JOURNAL OF ADVANCED TRANSPORTATION
J. Adv. Transp. 2010; **44**: 162–175
Published online 11 June 2010 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/atr.130

The analysis of motor vehicle crash clusters using the vector quantization technique

Lorenzo Mussone[1]* and Karl Kim[2]

[1]*Politecnico di Milano BEST, Via Bonardi 3, 20133 Milano, Italy*
[2]*Department of Urban & Regional Planning, University of Hawaii at Manoa, Honolulu, HI 96822, U.S.A.*

SUMMARY

In this paper, a powerful tool for analyzing motor vehicle data based on the vector quantization (VQ) technique is demonstrated. The technique uses an approximation of a probability density function for a stochastic vector without assuming an ''*a priori*'' distribution. A self-organizing map (SOM) is used to transform accident data from an N-dimensional space into a two-dimensional plane. The SOM retains all the original data yet provides an effective visual tool for describing patterns such as the frequency at which a particular category of events occurs. This enables new relationships to be identified. Accident data from three cities in Italy (Turin, Milan, and Legnano) are used to illustrate the usefulness of the technique. Crashes are aggregated and clustered crashes by type, severity, and along other dimensions. The paper includes discussion as to how this method can be utilized to further improve safety analysis. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: vehicular accidents; data transformation; artificial neural networks; self-organizing maps; clustering

1. INTRODUCTION

The multidimensional nature of motor vehicle crashes creates two problems for safety analysts. First, it is often difficult to design and implement effective sampling strategies because the underlying distributions of crash involved motorists, vehicles, roadways, environments, and factors are not known. Second, it is computationally intensive to analyze large administrative databases such as police crash report files which may contain thousands of cases. An important step in data analysis is to understand the underlying structure of the information being analyzed. This can help to determine whether the data are sufficient in a statistical sense and which statistical methods and models can be appropriately applied. There are many examples of research in which these issues have been addressed [1,2].

Pattern recognition as an initial activity in data analysis is intuitively appealing. Pattern recognition involves classifying a sample of cases or observations into a smaller number of mutually exclusive groups or clusters based on similarities among their attributes. With this type of analysis there are no dependent and independent variables. Instead, the approach is largely descriptive. There are two general ways of recognizing patterns in data: statistical techniques and machine learning (artificial intelligence) approaches. Statistical clustering techniques are widely known and have been discussed extensively elsewhere [3–5].

This paper is focused on the use of artificial neural networks (ANN) for pattern recognition. We use motor vehicle crash data from three Italian cities to demonstrate this technique. It builds on earlier work by Mussone *et al*. [6] and proposes an application of the vector quantization (VQ) technique. It also builds on the work of Kim *et al*. [7] and others who utilize loglinear modeling techniques and other

*Correspondence to: Lorenzo Mussone, Politecnico di Milano BEST, Via Bonardi 3, 20133 Milano, Italy.
E-mail: mussone@polimi.it

methods such as Rough Set Analysis [8] for the analysis of cross-classified categorical data in traffic safety research. The technique employed in this paper, however, avoids the typical problems and limitations that arise with correspondence analysis, factorial analysis, and principal component analysis [9]. Quantization is the mapping of an input value on to a finite number of output values. Vector quantization is a version of ''scalar quantization,'' used in data compression and pattern recognition. It is popular with computer applications involving the compression of image or voice data, but it is also a useful tool for cluster analysis. With VQ, an input pattern or word is matched to a set of stored patterns or words and the best match is chosen.

The underlying purpose of these techniques is to discover the similarity or ''distance'' between cases when each case is represented by a collection of many different variables.

This paper strives to find the following:

- prevailing causes of crashes for a road or set of roads; this can be useful when inspecting and reviewing roadway geometry.
- location of a critical segment or location where attention is needed in terms of engineering, enforcement, or education.
- relationships between fatal crashes and roadway attributes; this can help to determine the degree of randomness of fatal crashes for certain roads.

Cluster analysis may not be an efficient tool when N-dimensional space is involved, that is, unless data are projected, preserving their statistical features, in a two- or three-dimensional space before conducting the analysis. The VQ technique is used to create clusters of similar types of motor vehicle crashes which are, in turn, compared to crashes occurring on similar roads. By grouping data into classes, VQ helps to illustrate patterns and relationships. When there are many cases, the frequency of each class can be used as an accident probability, and VQ can be used to predict or study new scenarios (see Refs. [10–14] for additional details regarding the technique).

2. METHOD

VQ is a classical approximation of a probability density function (pdf), $p(\mathbf{X})$, for a stochastic process, $\mathbf{X} \in R^n$. A set of codebook vectors (CVs), $\mathbf{m}_i \in R^n$, $i = 1,2 \ldots, k$ (k is the number of CVs) is constructed. A CV is made up of two components: a vector which has the same dimension of the process $\mathbf{X}$ and a pointer to a map of a space, usually of two (or sometimes three) dimensions on which the CVs are projected. The set of CVs and their pointers summarize the information contained in $\mathbf{X}$.

The number of CVs corresponds to the number of elements (cells) of the space and it is indirectly proportional to the degree of aggregation of data and then to the expected number of data clusters. There are no rules to define the number of CVs (or cells) since it depends on the specific data set and the process being analyzed. A ''trial and error'' approach is used. At the end of the learning phase, if the information contained in a single CV is too general, the number of CVs is increased. If on the other hand, the results are too specific, the number of CVs is reduced. Working with the CVs, therefore, is the central activity of the technique.

Cells are mathematical entities whose boundaries are not necessarily regular. The distance from the center of one cell to the centers of neighboring cells can be calculated with different formulas. VQ can be formulated in Euclidean space and calculated with the Euclidean norm (as in 1) for every cell meaning perfectly regular boundaries. Further VQ uses ''nearest neighbor'' routine, characterized by CVs in which the relative final distance between cells is as low as possible. The final values of $\mathbf{m}_i$ that best fit the stochastic process $\mathbf{X}$ are those pointing to cells in which relative distances have been minimized.

The approximation of $\mathbf{X}$ entails finding the best or closest CV $\mathbf{m}_c$ for each vector $\mathbf{x}$ so as to minimize their total distance, as Gersho and Gray [11] have described. More than one vector $\mathbf{x}$ can be linked to the same $\mathbf{m}_c$.

