Problems on Sequences: Information Theory & Computer Science Interface

J. C. Kieffer, W. Szpankowski, and E-H. Yang

I. Introduction

Recent years have seen a proliferation of research in "Problems on Sequences" which has benefited from the interplay between information theory and computer science – each of these two fields has had an impact upon the other in providing design paradigms and in providing ways of obtaining performance bounds. Because of the continued expansion of this research interest, it was deemed to be an opportune time for this special issue, devoted to this interface.

The papers in this Special Issue well illustrate this theme. We have organized our discussion of these papers into the following subareas:

- Analysis of Algorithms in Problems on Sequences
- Data Structures in Problems on Sequences
- Complexity Issues
- Estimation/Prediction in Problems on Sequences

Some of the contributed papers well exhibit more than one of these four subareas. Our classification was made on what we felt was the principal emphasis of each paper.

II. Analysis of Algorithms in Problems on Sequences

Analysis of algorithms, a field established by D. Knuth, consists of powerful techniques for analyzing the asymptotic performance of algorithms on sequences. Complex analysis is a prime tool in the analysis of algorithms "toolbox". Two papers in this special issue, by Jacquet and Szpankowski [8] and Conrad and Mitzenmacher [3], are prime illustrations of the use of analysis of algorithms to attack problems on sequences.

A. Asymptotics of Minimax Redundancy

In the 1970's, it was established that the worst case block minimax redundancy of an individual sequence of length n with respect to the family of r-th order Markov information sources on a finite alphabet of size m can be expanded to first order as

$$(1/2)m^r(m-1)\log_2 n + O(1). \tag{1}$$

Finally, after many years, Jacquet and Szpankowski [8] have obtained the second order asymptotics in (1). They determine an explicit constant $C(m, r)$ depending on m, r such that the minimax redundancy just discussed can be expanded more sharply as

$$(1/2)m^r(m-1)\log_2 n + C(m,r) + o(1). \tag{2}$$

The techniques they employ are so powerful that it is believable that with a little more work the third order $o(1)$ term in (2) can also be elucidated. Their approach involves getting exact and asymptotic formulas for the number of individual sequences belonging to a Markov type class; they do this in a clever way by reducing the problem to the more managable problem of counting Eulerian paths in a multigraph.

B. Power Laws in Language Models

Let there be given a memoryless source with finite alphabet A of size $|A| > 1$, and let A^+ be the set of all nonempty finite strings over A. For each string $x \in A^+$, let $p(x)$ be the probability with which the source generates x. (To avoid trivialities, we suppose that $p(x) > 0$ for all $x \in A$.) Let

$$x_1, x_2, x_3, \cdots$$

be a ranking of the elements of A^* in which

$$p(x_1) \geq p(x_2) \geq p(x_3) \geq \cdots.$$

Using complex analysis tools from the analysis of algorithms, Conrad and Mitzenmacher [3] establish a power law for the sequence $\{p(x_j)\}$: that is, they find positive constants c_1, c_2, α for which

$$c_1 j^{-\alpha} \leq c_2 j^{-\alpha}$$

for all j sufficiently large. In [3], this power law result is offered in the context of an artifical language "monkey typing" model introduced by Mandelbrot as a counterpoint to natural language models. We point out here another interpretion of this problem in the context of Tunstall codes for the given source. For each j, the Tunstall algorithm builds a rooted tree with $j|A| - (j-1)$ leaf vertices, j non-leaf vertices, and j outgoing edges per non-leaf vertex, such that at each non-root, non-leaf vertex there is a unique string belonging to $\{x_1, x_2, \cdots, x_j\}$ that is represented at that vertex by following the path from root to that vertex and writing down a label from A on each edge along the way. This rooted tree is used to build the variable-length to fixed-length Tunstall code having $j|A|-(j-1)$ binary codewords for the given source; this is the lossless variable-length to fixed-length code having this many codewords that yields minimal expected compression rate. The Conrad-Mitzenmacher result potentially has implications regarding the asymptotic behavior of Tunstall code performance.

III. Data Structures in Problems on Sequences

Papers [7], [11], [18], [14] [17] illustrate the use of data structures such as grammars, graphs, trees, and patterns in connection with problems on sequences.

