Available online at www.sciencedirect.com
SCIENCE DIRECT
Neural Networks 19 (2006) 186–195
Neural Networks
www.elsevier.com/locate/neunet

2006 Special issue

Machine learning in soil classification

B. Bhattacharya *, D.P. Solomatine

Hydroinformatics and Knowledge Management Department, UNESCO-IHE Institute for Water Education, P.O. Box 3015, 2601 DA Delft, The Netherlands

Abstract

In a number of engineering problems, e.g. in geotechnics, petroleum engineering, etc. intervals of measured series data (signals) are to be attributed a class maintaining the constraint of contiguity and standard classification methods could be inadequate. Classification in this case needs involvement of an expert who observes the magnitude and trends of the signals in addition to any a priori information that might be available. In this paper, an approach for automating this classification procedure is presented. Firstly, a segmentation algorithm is developed and applied to segment the measured signals. Secondly, the salient features of these segments are extracted using boundary energy method. Based on the measured data and extracted features to assign classes to the segments classifiers are built; they employ Decision Trees, ANN and Support Vector Machines. The methodology was tested in classifying sub-surface soil using measured data from Cone Penetration Testing and satisfactory results were obtained.

Keywords: Soil classification, cone penetration testing, machine learning, ANN, decision trees, SVM

1. Introduction

In a number of engineering problems, there is a necessity to classify contiguous intervals (segments) of series data (signals). Series data has an additional index variable (distance or time) associated with each data value. Standard classification algorithms in these situations are often inadequate due to the additional contiguity constraint. Examples from the following domains can be mentioned: classification of sub-soil layers using Cone Penetration Testing (Bhattacharya & Solomatine, 2003; Coerts, 1996; Huijzer, 1992), well-log analysis in petroleum engineering (Kerzner, 1986), palaeoecology (Gordon, 1996), etc. In these cases, measurements are taken from a vertical bore or with a test apparatus, which is pushed down the earth and it is required that the stratigraphical information is preserved in the classification. The problem is solved in two phases: firstly, a segmentation algorithm is used to cluster contiguous blocks of instances and secondly, these segments are classified by domain experts.

We investigated the problem with a specific interest of automating classification of soil layers from measured data. In civil engineering, it is a prerequisite to know the soil classes up to some depths prior to any construction. The direct method to identify the soil classes by drilling boreholes and testing soil samples is very expensive. A cost-effective alternative is the so-called Cone Penetration Testing (CPT), which is one of the most popular soil investigation methods (Coerts, 1996). In CPT, a metallic cone is pushed into the soil and an indication of the *in-situ* soil strength is obtained by measuring the force needed to let it advance at a constant rate. A CPT recording is a quasi-continuous picture of the subsurface at the test location. It contains the vertical variations of the mechanical characteristics of the subsoil. These variations in turn indicate variations in geological layers and their properties. During a test, two primary signals are recorded: (1) the cone tip resistance stress (q_c), (2) the frictional stress (f_s), which are used to derive the more widely used friction ratio $R_f = f_s \times 100/q_c$. Additionally, information is available from borehole drilling in the proximity of CPTs typically with the frequency of 1 borehole for 10 CPTs. Observing the variations of q_c and R_f (Fig. 1) and using the nearby borehole information, an expert firstly segments the logs, i.e. finds boundaries of layers (class boundaries), and secondly, using the domain knowledge assigns a soil class C_i to each segment (where $i = 1, 2, \ldots, I$ and $I =$ number of classes).

In practice, a manual segmentation and classification procedure is followed. This procedure requires expertise, and is expensive, time consuming, subjective and not completely reproducible. The challenge is to automate this procedure. In order to achieve this a new algorithm called CONCC (CONstraint Clustering and Classification) was developed (Bhattacharya & Solomatine, 2003, 2005). It can be used in automatic classification with the constraint of contiguity and includes the following two steps:

* Corresponding author. Tel.: +31 15 215 1749.
E-mail addresses: b.bhattacharya@unesco-ihe.org (B. Bhattacharya), d.solomatine@unesco-ihe.org (D.P. Solomatine).

0893-6080/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2006.01.005

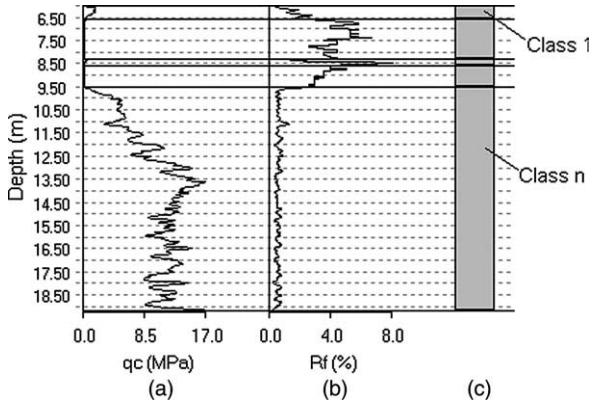

Fig. 1. Variation of cone tip resistance (q_c) (a) and friction ratio (R_f) (b) along depth of a Cone Penetration Testing, which is used to segment the logs and assign classes to the segments (c).

- *Segmentation*: to find J segments of data from a single series data (e.g. CPT);
- *Classification*: to build a classifier to assign classes to these segments; it is built using measured data and extracted features from segments of a number of series data from a region and trained with classes labelled by experts.

This paper presents segmentation and classification of series data using CONCC. Using the CONCC algorithm, segments are determined and then the found segments are classified using three Machine Learning (ML) methods: Decision Tress (DT), ANNs and Support Vector Machines (SVM). Application of ML in geostatistical problems is quite limited, some applications are reported by Bhattacharya and Solomatine (2003, 2005), Juang, Huang, Holtz, & Chen (1996), Kumar, Konno, & Yasuda (2000) and Zhang & Tumay (1999).

2. Segmentation of series data

Segmentation can be defined as the clustering of series data where the constraint of contiguity has to be maintained. If the measured instances are labelled 1, 2,…, N according to depth, and J segments are sought, then $J-1$ 'markers' are needed in some of the $N-1$ gaps between pairs of neighbouring measurements to produce J segments of contiguous block of data. The number of possible partitions in this case is considerably smaller than that of the unconstraint clustering case. Accordingly, most of the available algorithms employ the exhaustive or semi-exhaustive search within this reduced search space.