Figure 1 contains an example of a two-dimensional quantifier (belonging to R^2) is presented. The first box contains initial data in their original space (R^n). The second box contains a set of CVs chosen on a map on a plane, the third one contains transformed data in the auxiliary space (R^2) after application of the VQ technique (that is after learning of $\mathbf{m}_i$) in which each vector $\mathbf{x}$ is linked to its best CV. Vector $\mathbf{x}$ belonging to the same CV or located close to it, can be considered belonging to the same cluster (in this

DOI: 10.1002/atr

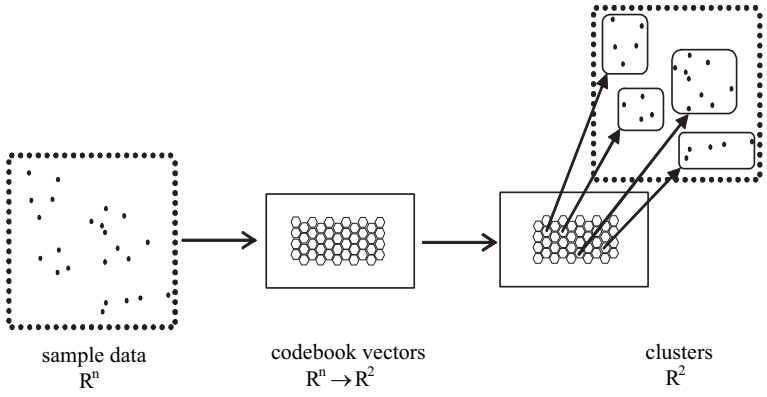

Figure 1. Summary diagram illustrating vector quantization technique.

R^2 space). The image appears more regular because of the use of a hexagonal lattice. The same could be obtained by using a rectangular or square lattice (changing the number of neighbor cells).

The mathematical basis for VQ involves determination of the distances between $\mathbf{x}$ and CVs and finding the best c (of $\mathbf{m}_c$) for each $\mathbf{x}$ that minimizes the overall distance. The index c can be defined by the decision process:

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \tag{1}$$

or equivalently

$$c = \arg\min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \tag{2}$$

where the norm is assumed to be Euclidean. The first member of (1) is the quantization error for the CV c. An approach for selecting CVs is to minimize the quantization error, E, that is:

$$E = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \|\mathbf{x}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) - \mathbf{m}_c\|^2 p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \mathrm{d}\mathbf{x}_1 \mathrm{d}\mathbf{x}_2, \ldots, \mathrm{d}\mathbf{x}_n \tag{3}$$

where $p(\mathbf{x})$ is the joint pdf of $\mathbf{x}$ [the integral is generally over $(-\infty, +\infty)$, that is, the interval for the pdf of $\mathbf{x}$]; n is the dimension of data space R^n. E depends not only on $\mathbf{m}_c$ but also on $\mathbf{x}$ conditioning the best coupling with the initial $\mathbf{m}_i$. The c index is a function of $\mathbf{x}$ and all $\mathbf{m}_i$, as the gradient of E with respect to each CV $\mathbf{m}_i$ is unknown. If the value of the CVs $\mathbf{m}_i$ changes, the c index can also change, moving discontinuously to single out a new CV nearer to $\mathbf{x}$.

A local approximation rather than the entire form of the pdf of a CV is of interest. The best choice, therefore, of the values of the CVs, $\mathbf{m}_i$, is such that their local density approximates the function $[p(\mathbf{x})]^{n/(n+2)}$ as long as the number of CVs is large enough where n reflects the dimension of $\mathbf{x}$ (see Ref. [15]). In practical terms, it is often that $n > 2$. For this reason, VQ approximates the pdf with a limited set of discrete parameter vectors.

Closed-form solutions to determination of CVs have not, until recently, been proposed. Iterative methods of approximation have been utilized. Kohonen [12] proposed a solution by assuming that the $p(\mathbf{x})$ function is continuous and the functions including the c parameter (integer and therefore discontinuous in R) could be differentiated. The norm is a scalar, therefore, the product of the scalar is an admissible operation which assumes the square of (1):

$$\|\mathbf{x} - \mathbf{m}_c\|^2 = [\min_i \{\|\mathbf{x} - \mathbf{m}_i\|\}]^2 = \lim_{r \to -\infty} \left(\sum_i \|\mathbf{x} - \mathbf{m}_i\|^r \right)^{2/r} \tag{4}$$

where the limit is a positive real value. The function inside the limit is continuous single valued and differentiable except for those singular values where $\mathbf{x}$ is equal to $\mathbf{m}_i$, which is unlikely to occur. The

DOI: 10.1002/atr

gradient of (3) (the gradient of a scalar function $f(\mathbf{x})$ with respect to a vector variable $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, is a vector field whose components are the partial derivatives of f with respect to $\mathbf{x}$) becomes, by using (4):

$$\nabla_{\mathbf{m}_j} E = \int \lim_{r \to -\infty} \nabla_{\mathbf{m}_j} \left(\sum_i \|\mathbf{x} - \mathbf{m}_i\|^r \right)^{2/r} p(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{5}$$

Taking into account that:

$$\lim_{r \to -\infty} \left(\frac{\|\mathbf{x} - \mathbf{m}_j\|}{\|\mathbf{x} - \mathbf{m}_c\|} \right)^r = \delta_{cj} \tag{6}$$

where δ_{cj} is the Dirac function (it is 1 for $c = j$, 0 elsewhere), (5) can be written as:

$$\nabla_{\mathbf{m}_j} E = -2 \int \delta_{cj} (\mathbf{x} - \mathbf{m}_j) p(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{7}$$

An approach to the solution of (7) implies an iterative approach to obtain the desired solution with a certain approximation. Let t be the index of such iteration which can represent time, such that the sample function of gradient in (7), at time t, is:

$$\nabla_{\mathbf{m}_j} E|_t = -2\delta_{cj} [\mathbf{x}(t) - \mathbf{m}_j(t)] \tag{8}$$

In the E space the steepest descent occurs in the opposite direction of the gradient. By denoting with $\alpha(t)$ the updating factor, the time evolution of CV, i, becomes:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) \delta_{ci} [\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{9}$$

This is also the learning law for the connection weights in self-organizing maps (SOMs), as shown in the next section. The learning law of (9) can be rewritten as, assuming $\rho = \alpha(t)\delta_{ci}$:

$$\mathbf{m}_i(t+1) = [1 - \alpha(t)\delta_{ci}] \mathbf{m}_i(t) + \alpha(t)\delta_{ci} \mathbf{x}(t) = (1 - \rho)\mathbf{m}_i(t) + \rho \mathbf{x}(t) \tag{10}$$

which represents a well-known learning algorithm [16] where, under some hypotheses on ρ, the learning of $\mathbf{m}_i$ leads to the best approximate of $\mathbf{x}(t)$.