A. *Grammar Based Coding*

Grammars are the key data structures employed in the papers by He and Yang [7] and Kieffer and Yang [11] appearing in this issue. Specifically, these papers concern grammar based lossless source coding, in which the data sequence to be compressed is represented as a context-free grammar whose language consists uniquely of that data sequence.

We first discuss the He-Yang paper [7]. This paper reconsiders the irreducible grammar based lossless source codes introduced in [10] from a new point of view. We put forth a general framework useful for understanding the contribution of this paper. Let A be a finite source alphabet, and let W be a complete prefix set of nonempty finite strings over A. (*Prefix* means no word in W is a prefix of any other word in W, and *complete* means every infinite sequence from A has a prefix in W.) Then, each string x in A^+ has a unique parsing

$$(\pi_1, \pi_2, \cdots, \pi_t),$$

where each π_i for $i < t$ belongs to W and π_t has no prefix in W. One can attempt to define a k-th order empirical entropy of x relative to W as

$$h_k(x|W) = \inf_p \min_{s_0} \left\{ -\log_2 \sum_{s_1, s_2, \cdots, s_t} \prod_{j=1}^{t} p(\pi_j, s_j | s_{j-1}) \right\},$$

where, for some finite set S with k elements, the following are true:

- The minimum is over all $s_0 \in S$, and the joint summation is over all $s_1, s_2, \cdots, s_t$ in which each s_i belongs to S.

- p represents a conditional probability function $\{p(\pi, s'|s)\}$ defined for s, s' in S and π in W (with $p(\pi_t, s_t|s_{t-1})$ denoting the summation of $p(\pi, s_t|s_{t-1})$ over all $\pi \in W$ such that π_t is a prefix of π).

There is something problematic with the definition of $h_k(x|W)$: the conditional probability functions p that are allowed must be restricted appropriately. For one thing, as the length of x goes to infinity, it is desired that the number of phrases t also go to infinity. Also, the effect of the last phrase π_t on $h_k(x|W)$ should not be too great. (The precise manner in which the functions p must be restricted will be clear to the reader from a reading of [7].) Having appropriately defined $h_k(x|W)$, one can define the n-th order maximal redundancy of grammar based codes relative to this empirical entropy as

$$Red_n(k|W) = \max_L \max_{x \in A^n} \{L(x) - h_k(x|W)\}$$

where the outer maximum is over codeword length functions L of the irreducible grammar based codes that were introduced in the paper [10]. He and Yang prove that

$$Red_n(k|W) = O\left(\frac{n \log_2 \log_2 n}{\log_2 n}\right)$$

for all positive integers k, where W is the prefix set associated with the run-length parsing of strings over A. This result is stronger than the grammar based coding theorem proved in [10]. We explain why this is so. Suppose we interpret $h_k(x|W)$ as $\inf_\mu\{-\log_2 \mu(x)\}$, where μ ranges over a class of alphabet A "reference sources". Then, we can say that He and Yang have extended the class of reference sources from the class of k-state finite-state sources on A (the class of sources considered in [10]) to a class of sources including all k-state renewal processes on A. The information theory associated with renewal processes has been of much interest in recent years; the paper of He and Yang provides a useful addition to that literature. As a final remark, we note that the coding theorem proved in [7] very likely extends to other complete prefix sets W than the run-length one. By allowing W to range over other complete prefix sets, one obtains classes of reference sources which follow a "variable-length shift" model. Variable-length shift source models were considered in the late 1970's and early 1980's in the development of the theory of asymptotically mean stationary processes; He and Yang have made an important extension of the variable-length shift source model in allowing the complete prefix set W to be possibly countably infinite rather than finite.

