Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Discrete Mathematics 21 (2005) 219–225
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

Bridging Lossy and Lossless Compression by Motif Pattern Discovery

Alberto Apostolico, Matteo Comin, Laxmi Parida

[a] *Department of Computer Sciences, Purdue University, Computer Sciences Building, West Lafayette, IN 47907, USA* and *Dipartimento di Ingegneria dell' Informazione, Università di Padova, Padova, Italy. Work performed in part while on leave at IBM T.J. Watson Center.* axa@dei.unipd.it

[b] *Dipartimento di Ingegneria dell' Informazione, Università di Padova, Padova, Italy.* ciompin@dei.unipd.it

[c] *IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA.* parida@us.ibm.com

Abstract

We present data compression techniques hinged on the notion of a *motif*, interpreted here as a string of intermittently solid and wild characters that recurs more or less frequently in an input sequence or family of sequences. This notion arises originally in the analysis of sequences, particularly biomolecules, due to its multiple implications in the understanding of biological structure and function, and it has been the subject of various characterizations and study. Correspondingly, motif discovery techniques and tools have been devised. This task is made hard by the circumstance that the number of motifs identifiable in general in a sequence can be exponential in the size of that sequence. A significant gain in the direction of reducing the number of motifs is achieved through the introduction of *irredundant* motifs, which in intuitive terms are motifs of which the structure and list of occurrences cannot be inferred by a combination of other motifs' occurrences. Although suboptimal, the available procedures for the extraction of some such motifs are not prohibitively expensive. Here we show that irredundant motifs can be usefully exploited in lossy

1571-0653/$ – see front matter
doi:10.1016/j.endm.2005.07.029

compression methods based on textual substitution and suitable for signals as well as text. Actually, once the motifs in our lossy encodings are disambiguated into corresponding lossless codebooks, they still prove capable of yielding savings over popular methods in use. Preliminary experiments with these fungible strategies at the crossroads of lossless and lossy data compression show performances that improve over popular methods (i.e. GZip) by more than 20% in lossy and 10% in lossless implementations.

Traditionally data compression methods are partitioned into lossy and lossless. Typically, lossy compression is applied to images and more in general to signals susceptible to some degeneracy without lethal consequence. On the other hand, lossless compression is used in situations where fidelity is of the essence, which applies to high quality documents and perhaps most notably to textfiles. Lossy methods rest mostly on transform techniques whereby, for instance, cuts are applied in the frequency, rather than in the time domain of a signal. By contrast, lossless textual substitution methods are applied to the input in native form, and exploit its redundancy in terms of more or less repetitive segments or patterns.

NARRATIVE:

Let $s = s_1 s_2 ... s_n$ be a *string* of length $|s| = n$ over an alphabet Σ. A character from Σ, say σ, is called a *solid* character and '.' is called a "don't care" character. A *motif* is any element of Σ or any string on $\Sigma \cdot (\Sigma \cup \{.\})^* \cdot \Sigma$.

Definition 1.1 (k-Motif m, Location list $\mathcal{L}_m$) Given a string s on alphabet Σ and a positive integer k, $k \leq |s|$, a string m on $\Sigma \cup$ '.' is a motif with location list $\mathcal{L}_m = (l_1, l_2, \ldots, l_q)$, if all of the following hold: (1) $m[1], m[|m|] \in \Sigma$, (2) $q \geq k$, and (3) there does not exist a location l, $l \neq l_i$, $1 \leq i \leq q$ such that m occurs at l on s (the location list is of maximal size).

Definition 1.2 (Maximal Motif) Let m_1, m_2, ..., m_k be the motifs in a string s. A motif m_i is maximal in composition if and only if there exists no m_l, $l \neq i$ with $\mathcal{L}_{m_i} = \mathcal{L}_{m_l}$ and $m_i \preceq m_l$. A motif m_i, maximal in composition, is also maximal in length if and only if there exists no motif m_j, $j \neq i$, such that m_i is a sub-motif of m_j and $|\mathcal{L}_{m_i}| = |\mathcal{L}_{m_j}|$. A maximal motif is a motif that is maximal both in composition and in length.