2.1. Self-organizing maps

An SOM is a particular type of ANN. A neural network (NN) consists of both processors and connections. Processors are linked to each other by connections which make up a network of one or more layers. Processors are usually called neurons which are characterized by a transfer function converting signals coming from NN inputs or other neurons (see Ref. [17]).

The special two-dimensional case of an SOM is drawn in Figure 2 (see Section 4.1) which represents a map of input data (belonging to R^n) on to a two-dimensional array of nodes, built up by output neurons. In this figure, the NN is represented by the X–Y plane, while the Z-axis represents the accident count for each CV.

Connections between neurons can be formed using either a rectangular lattice where each neuron has four possible connections or a hexagonal one (as shown in Figure 1) with six sides. The hexagonal lattice limits vertical and horizontal learning directions more than the rectangular lattice. Square lattices are generally to be avoided in order to obtain a more stable orientation in the data space.

SOMs are characterized by a learning rule of connection weights $\mathbf{m}_i$:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t) [\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{11}$$

The $h_{ci}(t)$ function is called the neighborhood function and measures the interaction between the two neurons i and c. It must satisfy conditions for the convergence of the learning algorithm such that

DOI: 10.1002/atr

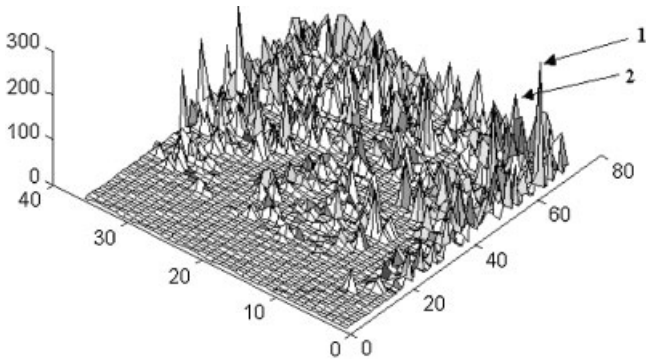

Figure 2. Map of codebook cells classified by index (represented by gray colored faces) and with the number of crashes (Z-axis), Turin.

$h_{ci}(t) \to 0$ when $t \to \infty$. A frequent choice is based on the use of a Gaussian function:

$$h_{ci}(t) = \alpha(t)\exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right) \tag{12}$$

where $\alpha(t)$ ($\in]0,1]$) has the role of the learning factor (or step) and $\sigma(t)$ controls the width of interaction between cells, while $\alpha(t)$ and $\sigma(t)$ are decreasing monotonic functions. Vectors $\mathbf{r}_c$ and $\mathbf{r}_i$ ($\in R^2$) represent the coordinates of nodes i and c, respectively.

Equations (9) and (11) differ only in the use of a Gaussian function instead of a Dirac δ function. For this reason, the solution of the VQ problem obtained by a two-dimensional SOM cannot be considered exact. The error is minor compared to the advantages associated with simplification.

SOMs preserve the topological structure of the original information. Clusters, samples, and their relationships, while expressed in N-dimensional space, are preserved in the two-dimensional space of the SOM. This representation does not depend on the number of independent and dominant variables. The dimension of the map or the number of CVs is greater than the number of cluster the operator decides to identify. It is possible to have either a high degree of aggregation or a high level of detail.

After reformulating the data into an SOM, finding clusters becomes easier because of the two-dimensional space. Data reduction also improves computation efficiencies. The SOM filters out distortions and random variations present in the data because CVs are based on averages and less sensitive to fluctuations.

The use of SOMs, like other ANN models, begins with an iterative learning phase. The learning phase provides the setup of an NN which can then be later used for analysis. This phase changes according to the type of NN. With SOMs, learning is described by (11) which provides the rule for updating connection weights between neurons. The SOM learning phase determines the dimensions of the computational problem and the structure of the map. This phase involves evaluation and comparison of total quantization error (see Ref. [15]). The problem is to determine which function should be used for $\alpha(t)$. The learning process can be compromised if it decreases toward zero too quickly. There is no validation since the final result is just the product of an assignment in order to produce the best cluster configuration (in the sense of minimum quantization error).

The use of average quantization error may be useful with small maps in order to identify that random initialization that has produced the best result. This approach is not practical with larger maps as the number of cases and dimensions increase. It is more useful to use a weighted distance measure, such as:

$$\sum_i h_{ci}\|x - m_c\|^2 \tag{13}$$

where h_{ci} is the neighborhood function in (12). This index can also be used to determine the best dimension of the map.

A label (which is information present in the database) can be linked to each CV without using it in the learning phase. These labels help to visualize the results. Once the best coupling with the CVs is found and, correspondingly, with each neuron or cell of the network, the list of addressed data vectors for each cell is inspected and labels are evaluated (generally each CV addresses more than one data

DOI: 10.1002/atr

vector). The strategy involves linking labels to neurons which are then used to represent the whole cell. The simplest method for analyzing results is to visually inspect the maps of the codebooks and the other information linked to them (see Ref. [18] for more detailed discussion). One type of map depicts the distance between neighboring cells marked by different labels on a single plane (e.g., see Figures 3–5 in Section 4). Another one plots on a three-dimensional surface, the number of vectors mapped onto