A segment g_j to be identified is defined as

$$g_j = \{\{x_{1,1},\ldots,x_{1,K}\},\ldots,\{x_{l,1},\ldots,x_{l,K}\},\ldots\{x_{nj,1},\ldots,x_{nj,K}\}\}_j \quad (1)$$

where

$x_{l,k} \in \Re^n$ and represents the measured data;
$l = 1, 2, \ldots, n_j$, where n_j is the number of instances in segment g_j;
$k = 1, 2, \ldots, K$, where K is the number of dimensions (signals);
$j = 1, 2, \ldots, J$, where J is the number of segments.

After segmentation, the classification problem is solved when segments are attributed to classes:

$$g_j \rightarrow C_i \quad (2)$$

Review of methods for segmentation (constraint clustering) is given by Bhattacharya and Solomatine (2003), Wagstaff (2002) and Gordon (1996). The reported methods partition a dataset into J groups by minimising the criteria of within-group sum of squared deviations from the segment mean, i.e. dispersion. The reported methods follow these approaches: (i) a dynamic programming (computationally demanding) method where the solution is obtained recursively (Hawkins & Merriam, 1973); (ii) a split moving window of fixed width that is moved along the data sequence and a marker is placed at points where a substantial change in some statistical criteria is observed (Webster, 1978).

Segmentation problem has more particular features that make it different from clustering problem. The segmentation algorithm follows to some extent the logic of a domain expert that performs manual segmentation. This poses additional requirements to the segmentation algorithm, which are considered below:

1. Firstly, there could be data points (measurements) that obviously belong to class C_1 but are within the segment corresponding to class C_2 and should be attributed to class C_2. Such points (vectors) could be the result of noise or instrumentation errors but in case of CPT logs, they appear due to the small inclusions of another soil class in the otherwise predominantly homogeneous soil layer and are observed mostly in the proximity of another soil layer. Such points will be called *aliens*. The segmentation algorithm should be able to process them.
2. In practice, classification problem can be formulated differently, depending on the user preference to have small or large number of classes. Less the number of classes, more inclusive (broader) they are. Segmentation algorithm should be able to identify the number of segments from a user-defined parameter, expressing the need in a particular application for small or large number of classes.

2.1. Characterizing segments by representative intervals

Segment g_j is composed of the measured signals and may contain aliens. It may happen that not all the elements of g_j are necessarily attributable to a single class C_i. It is proposed to build a smaller subset $h_j \subset g_j$ in such a way that it would have less aliens. Such subset can be called a representative subset of the segment g_j. An empirical method is proposed to construct h_j: to include in it only those points that are within the constraints $[L_{j,k}, U_{j,k}] : h_j = \{\forall x_{l,k} : L_{j,k} \le x_{l,k} \le U_{j,k}\}$. To present it formally, a mapping $\xi(\theta_k)$ is introduced:

$$[L_{k,j}, U_{k,j}] \leftarrow \xi(\theta_k)(g_j) \quad (3)$$

where $L_{k,j}$ is the lower bound in dimension k for the segment j, $U_{k,j}$ is the upper bound in dimension k for the segment j, $k = 1, 2, \ldots K$ and K is the total number of signals. For simplicity we assume that ξ has only one user-defined parameter θ_k for each dimension.

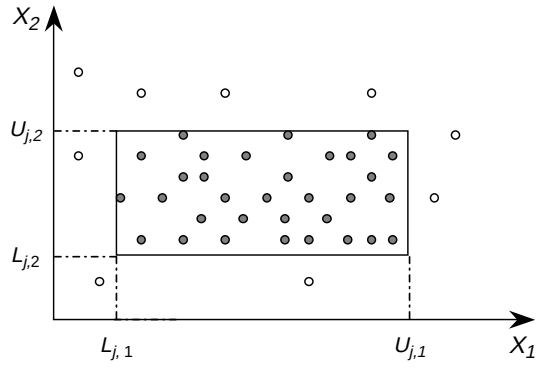

Fig. 2. Representative intervals for dimension $K=2$. Data points of hj are shaded.

An illustrative example of the representative set of segment g_j obtained with the mapping $\xi(\theta_k)$ is shown in Fig. 2 for dimension $K=2$. $L_{j,k}$ and $U_{j,k}$ may be found by truncating θ_k percent of lower and upper mass from a frequency distribution of the signal k. A less computationally demanding alternative could be by expressing the measured data within a segment as a fuzzy number with a membership computed from a chosen membership function. The intervals can be computed using an alpha-cut with a value of alpha as θ. As an example, Fig. 3(a) shows measurements of the three contiguous segments of a CPT ($K=2$) where overlapping of the segments can be seen. There is no overlap of the elements of the representative sets (h_j) of these segments (Fig. 3(b)).

We experimented with different α-cuts and mass percentages and the conclusion was that this empirical way of filtering out aliens works well. However, one could propose non-rectangular constraints for h_j. More research is needed into this issue.

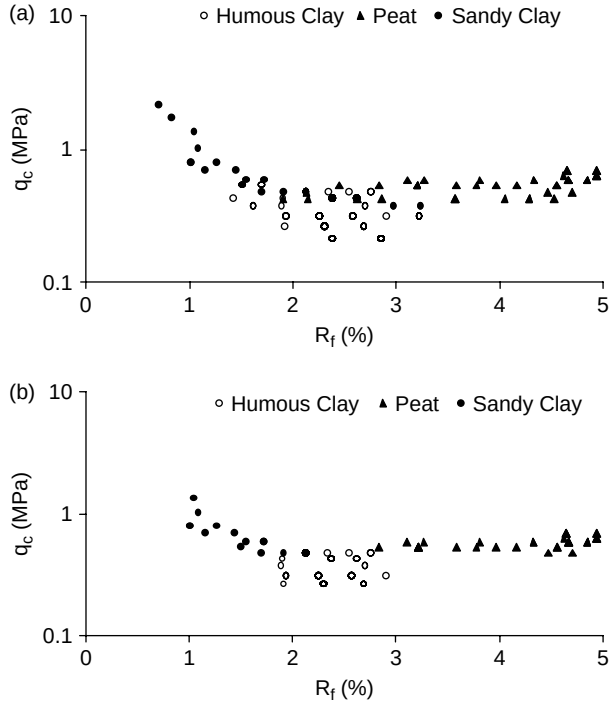

Fig. 3. (a) Data points belonging to the three contiguous segments; (b) Data points belonging to the representative sets of these three segments.

