Optimal partitioning of data-logged soil profile

Rocky Wang & Roger Skirrow
Technical Standards Branch, Alberta Transportation, Edmonton, AB, Canada

2011 Pan-Am CGS Geotechnical Conference

ABSTRACT
Optimal partitioning of data-logged soil profile can be implemented using a spreadsheet and VBA programming. The basic assumption is that geo-stratification of soils is closely related to the sedimentary cycle or depositional environment. This stratification is reflected in various soil property test and indexes, and can be determined through geophysical testing, or the CPT test. Within the same geological layer, the dispersion of various/multi-parameter readings is smaller than that between neighbouring layers. Optimal stratification is equivalent to minimizing the summation of the dispersion of 'internal' indexes for each individual layer, and maximizing the summation of the dispersion of 'external' indexes of each individual layer relative to all layers. The calculation includes multiple iterations. The methodology may also be used for geotechnical zoning for site characterization. Examples are presented for illustrative purposes.

RÉSUMÉ
Partitionnement optimale du profil de sol de données enregistrées peut être implémenté à l'aide d'une feuille de calcul et de la programmation VBA. Le postulat de base est que geo-stratification des sols est étroitement liée au cycle sédimentaire ou milieu sédimentaire. Cette stratification se reflète dans divers de test de propriété du sol et d'index et peut être déterminée au moyen de tests géophysiques, ou à l'épreuve du CPT. Dans la même couche géologique, la dispersion des divers/multi-parameter lectures est plus petite que celle entre les couches voisines. La stratification optimale est équivalente à minimiser la sommation de la dispersion des index « internes » pour chaque couche individuelle et en maximisant la sommation de la dispersion des index « externes » de chaque couche individuelle par rapport à tous les calques. Le calcul comprend plusieurs itérations. La méthodologie peut également être utilisée pour géotechniques zonage pour la caractérisation des sites. Des exemples sont présentés à titre indicatif.

1 INTRODUCTION

For design and contracting purposes, site-specific geotechnical investigations are required for civil engineering projects. Geophysical and in-situ probes are commonly used techniques for geotechnical studies. These 'indirect' methods of determining geo-stratification offer good cost-benefit, accuracy and speed. Since soil samples are not obtained indirect methods of data interpretation must be utilized in evaluating various soil types of the strata encountered (Mayne, 2009).

An interested example was reported by Tumay (2000), which developed a continuous intrusion miniature cone penetration test (CIMCPT) system for use in rapid and economical shallow-depth characterization of sites, especially at rapidly locating thin slip surface by a 4 mm-depth increments sounding. For such case, delicate data interpretation is essential. Optimal partitioning provides a tool to aid in the interpretation of fine scale geo-stratification.

The fundamentals introduced in this paper are not new in mathematics and the geosciences but may not be familiar to geotechnical engineers. Readers can find an algorithm for calculating optimal partitions of one-dimensional data sets as described by Fisher (1958), and some other implementations of the algorithm, e.g., FORTRAN by McRae (1971), Turbo Pascal by Lindberg (1990), and Hawkins & Merriam (1973) for segmenting well-log data.

2 FUNDAMENTALS

In Figure 1, tip resistances, q_t changes with depth. This kind of soil behaviour may be a reflection of the soil genesis and geological forces, operational uncertainties of data acquisition system, and a multitude of random unknown. In terms of analytics, the phenomenon can be decomposed into three components:

$$x_{ij} = \mu + a_i + \varepsilon_{ij}$$
(i =1, 2, 3; j = 1, 2,…, n_i) [1]

Where μ is the *grand mean*; a_i is the contribution by behavior of layer i itself; and ε_{ij} is random error of jth observation in i layer, which generally follows $N(0, \sigma^2)$ for most of known soil types. In fact, almost all in situ soil soundings aim at finding { a_i }, which coincides with *SSY* = *SSE* + *SSR* in the sense of the analysis of variance, where, *SSY* is the total sum of squares of the deviations of the reading around the *grand mean* corresponding to the sum of (x_{ij} -μ) in Equation [1]; *SSE* is the error sum of squares of the deviations of the reading around the three separate group means { a_i } corresponding to the sum of the $\varepsilon_{ij,}$ also called squared distance; *SSR* is the sum of squares of the deviations of the group means from the *grand mean* corresponding to the sum of the a_i.

