Exploring a region classifier based on Kolmogorov complexity

Kingsley HAYNES
School of Public Policy, George Mason University
Fairfax, VA 22030, USA

Rajendra KULKARNI
School of Public Policy, George Mason University
Fairfax, VA 22030, USA

Roger STOUGH
School of Public Policy, George Mason University
Fairfax, VA 22030, USA

Laurie SCHINTLER
School of Public Policy, George Mason University
Fairfax, VA 22030, USA

ABSTRACT

In this paper we explore the development of a parameter free region classifier based on Kolmogorov complexity. Given a set of regions described by unlimited but fixed number of attributes for each region, the region classifier will be able to build a classification tree which will help identify which regions are similar/dissimilar to each other based on a relative distance measure derived from Kolmogorov complexity. The region classifier is tested with the block level U.S. Census demographics data as well as hitech establishment data for a subset of metropolitan regions. Preliminary results are presented for the census data as well as for the hitech sector for three different time periods.

Key Words: Kolmogorov Complexity, Regional classification, Hitech

1. INTRODUCTION

Back in the days when Silicon Valley was still considered as a gold-standard for regional growth; cities/regions all over the world desired to be referred to as the next Silicon Valley. Such emulations/comparisons are of course more rhetorical than formal. In fact, it's not uncommon even for a third party observer, studying an object/phenomenon to compare it with some other known object/phenomenon and then to find patterns/similarities/dissimilarities. Again, such comparisons are seldom based on robust statistical analysis. The later may be due to dearth of data as well as lack of a universally acceptable methodology that could serve as a general purpose classifier that puts different objects/phenomena in different boxes. Even if such data together with a robust method is available, comparisons based on different attributes give different output. What might be desirable is a universal classifier that is easy to use and whose output offers a relative degree of similarity/dissimilarity between large number of objects/phenomena.

The following sections describe theoretical basis of a general purpose classifier and a practical approach that approximates it.

2. KOLMOGOROV COMPLEXITY

Every object/phenomenon that can be described in a human recognized language can also be described as a binary sequence of zeros and ones. In that case, comparing two or more such objects reduces to comparing their respective binary sequences with each other, in other words finding a "distance" between the two sequences. For example one could compare two sequences by computing Hamming distance, where comparisons are made at the bit level for corresponding bits in each sequence. However, in some cases, such a distance may give misleading answer. For eg. take a random sequence A = "00101101010" and generate another sequence B of same length where every corresponding bit is reversed, thus B = "11010010101". A distance measure based on Hamming distance would result in the maximum distance between the two indicating they are unlike each other, however, in reality one is just the "negative" copy of the other. And as pointed out in [1] Bennett et. al if these were bits representing an image, the two images are still the same, except that one is a negative of the other. Similar arguments apply for other distance measures such as Manhattan or Euclidean distance. Let's revisit this point after describing what Kolmogorov complexity and a distance measure based on it.

Given a phenomenon/object the Kolmogorov complexity gives the most concise description of that phenomenon/object such as to fully describe it. However, there can be many different ways to provide a concise description. Therefore, one must specify what is meant by a concise description. This is accomplished with the help of a *formal description* language based on binary alphabets as shown below.

Formal Descriptive Language

Assume that every letter, word, symbol and expression is described in terms of strings made of binary alphabets of '0' and '1.' With this assumption one can build a lexicographical ordering, a sort of lookup table or an interpreter that generates the strings for each letter, word and symbol. Thus a description of an object now consists of a series or string of '0's and '1's. Here is an example of a lookup table that has lexicographical ordering,

$$U = [\ (\varepsilon;0), (0;1), (1;2), (00;3), (01;4), (10;5),\ldots] \quad (1)$$

where, the first symbol of the pair of symbols inside the parenthesis represents binary equivalent string of the second symbol and ε represents a NULL or a blank. Using U, an object O may be described by variable number of distinct strings Si such that:

$$D = [\ S_1, S_2, \ldots.S_i.\ ] \quad (2)$$

Where D is a set of descriptions of object O.