2.2. Distance measure

As a measure of within-group similarity, typically, the dispersion is used; it is calculated as the sum of squared distances of data points to a cluster centre. Due to the specifics of the considered problem, we are not interested in the variation within h_j representing cluster g_j (and the class corresponding to it). If we consider all data points belonging to a class as full members of that class, then dispersion should be set to zero. There could be different ways of defining such 'insensitive' distances; we adopted the following method: distance with respect to dimension k of a data point $x_{l,k}$ to a cluster is computed to the boundaries limiting h_j, that is to the hyperplanes $x_k=L_{k,j}$ and $x_k=U_{k,j}$ instead of measuring it from the centre:

$$d_{l,k} = \max((L_{j,k}-x_{l,k}),(x_{l,k}-U_{j,k}),0) \tag{4}$$

where, $d_{l,k}$ is the distance of a data point $x_{l,k}$ in dimension k, $l=1,2,\ldots,n_j$ and n_j is the number of data points in the segment g_j.

Thus, data points from h_j do not contribute to the dispersion and other data points from this segment do. In Fig. 2, all the data points belong to the same segment, but the shaded circles do not contribute to computing dispersion of the segment, while the points shown as empty circles do contribute.

2.3. Distance between segments

Distance $Dist_{jp,k}$ with respect to dimension k between segments g_j and g_p could be calculated in different ways; we use simply the distance between the boundaries limiting h_j:

$$\mathrm{Dist}_{jp,k} = \max((L_{j,k}-U_{p,k}),(L_{p,k}-U_{j,k})) \tag{5}$$

For the segmentation algorithm that follows it is also important for each x_k to introduce the notion of the distance threshold $D_{jp,k}$ that would determine if two clusters g_j and g_p are close or not in dimension k

$$D_{jp,k} = m_k\left[Abs\left(\frac{L_{j,k}+U_{j,k}}{2}\right)-\left(\frac{L_{p,k}+U_{p,k}}{2}\right)\right] \tag{6}$$

where m_k is a constant with respect to dimension k and $0 \le m_k \le 1$. The distances between segments are schematically shown in Fig. 4.

2.4. The algorithm

The algorithm uses within-group dispersion w_j defined in Eqn.(7) as a measure of homogeneity of the segment g_j and the total dispersion W defined in Eqn.(8) as a global measure of the effectiveness of a partition.

$$w_j = \sum_{k=1}^{K}\sum_{l=1}^{n_j} d_{l,k}^2 \tag{7}$$

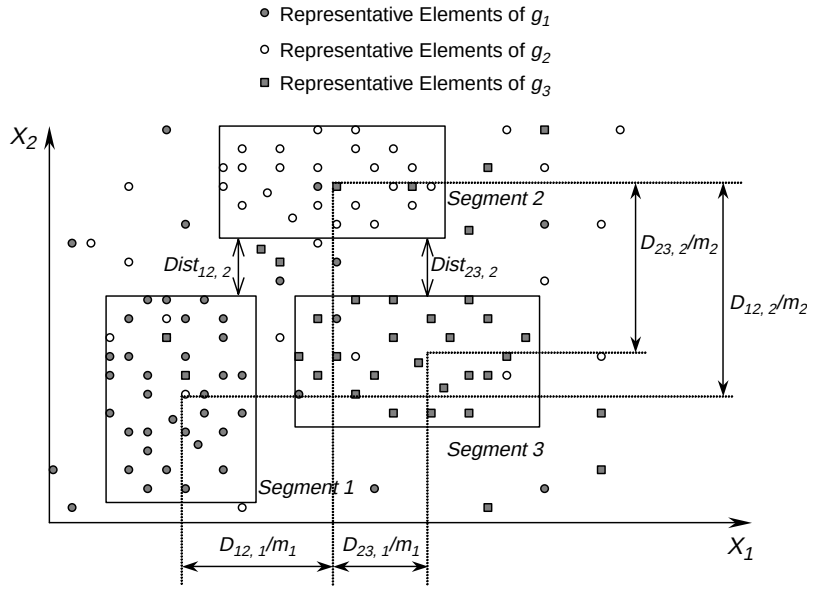

Fig. 4. Schematic diagram showing distances between three consecutive segments.

$$W = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{l=1}^{n_j} d_{l,k}^2 \tag{8}$$

where $l=1, 2, \ldots, n_j$, n_j is the number of data points in segment g_j; $k=1, 2, \ldots, K$; $j=1, 2, \ldots, J$, and $d_{l,k}$=distance of a data point to the segment it belongs (defined in (4)).

Characteristic of the existing divisive segmentation algorithms is that once a segment boundary is found, usually, it cannot be updated later. Due to the high computing cost involved, it is not possible to consider updating all the segment boundaries when a new segment is found. The following alternative is proposed: when a segment g_j has to be partitioned, it is joined to its preceding segment g_{j-1} and the following segment g_{j+1} to form a new segment g_j, which is then partitioned into four segments; this operation increases the number of segments by one. If g_j happens to be the first or the last segment then it is joined respectively with segment g_{j+1} or g_{j-1} segment and then partitioned into three segments.

The following is the outline of the segmentation algorithm:

1. Divide the dataset into $J=3$ segments for which the total dispersion W is minimum and for which the following constraints to segmentation between any two neighbouring segments g_j and g_p are valid:

$$\text{Dist}_{jp,k} \geq D_{jp,k} \tag{9}$$

$$b_j \geq a \tag{10}$$

$$b_p \geq a \tag{11}$$

where b_j and b_p are the number of data points in segments g_j and g_p, respectively, a is a constant defining the desired minimum number of data points in a segment, $k=1,\ldots,K$.