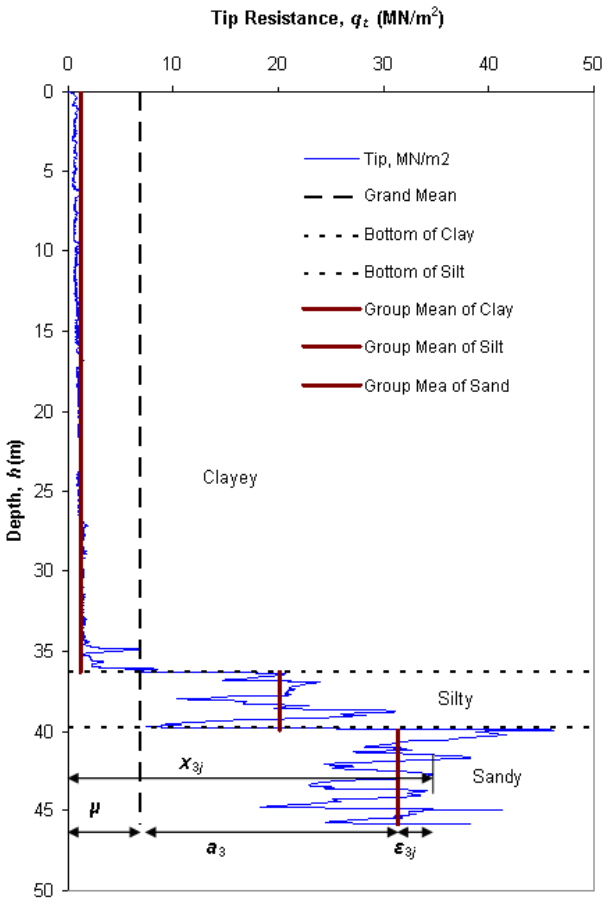

Figure 1. Partial CPT curve (data source: http://www.coe.lsu.edu/cpt/) and analytical form

For the three soil layers profiled in Figure 1, a simple ANOVA – single factor analysis can find the total sum of squares *SSY* = 260718, the sum of squares of between-groups *SSR* = 242087, and the error sum of squares *SSE* = 18631. *F*-Test shows the least squares partition is true. However, the conventional ANOVA only can help to determine if all the three means { a_i } in the profile are equal. The task here can be described with reference to statistics as: given a set (or some sets) *n* of numerical readings, and a positive integer *K* (<*n*), to find a set of cut-points for grouping the *n* into *K* mutually exclusive and exhaustive subsets such that the *SSE* is minimized. Since *SST* is a constant for a given system, to minimize the *SSE* is equivalent to maximizing the *SSR*.

In practical terms, during soil explorations, one or several sets of sequential readings at certain vertical depth intervals are recorded. If several sets are recorded a multivariate geo-stratification problem is encountered. Which set of readings better reflects the stratification? What is the relevant system of weightings to be applied? In this regard, geotechnical engineer of the project has no equal. As a result, a complex matrix of initial measurements or datasheet would be required for soil profiling:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \qquad [2]$$

In which, *n* is sample size, *m* is criteria. The following steps present how to reach an optimization of least squares partition.

2.1 Data Normalization

Data normilzation is a form of data pre-processing. This step 'cleans' the data. This step aims at scaling the attribute data to fit in a specific range. There are different types of data normalization available. In this paper, the technique of Min Max Normalization is used. The method transforms a data reading to a dimensionless number which fits in the range [z_{min}, z_{max}], herein for [0,1] and obtain a matrix $\boldsymbol{Z}_{nxm}$= [z_{ij}]:

$$z_{ij} = \left[\frac{x_{ij} - \min\limits_{1\le i\le n}\{x_{ij}\}}{\max\limits_{1\le i\le n}\{x_{ij}\} - \min\limits_{1\le i\le n}\{x_{ij}\}} \right] \cdot \left(z_{\max} - z_{\min}\right) + z_{\min}$$

(i = 1, 2, …, n; j = 1, 2, …, m) [3]

In another usage in statistics, normalization refers to the division of multiple sets of data by a common variable in order to negate that variable's effect on the data, thus allowing underlying characteristics of the data sets to be compared. Normalization allows data on different scales to be compared, by bringing them to a common scale. In terms of levels of measurement, these ratios only make sense for ratio measurements where ratios of measurements are meaningful but, not for interval measurements where only distances are meaningful.