Like the He-Yang paper [7], the Kieffer-Yang paper [11] also uses context-free grammars as the key data structure. The difference between these two papers is that paper [11] designs a new class of grammar based codes, instead of reconsidering a previously considered class. We spell out the background needed to understand [11]. Let $x = (x_1, x_2, \cdots, x_n)$ be a data sequence over the finite alphabet A, and let $y = (y_1, y_2, \cdots, y_n)$ be a sequence over the finite alphabet B which is a coarsening of x, meaning that there is a function from A into B that carries x into y when applied to x coordinate by coordinate (in obtaining coarsenings of different data sequences, this "coarsening function" is allowed to change arbitrarily). It is assumed that an encoder possesses x and knows y, and that a decoder possesses y and knows only y. The lossless refinement source coding problem is the problem of determining a universal procedure via which the encoder sends code bits to the decoder that enable the decoder to "refine" y and obtain x. Kieffer and Yang [11] determine such a universal procedure via a grammar based approach, which we can summarize as follows. The encoder first represents x via a context-free grammar G_x such that the language generated by G_x is $\{x\}$. Secondly, the encoder sends the decoder $o(n)$ bits which enable the decoder to build a context-free grammar G_y representing y for which there is a mapping carrying G_x into G_y that preserves the production rule structure. Finally, the encoder sends to the decoder some additional bits (these are the key bits generated by the algorithm) which the decoder uses to "refine" G_y to obtain G_x, from which the decoder "grows" x. There is introduced a notion of the conditional entropy of the grammar G_x given the grammar G_y, which proves to be approximately equal to the number of key bits that are transmitted. This grammar based algorithm is universal, yielding $O(1/\log_2 n)$ redundancy per sample with respect to any finite class of finite-state refinement compression schemes when maximized over all pairs (x, y). No lossless refinement compression scheme other than the one in [11] is presently known which has redundancy per sample performance decaying faster than $O(1/\log_2 n)$.

B. Compression of Words Constrained by a Graph

Let A be a finite source alphabet, and for each positive integer n, let $\sim_n$ be an equivalence relation on n. We suppose that the family $\{\sim_n\}$ of equivalence relations satisfies the following subadditivity property: For each pair of positive integers m, n, if $x, x' \in A^m$ are equivalent under $\sim_m$ and $y, y' \in A^n$ are equivalent under $\sim_n$, then $x * y, x' * y' \in A^{m+n}$ are equivalent under $\sim_{m+n}$. (The $*$ operation denotes left-to-right concatenation of strings.) Suppose A-valued random variables $X_1, X_2, X_3, \cdots$ are generated by a stationary ergodic source with alphabet A, and for each n, let $\tilde{X}^n$ be the random variable which is the $\sim_n$ equivalence class of $(X_1, X_2, \cdots, X_n)$. First Körner and then Kieffer laid down successive steps in a source coding theory for the sequence of variables $\{\tilde{X}^n\}$ (see the references in in [18]); in this theory, one reconstructs at the decoder not $(X_1, X_2, \cdots, X_n)$ but its equivalence class $\{\tilde{X}^n\}$ instead, or, equivalently, one reconstructs one fixed sequence belonging to this equivalence class. One consequence of this theory is that the AEP holds for the sequence $\{\tilde{X}^n\}$, namely, there is a nonnegative constant H (less than or equal to the entropy rate of the given stationary ergodic source) such that

$$-n^{-1} \log_2 P(\tilde{X}^n) \to H \ \text{ almost surely } (n \to \infty), \tag{3}$$

where $P(\tilde{X}^n)$ is the usual probability function which takes the value $P(\tilde{X}^n = y)$ if $\tilde{X}^n$ takes equivalence class y as its value. (This AEP is a corollary of Theorem 2 of [9].) Motivated by the Kolmogorov complexity concept, define $C(\tilde{X}^n)$ to be the length of the shortest binary program that can be run on a universal computer to yield $\tilde{X}^n$. In the paper [18] in this issue, Savari proves the new result that

$$n^{-1} C(\tilde{X}^n) \to H \ \text{ almost surely } (n \to \infty), \tag{4}$$

(This can be regarded as a universal coding result since the left side of (4) does not depend on the source, although the universal code given by this result is nonconstructible.) This result can be proved by relating the left side of (4) to the left side of (3).