2. Choose the segment g_j for which w_j is the highest. Join the preceding segment g_{j-1} and the following segment g_{j+1} to g_j to form a new segment g_j. If g_j is the first segment then merge only g_{j+1} with it. If g_j is the last one then merge only g_{j-1} with it.
3. If g_j is the first or the last segment then break this new segment g_j into three segments; otherwise break g_j into four segments. Make a search of possible partitions using a user-defined parameter step such that for step=1 an exhaustive search is made and for step=z (an integer) every zth data point within g_j is considered for a partition. Find the partition for which W is minimum and for which Eqs. (9–11) are valid between any two neighbouring segments g_j and g_p. The number of segment J increases by one. If no new segment can be formed satisfying Eqs. (9–11) then go to step 5.
4. Repeat steps 2–3 until no new segment can be formed.
5. Stop.

Combination of inequalities Eqs. (9–11) are used as a stopping criterion for the segmentation algorithm, which otherwise needs the number of segments J to be fed as an input. In some problems, satisfying Eqn.(9) with respect to one dimension can be sufficient for a change of a class. Otherwise, it needs to be satisfied for all the dimensions.

The CONCC algorithm addresses the shortcomings of the existing segmentation algorithms. It uses elements of fuzzy logic to address the imprecision in the measured data and uses a particular threshold-based distance measure between instances and segment centres. Initial tests of the CONCC algorithm were performed and a satisfactory performance was achieved (Bhattacharya & Solomatine, 2003).

3. Feature extraction using boundary energy

After the segments are found, the next task is to assign classes to the segments. However, for different locations, the mapping (2) could be different due to the spatial variability. When data from several test locations are combined, an overlap of instances

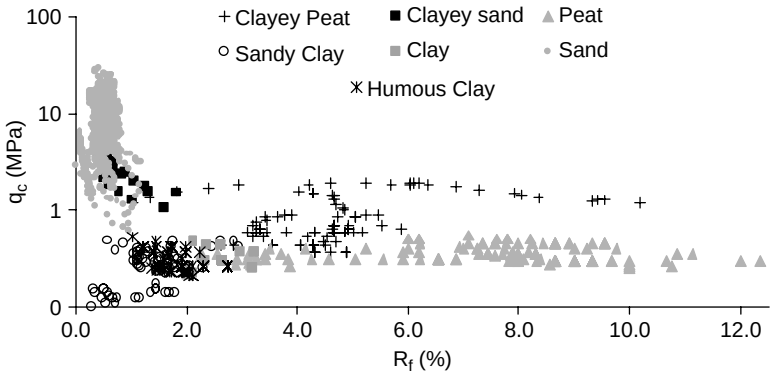

Fig. 5. Scatters of the measured data from several Cone Penetration Tests in the cone tip resistance (q_c) — friction ratio (R_f) space. Overlapping of different classes is discernible.

of different classes is usually observed (Fig. 5). The spatial variability has several reasons such as site-specific conditions, location and depth, measuring instruments, etc. To solve the problem, it needs to be brought to a higher dimension by bringing in additional features so that a partition of the input space is possible by a classifier for assigning a class to that subset of inputs.

In engineering, experts use subjective criteria, so the automated classification methods should select appropriate features and be compatible with experts. During this research, several experts in geology were interviewed and their manual classification procedures were recorded. It was observed that the experts assign high importance to the shape, in addition to the magnitude of the data. This led us to conclude that in automating the classification procedure, experts' perception about the shape of signals needed to be parameterised.

Shapes of signals can be represented using multi-scale transforms such as Fourier transform, *w*-representation using Marr wavelet and Morlet wavelet, Gabor transform, etc. (Costa & Cesar, 2001). These representations can be used to derive shape measures using multi-scale energy methods such as boundary energy, multi-scale wavelet energy, etc. In this research the boundary energy has been used in parameterising the shape effects.

3.1. Boundary energy

Boundary energy is defined as the amount of energy required to modify the shape of a contour to its lowest energy level (a circle), with the same perimeter as the original object. The concept of boundary energy originated from the theory of elasticity and was first applied in biological shape characterisation (Young & Calvert, 1974). Since then, it has been widely used as a global shape measure for classification of a variety of shapes. Boundary energy is defined as follows

$$B_{a,k} = \frac{1}{N}\sum_{n=0}^{N-1} c(a,n)^2 \tag{12}$$

where $B_{a,k}$ denotes the boundary energy of a signal at scale a along the dimension k, c is the curvature at point n, $n=1,2,\ldots,N$; and N is the number of discrete observations. A detailed description on boundary energy can be found in Costa and Cesar (2001).

The multi-scale dimension of boundary energy is brought by successive low-pass filtering of a series data and by computing boundary energy of each of these filtered series data. Gaussian filter is the most common one and the value of 'sigma' in the Gaussian expression is changed gradually from low values to very high values. As a result, the curvature of a series data is computed at different sigma values, i.e. at different analysing scales leading to a multi-scale representation of the series data. Such a multi-scale representation of curvature of a series data is called a *curvegram* (Costa & Cesar, 2001). Successive low-pass filtering of a series data with varying sigma values lead to a multi-scale characterisation of the energy contained in the series data. With increasing scales, the small scale details of a series data vanishes and the most important features become prominent, which can be utilised in subsequent classification. Boundary energy has been successfully applied in biomedical engineering, neuroscience, and in general, in analysing images from a diverse domain (see, e.g. Vliet & Verbeeck, 1993) and has been recognised as an effective tool that can be used as a global shape measure.

It is evident that computation of boundary energy depends upon an accurate estimation of multi-scale curvature of a series data, which for a discrete signal is not an easy task. Commonplace computation techniques of curvature, based on finite difference methods can lead to high errors, effectively thwarting the possibility of using boundary energy as a shape measure. In this regard, a Fourier-based curvature computation technique, introduced in Young and Calvert (1974), which is much more accurate than the traditional curvature computation methods, has been used.