2.2 Error Sum of Squares (*SSE*) Matrix Determination

Any portion of the soil profile with multi-index/parameter readings can be considered as a single soil layer numerically. Numerical approaches of geo-stratification are in essence based on data manipulation regarding 'one dimensional' or 'single parameter' observation. Therefore any multi-folded observations within any range of depth should be converted or reduced into a column vector of observations or single measures on each element reading with depth as its dimension, say, $\boldsymbol{X}_{nxm} \rightarrow \boldsymbol{X}_{nx1}$, or $\boldsymbol{Z}_{nxm} \rightarrow \boldsymbol{Z}_{nx1}$. Such conversion is indirectly carried forward in the determination of the *SSE* matrix $\boldsymbol{D}$ (= [d_{ij}]). Also, the first principle component, or factor analysis method, can be used to address any multivariate stratification problem.

The *SSE* of a length [*i*, *j*] of segment of multi-index readings is calculated as d_{ij},

$$d_{ij} = \sum_{\alpha=i}^{j}\sum_{\beta=1}^{m}\left[z_{\alpha\beta} - \overline{z_{\alpha}(i,j)}\right]^2$$

$(1 \le I \le j \le n)$ [4]

Where $\overline{z_{\alpha}(i,j)} = \sum_{\alpha=i}^{j}\left[z_{\alpha\beta}/(j-i+1)\right]$ is the average of $z_{\alpha\beta}$ given β (index) for [i, j] segment, which is the separate group mean in terms of statistics.

Equation [4] represents the deviation of the raw data around its average value for the separate segment. In terms of the analysis of variance, d_{ij} is called the sum of squares within groups, or simply the squared distance.

Since $d_{ij} = 0$ for $i = j$ and, $d_{ij} = d_{ji}$ for $i \neq j$, the ordered set of n 'readings' (one-parameter) can be partitioned into at most $C_{n-1}^{1} + C_{n-1}^{2} + ... + C_{n-1}^{n-1} = 2^{n-1} - 1$ contiguous segments, thus it is only necessary to calculate $n(n\text{-}1)/2$ of d_{ij}, say:

$$D = \begin{bmatrix} d_{12} & d_{13} & \dots & d_{1n} \\ & d_{23} & \dots & d_{2n} \\ & & \dots & \dots \\ & & & d_{n-1,n} \end{bmatrix} \quad [5]$$

2.3 Set Target Function

Assuming the grouping problem being by way of k layers out of n readings, say, $\{i_1 = 1, i_1+1, \dots, i_2-1\}$, $\{i_2 = 1, i_2+1, \dots, i_3-1\}$, …, $\{i_k = 1, i_k+1, \dots, n\}$, the target function will be,

$$\tilde{e}[p(n,k)] = \sum_{j=1}^{k} D(i_j, i_{j+1}-1) \quad [6]$$

For a given n and k, $i_1 = 1 < i_2 < \dots < i_k < n$, $i_{k+1} - 1 = n$. The problem reduces to a search for the contiguous partitions determined by k-1 cut-points that minimised the target function the 'tilde' e, for which noted as $e[p(n,k)]$.

2.4 Optimization Calculation

The optimization algorithm involves multiple iterations. To find an optimal partition of 2 contiguous layers, the equation [6] has,

$$e[p(n,2)] = \min_{2 \le j \le n}\left[D(i, j-1) + D(j,n)\right] \quad [7]$$

By manipulating the cut-point j, subject to the constraint $2 \le j \le n$, the target function is minimized, and the j thus obtained is the optimal partitioning point.

For k optimal partitions, firstly assuming

$$e[p(n,k)] = \min_{k \le j \le n}\left\{e[p(j-1,k-1)] + D(j,n)\right\} \quad [8]$$

This is equivalent to stating that the ordered set of n readings is initially partitioned into 2 contiguous layers. Then, the {1, 2,…, j -1} partition, will be partitioned into k - 1 layers, while the {j, j+1, …, n} itself constitutes a single layer. Note that $k \le j \le n$.