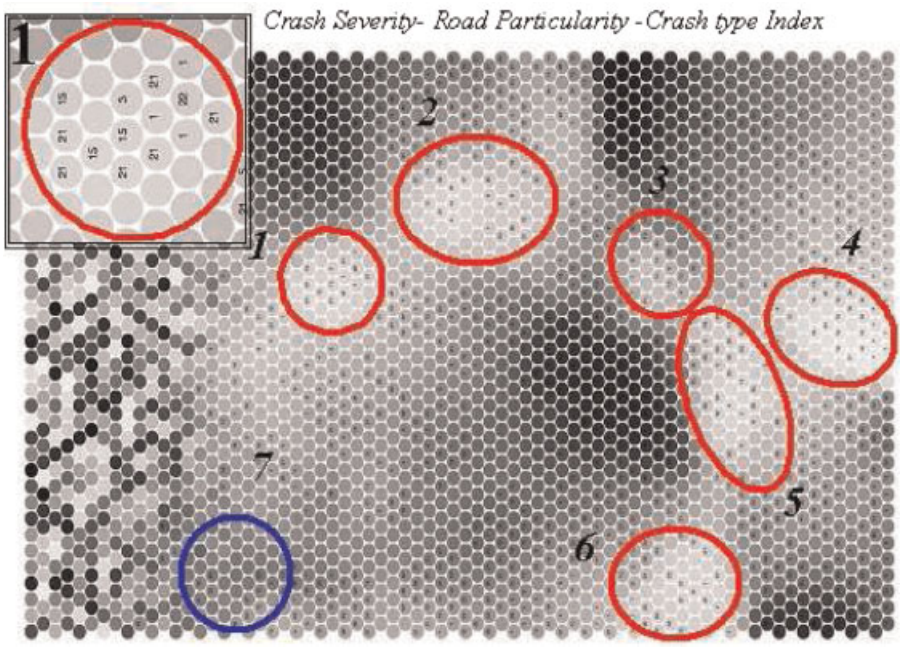

Figure 3. Main clusters for crash distribution according to index and road name, Turin.

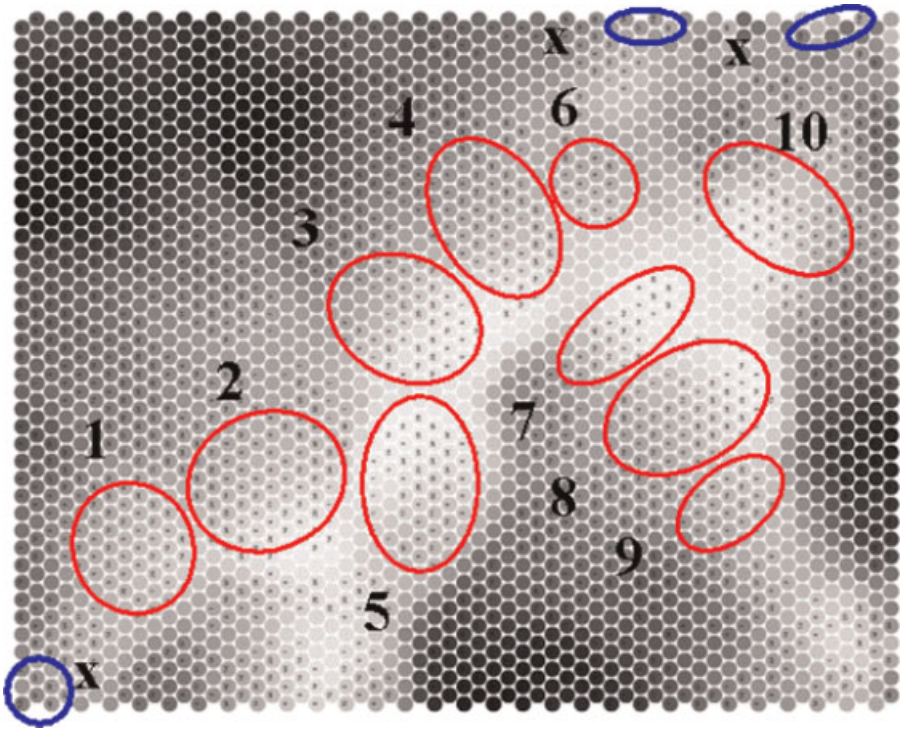

Figure 4. Main clusters for accident distribution according to index, Milan.

DOI: 10.1002/atr

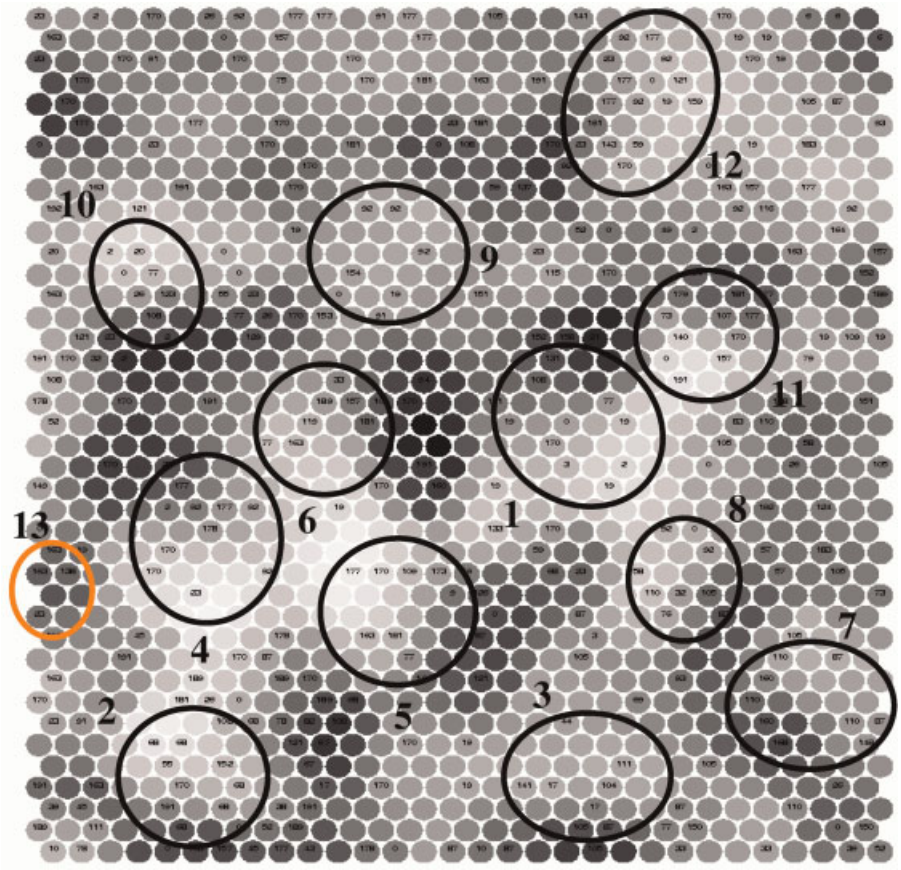

Figure 5. Main clusters for accident distribution according to index and road name, Legnano.

each CV, or the quantization error (e.g., in Figure 2). *X*- and *Y*-axes simply identify the coordinates of CVs. *Z*-axis identifies the number of observations associated with a CV, or the data cluster associated with that CV.