Let l_i represent the length in binary bits of string S_i such that

$$l_i = |\ S_i\ | \quad (3)$$

And

$$L = [\ |S_1|, |S_2|,\ldots.|S_i|\ldots] = [l_1, l_2,\ldots l_i,\ldots] \quad (4)$$

Then the string S_i with the smallest length l_i represents the most concise description and the size of such a string is called the Kolmogorov complexity, also known are K complexity of that object [2], [5].

$$K(O) = \textit{min}\ [\ |S_1|, |S_2|,\ldots.|S_i|\ldots] \quad (5)$$

or

$$K = K(O) = \textit{min}\ [l_1, l_2,\ldots l_i,\ldots] \quad (6)$$

Where *min* represents the minimum over all possible values.

Next we introduce the notion of computational complexity. Various descriptions of object O can be coded using computational programs (formal languages), then all such descriptions represent object O's complexity. However, the one program with the shortest possible description measured in terms of the length of the program represents the computational complexity of that object/phenomenon. In other words, the program with the least number of bits represents the most compressed description of the object O. And yet, such a program is non-computable [1], [2], [5]. In other words, it's a theoretical concept that sets the lowest threshold in terms of size of the full and concise description of that object/phenomenon. However, all is not lost because one may use programs based on Lempel-Ziv [6] compression scheme to achieve the smallest or the most compressed description and retrieve the original by decompression.

3. KOLMOGOROV COMPLEXITY DISTANCE

Consider two objects O_1 and O_2, with K complexities of K_1 and K_2 respectively derived from the methodology shown in Eq. (1) through (6), then $K(O_2|O_1)$ is called the conditional or relative K-complexity of object O_2 given O_1. Thus

$$K(O_2|O_1) = K(O_1cO_2) - K(O_1), \quad (7)$$

where $K(O_1cO_2)$ refers to K-complexity when O_1 and O_2 are *concatenated* in that order, and 'c' is a concatenation operator.

$$K(O_2|O_1) <> K(O_1|O_2) \quad (8)$$

In order to compute symmetric quantity that does not depend on the order in which O_1 and O_2 are concatenated one may use the formulae from [3] to get the following sequence of Eq (9) through Eq. (12):

$$K_d = \{ [K(O_1|O_2) - K(O_2|O_2)] / (K(O_2|O_2) \} + \{ [K(O_2|O_1) - K(O_1|O_1)] / K(O_1|O_1))\} \quad (9)$$

Where $K(O_i|O_i)$ is K complexity of string generated out of concatenation of Oi with itself. Rewriting K_d as:

$$K_d = \{ [K(O_1cO_2) - K(O_1)] - [K(O_2cO_2) - K(O_2)]\}/[K(O2cO_2) -K(O_2)] \}+\ldots\{ [K(O_2cO_1) - K(O_2)] - [K(O_1cO_1) - K(O_1)]\}/[K(O_1cO_1) -K(O_1)] \} \quad (10)$$

By using appropriate compressor then the following can be computed:

$$\{[L(l_1cl_2) - L(l_1)] - [L(l_2cl_2)] - L(l_2)] / [L(l_2cl_2)] - L(l_2)]\} + \ldots \{[L(l_2cl_1) - L(l_2)] - [L(l_1cl_1)] - L(l_1)] / [L(l_1cl_1)] - L(l_1)]\} \quad (11)$$

Another way suggested in [1] is to compute the normalized distance between O_1 and O_2 as:

$$[L(l_1cl_2) - \min (L(1_1), L(l_2))]/[\max(L(l_1), L(l_2))] \quad (12)$$

Note that subtracting min of the two sequences and division by max among the two, guarantees that sizes of the sequences does not affect the distance measure. Without these two terms, for eg., for same number of differences in the sequences, smaller sequences will appear to have large difference compared to longer sequence.