By changing j in equation [8], the kth layer with j_k cut-point can be sorted out by optimization iterations, say, {j_k, j_{k+1}, …, n}. Then equation [9] is used to find the next layer, { j_{k-1}, j_{k-1}+1, …, j_k -1} .

$$e[p(j_k - 1, k-1)] = \min_{k-1 \le j \le j_k - 1}\begin{Bmatrix} e[p(j_{k-1} - 1, k-2)] \\ + D(j_{k-1}, j_k - 1) \end{Bmatrix} \quad [9]$$

These operations are repeated until all optimal partitions are determined.

Such optimal grouping problem can also be simply described as: given n readings and g partitions ($g < n$), to group the n into g mutually exclusive and exhaustive subsets, say, {1, 2,…, p}, {p+1, …, q}, {…}, …, {v+1, n}, such that the sum of squares within the individual groups, $SSE = D(1,p) + D(p+1,q) + \dots + D(v+1,n)$ is minimized.

The appendix presents the VBA code which has been thoroughly tested. As a benchmark, interested readers can exercise a short problem: (3, 2, 3.05, 1, 4.05, 5), the answers are shown in Table 1. Note that cut-points themselves should belong to the immediately upper layers.

Table 1. Benchmark example by hand calculation.

Partitions	Cut-points	*SSE*
2	4	0.205
3	3, 4	0.072
4	3, 4, 5	0.043
5	1, 2, 3, 4	0.029

3 EXAMPLE

Benchmarking data can be found at http://www.coe.lsu.edu/cpt/La3059T1_English.txt. Table 2 shows the results of different systems of segmenting expressed by cut-points starting from the ground surface. Figure 2 presents the *SSE* versus soil layers partitioned. The figure shows that four layers are sufficient to optimally partition the soil strata. Figure 3 shows the four soil layers profiled.

Table 2. Benchmark example by spreadsheet.

Layers	Cut-points								
2	1537								
3	1688	1537							
4	1721	1691	1537						
5	1948	1721	1691	1537					
6	1948	1723	1688	1669	1537				
7	1948	1723	1688	1665	1639	1537			
8	1948	1781	1721	1688	1665	1639	1537		
9	1948	1906	1881	1723	1688	1665	1639	1537	
10	1948	1906	1881	1721	1691	1688	1665	1639	1537

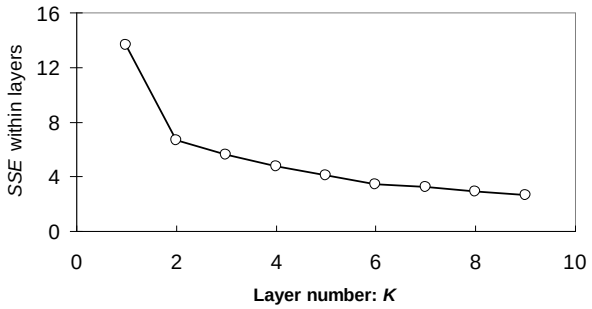

Figure 2. *SSE* versus segments

4 SUMMARY

The approach described in this article has roots in statistic theory. The approach aims to help detect changes in subsurface strata numerically, and as such is best applied to in-situ data acquisition systems such as the CPT.

VBA was used to implement the optimal partition method described in this paper. The built-in optimization tool *Solver* that resides in Microsoft Excel could be used to reduce the programming effort. An example of the use of Solver for an optimal search application is presented by Low and Tang (1997).

To apply the optimal partitioning approach efficiently and appropriately, users need to exercise judgement such as is required for any data interpretation method, e.g., are the partitions reasonable, do they provide partitions that are meaningful for geotechnical design. It is also important and basic to implement an approach such as this based on high quality and reliable raw data, e.g., any outlier and abnormal data resulting from malfunction of the data acquisition system should be dismissed at the data pre-processing stage.

The optimal partition method discussed herein is relative to only normally distributed one-dimensional data for each individual group. For other modes of data, see Fitzgibbon et al (2000). A 2D-type least squares partition is possible but may not be computationally practical.