By specializing $\{\sim_n\}$ to a specific set of equivalence relations motived by computer science considerations, Savari is sometimes able to find bounds for H or even to compute H; this is the major contribution of [18]. In this part of Savari's work, a finite undirected graph G with vertex set A is used, and two blocks x,y from A^n are delared to be equivalent under $\sim_n$ if there is a finite sequence of blocks from A^n, starting with x and ending with y, such that each block after the first one is obtained by transposing two consecutive entries of the preceding block that form an edge in G. The structure of the graph G dictates what Savari is able to learn about the value of H (which is called "interchange entropy" in this context).

C. Faster Tree Source Coding

A recursively generated conditional probability function gives rise to an arithmetic coder which may yield efficient compression but be slow in a standard implementation. We suppose that the conditional probability functions arise from a tree source model. For encoder/decoder processing of the current source sample, one starts at the root of the tree with the first past source sample and follows a path generated by moving backward in the past until a tree leaf is reached, which is labelled by a state that determines the conditional probability function to be used for the current sample. Once the current sample is processed, the next source sample then plays the role of "current sample," and its past is obtained by appending the previous current sample to the beginning of the previous past, the tree then again being used to find the conditional probability function that is now to be used via following a root-to-leaf path. This procedure is slower than it need be because the paths followed in the tree (the "contexts") in processing one source sample to the next have a common segment that is traversed twice. A more computationally efficient method could be devised which would achieve a speedup by taking advantage of this common segment. Martín et al., in their paper [14] in this issue, obtained a faster implementation for tree source coding. To accomplish this, they use a combination of techniques involving several key ideas, some of which are

- use of a finite-state machine (called the FSM closure of the tree model) to model the above-described transition between contexts;
- compactification of the tree in which some edges can be labeled with a string consisting of more than one letter from the source alphabet; and
- use of suffix links, a tool used in suffix tree generation methods (such as in Ukkonen's algorithm; see [19] and Chapter 6 of [6]).

Martín et al. [14] are thus able to obtain linear encoding/decoding time in the semi-predictive approach to the Context Algorithm, an algorithm previously established to be a universal lossless coding scheme [21].

D. Universal Compression over an Unknown Alphabet

A *pattern* is a sequence whose first entry is 1, whose second distinct entry is 2 (reading left to right), whose third distinct entry is 3, etc. For example, the patterns of length three are

$$111, 112, 122, 123.$$

Let $x = (x_1, x_2, \cdots, x_n)$ be a sequence of length n over any alphabet. Let k be the number of distinct symbols appearing in x, i.e., k is the cardinality of the set $\{x_1, x_2, \cdots, x_n\}$. Then, there is a natural pattern of length n with alphabet $\{1, 2, \cdots, k\}$ (called the pattern induced by x) which reflects the relative positions of the entries of x. For example, the pattern induced by the sequence (d, c, c, a, b, c, d, a) is $(1, 2, 2, 3, 4, 2, 1, 3)$. In general, it is impossible to losslessly encode the set of all sequences of length n over an arbitrary alphabet (because the alphabet may be uncountable and there are only countably many binary codewords that can be used by the lossless encoder). However, it is possible to losslessly encode the set of patterns induced by these sequences, because this is a countable set. Each memoryless probability distribution on the set of sequences of length n over an arbitrary alphabet can be carried over to a probability distribution on the set of patterns of length n by the function which maps each sequence into its induced pattern. In the paper by Orlitsky et al. in this issue [17], a remarkable $O(\sqrt{n})$ bound is proved on the worst case block redundancy of this family of probability distributions on the set of patterns of length n, and a linear time sequential lossless pattern coder achieving worst case block redundancy bound $O(n^{2/3})$ is given. These strong results suggest several open questions for the future:

- Can a linear time pattern coder be found which closes the gap between the bound $O(n^{2/3})$ and the bound $O(\sqrt{n})$?
- What are the implications for grammar based coding? (Grammar based codes encode the concatenation y of the right sides of the production rules of a grammar representing a

data sequence; the pattern of y is the key feature of y since y consists mostly of variables of the grammar rather than terminal symbols and it does not matter what notation is used to denote these variables —they are "dummy variables".)

