Sparse p-Adic Data Coding for Computationally Efficient and Effective Big Data Analytics

Fionn Murtagh, Department of Computing and Mathematics,
University of Derby, Derby DE22 1GB, UK; and
Department of Computing, Goldsmiths, University of London,
London SE14 6NW, UK. Email: fmurtagh@acm.org

April 26, 2016

Abstract

We develop the theory and practical implementation of p-adic sparse coding of data. Rather than the standard, sparsifying criterion that uses the L_0 pseudo-norm, we use the p-adic norm. We require that the hierarchy or tree be node-ranked, as is standard practice in agglomerative and other hierarchical clustering, but not necessarily with decision trees. In order to structure the data, all computational processing operations are direct reading of the data, or are bounded by a constant number of direct readings of the data, implying linear computational time. Through p-adic sparse data coding, efficient storage results, and for bounded p-adic norm stored data, search and retrieval are constant time operations. Examples show the effectiveness of this new approach to content-driven encoding and displaying of data.

Keywords: big data, p-adic numbers, ultrametric topology, hierarchical clustering, binary rooted tree, computational and storage complexity

1 Introduction

We start with a description of the context for this work. In [24], we provide background on (1) taking high dimensional data into a consensus random projection, and then (2) endowing the projected values with the Baire metric, which is simultaneously an ultrametric. The resulting regular 10-way tree is a divisive hierarchical clustering. Any hierarchical clustering can be considered as an ultrametric topology on the objects that are clustered. An ultrametric topology also expresses an r-adic number representation. We require r integer, $r \geq 2$. A 10-way tree, derived from decimal numbers, can be considered as an r-adic number visualization, where $r = 10$. In [7] we discussed a number of applications, and accompanying experimental evaluation. How our random projection, based on uniformly distributed random axes, differs from other work on random projection that requires Gaussian axes, is discussed in [27]. In [25], the consensus random projection was related to the principal eigenvector, addressed also was how the consensus projection is a seriation with clustering properties, and this process of seriation and clustering is closely related to the methods of spectral clustering, and power iteration clustering. This is extended in [26] with particular reference to Correspondence Analysis.

In [24, 27], the context for the use of the following data is described: 34,352 research funding proposals indexed in the open source Apache Solr package for (server-side) indexing, and (client-side) querying, search and retrieval. This package supports search and retrieval in large document collections, consisting of various fields (such as title, authors, abstract, and any other field that is defined for the document collection). The Solr packages determines and uses a similarity between documents that it calls MLT ("more like this"). This similarity has fixed field weights, and otherwise weights the document/term indexed data (see further discussion in [23]). We used scores generated by Solr for the top 100 matching proposals for each of a selection of 10,317 of the proposals set.

Using a regular 10-way tree, Figure 1 shows the hierarchy produced, with nodes colour-coded (a rainbow 10-colour lookup table was used), and with the root (a single colour, were it shown), comprising all clusters, to the bottom. The terminals of the 8-level tree are at the top.

This divisive hierarchical clustering algorithm uses the Baire, or longest common prefix, distance, which is also an ultrametric, on the consensus random projection values.

While this Figure 1 is an unorthodox display of a dendrogram, or hierarchical clustering, its role as a display is used here for exposition and discussion, and it is better depicted in this way, compared to a regular 10-way tree.

The first Baire layer of clusters, displayed as the bottom level in Figure 1, was found to have 10 clusters (6 of which are very evident, visually.) The next Baire layer has 87 clusters (the maximum possible for this 10-way tree is 100), and the third Baire layer has 671 clusters (maximum possible: 1000).

The Baire hierarchy has a limitation in regard to storage. We need to allow for a regular r-way tree, here a 10-way tree, for storage. For finding a path in a hierarchical descent from the root of the tree, traversal length is logarithmic in the number of objects clustered, i.e. associated with the terminal nodes. However storage of all nodes in the tree, when the precision and hence the number of digits per object is not fixed, is exponential.

We propose the following solution in order to linearize storage. First, we invoke the natural sparse coding of a p-adic (p prime) or more general r-adic (r integer) representation. We require checking of the general class of data that is at issue, in order to limit the precision. Secondly, we reduce the base r of the number representation used, in order to further economize on Baire tree information. We do this by means of a best approximation of the data in the new base, relative to the data in the former base.

Motivation for this work includes the varied perspectives that are at issue in the following: [4, 8, 11, 15, 18].

This paper is structured as follows.

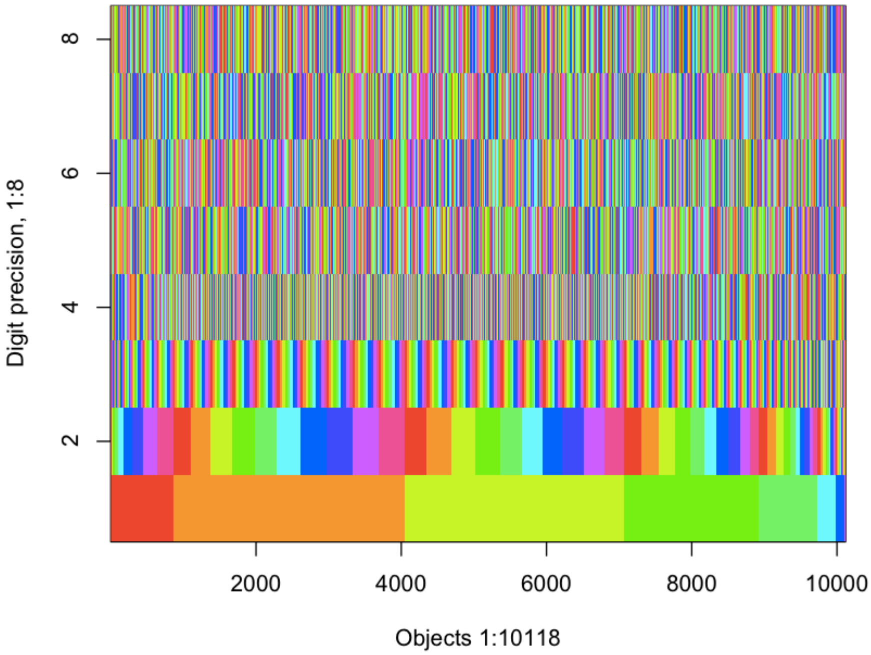

Figure 1: Means of 99 random projections. Abscissa: the 10118 (non-empty) documents are sorted (by random projection value). Ordinate: each of 8 digits comprising random projection values. A rainbow colour coding is used for display.