Although, Eq. (11) and Eq. (12) satisfy positivity (distance is >= 0) and symmetry, they may not satisfy the triangular inequality. Neverthless, for our purpose, where we compare each of the regions with a reference region, the relative distance measure in terms of Kolmogorov distance does compute.

Next we will illustrate with a toy example how to compute distance based on Eq (11). Following that a more comprehensive test results will be given based on block level Census 2000 demographics data (http://www.census.gov) as well as Hitech business establishment data for the years 1999, 2002 and 2006 (ESRI Business Analyst) for more than two dozen regions (Metropolitan divisions) consisting of 104 counties in the lower 48 states.

4. EXAMPLES

Example 1: This example consists of a toy problem. Consider three 5x5 grids G1 G2 and G3, where each grid cell has a value of either 0 or 1.

0	1	1	0	0
1	0	0	1	0
0	0	0	0	1
1	1	0	0	1
1	0	1	0	0

G1

0	1	0	0	0
0	0	1	1	0
1	0	0	0	1
1	1	0	0	1
1	0	1	0	0

G2

0	0	1	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	1

G3

Rewriting grid values as binary string S_1, S_2 and S_3 by concatenating each row we obtain:

S_1 = 0110010010000011100110100
S_2 = 0100000110100011100110100
S_3 = 0010000000000000000000001

To apply Eq. (9) we generate the following:

(S_1cS_2)=
0110010010000011100110100010000011010001110011010 0
(S_1cS_3)=
0110010010000011100110100010000000000000000000001
(S_2cS_1)=
0100000110100011100110100011001001000001110011010 0
(S_3cS_1)=
0010000000000000000000001011001001000001110011010 0
(S_1cS_1)=
0110010010000011100110100011001001000001110011010 0
(S_2cS_2)=
0100000110100011100110100010000011010001110011010 0
(S_3cS_3)=
0010000000000000000000001001000000000000000000001

where '*c*' is a concatenation operator.

Then $Kd(S_1S_2)$ is given by:

$$\{ [K(S_1cS_2) - K(S_1)] - [K(S_2cS_2) - K(S_2)]\}/[K(S2cS_2) -K(S_2)] \}+\ldots\{ [K(S_2cS_1) - K(S_2)] - [K(S_1cS_1) - K(S_1)]\}/[K(S_1cS_1) -K(S_1)] \} \quad (13)$$

Which results in a value of 1.166667 and $Kd(S_1S_3)$ is given by:

$$[K(S_1cS_3) - K(S_1)] - [K(S_3cS_3) - K(S_3)]\}/[K(S3cS_3) -K(S_3)] \}+\ldots\{ [K(S_3cS_1) - K(S_3)] - [K(S_1cS_1) - K(S_1)]\}/[K(S_1cS_1) -K(S_1)] \} \quad (14)$$

giving a value of 0.291667, indicating that G1 and G3 are more similar than G1 and G2.

As was stated previously that the Kolmogorov distance is a theoretical concept that is not easy to compute. Hence one must use methods that are approximations to the theoretical Kolmogorov distance, some of which are expressed as the compressors based on Lempel-Ziv techniques.

The next several examples deal with large datasets such as Census demographics and business listing across 29 metropolitan divisions comprising 104 counties in the lower 48 states (see Appendix A for the listing of the census division). To handle such large datasets, we use a methodology outlined in [4], represented in Eq. (10). The metropolitan divisions are based on Census boundary data files (http://www.census.gov/geo/www/cob/mmsa2003.html) as shown in Map 1. The metropolitan divisions are defined by the US Office of Management and Budget (http://www.census.gov/population/www/estimates/00-32997.pdf, Section 7, Pg 10).