The software used to develop this application is cited in Refs. [12,13]. It consists of three modules for map initialization, training, and quantization error evaluation. Information is extracted using the different programs in order to match input data to the best codebook, to calculate and visualize distances between CVs, to generate what is known as a ''Sammon'' map of input data, to select and visualize selected planes present in multidimensional space. The computational time is typically around 5 minutes per network. Obviously, the formulation and investigation of the best network structure can take much longer.

3. THE ANALYSIS OF CRASH DATA

The data used in this section came from three cities in Northern Italy (Milan, Turin, and Legnano). Data were collected by the police and include all injury-producing crashes. Property damage only crashes were excluded from the analysis. Each database is unique and contains different variables and coding schemes. Average daily traffic, roadside conditions, and other variables of interest that influence crashes are contained in the databases.

The first step is to construct a database for the learning phase. Some fields are coded into tables containing items which cannot be classified as cardinal values because each item has its own information content. After grouping similar items, their content was transformed with a binary code representing each item. The number of bits necessary for each field is a function of the number of groups. Other variables containing descriptive information involved the creation of an ordinal

DOI: 10.1002/atr

classification based on linear or nonlinear weights assigned to each item. The second step involves the construction of an SOM according to rules given in the previous section. The third and final step entails the analysis of the map.

The meaning of some variables (time, number of injured people, number of deaths, etc.) is apparent while others require further explanation. ''Road code'' is the name of the road and in the case of an intersection is the name of the most important road; it is not used for learning. ''Crash type'' refers to the type of collision and whether it is involved one or two vehicles or a pedestrian. It can be head-on, frontal–lateral, lateral, bump, collision with pedestrian, crash with fixed object, etc. ''Road characteristics'' describe whether or not the crash occurred at an intersection and the type of traffic control device such as a traffic signal or roundabout. It also describes the roadway alignment or (straight or curved, sloped or level) or if the crash occurred within a tunnel. The variable ''road type'' describes the number of lanes and the administrative type of road; while ''type of traffic'' reports on the volume of traffic in terms of ''no traffic,'' ''low traffic level,'' ''medium traffic level,'' and ''high traffic level.''

3.1. Data from three Italian cities: Turin, Milano, and Legnano

The data from the three cities used in this study are described in Table I. For Turin, the data cover a period of approximately 4 years, from January 1997 to May 2000 (31 864 records). In Table I, the fields used in the analysis are listed. Fields such as crash type, road characteristics, and road type are transformed after grouping similar items using a specific binary code. Fields such as crash severity, meteorological condition, roadbed condition, light condition, and type of traffic are included using an ordinal classification.

Twenty-four variables are used to represent the vector of an accident occurring in Turin. In Table I each byte represents one variable. An index is then created by grouping the crash severity (no. 4), crash type (no. 5), road characteristics (no. 16) variables enumerating them and labeling the index variable

Table I. Fields used to describe crashes in Turin, Milan, and Legnano road networks.

	Field	Number of bytes	Turin	Milan	Legnano
1	Time of crash	1	X	X	X
2	Lighting condition	1	X	–	–
3	Meteorological conditions	1	X	X	X
4a	Crash severity 1	1	X		–
4b	Crash severity 2	2	–	X	–
5a	Crash type 1	2	X	X	–
5b	Crash type 2	3	–	–	X
6	Number of drivers	1	X	–	–
7	Number of passengers	1	X	–	–
8	Number of two wheel vehicles	1	X	–	X
9	Number of other vehicles	1	X	–	–
10	Number of people involved	1	–	X	–
11	Number of vehicles involved	1	–	X	X
12	Number of heavy vehicles	1	X	–	–
13	Number of uninjured people	1	X	–	–
14	Number of injured people	1	X	X	X
15	Number of deaths	1	X	X	X
16a	Road characteristics 1	4	X	X	–
16b	Road characteristics 2	3	–	–	X
17a	Road type 1	3	X	X	X
17b	Road type 2	2	–	–	X
18a	Road bed 1	1	X	X	–
18b	Road bed 2	2	–	–	X
19	Type of traffic	1	X	X	–
20	Road code (road name)	1	X	X	X

DOI: 10.1002/atr

with a progressive number: 54 possible combinations are obtained. For example the following case for Turin data:

Severity = fatal accident = code 30
Road type = signalized intersection = code 0100
Crash type = frontal lateral = code 10

has Index = 52. For simplicity we built the indices (for Turin and Milan) so that an increasing value means a higher severity. This information is not used for learning in order to avoid using them for both learning and for deriving results. The numbers present in Figures 3–5 refer to these numbers. After these operations a total of 17 variables are used in the learning data set.

At the end of the validation process, a rectangular 36×7 lattice was constructed. The average quantization error is 0.1759. The quantization error represents the average distance between the data vectors and the CVs. Because all 17 variables are normalized to 1, the average percentage error is approximately 1% for each variable.

For Milan, the data were also furnished by the local police and include crashes which occurred in the city in 1995 (21 822 records). The data structure is shown in Table I.

Twenty different variables are used to represent the vector of an accident. An index is used to label data and is made up by three variables: crash severity (no. 4), crash type (no. 5), and road characteristics (no. 16), allowing for 103 combinations. After these operations, there are 12 variables used in the learning data set.

At the end of the validation process, a rectangular lattice with 40×60 neurons with a hexagonal connection type was selected. Its average quantization error is 0.0953 and the weighted QE is 0.64346. Because all 12 variables are normalized to 1, the average percentage error is less than 1% for each variable.

Legnano is a medium size Italian city not far from Milan with approximately 60 000 inhabitants. The accident data were collected by the police and include all crashes except for property damage only crashes. Data were collected over a 2-year period, from 1999 to 2000, and amount to 1339 records. Twenty different variables were used to represent the vector of an accident (see Table I). The labels used to investigate maps are based on the road name, the type of accident, time weather and roadbed type. In this application all variables except road name are used for the learning phase.