In [20], some of the ways are reviewed that we can p-adically encode a hierarchical clustering, a classification tree, that is conventionally a rooted, binary, labelled and ranked tree.

In section 2 we introduce sparse coding. General motivation for r-adic and p-adic (r integer, p prime) data coding is provided in section 3. Then specification is provided in section 4. Our first method for implementing p-adic data encoding is set out in section 5. In section 6 a further algorithm is proposed of the p-adic or m-adic ($p, m < r$ where r-adic is the starting context) representation.

In this work we are not just dealing with a tree structuring, viz. a hierarchical clustering, followed by encoding, but rather a tree structuring that represents the data. In this particular way, data content is taken into account. In section 6, with the aim of storage economy, we do take the hierarchic representation alone, and simplify that, through p-adic or m-adic mapping (p prime, m an integer, with p or m < r, r the integer that is used in our initial r-adic representation).

2 Introduction to Sparse Coding

Sparse coding is used to support data compression. Additionally compressive sampling is the use of appropriate sparse coding in order to support data reconstruction. Widely used examples of sparse coding are run length encoding, use of orthonormal mapping through eigen-reduction, and the use of sparsifying signal transforms including the Fourier, wavelet, discrete cosine, and other multiple resolution transforms. In this work, we demonstrate how p-adic number representation provides a most advantageous way to sparsely encode big data sets.

A short introduction to sparse coding is [28]. The ith input vector $x_{iJ} \in \mathbb{R}^J$ is written as a linear combination of a possibly over-complete (i.e., $K > J$) basis, ϕ, with each a_{ik} a scalar coefficient: $x_{iJ} = \sum_{k=1}^{K} a_{ik}\phi_{kJ} = a_{iK}\phi_{KJ}$. Due to over-determination of the problem of estimating the coefficients, a sparsity cost function is also taken into consideration. A standard objective for such sparse coding is as follows (where we are rewriting in matrix notation, and replacing norm with distance):

$$\begin{aligned}
&\min_{a_{iK},\phi_{KJ}} \sum_{i=1}^{I} \|x_{iJ} - \sum_{k=1}^{K} a_{ik}\phi_{kJ}\|_2^2 + \lambda \|a_{iK}\|_0 \\
&= \min_{a_{iK},\phi_{KJ}} \sum_{i=1}^{I} \|x_{iJ} - a_{iK}\cdot\phi_{KJ}\|_2^2 + \lambda \|a_{iK}\|_0 \\
&= \min_{a_{iK},\phi_{KJ}} \sum_{i=1}^{I} d_2(x_{iJ} - a_{iK}\cdot\phi_{KJ})^2 + \lambda \|a_{iK}\|_0 \qquad (1)
\end{aligned}$$

i.e., minimize in a least squares sense, using Euclidean, L_2, distance, the separation of x_{iJ} from its approximation by the linear combination of ϕ_k terms, $1 \leq k \leq K$, each term weighted by a_{ik}; and that minimization subject to the L_0

pseudo-norm, implying that the number of non-zero terms is to be minimized. For this optimization, λ is a Lagrange multiplier.

Due to its differentiability and convex optimization to provide a solution for it, $\|a_{iK}\|_0$ is generally replaced, as an approximation, by $\|a_{iK}\|_1$. Since the a_{ik} terms may decrease in value, but then the ϕ_k terms could be large, an additional constraint can be applied: $\|\phi_k\| \leq C$, for all k, and for constant C.

We restate the foregoing by saying that we want to represent all our x_{iJ} data vectors by a basis expansion, the ϕ_k being the basis, and having K coefficients, also termed loadings or projections. The coefficients are the scalar product, $a_i^t \phi$, of these two K-valued vectors. Our aim is to have a small number of non-zero coefficients in the vector of coefficients, a_{iK}.

As discussed briefly in [28], encoding a new vector x_{iJ} typically requires re-optimizing the vector of coefficients, a_{iK}.

In general applications, the significant coefficients are of most interest. The signal is made more sparse by using some compactifying transform (such as a wavelet or other multiresolution transform). Then, in transform space, small-valued transform coefficients are thresholded. This also can be applied for noise filtering, using hard or soft thresholding of wavelet coefficients. (That is, setting wavelet coefficients to zero if they are less than, for example, a statistically determined significance threshold, and then subsequently applying the inverse wavelet transform in order to reconstruct the data.) In compressive sensing [12], (1) the data is sampled through a sensing protocol; and then (2) the basis to be determined, ϕ above, is the product of the sensing protocol matrix and appropriate dictionary atoms; (3) since in a multiresolution transform dictionary such as wavelets, the coefficients have a power law decay, the estimation of these coefficients, a_i above, retains a set number of the largest entries. One arrives at a c-sparse encoding, such that $\|a_{iK}\|_0 \leq c$ (i.e. $\leq c$ non-zero coefficients retained).

Our new perspectives include the following.

1. Instead of a signal compactifying transform (such as a wavelet or other multiresolution transform), we use a tree or ultrametric topology in which our data is embedded. The partial order needs to take fully into account the importance or significance properties of the data used. We achieve this objective through p-adically encoding our data. (In [1], wavelet transform compactifying makes use of sets of branchings connecting wavelet scales. This then takes account of e.g. image or signal edges showing up as a succession of high wavelet coefficients along the branch of a wavelet tree structuring of the image; and the alternative, low wavelet coefficient values, in the case of smooth regions.)

   Our embedding in an ultrametric topology, expressed p-adically, is, by design, compactifying. That is, we take account of the given proximity (or metric) properties of our data.