Example 2: The Census demographic data at block level consists of 40 variables per metropolitan division. The results are based on Eq. (11) and were computed using Complearn toolkit (http://complearn.sourceforge.net), the output is shown in Figure (4). There are three distinct regimes that emerge in terms of normalized distance matrix, group 1 consists of New York, Los Angeles, Chicago, Philadelphia, Newark, in decreasing order of similarity distance, while

Dallas and Wash D.C. are nearly similar in distance from this group 1. Group 2 consists of Detroit, Seattle, Troy (MI) and Anaheim. While group 3 consists of Okland, Boston, Fortworth and Miami, San Francisco, Cambridge (MA), Camden (NJ), Fort Lauderdale (FL), Bethesda (MD), West Palm (FL), …Takoma (WA), Gary (IN), Wilimigton, (DE,MD, NJ). Members of group 3 are similar to members of group 1 members in decreasing order of similarity. In fact Gary (IN) and Wilmington (DE) are the farthest in demographic similarity compared to Los Angeles and New York. The results are shown in Figure (4).

Example 3: This example consists of comparing the same 29 metropolitan divisions as in previous example but data on the hitech sector services (combination of Sofware/IT and Engg/Management services). The data for 2006, 2002 and 1999 was extracted from ESRI's Business Analyst 2.1 products.

Comparative Kolomogorov complexity for year 2006, 2002 and 1999 are shown in Figure (1), (2) and (3) respectively. Each of the three figures, show three distinct groups of MSA. Of these, 1999 group consisting of New York, Los Angeles etc has 9 MSAs, this group doubles in size in 2002 and then shrinks back by about a third, yet the top five MSAs maintain stable relative positions. Opposite it true for the MSAs in the group at the other extreme.

5. CONCLUSION AND FUTURE RESEARCH

With a very simple tool consisting of *off-the-shelf* compression software such as gzip or zip, and applying it to regional datasets, one can emulate Kolmogorov complexities and Kolmogorov distances and quickly explore and determine whether there are hierarchical relationships between different regions and if a clear hierarchy cannot be determined then one can get a visual diagram of the relationships in terms of similarity distances among the regions. Note that this non-parametric analysis was carried out using little to no pre-processing of the datasets. One limitation of the current method is that, since this analysis is based on relative distance measure, comparative analysis across different time periods cannot be carried out unless one assigns a suitable objective desirability/undesirability score to each of the actors in the diagram. We hope to address these issues in the future. And yet, the importance of K distance diagrams in providing visual cues/information in alternate scenarios cannot be ignored. Especially for comparing how relative positions change with the addition of new actor(s) or deletion of existing actor(s) over different time periods. We plan to apply this methodology to number of other fields such as to compare changing political power structures; to compare adoption of public policies in different regions; to compare how foreclosure rates and housing prices change across regions.

Acknowledgements
The authors would like to thank Dr. J. Paelinck, Dr. S. Gorman and Dr. Acharya for their valuable comments.

6. REFERENCES

[1] H Bennett, P Gacs Ming L, Paul Vitanyi and W Zurek W; "Information Distance", **IEEE Transactions on Information Theory**, Vol 44 , No. 4, July 1998., pp. 1407-1423.

[2] G. J. Chaitin, "On the length of programs for computing finite binary sequences", **J. Assoc. Comput. Mach**. Vol. 13, 1966, pp 547-569.