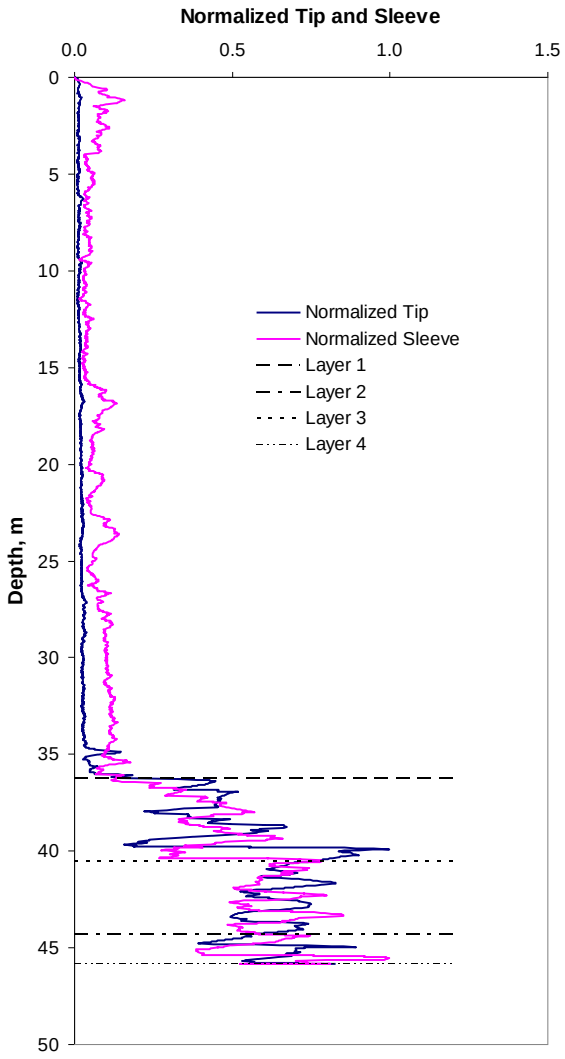

Figure 3. Four layers profiled

REFERENCES

Fisher, W.D. 1958. On grouping for maximum homogeneity, *Jour. Am. Stat. Assoc.,* 53(284):789-798.

Fitzgibbon, L.J., Allison L. and Dowe D. 2000. Minimum message length grouping of ordered data, *Proceedings of the 11th International Conference on Algorithmic Learning Theory*, Sydney, Australia, 56 – 70.

Hawkins, D.M. and Merriam, D.F. 1973. Optimal zonation of digitized sequential data, *Jour. Math. Geology,* 5(4): 389-395.

Lindberg, M.B. 1990. Fisher: A Turbo Pascal unit for optimal partitions, *Computers & Geosciences,* 16(5):717-732.

Low, B.K., and Tang, W.H. 1997. Reliability analysis of reinforced embankment on soft ground, *Can. Geotech. J.* 34:672-685.

Mayne, P.W. 2009. *Engineering design using the cone penetration test – geotechnical applications guide*. ConeTec. Pp. 165.

McRae, D.J., 1971. MIKCA: A Fortran IV iterative K-means cluster analysis program, *Behavioral Science* 16: 423–424.

Tumay T.M., and Titi, H.H. 2000. Louisiana's Continuous Intrusion Miniature Cone Penetration Test System, In: http://onlinepubs.trb.org/onlinepubs/trnews/rpo/rpo.trn207.pdf .