2. For c-sparse encoding, hence minimizing $\|a_{iK}\|_0$, we will instead implement this constraint using the p-adic norm.

3. To address the issue raised in [28], whereby new data implies re-optimization, we will require the following: the ultrametric tree topology corresponding to our p-adic representation is a *regular* tree topology; also, this tree topology is a rooted, *ranked*, tree, viz. for each node in the tree, there is a real-valued level value.

3 Motivation for p-Adic Sparse Coding

p-Adic numbers are endowed with a very different order structure compared to real numbers. Following a short introduction to p-adic numbers, our objective in this work is to exploit the natural (topological rather than Hilbert space geometric) ordering of p-adic numbers in sparse coding.

p-Adic numbers were introduced by Kurt Hensel in 1898. The p-adic numbers are base p numbers, where p is a prime number. The reals are expressed in terms of a p-adic number systems where p is infinity. The ultrametric topology was introduced by Marc Krasner in 1944 [16], the ultrametric inequality having been formulated by Hausdorff in 1934. Essential motivation for the study of this area is provided by Schikhof [31] as follows. Real and complex fields gave rise to the idea of studying any field K with a complete valuation $|.|$ comparable to the absolute value function. Such fields satisfy the "strong triangle inequality" $|x + y| \leq \max(|x|, |y|)$. Given a valued field, defining a totally ordered Abelian group, an ultrametric space is induced through $|x - y| = d(x, y)$. Various terms are used interchangeably for analysis in and over such fields such as p-adic, ultrametric, non-Archimedean, and isosceles. The natural geometric ordering of metric valuations is on the real line, whereas in the ultrametric case the natural ordering is a hierarchical tree. p-Adic numbers, which provide an analytic version of ultrametric topologies, have a crucially important property resulting from Ostrowski's theorem. Each non-trivial valuation on the field of the rational numbers is equivalent either to the absolute value function or to some p-adic valuation ([31], p. 22). Essentially this theorem states that the rationals can be expressed in terms of (continuous) reals, or (discrete) p-adic numbers, and no other alternative system.

A generalization of integer coding, as well as Huffman (prefix) and Golomb-Rice (two-symbol alphabet) coding, is studied in [29]. While being acknowledged as being very close in operation to these widely used entropy coding algorithms, in [30] the authors point to how all are special cases of one p-adic coding algorithm. Many examples are provided in [30], using implementation in the Ruby programming language.

In [3], the so-called split-LBG clustering algorithm is considered in a p-adic framework. This clustering algorithm, [17], is used for data quantization. Motivation for this work is dynamic contexts, where cluster centres and data elements considered, can change over time. Another top-down, or divisive, hierarchical clustering is proposed in [19]. Our work, that uses the Baire hierarchy, seeks digit, and hence integer, representatives of clusters or nodes in the tree. In adopting such an approach, we very easily arrive at p-adic coding.

4 Approximation of r-Adic (r Integer) Representation

We proceed as follows. Express our data as r-adic numbers. Take a usual decimal representation of real valued numbers, to finite precision, base $r = 10$ and precision is the cth integer value.

A direct expansion of $x = \sum_{k=0}^{c} a_k p^k$ offers no guarantee of a smaller, resulting number of digits, compared to the original base, r. Expansion can be considered for $p < r$ or $p > r$, and for representation, digit-wise, of the given x, base r, and the full number. Given our use of coding to expedite search and retrieval, we therefore determine close approximation to our data. This is analogous to a transform coding, referred to in section 2, used subsequently for sparse coding.

Rewriting a c-digit, base $r = 10$, number in some other base, $x = \sum_{k=0}^{c} a_k p^k$, is not of benefit, because of computation required (more so than the stages of our Baire-based algorithm which, by design, has linear time requirements), and also due to the undetermined digits of precision, and hence overall storage, required.

Motivated by the problem specification of equations (1), we ask what benefits could there be if, instead of the L_2, together with L_0 and L_1 metrics, we use the Baire metric, which is simultaneously an ultrametric. See [7, 22], the former for applications, and the latter for roles in computational science. For vectors $x_{iJ}, x_{i'J}$, consider a basis r, which can be a prime, notationally p, or non-prime, for example, decimal, $r = 10$. Definition of the Baire metric: $d_r(x_{iJ}, x_{i'J}) = r^{-\beta}$ such that $\beta = \text{argmax}_j \{x_{ij} = x_{i'j} | 1 \leq j \leq J\}$.

We seek to minimize

$$\sum_{i=1}^{I} d_2^2(x_{iJ} - a_{iK} \cdot \phi_{KJ})$$

$$= \sum_{i=1}^{I} \sum_{j=1}^{J} \left(x_{ij} - \sum_{k=0}^{c-1} a_{ik} \phi_j^k \right)^2 \text{ with } \|x_i\|_r \leq r^{-c} \text{ and } a_{ik} < r \tag{2}$$

In seeking this best approximation, given ordering of r-adic expansion through the ϕ^k term, this natural ordering allows, for the r-adic digits, the best 1-, 2-, ... c-approximation to be derived from the r-adic representation.

Informally expressed, we have the following. At each level of the regular tree, we approximate the values that we have at that level by a best match to all of these values. Also the number of levels in the regular tree is bounded by design.

5 p-Adic Encoding Algorithm Description and Implementation

Given data in a space of any dimensionality, (1) we apply a consensus of random projections, and (2) we induce a Baire hierarchical clustering on these projections. Given that the projections are of digit precision J, in a real number and hence decimal by default system (i.e. $r = 10$), the projected values are endowed with the Baire distance (and ultrametric). This Baire distance is the longest common prefix, of the digits, ordered by precision.

We can display the Baire hierarchy, for I data points, and for digit precision J, as an $I \times J$ array. Each value in this array is a base $r = 10$ value. We have: $A : I \times J \rightarrow \{0, 1, 2, \ldots, r-1\}$ where $r = 10$. Consider any digit of precision, j_α. Fix the value of $j_\alpha \in \{0, 1, \ldots, r-1\}$. Consider all child nodes of this node. Unless any node is empty, there are r child nodes of this selected node. Cf. Figure 1 and later figures also.