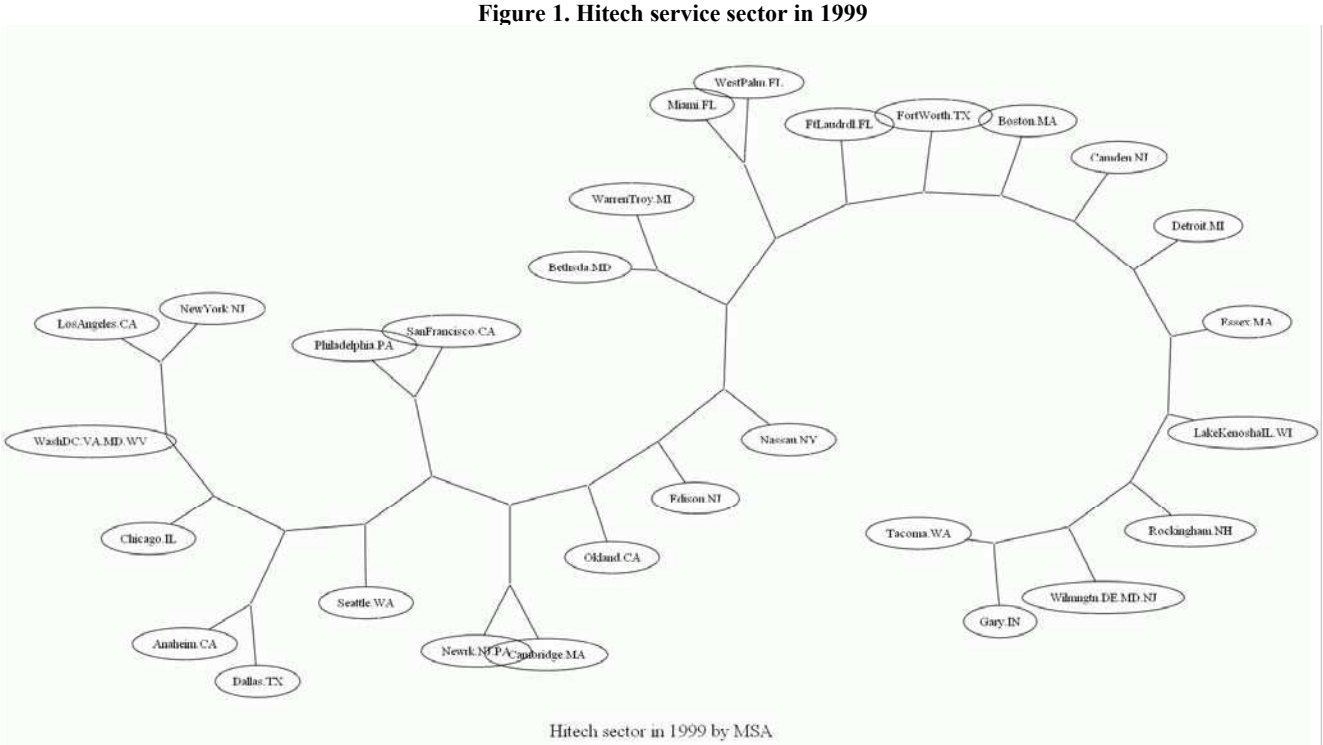
[3] Baronchelli, E. Caglioti, V. Loreto, "Artificial sequences and complexity measures", **IOP Electronic Journals**, http://www.iop.org/EJ/article/1742-5468/2005/P04002/jstat5_04_.

[4] Ming Li, Xin Chen, Xin Li, Bin Ma and Paul Vitanyi, "The Similarity Matrix", **IEEE Transactions on Information Theory**, Vol 50, No. 12, December 2004, pp. 3250-3264.

[5] Ming Li and Paul Vitanyi, **An Introduction to Kolmogorov Complexity and its Applications** in Graduate Texts in Computer Sciencel Editors: David Gries and Fred Schneider, 2nd Edition, Springer, NY, 1997.

[6] Jacob Zip and Neri Merhav, "A Measure of Relative Entropy between individual sequences with application to Universal Classifiction", **IEEE Transactions on Information Theory**, Vol. 39, No. 4, July 1993, pp. 12701279.