- What are the implications for the lossless compression of data objects with "rich alphabets"? (An 8-bit per pixel 256×256 gray level still image has rich alphabet $A = \{1, 2, 3, \cdots, 255\}$, meaning that the alphabet size is too large compared to the size of the image for there to be sufficient repetitions of alphabet strings that can be exploited by a finite-state encoder designed for alphabet A. In other words, for alphabet $A = \{1, 2, \cdots, 255\}$, you'd have to have an $N \times N$ image with $N >> 256$ in order to expect the kind of asymptotic performance that such finite-state encoders are capable of achieving. Pattern coding might help since with 8 bits per pixel, an image could be recovered from its pattern with only an additional $8 * 256 = 2048$ bits, increasing the compression rate in a 256×256 image by only $2048/(256)^2 = 1/32$ bit/pixel.)

IV. Complexity Issues

A. NP-Hard Lossless Source Coding Problems

Garey et al. [5] showed that optimal lossy source coding can be an NP-complete problem. In this issue, M. Mitzenmacher [15] investigates how hard certain lossless source coding problems are. Specifically, he shows that the following problems are NP-hard:

- Given a finite set of data files, find the pair of Huffman dictionaries that minimizes the total number of code bits in the compressed files taken together.
- Find multiple preset dictionaries for LZ77 coding [23].

B. Computational Complexity in Image Retrieval

Databases of various types (e.g., text-only databases, multimedia databases, bio-informatics databases) have proliferated in recent years. One of the concerns of computer science is the efficient management of tasks associated with databases. Retrieval of data from a database is one such task. For text-only databases, there are some well established approaches to the data retrieval problem. On the other hand, for multimedia databases, more research into data retrieval methods needs to be done. Vasconcelos, in his paper [20] in this issue, concentrates on the problem of image retrieval. A probabilistic formulation of the image retrieval problem is employed, in which there are finitely many database image classes, each governed by a probabilistic model, as well as a query image class governed by a proabilistic model. In this framework, image retrieval in response to a query or sequence of queries can be regarded in a Bayesian decision theory context. Maximum aposteriori probability retrieval (MAP retrieval) is the retrieval method yielding the minimum probability of image retrieval error. Unfortunately, implementation of the MAP method involves an optimization that might entail a high degree of computational complexity. The paper [20] addresses this problem by deriving computationally efficient ways to evaluate the MAP function. Kullback-Leibler divergence is a principal tool in the analysis. It is interesting that Kullback-Leibler divergence plays a role in this computer science problem as well as in information theory, exhibiting another instance of the interface with which this issue is concerned.

C. Computational Complexity of MAP Decoding

As we have just discussed, the paper of Vasconcelos is concerned with the MAP method in image retrieval. The MAP method is also a well known method in communication engineering, where it is called MAP decoding. MAP decoding is implemented at the output of a channel to estimate a transmitted data sequence that is corrupted by channel noise; the MAP decoder output is chosen as that data sequence which is most likely to have been transmitted, given the observed channel output sequence. The paper of Wu et al. in this issue [22] is concerned with reducing the computational complexity associated with a MAP decoder. Specifically, the paper [22] addresses MAP decoding of Markov sequences corrupted by transmission over a memoryless binary channel that produces independent substitution and erasure errors. In this context, MAP estimation becomes the problem of finding a longest path in a weighted acyclic directed graph. A matrix formulation of this computational problem is presented, which, for a Gauss-Markov input sequence, reduces the computational complexity associated with the MAP decoder. The authors point out that their algorithmic approach, whose origins can be traced back through the computer science literature, could be potentially useful in other MAP estimation problems in communication engineering.