In [7] we noted how digits at differing precision levels display different distributions. In that work we noted how different digit distributions can be used as a novel discriminating feature, and we exemplified this on astronomical spectrometric and photometric redshift values. This was in the context of nearest neighbour regression. A consequence for this work in this paper is that we will approximate our given A values independently by precision digit. A is our array display of the hierarchy.

More generally, patterns and, indeed, anomaly in data digits can be of major benefit in, e.g., forensic data analytics. Cf. Benford's law [5, 13, 6].

We map the base $r = 10$ digit values onto a base p system. We examined $p = 7, 5, 3, 2$, and we report below on some of these. It is not a requirement that p be prime. What does follow from our algorithm is that the closer p or an alternative target number base is to $r = 10$, then the better the approximation, that we will form to A, will be.

In order not to overload our notation, we will consider any one given digit of precision, $j = 1, 2, \ldots, J$. See Figure 1, where we are taking into consideration at any given time, one row, or one level of digit precision. We do this in view of the observed different distributional characteristics of digits that we reported in [7], and in view of Benford's law. Were it the case of compression as our sole objective, then we would take all digits into consideration, hence all j.

From equation (2), we seek to determine the minimum of:

$$(x_i - \sum_{k=0}^{K-1} a_{ik} \phi^k)^2 \tag{3}$$

for all objects, i, hence, we seek to determine the minimum of:

$$\sum_{i=1}^{I} (x_i - \sum_{k=0}^{K-1} a_{ik} \phi^k)^2 \tag{4}$$

Write x_i by digit of precision, x_{ij}. Further we specify that the set of objects, i, must be in a partition, to be determined, of the object set. Call this partition of the objects, $o_0, o_1, \ldots o_{K-1}$. Alernatively expressed, for some k, each index $i \in o_k$. So, having $K - 1$ values at this, or any, level, then the partition class of value $0 \leq k \leq K - 1$ is o_k. Following this additive decomposition (by partition) of x_i, we replace equation (4) by equation (5):

$$\sum_{k=0}^{K-1} \sum_{i=1}^{I} (x_{io} - a_{ik}\phi^k)^2 \text{ with } o = o_k \tag{5}$$

(We write $o = o_k$ here solely for notational clarity.) The first term in this expression is a real value, and the second term is a K-adic value. If p-adic, then this second term is in the field of p-adic numbers, $\mathbb{Q}_p$. Therefore as such, we must re-map the rightmost term into the reals. In order to do this, consider what is done in quantization for compression tasks: codewords are used, assembled in a codebook, and accessed as a function of the given data values. This we do now as follows: for all $i = 1, 2, \ldots, I$, $a_k\phi^k \to c_k$, with $c_k \in \mathbb{R}$. Thus, equation (5), which we seek to optimize through choice of codewords, c_k, and partition of objects, o_k, for $k = 0, 1, \ldots, K-1$, becomes:

$$\sum_{k=0}^{K-1} \sum_{i=1}^{I} (x_{io} - c_{ik})^2 \text{ with } o = o_k \tag{6}$$

This is the classical K-means clustering problem where we seek clusters of the object set, the union of these clusters is the object set, and the codewords are the cluster centres or centroids. Iterative optimization, from random initialization of cluster centres, provides the solution. See e.g. [2].

For completeness of exposition, we can return to the levels j corresponding to digit precision, in equations (5), (6):

$$\sum_{k=0}^{K-1} \sum_{i=1}^{I} (x_{ijo} - c_{ijk})^2 \text{ with } o = o_k, 1 \leq j \leq J \tag{7}$$

Thus:

$$\sum_{k=0}^{K-1} \sum_{i=1}^{I} (x_{ijo} - a_{ijk}\phi^k)^2 \text{ with } o = o_k, 1 \leq j \leq J \tag{8}$$

As we have noted above, we retain the separateness of digit precision, in view of our possible interest in their distributional properties. (Also as noted above, if compressibility were the sole objective, then the optimization of equations (6), (7) or (8) would be summed over j.)

An optimal solution to equation (6) implies an optimal representation of our set of reals expressed as a K-valued, K-adic expansion, equation (5). In practice, our iterative optimization of K-means in regard to the optimand, equation (6), is suboptimal. This is simply due to K-means partitioning being NP-complete. In usual operation, firstly a large number of iterations are permitted, and secondly, a number of different initialization configurations are used to provide the best overall result, or a consensus result. (In our implementation, R function `kmeans` was used with maximum 500 iterations and 50 random starts.)

Figure 2 shows the 3-adic representation that results for Figure 1. For computational convenience, these are digits 1, 2, 3 representing 0, 1, 2 in the 3-adic representation. Mean squared error (MSE) is determined from the 3-adic expansion, together with the codewords from the codebook. (These are the cluster centres determined by the K-means algorithm.) The MSE at digit precisions 1, 2, . . . , 8 were as follows: 0.26, 0.90, 0.88; 0.88; 0.89; 0.89; 0.90; 0.90. The p-adic Baire tree in this figure is a regular 3-way tree or hierarchy.

Figure 3 shows the 2-adic or binary representation of Figure 1. For computational convenience, these are digits 1, 2 representing 0, 1 in the 2-adic representation. The MSEs at digit precisions 1, 2, . . . , 8 were as follows: 0.56, 1.96, 1.98, 2.01, 1.97, 2.03, 1.97, 1.99. The p-adic Baire tree in this figure is a regular 2-way, or binary, tree or hierarchy.

Computationally all stages of the processing are a linear function of cardinality of the object set, and of the levels or digits of precision, of numbers of iterations in the K-means work. Thus overall our preprocessing, i.e. data structuring, is of linear computational time. Through application of filtering in the sparsely encoded data, i.e. limiting the p- or r-adic expansion to a fixed number of terms (alternatively expressed: limiting to a fixed number of digits of precision in the chosen number base), we also have storage that is linear in the number of objects.