At the end of the validation process, a lattice with 30×40 neurons with a hexagonal connection type was selected. Its average quantization error is 0.2168 and the weighted QE is1.027. With 19 variables normalized to 1, the average percentage error is approximately 1.1%. Some variables such as accident type and number of vehicles were not always filled out for all crashes. This may partially explain the poor results in terms of QE. Nevertheless the number of CVs used is quite high.

4. RESULTS

4.1. Turin

In Figure 2, the crash data mapped onto each CV is drawn. The gray color of each cell is related to a classification of the ''index'' (the combination of crash severity, road characteristics, crash type) used to label the map.

In Figure 3, a map showing the distance between vectors is drawn. Labels are coded by numbers and refer to the above-mentioned field ''index.'' In the map, six main clusters can be observed. By identifying the labels present in each cluster, the main relationships between cluster and crash type can be extracted. The following crash clusters can be readily recognized:

Cluster 1: non-signalized intersections,
Cluster 2: signalized intersections,
Cluster 3: crashes at a non-signalized intersection,
Cluster 4: bumps and other crash between two vehicles,
Cluster 5: frontal–lateral at non-signalized intersection, and
Cluster 6: frontal–lateral crashes involving pedestrians.

Cluster 6 contains the highest number of crashes and exhibits two peaks (more than 100 crashes mapped into a single cell) for frontal–lateral crashes occurring at intersections.

DOI: 10.1002/atr

Crashes occurring on straight roads do not form recognizable clusters and are scattered throughout the map. Fatal crashes are grouped into area 7 which cannot be considered a cluster because the distances between CVs are great. Serious injury crashes are present in area 7 or in another part of the map, and are always distant (in the sense of vector distance) from nearby cells. In the left high corner of Figure 3, detailed view of cluster 1 is shown. The number in each cell corresponds to a combination of the index (severity, crash type, and road type).

In other maps, different features of the data can be investigated. By using the name of the roads, those with similar crash types and patterns (in terms of distance between vectors) could be isolated and identified.

In addition to distances and labels, the map in Figure 3 provides other information about the data transformation. The number of crashes mapped onto each CV gives a measure of occurrence of those crashes and can be interpreted as the probability of an accident occurring. The more data in the database the more realistic is the interpretation.

By using the map labeled by road name, the homogeneity (or non-homogeneity) of crashes for each road can be studied. In Table II, a list of the top 20 roads (on the basis of the number of crashes) is presented. The table reports for each road the number of crashes, the number of CVs used to map their crashes, and the crashes actually mapped by those vectors.

If the crashes are homogenous, that is, the same road is affected by a few causes or types of crashes, many crashes can be mapped by a few CVs (as in Vittorio Emanuele II). On the other hand, when there are many causes or types of crashes, data are scattered on the map without a specific relationship (as in Lecce).

A difference between the columns ''crashes'' and ''mapped crashes'' may exist because some crashes can be mapped by other CVs with different labels. If there are no vectors for a certain road, it means that its crashes may be scattered throughout the map and thus mapped separately by other CVs. The ratio between the mapped crashes and the total number of crashes gives another criterion with which to evaluate the degree of homogeneity of crashes on a single road. The higher the value of the ratio, the greater the homogeneity. In Table II the ratios of the number of mapped crashes to the number of CVs (C/B) and the ratio of the number of mapped crashes to total crashes (C/A) are derived. This helps to identify the following ''best'' roads: avenue Peschiera (35, 0.55), avenue Vittorio Emanuele II (25, 0.50), avenue Regina Margherita (14, 0.18), and avenue Unione Sovietica (14, 0.14).

Table II. List of the top 20 roads ranked by the number of crashes (Turin data).

Road name	Crashes (A)	Codebook vectors (B)	Mapped crashes (C)	C/A	C/B
Regina Margherita	954	13	176	0.18	14
Vittorio Emanuele II	859	17	433	0.50	25
Unione Sovietica	676	7	95	0.14	14
Giulio Cesare	662	4	35	0.05	9
Nizza	504	8	77	0.15	10
Orbassano	480	7	39	0.08	6
Francia	467	5	54	0.12	11
Vercelli	435	2	14	0.03	7
Moncalieri	383	3	14	0.04	5
Trapani	323	5	10	0.03	2
Casale	315	4	36	0.11	9
Grosseto	306	2	2	0.01	1
Novara	275	5	29	0.11	6
Ferraris Galileo	267	3	33	0.12	11
D'azeglio Massimo	263	1	2	0.01	2
Siracusa	256	2	12	0.05	6
Giordano Bruno	253	2	7	0.03	4
Peschiera	251	4	138	0.55	35
Lecce	249	0	0	0.00	–
Cossa Pietro	243	1	3	0.01	3

DOI: 10.1002/atr

4.2. Milan

In Figure 4 the map shows the distances between vectors and data clusters. Labels are coded by numbers and refer to the field ''index.'' Ten main clusters can be extracted. Type of crash can also be extracted. Clusters 1, 2, and 5 contain the highest number of crashes for frontal–lateral in straight roads and evening hours.

Another method to show distance between CVs is based on the use of a Sammon map. This method is useful when extracting clusters. The Sammon map of the data mapped in Figure 4 helps to identify four large clusters. They can be linked, respectively, to the clusters in Figure 4:

1. Clusters 3, 4, and 6
2. Clusters 1, 2, and 5
3. Clusters 7, 8, and 9
4. Cluster 10.

The first group represents crashes with low severity both on straight roads and at intersections in the afternoon and evening. These crash clusters are characterized mainly by frontal–lateral crashes. Cluster 4 represents pedestrian crashes and cluster 6 captures rear-end crashes. The second grouping represents crashes with low severity on straight roads and evening hours (in this area there are also clusters with crashes occurred in the morning but with less significance); cluster 1 is characterized by many rollover crashes and clusters 2 and 5 by frontal–lateral crashes. The third grouping includes crashes at intersections in the afternoon and evening and mainly frontal–lateral. The fourth includes crashes on straight roads and at intersections frontal–lateral and involving pedestrians. Fatal crashes are grouped into areas marked by little circles (called ''x'') at the top and bottom edges of the map.