D. Program Plagiarism Detection Systems

The originators of software programs need to have good ways of detecting whether someone has modified one of their programs for their own use. A system for accomplishing this task is called a program plagiarism detection system. Chen et al., in their paper [2] in this issue, put forth a method for designing such a system. The idea is to use a nonnegative distance function $d(\cdot,\cdot)$ defined for pairs of programs, such that as $d(x,y)$ becomes closer to zero for two programs x, y, the more likely it is that one of the two programs is a plagiarized version of the other. In an earlier work, such a distance function $d^*(\cdot,\cdot)$ was put forth, based on Kolmogorov complexity. The distance function $d^*(\cdot,\cdot)$ was shown to satisfy a universality property in that it minorized computable distance functions of a certain type (see [2] for the precise sense of this minorization). Roughly speaking, if a computable distance function $d(\cdot,\cdot)$ is such that $d(x,y)$ is small for a pair of programs x, y, then $d^*(x,y)$ will also be small; that is, if any distance measure detects plagiarism, then so does $d^*(\cdot,\cdot)$. This universality result indicated that the distance function $d^*(\cdot,\cdot)$ should be the best distance function to use in detecting program plagiarism. Unfortunately, the distance function $d^*(\cdot,\cdot)$ is noncomputable (because Kolmogorov complexity is noncomputable). In the paper [2], a way around this problem is investigated in which $d^*(\cdot,\cdot)$ is approximated by a computable distance function. (A compression algorithm is used to heuristically approximate Kolmogorov complexity in the definition of $d^*(\cdot,\cdot)$.) Using this computable distance function, Chen et al. design a practical plagiarism detection system; results of experiments are reported in [2] that show advantages of this system over other practical plagiarism detection systems.

V. Estimation/Prediction in Problems on Sequences

A. Entropy Estimator Based on Block Sorting

The paper by Cai et al. [1] presents a sequence of entropy estimators which operate on longer and longer initial segments of an infinite individual sequence over a finite alphabet A. A universal estimation result is proved in the following sense: almost surely, when applied to the sequence of random outputs generated by any stationary ergodic source with alphabet A, the proposed sequence of entropy estimators yields entropy estimates converging to the entropy rate of the source. Specializing to a stationary ergodic finite-state Markov source, a result on the speed of convergence of the entropy estimates is obtained. The Burrows-Wheeler Block Sorting Transform is a key feature of these entropy estimators: the entropy estimate based on an initial source segment is formed in a natural way from the sequence which is the Burrows-Wheeler transform of that segment. A question not addressed in [1] is how the performance of the universal sequence of entropy estimators of this paper compares to the performance of other universal sequences of entropy estimators. This is an open question for the future.

B. Pseudorandomness via Prediction Theory

The question concerning whether one can construct a truly random individual sequence has attracted the interest of many people (see [12], Section 1.5.1 of [13], and the references in [16]). The answer to the question (due to Kolmogorov complexity considerations) is that the very fact that a pseudorandom sequence is constructible dictates that the sequence can obey only a limited number of randomness properties. Within this limitation, many constructions of pseudorandom sequences have been proposed. Prediction theory has been one tool that has been used to obtain pseudorandom individual sequences. For example, Ehrenfeucht and Mycielski [4] construct an infinite pseudorandom binary sequence by making each subsequent entry to be the complement of what would be predicted by a certain universal predictor applied to the previous entries; the pseudorandomness properties of their sequence have of this date not yet been completely worked out. Nobel, in his paper [16] in this issue, also investigates the problem of pseudorandom individual sequence construction from the prediction theory point of view. He formally defines a memoryless sequence to be an individual sequence of real numbers on which no continuous Markov prediction scheme can outperform the best constant predictor under square loss. He then shows that memoryless sequences obey asymptotic behavior that we would expect of "typical sequences" generated by a memoryless information source (a law of large numbers and a central limit theorem are established). He also discusses the construction of memoryless sequences.

C. *Feder's Paper?*

VI. Conclusions

Below is an incomplete list of topics illustrating some of the ways in which information theory (IT) and computer science (CS) impact each other.

- Analysis of algorithms applied to analyze algorithms for data compression, prediction, pseudorandom number generation, and classification.
- Computational or descriptive complexity applied to examine the complexity/performance tradeoff in lossless/lossy data compression.
- CS data structures (grammars, trees, and graphs) applied to design codes for sequences.
- Exact and approximate pattern matching techniques applied to context modeling/coding.
- Entropy, mutual information and information distance concepts applied to problems of computational biology and bioinformatics.
- IT concepts applied to provide bounds in computational learning theory.
- Applications of IT to computational linguistics.
- Applications of IT to information retrieval (especially data mining).