A binary tree representation (Figure 3) is particularly appropriate for decision making and related binary routing, and such a clustering display, determined in a top-down or divisive way can be contrasted with an agglomerative hierarchical clustering algorithm.

To check on this, the following assessment was carried out. The mean random projection set, 10118 values, was hierarchically clustered, using Ward's minimum variance method. The input data (cf. Figure 1) were to full available precision, viz. 9 digits of precision. The real values representing the binary expression from Figure 3 were also hierarchically clustered using Ward's method. For the projections, the correlation between input distances and cophenetic (or ultrametric, or tree) distances determined from the hierarchical clustering was 0.6137619. For the binary, 2-adic hierarchy, the correlation between input distances (i.e. from the reals directly expressing the 2-adic, 8-level hierarchy), and cophenetic distances from the Ward hierarchical clustering, was 0.7257293. The correlation between the two sets of input distances was only 0.4872135. But actually the correlation between the values themselves (recalling that both sets of values comprised vectors of 10118 values) was 0.7102683. Finally the correlation between the two sets of cophenetic distances was 0.6154983.

The correlation between the two sets of values (mean random projection, as noted, of 9 digits of precision, and its 2-adic, 8-term expansion, that we determined through original digit optimized fit), viz. 0.7102683, is our most relevant finding. Note how levels of precision differ (9 versus 8), that in the former case we have digits that range over 0 to 9, whereas in the 2-adic representation case, we have digits that range over 0 and 1 (although, whenever convenient for implementation, we express them as, respectively, 1 and 2). So in the former case we have to consider, for each object, 10 possible values for each of 9 digits; and in the latter case, we have, for each object, 2 possible values for each of 8 digits.

For proximity matching or other such operations, we therefore require:

1. The random projection vector.

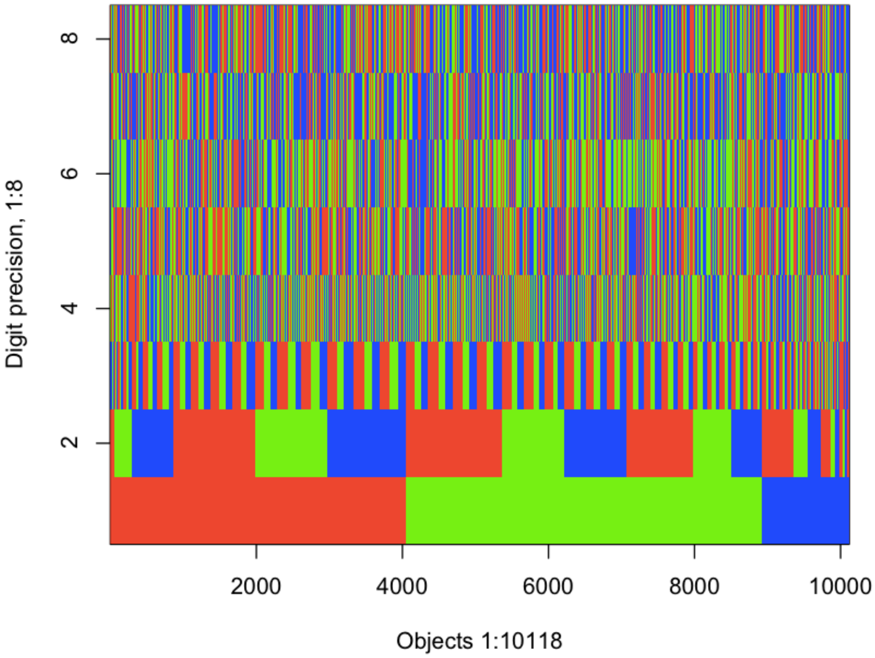

Figure 2: Abscissa: the 10118 (non-empty) documents are sorted (by random projection value). Ordinate: each of 8 digits comprising random projection values. 3-adic representation, displayed using three colours.

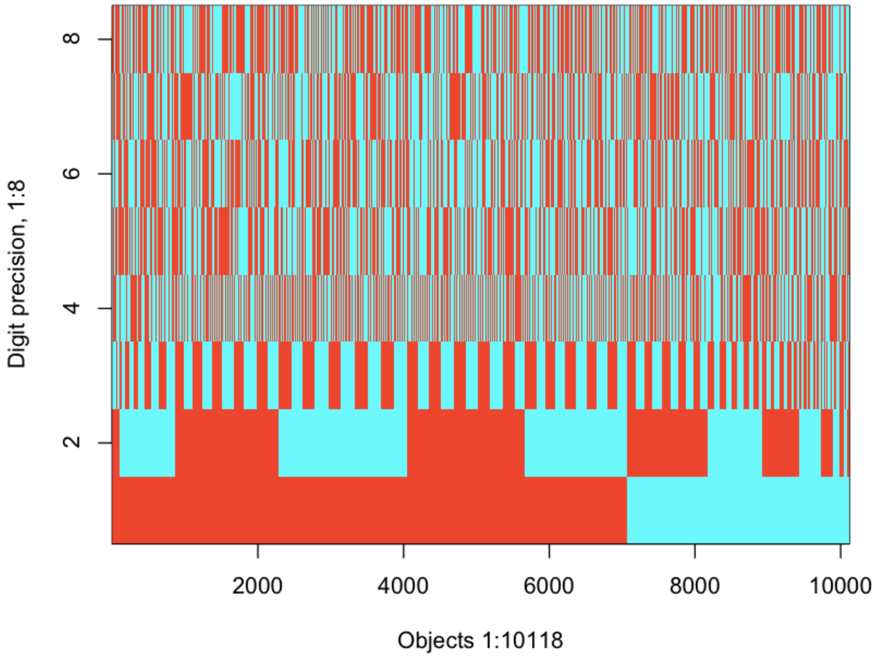

Figure 3: Abscissa: the 10118 (non-empty) documents are sorted (by random projection value). Ordinate: each of 8 digits comprising random projection values. 2-adic, or binary, representation, displayed using two colours.