Fig. 5 shows instances from several segments taken from a number of CPTs ($K=2$) where segments were labelled by the experts. From the fact that instances overlap it can be concluded that partitioning of the data to build a classifier in this space ($q_c(R_f)$) may not be possible. Fig. 6 shows R_f and the corresponding boundary energy for these segments, whereas Fig. 7 shows q_c and the corresponding boundary energy for these

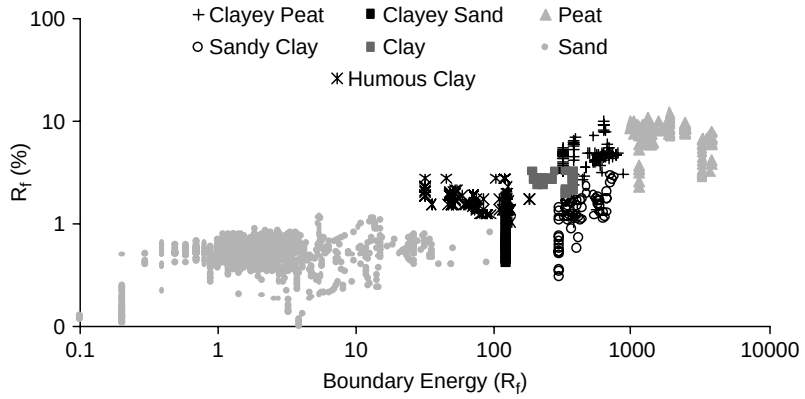

Fig. 6. Variations of friction ratio (R_f) with the boundary energy of R_f using data from several Cone Penetration Tests. Several clusters of instances corresponding to classes are now more or less disjoint.

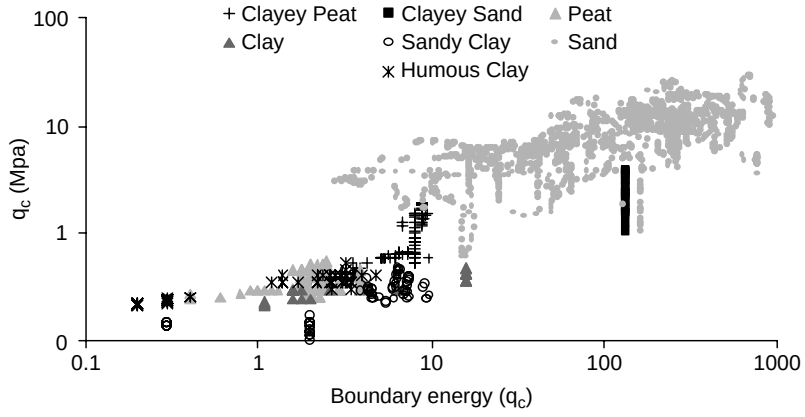

Fig. 7. Variations of cone tip resistance (q_c) with the boundary energy of q_c using data from several Cone Penetration Tests. Several clusters of instances corresponding to classes are now more or less disjoint.

segments. Clusters corresponding to classes are now more or less disjoint and present a much easier problem for a classifier.

4. Classification

The overall classification scheme is shown in Fig. 8 with the particular reference to the classification of soil based on CPT data. It can be easily amenable to classification problems of other domains (e.g. petroleum engineering) as well. The training data may be created by segmenting several series data by using the segmentation algorithm presented above or by involving an expert. The series data used in the training is then labelled by experts. Features from the labelled data (during training) or unlabelled data (during operation) are extracted using boundary energy (in the FE unit). Data preparation is carried out in the constructed feature space (in the DP unit). The classifier (CA) consists of two units. The pre-classification unit (PC) classifies each instance during the testing and operational use. The compaction unit determines a single class (called a 'compact class') for a segment.

Classifiers are trained using DT, ANN and SVM to learn the following mapping from the labelled training data:

$$\{x_{l,1}, B_{l,1}, x_{l,2}, B_{l,2}, \ldots, x_{l,K}, B_{l,K}\} \rightarrow C_i \quad (13)$$

where

l = index of instances, $l = 1, 2, \ldots, n_j$ and n_j is the number of instances in g_j;
K = number of signals (dimension);
$B_{l,k}$ = boundary energy at point l in dimension k for input variable $x_{l,k}$.

The classifier (Eq. (4)) learns to classify each instance. Finally the mapping (2) is ensured by compacting the classes of instances within a segment. This is required due to the presence of aliens (noise). The heterogeneity of the natural environment (e.g. soil) is the prime reason behind the presence of these mischievous measurements. For measurements in natural environments, such as CPTs or well-logs of petroleum engineering, the aliens are observed mostly in the proximity of another segment (i.e. another class). However, aliens can also be observed due to various other reasons such as instrument errors, etc. Experts ignore these aliens and pick up the general pattern of the segment. The compaction algorithm assigns weights to the classes determined by the PC unit. The weights (w_l) are computed as points on a Gaussian curve