Figure 1. Hitech service sector in 1999



Hitech sector in 1999 by MSA

Figure 2. Hitech Service sector in 2006

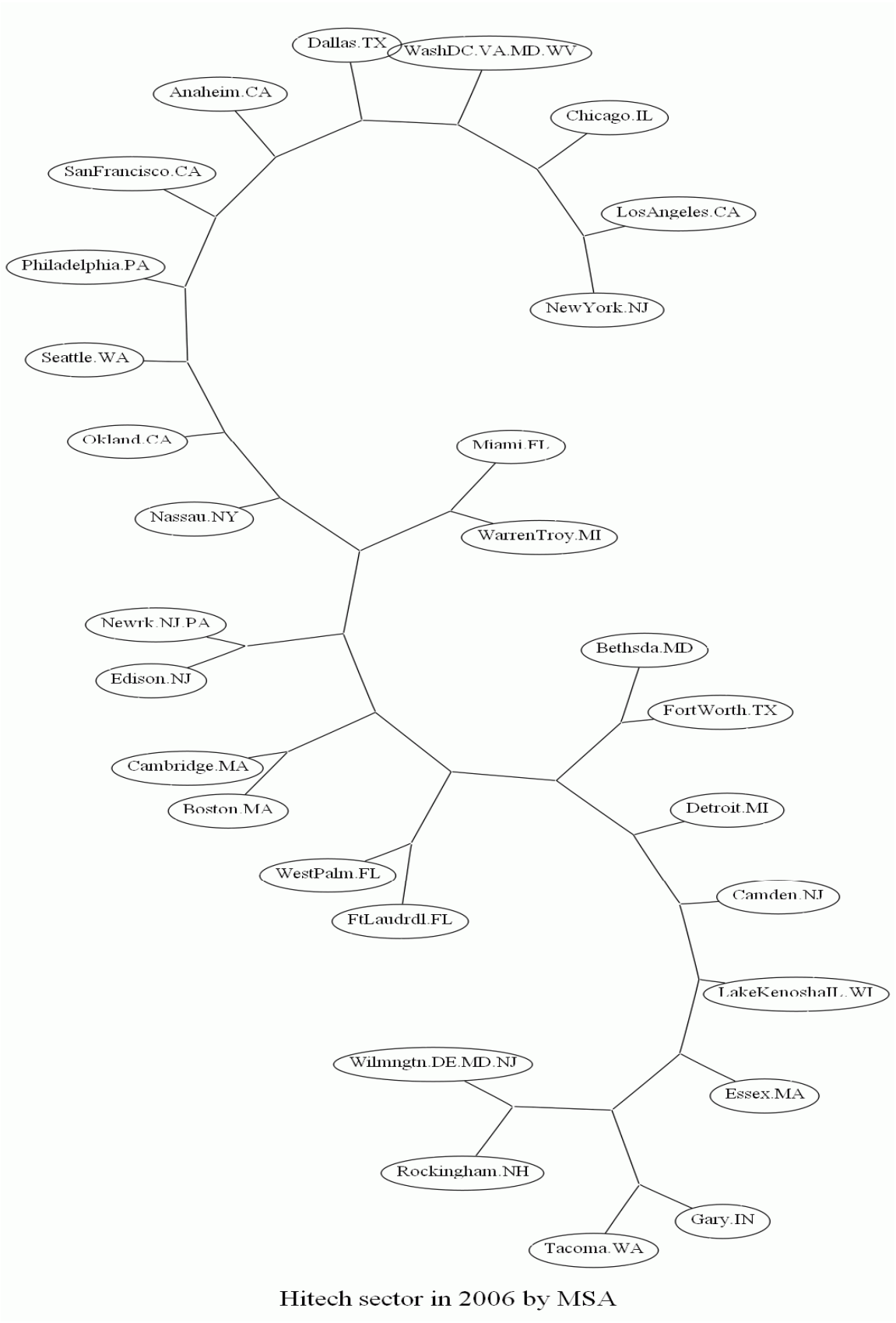