2. The codebook for each digit level.

We term the foregoing approach the codebook through cluster-based representation.

6 p-Adic and m-Adic Fitting through Approximation of Hierarchy Representation

We now seek to simplify the representation in the following way. Label the digits of precision, or layers, or levels, $j = 1, 2, \ldots$, which in the case of the study here are of maximum value, 8. The set of objects is indexed by i, and in the study here, $i = 1, 2, \ldots, 10118$. We will refer to the Baire hierarchy representation of objects crossed by digits, as the Baire array display, A, cf. section 5 above, with elements a. For digit set, J, and object set, I, we have the Baire array display defined by the mapping: $A : I \times J \rightarrow \mathbb{Z}_\nu$, where $\mathbb{Z}_\nu$ is the set of integers modulo ν. In the decimal, or base 10, number system, which is our point of departure, $\nu = 10$ and we consider the decimal digits, $\mathbb{Z}_\nu = \{0, 1, 2, \ldots, 9\}$.

1. For a given digit level, j, and given object neighours, i and i', we take (without loss of generality) $i' = i + 1$. We consider the Baire array display values, $a_{i,j}, a_{i',j}$.

2. As a first step we determine all neighbour pairs, $a_{i,j}, a_{i'j}$, that have the same parent digit value. That is, $a_{i,j-1} = a_{i',j-1}$.

3. The next step in our algorithm is to determine if our Baire array display values differ by 1, viz. $|a_{i,j} - a_{i',j}| = 1$.

4. If, instead, $a_{i,j} = a_{i',j}$, then no intervention is required.

5. If $|a_{i,j} - a_{i',j}| > 1$, then we will not intervene insofar as there is sufficiently clear difference between these Baire array display values.

6. We determine the number of neighbour values differing by 1. That is, we determine: $\mathcal{N} = |(a_{i,j}, a_{i',j})|, i = i^*, i' = i^* + 1,$ such that $|a_{i,j} - a_{i',j}| = 1$, i.e. the number of these $i^*, i^* + 1$ pairs.

7. We determine the minimum such values of these pairs: $\mathrm{argmin}_{i^*, i^*+1} \mathcal{N}$. For any representative values of the indices, i', j, call the larger value that realizes this, $a_{i',j} = w$. We have $w \in \mathbb{Z}_\nu$.

8. Update the Baire array display, a, as follows: if $k \geq w$, then for $a_{i,j} = k$, $a'_{i,j} = a_{i,j} - 1$. That is, for those Baire array display pairs, with the same parent digit value, that differ by 1 in their own digit values, we set the higher digit value to be equal to the lower digit value; then we make this update of those digit values throughout the entire Baire hierarchy representation.

9. It results that the Baire hierarchy representation, with values in $\mathbb{Z}_\nu$, has been transformed into an optimal (resulting from step 7) approximating Baire array display representation, with values in $\mathbb{Z}_{\nu-1}$.

10. This procedure, steps 1–9, can be repeated. On each such iteration, we decrease the number of values in the Baire array display by 1.

6.1 Example of Application

We use Figure 1. This figure is displayed with digit values 0 (taken here as a value, and not the absence of a value) to 9, and we have it as 10-adic encoded. The approximation algorithm described in the previous subsection is applied, stepwise. The initial 10-adic representation is approximated by a 9-adic approximation. From the 9-adic encoded hierarchy representation, an 8-adic encoded hierarchy representation approximates it. Then that is approximated by a 7-adic encoded hierarchy representation. Then a 6-adic encoding approximates that. Then a 5-adic encoding approximates that, and is shown in Figure 4. This is followed by a 4-adic encoding, a 3-adic encoding, and finally a 2-adic, binary, encoding. The last one here is shown in Figure 5.

6.2 Approximation Distance and Observed Convergence of m-Adic Approximation of the Baire Array Display

Figure 6 displays the squared distance, firstly, between the m-adic approximation and the given Baire array display. That is, we have 10-adic or decimal data to start with. See Figure 1. Just two of our m-adic approximations are displayed here in Figures 4 and 5. In Figure 6 there is the growing discrepancy between the m-adic approximation and the given decimal data.

Note that the approximation to the given Baire array display, as is the latter, are normalized, so that values are real-valued and bounded by 0 and 1.

Secondly, Figure 6 shows the convergence properties. For m $= 9, 8, \ldots, 3, 2$, we consider the squared error between the m-adic approximation and the m+1-adic approximation, or, initially, the given decimal data.

We note from Figure 6 that up to m = 5, which as a prime, we can write conventionally as p = 5, we have limited deviation from the given Baire array display data.

7 Conclusions

The Baire distance, simultaneously inducing an ultrametric as well as a metric, is a longest common prefix metric. It is just of interest to note that data transmission is commonly supported in computer networks through, as it is termed, longest prefix matching [9].

In this work, we have approximated any particular set of decimal data by a representation in any number basis that is from 2 (i.e. binary, prime), 3 (i.e.

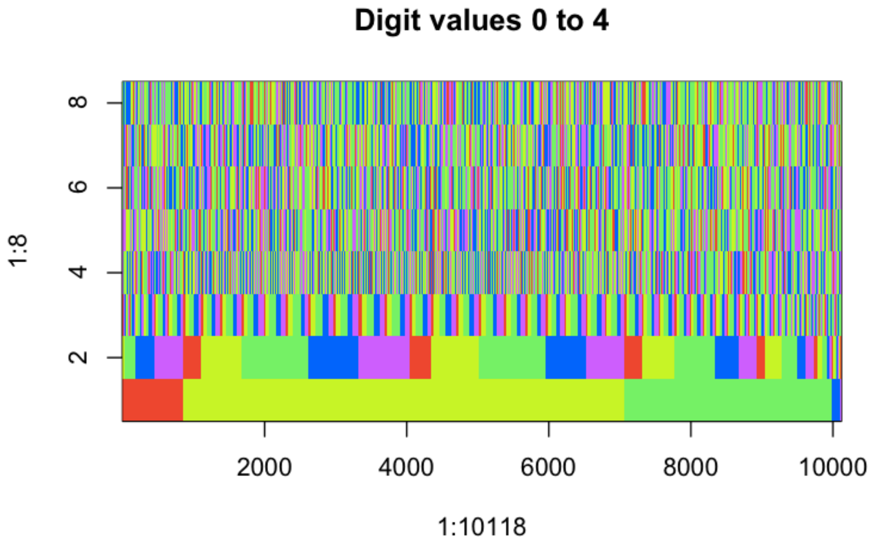

Figure 4: Fifth approximation by merging close neighbour pairs of a common parent node.