APPENDIX: VBA CODE

```
Public Function OptimalPartition(arra As Range, t)
   Dim tt, x, yy, xmax, xp, xmin, xm, s, d, e, z, xm3, i As Integer, j
As Integer, k As Integer, l As Integer, m As Integer, n As Integer,
suma As Double, sumc As Double, sumv As Double, q As
Integer, nh As Integer, ng As Integer, r As Integer, w As Double,
u As Double, xmm As Double, c As Integer
   Const rr = 1
   x = arra
   k = t
   r = arra.Rows.Count
   c = arra.Columns.Count

   nh = r - 1
   ng = r * (r + 1) * 0.5

   ReDim tt(1 To r, 1 To c)
   ReDim xmax(1 To 50)
   ReDim xp(1 To rr, 1 To t)
   ReDim xmin(1 To 50)
   ReDim xm(1 To 80)
   ReDim s(1 To 80)
   ReDim d(1 To 3200)
   ReDim e(1 To 80, 1 To 100)
   ReDim z(1 To r, 1 To c)
   ReDim xm3(1 To 100)

 'Data normalization
   For j = 1 To c
      For i = 1 To r
         tt(i, j) = x(i, j)
      Next i
      xmax(j) = Application.Max(tt)
      xmin(j) = Application.Min(tt)
   Next j

   For j = 1 To c
      xmm = xmax(j) - xmin(j)
      For i = 1 To r
         z(i, j) = (x(i, j) - xmin(j)) / xmm
      Next i
   Next j

   'To calculate the matrix of the sums of squares within
changeable layers
   For i = 1 To ng
      d(i) = 0
   Next i

   For j = 1 To r - 1
      For i = j + 1 To r
         sumc = 0
       For l = 1 To c
         suma = 0
         For q = j To i
          suma = suma + z(q, l)
         Next q
         suma = suma / (i - j + 1)

         For q = j To i
            sumc = sumc + (z(q, l) - suma) ^ 2
         Next q
       Next l
      sumv = r * (j - 1) + i - (j - 1) * j / 2
      d(sumv) = sumc
      Next i
   Next j

   'To find the min sum of squares over k layers divided into
   For i = 1 To nh
      For j = 1 To k
         e(i, j) = 0
      Next j
   Next i

   For j = 1 To r - 1
      For i = 1 To r - j
      sumv = r * i + r - j + 1 - i * (i + 1) * 0.5
      s(i) = d(i) + d(sumv)
      Next

      suma = s(1)
      e(j, 2) = 1
      For i = 1 To r - j
         If suma < s(i) Then GoTo 1
      suma = s(i)
      e(j, 2) = i
1:
      Next i
      xm(r - j) = suma
   Next j

   nextrow =
Application.WorksheetFunction.CountA(Range("A:A")) + 1
   Cells(nextrow, 1) = "L2="
   Cells(nextrow, 2) = Math.Round(xm(nh), 2)

   w = k - 1
   For l = 3 To k
      nh = r - l + 1
      For j = 1 To nh
         For i = 1 To nh - j + 1
```

```
            sumv = r * (l + i - 1) - j + 1 - (l + i - 1) * (l + i - 2) * 0.5
            s(i) = xm(i) + d(sumv)
        Next i

        suma = s(1)
        e(j, l) = l - 1
        For i = 2 To nh - j + 1
            If suma <= s(i) Then GoTo 2
            suma = s(i)
            e(j, l) = i + l - 2
2:
        Next i

        sumv = nh - j + 1
        xm(sumv) = suma
    Next j

  nextrow =
Application.WorksheetFunction.CountA(Range("A:A")) + 1
    Cells(nextrow, 1) = "L" & l & "="
    Cells(nextrow, 2) = Math.Round(xm(nh), 2)
  Next l

  xp(rr, 1) = Math.Round(xm(nh), 2)

  'To find the optimal partition of k layers divided into
  For l = 2 To k
  nextrow =
Application.WorksheetFunction.CountA(Range("C:C")) + 1
  Cells(nextrow, 3) = "k=" & l
    sumv = l - 1
    w = r
    For i = 1 To k
        xm3(i) = 0
    Next i
    For j = 0 To l - 2
        u = l - j
        sumv = r - w + 1
        w = e(sumv, u)
        u = j + 1
        xm3(u) = w
    Next j

    For i = 1 To l - 1
   nextrow =
Application.WorksheetFunction.CountA(Range("C:C"))
    Cells(nextrow, i + 3) = xm3(i)
        xp(rr, i + 1) = xm3(i)
    Next i
  Next l

 OptimalPartition = xp

  Columns("A:A").Select
  With Selection
    .HorizontalAlignment = xlRight
  End With

  Columns("B:B").Select
  With Selection
    .HorizontalAlignment = xlLeft
  End With

  Range("A1").Select

End Function
```