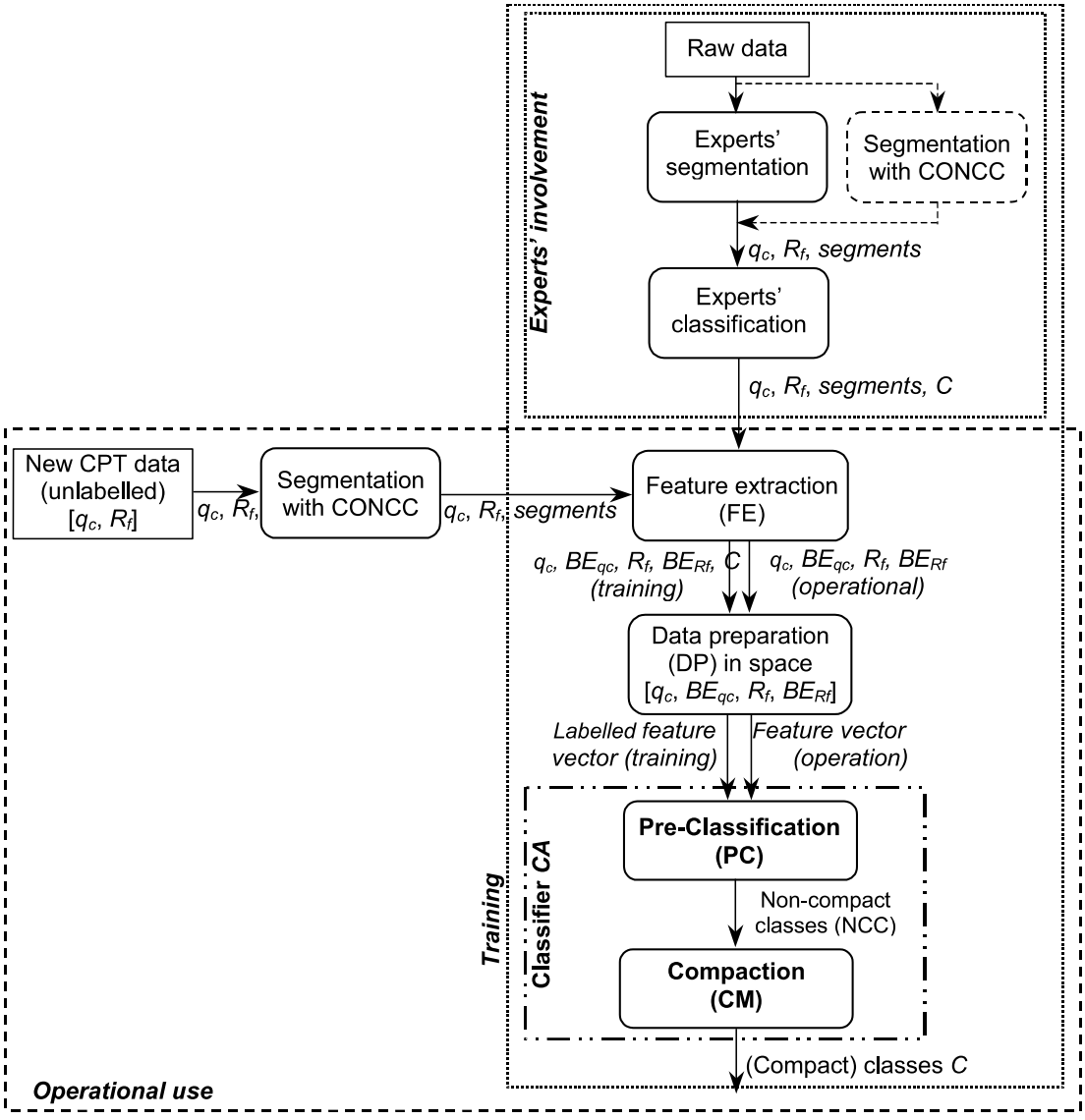

Fig. 8. The classification scheme during the training and operational phase. During training experts are involved in preparing the training data. Once the classifier CA is trained it replaces experts.

with zero mean and standard deviation = 2 (Fig. 9) and the compact class ($\bar{C}_{\mathrm{p}}$) is determined by:

$$\bar{C}_{\mathrm{p}} = w_l C_{\mathrm{p}} / \sum_{l=1}^{n_j} w_l \tag{14}$$

where n_j = number of instances in a segment and C_p is the class for the l^{th} instance.

5. Application to geotechnics

The above methodology was applied to classify soil on the basis of CPT data. The data that has been used to build the classifier is taken from the CPTs conducted at Nesselande, a residential zone under development near Rotterdam, The Netherlands. During 1996 to 2000 CPTs were conducted at 565 locations in an area of 2.5 × 3.5 km of Nesselande. At about 60 locations, boreholes were drilled as borehole information is

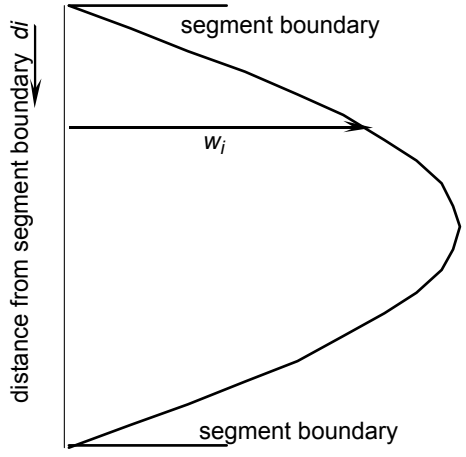

Fig. 9. The variation of weight w_l of an instance in relation to its position within a segment. The compaction algorithm of the classifier multiplies the class of each instance by w_l in order to determine a single class of a segment.

vital in determining the soil characteristics from CPTs. The Nesselande area is underlain by extensive peat and soft clay deposits. The Late Pleistocene and Holocene (sand) deposits predominantly make up the upper 10 m of soil in the Nesselande area. A detailed geological description of the area can be found in Weerts (1996). The study undertaken by the municipality of Rotterdam aims at finding the thickness of the soft sediments overlying (Pleistocene) sands, the presence and geometry of the sand bodies in the soft Holocene deposits, the hydrological contact of these sand-bodies with the (Pleistocene) sand, existence of peat within a few meters below the ground surface, engineering properties of the soil layers present in the subsurface, and their spatial distribution. Municipality of Rotterdam has the task of determining the soil classes of the subsurface of Nesselende using CPTs and borehole information.

5.1. Segmentation

The effectiveness and efficiency of the CONCC algorithm was investigated with several case-studies in classifying soil from CPT data. Fig. 10 shows the segmentation found by using the CONCC algorithm. Fig. 10(a) and (b) show the variation of the signals q_c and R_f for a particular CPT (number AAA27 of Nesselende, Municipality of Rotterdam, The Netherlands). Fig. 10(c) shows the experts' segmentation of the CPT using the borehole information in addition to the measured data. Five segments were found by the experts. Fig. 10(d) shows the segments found using the CONCC algorithm with $m_1=0.7$, $m_2=0.4$ (Eq. (6)), $a=5$ (Eq. (10) and (11)) and $step=1$. $\xi(\theta_k)$ was chosen as a function truncating 20% of lower and upper mass of the frequency distribution of the signal. The seven segments found by the CONCC algorithm (Fig. 10(d)) almost match the segments found by the experts. In the place of the third segment found by the expert (8.40–8.50 m) the algorithm found two thin segments, which, given the variation in R_f, can be justified. Instead of the long fifth segment the algorithm found two segments, which, given the variation in R_f, can also be justified. It can be readily concluded that the CONCC algorithm clearly finds segments comparable to the segments found by the experts. Note that Fig. 10(e) shows the segmentation using an existing algorithm (Huijzer, 1992) with the number of segments set to seven. These segments do not have resemblance with the experts' segmentation.