Many of these topics were well represented in the papers in this special issue. The reader who reads elsewhere will find examples of papers in other journals categorizable as being in the IT/CS interface. The range of topics at the interface is likely to continue to grow with time, driven by society's need to handle data more efficiently and more rapidly. Of course, one area at the interface that is going to continue to undergo an explosion in growth for many years to come is bioinformatics. In summary, researchers who choose the IT/CS interface as their research area will find a wealth of interesting problems.

References

[1] H. Cai, S. Kulkarni, and S. Verdu, "Universal Estimation of Entropy and Divergence Via Block Sorting," *IEEE Trans. Inform. Theory*, this issue.

[2] X. Chen, M. Li, B. McKinnon, and A. Seker, "A Theory of Uncheatable Plagiarism Detection and Its Practical Implementation," *IEEE Trans. Inform. Theory*, this issue.

[3] B. Conrad and M. Mitzenmacher, "Power Laws for Monkeys Typing Randomly: The Case of Unequal Probabilities," *IEEE Trans. Inform. Theory*, this issue.

[4] A. Ehrenfeucht and J. Mycielski, "A Pseudorandom Sequence—How Random is It?," *Amer. Math. Monthly*, Vol. 99, pp. 373–375, 1992.

[5] M. Garey, D. Johnson, and H. Witsenhausen, "The Complexity of the Generalized Lloyd-Max Problem," *IEEE Trans. Inform. Theory*, Vol. 28, pp. 255–256, 1982.

[6] D. Gusfield, *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, Cambridge, UK, 1997.

[7] D-K. He and E-H. Yang, "Performance Analysis of Grammar-Based Codes Revisited," *IEEE Trans. Inform. Theory*, this issue.

[8] P. Jacquet and W. Szpankowski, "Markov Types and Minimax Redundancy for Markov Source," *IEEE Trans. Inform. Theory*, this issue.

[9] J. Kieffer, "An ergodic theorem for constrained sequences of functions." *Bull. Amer. Math. Soc.*, Vol. 21, pp. 249–254, 1989.

[10] J. Kieffer and E-H. Yang, "Grammar-based codes: a new class of universal lossless source codes," *IEEE Trans. Inform. Theory*, vol. 46, pp. 737–754, 2000.

[11] J. Kieffer, "Grammar-Based Lossless Universal Refinement Source Coding," *IEEE Trans. Inform. Theory*, this issue.

[12] A. Kolmogorov and V. Uspenskii, "Algorithms and Randomness," *Theory of Probability and its Applications*, Vol. 32, pp. 389–412, 1987.

[13] M. Li and P. Vitanyi, *An Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, New York, 1993.

[14] A. Martín, G. Seroussi, and M. Weinberger, "Linear Time Universal Coding and Time Reversal of Tree Sources via FSM Closure," *IEEE Trans. Inform. Theory*, this issue.

[15] M. Mitzenmacher, "On the Hardness of Finding Optimal Multiply Preset Dictionaries," *IEEE Trans. Inform. Theory*, this issue.

[16] A. Nobel, "Some Stochastic Properties of Memoryless Individual Sequences, *IEEE Trans. Inform. Theory*, this issue.

[17] A. Orlitsky, P. Santhanam and J. Zhang, "Universal Compression of Memoryless Sources over Unknown Alphabets," *IEEE Trans. Inform. Theory*, this issue.

[18] S. Savari, "Compression of Words over a Partially Commutative Alphabet," *IEEE Trans. Inform. Theory*, this issue.

[19] E. Ukkonen, "On-line construction of suffix-trees," *Algorithmica*, Vol. 14, pp. 249–260, 1995.

[20] N. Vasconcelos, "On the Efficient Evaluation of Probabilistic Similarity Functions for Image Retrieval," *IEEE Trans. Inform. Theory*, this issue.

[21] M. Weinberger, A. Lempel, and J. Ziv, "A Sequential Algorithm for the Universal Coding of Finite Memory Sources," *IEEE Trans. Inform. Theory*, Vol. 38, pp. 1002–1014, 1992.

[22] X. Wu, S. Dumitrescu, and Z. Wang, "Monotonicity-based Fast Algorithms for MAP Estimation of Markov Sequences over Noisy Channels," *IEEE Trans. Inform. Theory*, this issue.

[23] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, Vol. 23, pp. 337–343, 1977.