Figure 3. Hitech service sector in 2002

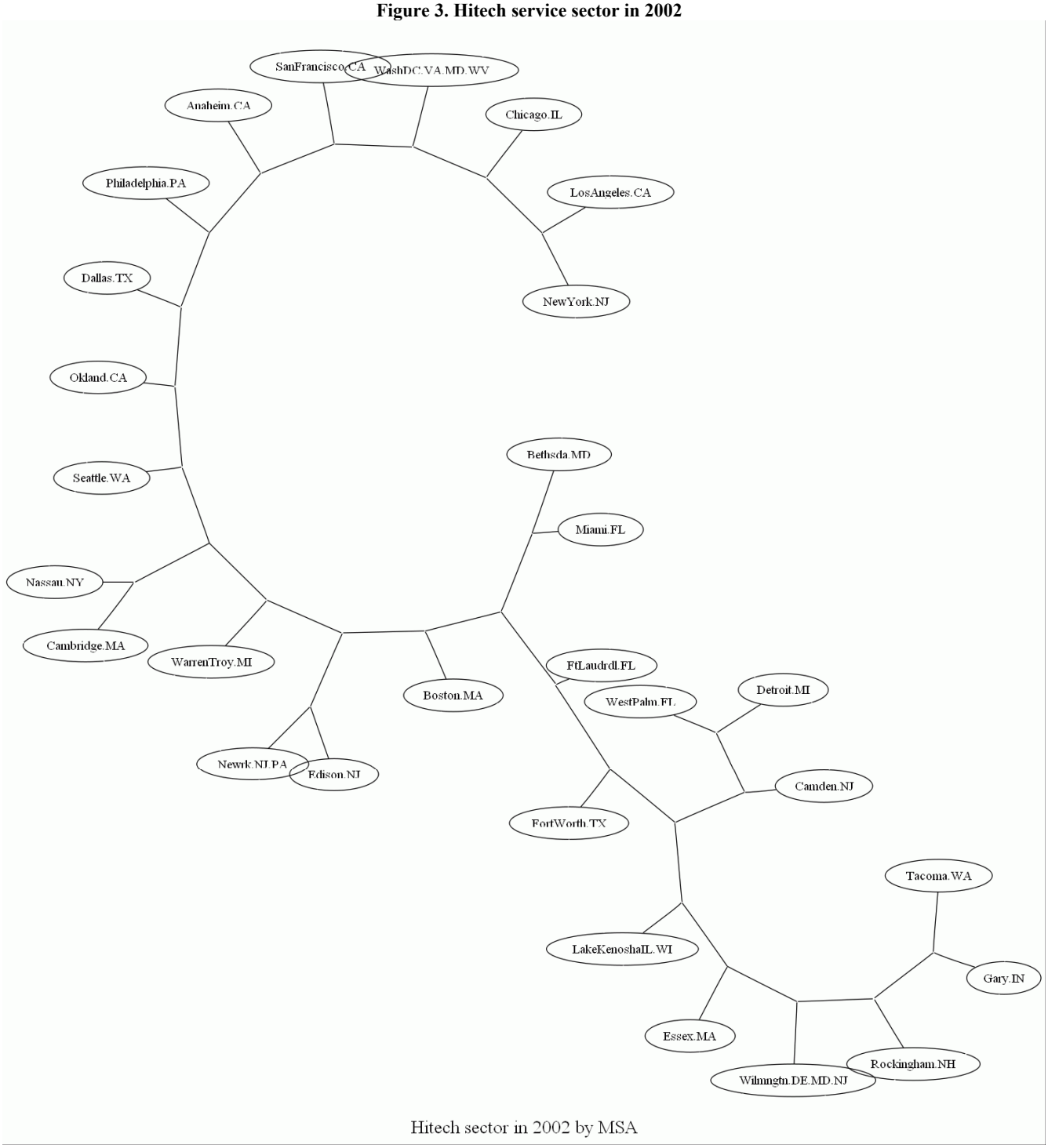


Figure 4. Census 2000 Demographics