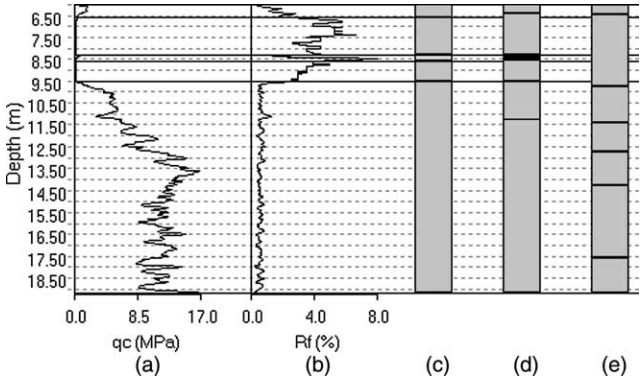

Fig. 10. (a) and (b) Measurements, (c) expert's segmentation of a CPT log (d) Segmentation using CONCC algorithm (e) segmentation using an existing algorithm of Huijzer (1992).

The values for the user-defined parameters in the segmentation algorithm used in the case-study were found by trial and error. Empirically, it was found that for a sandy and peaty soil type $m_1=0.6$ to 0.7, $m_2=0.3$ to 0.4 is a good estimate. For a peaty and clayey soil type, $m_1=0.05$–0.1, $m_2=0.05$–0.1 gave good results. The sensitivity of the results with respect to the thresholds still needs to be investigated.

5.2. Classification

Obtaining a proper dataset having more or less equal representation of all the classes as well as all the geographical regions of the site was a big problem. This is because the manual classification procedure of experts is time consuming and expensive. Therefore, building a classifier using a minimum amount of data was considered. In consultation with the experts in total, 7 CPTs of the area were chosen. Unfortunately, equal representation of all the classes could not be maintained. Four CPTs for training and three CPTs for testing were considered. Due to the scarcity of data, no cross-validation dataset was chosen. Each CPT of the testing group is statistically comparable to a CPT of the training group. The number of instances in training and testing were 5150 and 2830 respectively whereas the number of segments in training and testing were 72 and 46, respectively.

The requirement of details to be found in soil classes of an area varies as per the requirement of the end users. Often it may be enough to know the primary soil classes (sand, peat and clay for this area). Some other times, the presence of the secondary soil classes also need to be determined. The determination of just the sandy or non-sandy soil types poses an important geotechnical task. This may be due to the reason that the end user wants to model the spatial extent of the sand layers, or for the purpose of determining the initial settlements from the proposed constructions, or to study the drainage characteristics to assess the migration of contaminants.

Based on the above discussions, the following three classification problems were contemplated:

(a) binary classification, where the task of the classifier is to determine whether the soil is sandy or not;
(b) three-class classification, where the classifier has to identify the primary soil class only (i.e. sand, clay or peat);
(c) seven-class classification, where the classifier has to determine the appropriate class from the set of seven classes observed in this area.

Classifiers using DT, ANN and SVM were built for the above-mentioned three classification tasks. The experiments were conducted with WEKA (Witten & Frank, 2000) for DT, Neurosolutions (2005) and NeuralMachine (2005) for MLP ANN, and WEKA and RHUL (2005) for SVM. The number of nodes in the hidden layers of the ANN was optimised by exhaustive search. SVM was also optimised by running the classifier with different parameters' sets; however, this

Table 1
Classification accuracy of the binary classifiers (on the test dataset)

Soil class	% of correctly classified instances			% of correctly classified segments		
	DT	ANN	SVM	DT	ANN	SVM
Sandy soil	99.3	100.0	96.6	100	100	100
Non-sandy soil	96.3	96.4	99.5	100	100	100
Total	97.6	98.0	97.8	100	100	100

optimisation was not exhaustive. The misclassification costs were considered equal for all the classes.

5.3. Results and discussions

5.3.1. Binary classification

The results obtained with all the three methods were close to each other (Table 1). The instances with the sandy soil type were better classified by DT and ANN, whereas the instances of the non-sandy soil type were better classified by SVM (with WEKA). It was noticed that almost all erroneously classified instances were located near the segment boundaries. Measurements near a segment boundary are often noisy and it can be concluded accordingly that the performance of the classifiers was excellent.

Finally, the segment classes were determined using the compaction algorithm. It was observed that all the segments were correctly classified by the three methods, i.e. a classification accuracy of 100% was reached.

5.3.2. Three-class classification

The three-class classification problem is comparatively more difficult than the binary classification problem mainly due to the large overlap of the instances of the clayey soil with that of the other two classes. The performance of the classifiers were comparable; however, the SVM-based classifier (with WEKA) gave slightly better results (Table 2), with ANN and DT following closely. All the methods provided more accurate classification of the instances from the sandy soil. For the clayey soil, the SVM-based classifier was better than the others. All the methods provided poor results for the peaty soil. For each classifier, if the correct class was not predicted then the predicted, class was the geologically neighbouring class.

5.3.3. Seven-class classification

The results of the classifiers for the seven-class classification problem are shown in Table 3. The classifiers were very accurate for the silty sand and clayey sand. They were moderately accurate for the sandy clay, clay and humous clay, but were poor for the clayey peat. The ANN model was much better than the others in classifying the instances of the humous clay and peat. The sandy segments were correctly classified in all occasions, the clayey segments were classified with the accuracy of 3 out of 4 segments. The DT gave the best results for the segments of the clayey peat, still the accuracy was not high. The SVM-based classifiers gave poor results; these results are, however, preliminary since optimisation of the built SVMs (of regularisation constants and the kernels) was not exhaustive and further experiments should be undertaken. For each classifier, if the correct class was not predicted, then the predicted class was the geologically neighbouring class.