4.3. Legnano

In Figure 5, a map showing distance between vectors is drawn. Labels are coded by numbers and refer to the field ''road name.'' In the map, 12 clusters can be observed.

By inspecting the labels for each cluster, according to time, vehicles involved, roadbed and meteorological conditions, accident type, type of intersection, and number of injured people relationships between clusters and accident characteristics can be identified:

Cluster 1: very low severity, daytime, heavy traffic, and motorbikes with dry roadbed, calm weather, on straight roads;
Cluster 2: high severity, nighttime, vehicles and motorbikes, frontal–lateral crashes, with dry roadbed, calm weather, at intersections;
Cluster 3: low severity, daytime, heavy and passenger vehicles, frontal–lateral or isolated vehicle against obstacle with dry roadbed, calm weather, at intersections;
Cluster 4: low severity, daytime, motorbikes, with dry roadbed, calm weather, at intersections;
Cluster 5: low severity, daytime and nighttime, motorbikes, with dry roadbed, calm weather, at intersections;
Cluster 6: low severity, daytime and nighttime, heavy vehicles and motorbikes, with dry roadbed, calm weather, at intersections;
Cluster 7: low severity, daytime and nighttime, vehicle and motorbikes, with dry and wet roadbed, calm weather, at intersections;
Cluster 8: very low severity, daytime, vehicle and motorbikes, with dry and wet roadbed, on straight roads;
Cluster 9: high severity, daytime and nighttime, vehicles and motorbikes, lateral and pedestrian rolling over crashes, with dry roadbed, calm weather, on straight roads;
Cluster 10: high severity, daytime, vehicles, front-lateral crashes, with dry roadbed, calm weather, on straight roads;
Cluster 11: high severity, daytime and nighttime, vehicles, bumps, with wet roadbed and rain on straight roads;

DOI: 10.1002/atr

Cluster 12: very low severity, daytime and evening hours, vehicles, with wet roadbed and rain, on straight roads;
Cluster 13: no cluster but in this area there is an aggregation of crashes; low severity, daytime and nighttime, motorbikes with wet roadbed and rain and turning over in curve.

The list of the first 20 roads (in the same manner as Table II) is analyzed. The difference between the ''crashes'' and ''mapped crashes'' is more evident than what was shown in Table II. We found that roads are quite scattered among codebooks and only 36% of crashes has been mapped by codebooks labeled by them. Some roads have a good ratio C/A or C/B but generally the ratio C/B is not very high. This may lead to difficulty in analyzing the CV data but only for what concerns extracting information by using labels instead from CVs. In Legnano, we can deduce that the highest number of crashes is located in a few roads but there are many different causes. Generally this situation cannot be overcome by changing the size of the map. Increasing the map size may worsen the ratio, while reducing the map size, the data can be grouped together simply into other CVs. Therefore the size of the map should be based only on weighed quantization error. What may be useful to improve performance is to consider more variables, for example, vehicular flow, in order to explain more variance.

5. CONCLUSIONS

In recent years, ANN algorithms have been used to address a wide range of transportation problems. Notably, they have been used for incident detection [19], pavement management systems [20], traffic control systems [21], and traffic assignment algorithms [22]. The use of artificial intelligence for safety applications, however, has been more limited and recent [23–25].

In this paper, a new method of accident analysis based on VQ used with a particular type of ANN known as an SOM is presented. The approach facilitates the analysis of large, complicated databases with many different types of variables having complicated interrelationships. The procedure is most robust when there are many records and many different variables. While the process is computationally intensive, the availability of analytical, mapping, and display software greatly improves our ability to not only investigate the background human, vehicle, roadway, and environmental factors associated with motor vehicle crashes, but also provides a useful tool for evaluating potential interventions for reducing crashes and their severity.

As presented in this paper, there are three different uses for this set of algorithms and analytical tools. First, the method can be quite useful in cleaning data sets, in terms of data reduction, and in terms of helping to ''group'' or ''cluster'' cases or variables in preparation for more advanced modeling tasks. Second, the VQ and the class of NN models used in this paper are helpful in terms of the identification and recognition of patterns. These patterns may be expressed in terms of time, space, or perhaps in terms of clusters of related attributes. Pattern recognition is, no doubt, one of the more popular of the applications developed in fields other than accident analysis. Finally, the techniques used in this paper provide us with a powerful new way of simulating or modeling complex realities. While in this paper, the focus was on three Italian cities and the problem of motor vehicle safety, there are, no doubt, many other real world problems which could be modeled using this approach. The challenge, of course, is not only to accurately describe the world as we observe it, but also to simulate alternative scenarios or futures associated with changes in policies, laws, or programs that may have an impact on the particular outcome or phenomenon at hand.

In addition to general contributions to the field of ANN modeling, SOMs, and VQ techniques, this paper offers a number of specific contributions to the analysis of traffic safety questions. In addition to improving our knowledge of clustering techniques, the method, through the provision of a vector process, actually provides a way of introducing much more detail into the analytical process than other techniques. As pointed out, the two-dimensional maps produced through the SOM procedure preserve the N-dimensional structure of the original data. We can, therefore, use this technique to approximate the pdf of crashes. This provides another approach to estimate the interrelationship between various human, vehicle, roadway, and environmental factors associated with crashes. By including spatial and temporal variables, the procedure provides an alternative approach for conducting ''black spot'' or ''black zone'' analyses, developing works on this subject, i.e., Ref. [26]. Future research should focus

DOI: 10.1002/atr

on increasing the connectivity between these methods and other spatial analytical tools (e.g., GIS, mapping software, spatial statistics); other traffic safety applications are possible, such as:

- finding relationships involving various vehicle types;
- finding relationships that demonstrate the role of driver characteristics;
- finding relationships pertaining to weather conditions.

The case studies in this paper serve to demonstrate the power and versatility of the technique. At the same time, we learned that there will need to be additional refinements and adjustments in the technique itself. In Milan, for example, problems may have arisen because of the particular data structure that was chosen to represent crashes. Some of the variables had a very large average value. It might have been better to normalize the data, removing some of the extreme values, and preserving a more constant variance. While the number of cases does not seem to impose a limit on the method itself, it may lead to problems due to the particular nature of the problem being investigated. While 3 years of data from Turin might be considered optimal for some types of analyses, for others, there may not have been enough cases. Fatal or high severity crashes, for instance, were always located at the edges of the map. While in close proximity to each other, they were distant in terms of the CVs. This makes sense in that they are, comparatively speaking, more rare events than other types of crashes. This shows us, moreover, that these types of crashes are really different from others and that, furthermore, we might need a different data structure or set of cases to more fully investigate this type of crash.