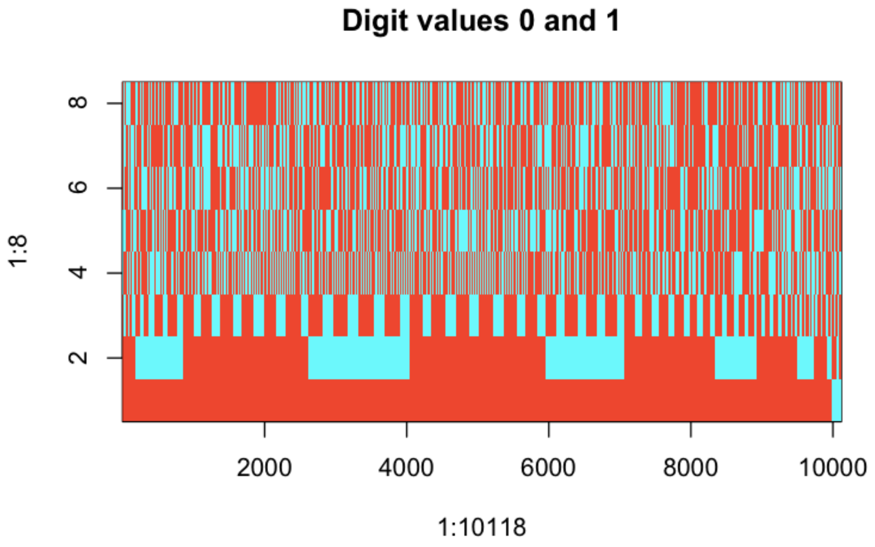

Figure 5: Eighth approximation by merging close neighbour pairs of a common parent node. Display of a binary tree.

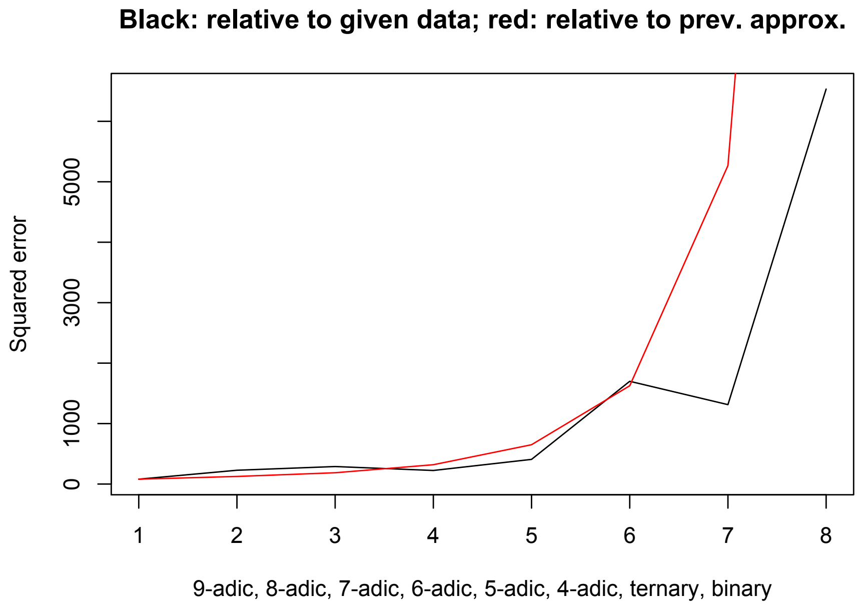

Figure 6: Black curve: measure of approximation by the succession of m-adic approximations, starting with the 10-adic Baire array display, proceeding with the 9-adic, 8-adic, 7-adic, 6-adic, 5-adic, 4-adic, 3-adic or ternary, 2-adic or binary. Approximation is measured by squared error. Red curve: measure of convergence, being the difference, viz. the squared error, between the m-adic approximation and the previous approximation. Approximations and the original Baire array display have normalized values, in [0,1].

ternary, prime), 4, 5 (prime), 6, 7 (prime), 8, and 9. Our methodology includes iterative optimization that is used for quantization. Such iterative optimization is not guaranteed to provide the global optimum, yet nonetheless with a fixed number of iterations, a practical sub-optimal outcome is obtained.

The relevance of bases other than 10, or 2 for binary arithmetic, can be addressed in terms of application domain. Following discussion of number base (and also scale) invariance, Hill [13] briefly comments on octal (base 8) and hexadecimal (base 16), as well as binary, systems for computation. It can also be recalled that a ternary, i.e. base 3, computer was developed and built by Nikolay Brusentsov in 1958, and remains a technology, i.e. having a ternary rather than a binary basis for computers, that is still the subject of interest [14].

The ability to map, through approximation, between number systems, that we have developed here, is valuable, firstly, to minimize storage, and, secondly, to manage the data representation. Such data representation in data analytics is for decision support, or, alternatively expressed, supporting actionable data.

Our work supports the following insightful statement. In his book [32] (p. 2015), Herbert Simon, the Nobel prize-winner (1978, Economics), noted the following, in the chapter entitled "The architecture of complexity: hierarchic systems": "How complex or simple a structure is depends critically upon the way in which we describe it. Most of the complex structures found in the world are enormously redundant, and we can use this redundancy to simplify their description. But to use it, to achieve the simplification, we must find the right representation."

References

[1] R.G. Baraniuk and V. Cevher and M.F. Duarte and C. Hegde, "Model-based compressive sensing", *IEEE Transactions on Information Theory*, 56 (4), 1982–2001, 2010.

