross, "A hybrid stochastic-connectionist architecture for gesture recognition," in *Proc. ICIIS*, 1999, pp. 336–341.

[21] G. Deboeck, "Public domain vs. commercial tools for creating neural self-organizing maps," *PC AI*, vol. 13, no. 1, pp. 27–33, 1999.

[22] C. Serrano-Cinca, "Self organizing neural networks for financial diagnosis," *Decision Support Syst.*, vol. 17, no. 3, pp. 227–238, 1996.

[23] A. Rauber and D. Merkl, "Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets," in *Proc. PAKDD*, 1999, pp. 228–237.

[24] N. Franken, "Visual exploration of algorithm parameter space," in *Proc. CEC*, 2009, pp. 389–398.

[25] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Computational Math. and Math. Physics*, vol. 7, no. 4, pp. 86–112, 1967.

[26] C. Spearman, "The proof and measurement of association between two things," *Amer. J. Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[27] L. A. Moyé, *Multiple Analyses in Clinical Trials: Fundamentals for Investigators*. Springer, 2003.

[28] D. W. Aha, C. L. Blake, S. J. Hettich, E. J. Keogh, C. J. Merz, and P. M. Murphy, "UCI repository of machine learning databases," 1998, univ. of California, Irvine, Dept. Inform. Comput. Sci.