At the same time, it is important to recognize that for some researchers, this type of approach which is so ''data-driven'' and reliant upon ''machine learning,'' may be a bit intimidating. Some could argue that pattern recognition software and algorithms such as the VQ technique should be limited to data cleaning operations or only for data reduction exercises – that without explicit formulations of relationships between dependent and independent variables and without the classical hypothesis testing approach, these tools are rather limited. However, another perspective suggests that particularly as the range of different variables and relationships increases to include not just human factors, but also vehicular, roadway, environmental, temporal, and spatial factors, additional tools capable of handling multidimensional categorical as well as continuous data are sorely needed. While some might want to relegate these advances to only the signal, image, and voice processing applications for which they were originally developed, the careful application and adaptation of these methods to address other more complex problems should not be ignored.

In time, the methods and techniques and applications described in this paper will become increasingly user friendly. Eventually, the ideas associated with ''machine learning'' and ANN will become more commonplace, not just within the field of accident analysis, but also in other areas of mathematical and statistical analysis. While the ideas and some of the images presented here may appear to be somewhat rudimentary and elementary in nature, we can expect that the models and approaches will become increasingly sophisticated over time, particularly as the availability and accessibility of large, complicated databases, and powerful new software programs increase.

6. THE LIST OF SYMBOLS AND ABBREVIATIONS

ANN	artificial neural network
CV	codebook vector
GIS	geographic information system
NN	neural network
pdf	probability density function
QE	quantization error
SOM	self organizing map
VQ	vector quantization

DOI: 10.1002/atr

ACKNOWLEDGEMENTS

We express our gratitude to the Urban Police of Turin, Milan, and Legnano who have allowed us to use data contained in this study. We thank Avis Morigawara for assistance with typing and formatting an earlier version of the manuscript.

REFERENCES

1. Tay R, Rifaat SM. Factors contributing to the severity of intersection crashes. *Journal of Advanced Transportation* 2007; **41**(3):245–265.
2. Rifaat SM, Chin HC. Accident severity analysis using Ordered Probit Model. *Journal of Advanced Transportation* 2007; **41**(1):91–114.
3. Anderberg MR. *Cluster Analysis for Applications*, Academic Press: New York, 1973.
4. Everitt B, Landau S, Leese M. *Cluster Analysis*, Arnold Press: London, 2001.
5. Kim K, Yamashita EY. Using a K-means clustering algorithm to examine patterns of pedestrian involved crashes in Honolulu, Hawaii. *Journal of Advanced Transportation* 2007; **41**(1):69–89.
6. Mussone L, Ferrari A, Oneta M. An analysis of urban collisions using an artificial intelligence model. *Accident Analysis and Prevention* 1999; **31**(1999):705–718.
7. Kim K, Nitz L, Richardson J, Li L. Personal and behavioral predictors of automobile and crash injury severity. *Accident Analysis and Prevention* 1995; **27**(4):469–481.
8. Kim K, Pant P, Yamashita E. Hit and run crashes: using rough set analysis and logistic regression to capture critical attributes and determinants. *Transportation Research Record* 2008; **2038**:114–121.
9. Hyvärinen A, Oja E. Independent component analysis: algorithms and application. *Neural Networks* 2000; **13**(4–5):411–430.
10. Gersho A. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory* 1982; **IT-25**:373–380.
11. Gersho A, Gray RM. *Vector Quantization and Signal Compression*, Kluwer Academic Publishers: Boston, 1992.
12. Kohonen T. *Self-organizing Maps*, Springer-Verlag: Berlin, 1995.
13. Kohonen T. Self-organizing maps of massive document collections. *Proceedings of IEEE-INNS-ENNS* Vol. 2, IJCNN2000, Como 24–27 July 2000; 3–9.
14. Vesanto J, Alhoniemi E. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks* 2000; **11**(3):586–600.
15. Zador P. Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Transactions on Information Theory* 1982; **IT-28**:139–149.
16. Blum JR. Multidimensional stochastic approximation methods. *The Annals of Mathematical Statistics* 1954; **25**: 737–744.
17. McClelland JL, Rumelhart DE. Distributed model of human learning and memory. In *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, Vol. **2**, Psychological and Biological Models. Mit Press: Cambridge, MA, USA, 1986; 170–215.
18. König A. Interactive visualization and analysis of hierarchical neural projections for data mining. *IEEE Transactions on Neural Networks* 2000; **11**(3):615–624.
19. Ivan J, Schofer J, Koppelman F, Massone L. Real time data fusion for arterial street incident detection. *Transportation Research Record* 1995; **1497**:27–35.
20. Taha M, Hanna A. Evolutionary neural network model for the selection of pavement management strategy. *Transportation Research Record* 1995; **1497**:70–76.
21. Hua J, Faghri A. Development of neural signal control system – toward intelligent traffic signal control. *Transportation Research Record* 1995; **1497**:53–61.
22. Mussone L, Matteucci M. An application of ant colony systems for DUE and SUE assignment in congested transportation networks. *Proceedings of the 11th World Conference on Transportation Research*, Berkeley, CA, USA, 25–28 June 2007.
23. Xie Y, Lord D, Zhang Y. Predicting motor vehicle collisions using Bayesian neural network models: an empirical analysis. *Accident Analysis and Prevention* 2007; **39**(2007):922–933.
24. Wei C-H, Lee Y. Sequential forecast of incident duration using artificial neural network models. *Accident Analysis and Prevention* 2007; **39**(2007):944–954.
25. Riviere C, Lauret P, Ramsamy JFM, Page Y. A Bayesian neural network approach to estimating the energy equivalent speed. *Accident Analysis and Prevention* 2006; **38**(2006):248–259.
26. Loo BPY. The identification of hazardous road locations: a comparison of the blacksite and hot zone methodologies in Hong Kong. *International Journal of Sustainable Transportation* 2009; **3**(3):187–202.

DOI: 10.1002/atr