[2] H.-H. Bock, "Origins and extensions of the k-means algorithm in cluster analysis", *E-Journal for History of Probability and Statistics*, 4 (2), 2008. http://www.jehps.net/Decembre2008/Bock.pdf.

[3] P.E. Bradley, "On p-adic classification", *p-Adic Numbers, Ultrametric Analysis and Applications*, 1 (4), 271–285, 2009.

[4] L. Brekke and P.G.O. Freund, "p-Adic numbers in physics", *Physics Reports*, vol. 233, 1993, pp. 1–66.

[5] F. Benford, "The law of anomalous numbers", *Proceedings of the American Philosophical Society*, 78, 551–572. 1938.

[6] A. Berger and T.P. Hill, *An Introduction to Benford's Law*, Princeton University Press, 2015.

[7] P. Contreras and F. Murtagh, “Fast, linear time hierarchical clustering using the Baire metric”, Journal of Classification, 29, 118–143, 2012.

[8] B. Dragovich and A. Dragovich, “p-Adic modelling of the genome and the genetic code”, *Computer Journal*, 53, 432–42, 2010.

[9] O. Erdem, A. Carus and Hoang Le, “Value-coded trie structure for high-performance IPv6 lookup”, *Computer Journal*, 2014, in press.

[10] F.Q. Gouvêa, *P-Adic Numbers*, Berlin: Springer, 2003.

[11] P. Hall, J.S. Marron and A. Neeman, “Geometric representation of high dimension, low sample size data”, *Journal of the Royal Statistical Society Series B*, 67, 427–444, 2005.

[12] K. Hayashi, M. Nagahara and T. Tanaka, “A user’s guide to compressed sensing for communications systems”, *IEICE Transactions on Communications*, E96-B (3), 685–712, 2013.

[13] T.P. Hill, “A statistical derivation of the significant-digit law”, *Statistical Science*, 10 (4), 354–363, 1995.

[14] D.W. Jones, *The Ternary Manifesto*, including “Standard ternary logic”, “Ternary arithmetic”, “TerSCII: ternary standard code for information interchange”, “Number representations for ternary computers”. http://homepage.cs.uiowa.edu/∼jones/ternary, 2012.

[15] A.Yu. Khrennikov, “Gene expression from polynomial dynamics in the 2-adic information space”, *Proceedings of the Steklov Institute of Mathematics*, 265, 131–139, 2009.

[16] M. Krasner, “Nombres semi-réels et espaces ultramétriques”, *Comptes-Rendus de l’Académie des Sciences, Tome II*, 219, 433–435, 1944.

[17] Y. Linde, A. Buzo and R.M. Gray, “An algorithm for vector quantization design”, *IEEE Transactions on Communications*, 28, 84–95, 1980.

[18] B. Mirkin (1997) Linear embedding of binary hierarchies and its applications, in B. Mirkin, F. McMorris, F. Roberts, and A. Rzhetsky (Eds.) *Mathematical Hierarchies and Biology*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, V. 37, American Mathematical Society, Providence, 331-356.

[19] B. Mirkin and E. Koonin (2003) A top-down method for building genome classification trees with linear binary hierarchies, in M. Janowitz, J.-F. Lapointe, F. McMorris, B. Mirkin, and F. Roberts (Eds.) *Bioconsensus*, DIMACS Series, V. 61, Providence: American Mathematical Society, 97-112.

[20] F. Murtagh, Symmetry in data mining and analysis: a unifying view based on hierarchy, *Proceedings of Steklov Institute of Mathematics*, 265, 177-198, 2009.

[21] F. Murtagh, "From data to the p-adic or ultrametic model", *p-Adic Numbers, Ultrametric Analysis and Applications*, 1, 58-68, 2009.

[22] F. Murtagh and P. Contreras, "Fast, linear time, m-adic hierarchical clustering for search and retrieval using the Baire metric, with linkages to generalized ultrametrics, hashing, Formal Concept Analysis, and precision of data measurement", *p-Adic Numbers, Ultrametric Analysis and Applications*, 4, 45–56, 2012.

[23] F. Murtagh, "MoreLikeThis and Scoring in Solr", technical report, 4 pp., 26 May 2013. http://www.multiresolutions.com/HiClBaireRanSpanPaths

[24] F. Murtagh and P. Contreras, "Linear storage and potentially constant time hierarchical clustering using the Baire metric and random spanning paths", in A. Wilhelm (ed.), *Analysis of Large and Complex Data*, Springer, Heidelberg 2016, in press.

[25] F. Murtagh and P. Contreras, "Clustering through high dimensional data scaling: applications and implementations", *Proceedings, ECDA 2015, European Conference on Data Analysis*, 2015, submitted.

[26] F. Murtagh, "Big Data scaling through metric mapping: Exploiting the remarkable simplicity of very high dimensional spaces using Correspondence Analysis", *Proceedings, IFCS 2015, International Federation of Classification Societies*, 2015, submitted.

[27] F. Murtagh and P. Contreras, "Random projection towards the Baire metric for high dimensional clustering", in A. Gammerman, V. Vovk and H. Papadopouous, (eds.) *Statistical Learning and Data Sciences*, Lecture Notes in Articial Intelligence, Volume 9047, pp. 424–431. Springer, Heidelberg, 2015.

[28] A. Ng et al., "Sparse coding", ufldl.stanford.edu/wiki/index.php/Sparse_Coding, last modified 8 April 2013 (accessed 2016-04-08).

[29] A. Rodionov and S. Volkov, "p-Adic arithmetic coding", *Contemporary Mathematics*, 508, 201–213, 2010.

[30] A. Rodionov and S. Volkov, "p-Adic arithmetic coding", 29 pp., 2007, http://arxiv.org/abs/0704.0834v1

[31] W.H. Schikhof, *Ultrametric Calculus*, Cambridge: Cambridge University Press, 1984. (Chapters 18, 19, 20, 21.)

[32] H.A. Simon, *The Sciences of the Artificial*, 3rd edn., MIT Press, 1996.