Table 2
Classification accuracy of the three-class classifiers (on the test dataset)

Soil class	% of correctly classified instances			% of correctly classified segments		
	DT	ANN	SVM	DT	ANN	SVM
Sand	100.0	99.8	98.5	100	100	100
Clay	82.9	96.5	98.5	90.9	90.9	100
Peat	58.7	60.5	58.7	66.7	66.7	66.7
Total	85.6	91.0	90.7	82.6	82.6	87.0

Table 3
Classification accuracy of the seven-class classifiers (on the test dataset)

Soil class	% of correctly classified instances			% of correctly classified segments		
	DT	ANN	SVM	DT	ANN	SVM
Silty sand	99.2	98.7	85.1	100	100	100
Clayey sand	83.3	70.8	54.2	100	100	100
Sandy clay	85.4	85.4	69.4	75	75	50
Clay	75.0	77.5	65.0	100	100	100
Humus clay	81.4	93.5	77.0	100	100	80
Clayey peat	65.1	64.6	49.7	57.1	42.9	28.6
Peat	74.1	84.0	65.4	100	100	50
Total	86.5	89.4	74.4	82.6	78.3	60.9

6. Conclusions

In this paper, an algorithm (CONCC) for segmentation and classification of series data where the constraint of contiguity has to be maintained is presented. Experiments were conducted to classify soil types based on cone penetration testing data. The main conclusions are:

- The presented segmentation algorithm provides segment boundaries coinciding with class boundaries. It takes care of the noise in data and finds segments similar to the ones found by experts.
- Due to the spatial variability of the measured parameters classification based on the measured parameters was not possible. Additional features were extracted by parameterising experts' perception about the shape of a series data using boundary energy. This novel approach proved to be effective.
- The proposed classification scheme effectively mimics experts' classification procedure and automates the classification task.
- In the case-study of soil classification using data from cone penetration testing, the predictive accuracy of the classifiers on the test set even for the most complex problem was found to be high (83%). When the correct class was not predicted, then the predicted class was a geologically neighbouring one. For many practical situations, such accuracy of prediction

was found to be sufficient by most experts and, if this error is allowed, the accuracy was 100%.

Possible improvements are seen in building a non-greedy optimal segmentation algorithm and in testing other methods of classification.

Acknowledgements

Part of this work was performed in the framework of the projects 'Predicting the structure of the subsurface: semi-automatic interpretation of cone penetration testing' and 'Data mining, knowledge discovery and data-driven modelling' of the Delft Cluster research programme supported by the Dutch government.

References

Bhattacharya, B., & Solomatine, D. P. (2003). An algorithm for clustering and classification of series data with constraint of contiguity. In A. Abraham, M. Köppen, & K. Franke (Eds.), *Design and application of hybrid intelligent systems* (pp. 489–498). Amsterdam: IOS Press.

Bhattacharya, B., & Solomatine, D. P. (2005). Machine learning in soil classification. *Proceedings of international joint conference on neural network*, Montreal, Canada, (pp. 2694–2699).

Coerts, A. (1996). *Analysis of static cone penetration test data for subsurface modelling—a methodology*. Amsterdam: Koninklijk Nederlands Aardrijkskundig Genootschap.

Costa, L. F., & Cesar, R. M. (2001). *Shape analysis and classification: Theory and practice*. Boca Raton, FL: CRC Press.

Gordon, A. D. (1996). A survey of constrained classification. *Computational Statistics and Data Analysis*, *21*, 17–29.

Hawkins, D. M., & Merriam, D. F. (1973). Optimal zonation of digitized sequential data. *Mathematical Geology*, *5*, 389–395.

Huijzer, G. P. (1992). *Quantitative penetrostratigraphic classification*. PhD thesis. Free University of Amsterdam, Amsterdam, The Netherlands.

Juang, C. H., Huang, X. H., Holtz, R. D., & Chen, J. W. (1996). Determining relative density of sands from CPT using fuzzy sets. *Journal of Geotechnical Engineering*, *122*(1), 1–6.

Kerzner, M. G. (1986). *Image processing in well log analysis*. Dordrecht, The Netherlands: Reidel Publishing Company.

Kumar, J. K., Konno, M., & Yasuda, N. (2000). Subsurface soil-geology interpolation using fuzzy neural network. *Journal of Geotechnical and Geoenvironmental Engineering (ASCE)*, *126*(7), 632–639.

NeuralMachine. (2005). http://www.data-machine.com/, 28.8.2005

NeuroSolutions. (2005). http://www.nd.com/, 28.8.2005

RHUL. (2005). Computer Learning Research Centre, Royal Holloway University of London, http://www.clrc.rhul.ac.uk/, 26/1/2005

Vliet, van L. J., & Verbeeck, P. W. (1993). Curvature and bending energy in digitised 2D and 3D images. In K.A. Hogda, B. Braathen, & K. Heia, (Eds.), *Proceedings of eighth Scandinavian conference on image analysis*, Norway, *vol. 2* (pp. 1403–1410).

Wagstaff, K. (2002). *Intelligent clustering with instance-level constraints*. PhD thesis. Cornell University, USA.

Webster, R. (1978). Optimally partitioning soil transects. *Journal of Soil Science*, *29*, 388–402.

Weerts, H. J. T. (1996). *Complex confining layers*. The Netherlands: Utrecht University.

Witten, I. H., & Frank, E. (2000). *Data mining: Practical machine learning tools and techniques with java implementations*. San Francisco, CA: Morgan Kaufmann.

Young, I. T., & Calvert, T. W. (1974). An analysis technique for biological shape. *Information and Control*, *25*, 357–370.

Zhang, Z., & Tumay, M. T. (1999). Statistical to fuzzy approach toward CPT soil classification. *Journal of Geotechnical and Geoenvironmental Engineering*, *125*(3), 179–186.