Requiring maximality in composition and length limits the number of motifs that may be usefully extracted and accounted for in a string. However, the notion of maximality alone does not suffice to bound the number of such motifs. It can be shown that there are strings that have an unusually large

number of maximal motifs without conveying extra information about the input. A maximal motif m is *irredundant* if m and the list $\mathcal{L}_m$ of its occurrences cannot be deduced by the union of a number of lists of other maximal motifs. Conversely, we call a motif m redundant if m (and its location list $\mathcal{L}_m$) can be deduced from the other motifs *without* knowing the input string s. More formally:

Definition 1.3 (Redundant/Irredundant motif) A maximal motif m, with location list $\mathcal{L}_m$, is redundant if there exist maximal sub-motifs m_i, $1 \leq i \leq p$, such that $\mathcal{L}_m = \mathcal{L}_{m_1} \cup \mathcal{L}_{m_2} \ldots \cup \mathcal{L}_{m_p}$, (i.e., every occurrence of m on s is already implied by one of the motifs $m_1, m_2, \ldots, m_p$). A maximal motif that is not redundant is called an irredundant motif

Definition 1.4 (Basis) Given a sequence s on an alphabet Σ, let $\mathcal{M}$ be the set of all maximal motifs on s. A set of maximal motifs $\mathcal{B}$ is called a basis of $\mathcal{M}$ iff the following hold: (1) for each $m \in \mathcal{B}$, m is irredundant with respect to $\mathcal{B} - \{m\}$, and, (2) let $\mathbf{G}(\mathcal{X})$ be the set of all the redundant maximal motifs generated by the set of motifs $\mathcal{X}$, then $\mathcal{M} = \mathbf{G}(\mathcal{B})$.

Theorem 1.5 *Every irredundant 2-motif in s is the meet of two suffixes of s.*

An immediate consequence of Theorem 1.5 is a linear bound for the cardinality of our set of irredundant 2-motifs: by maximality, these motifs are just some of the $n-1$ meets of s with its own suffixes. Thus

Theorem 1.6 *The number of irredundant 2-motifs in a string x of n characters is $O(n)$.*

With its underlying convolutory structure, Theorem 1.5 suggests a number of immediate ways for the extraction of irredundant motifs from strings and arrays, using available pattern matching with or without FFT. The construction used for our experiments must take into account additional parameters related to the density of solid characters, the maximum motif length and minimum allowed number of occurrences. The algorithm follows a *steepest descent* approximation to the optimal solution and is described in the full version of this paper. Each phase of our the paradigm alternates the selection of the pattern to be used in compression with the actual substitution and encoding. In practice, we estimate at $\log i$ the number of bits needed to encode the integer i (we refer to, e.g., [1] for reasons that legitimate this choice). In one scheme (hereafter, $Code_1$) [2], we eliminate all occurrences of m, and record in succession m, its length, and the total number of its occurrences followed by the actual list of such occurrences. Letting $|m|$ denote the length of m, f_m the number of

occurrences of m in the textstring, $|\Sigma|$ the cardinality of the alphabet and n the size of the input string, the compression brought about by m is estimated by subtracting from the $f_m|m|\log|\Sigma|$ bits originally encumbered by this motif on s, the expression $|m|\log|\Sigma|+\log|m|+f_m\log n+\log f_m$ charged by encoding, thereby obtaining: $G(m) = (f_m-1)|m|\log|\Sigma|-\log|m|-f_m\log n-\log f_m$. This is accompanied by a fidelity loss $L(m)$ represented by the total number of don't cares introduced by the motif, expressed as a fraction of the original length. If d such gaps were introduced, this would be:

$$(1)\quad L(m) = \frac{f_m d\log|\Sigma|}{f_m|m|\log|\Sigma|} = \frac{d}{|m|}.$$

Other encodings are possible (see, e.g., [2]). The table 1 is an example of 8-bit grey-level images as a function of the don't care density allowed (last column).

Fig. 1. Compression and reconstruction of images. The original is on the first column, next to its reconstruction by interpolation of two closest solid pixels. Black dots used in the figures of the last column are used to display the distribution of the don't care characters. Compression of "Bridge" at 1/4 and 1/3 (shown here) '.'/char densities yields savings of 6.49% and 17.84% respectively. Correspondingly. 0,31% and 12,50% of the pixels differ from original after reconstruction.

Ziv and Lempel designed a class of compression methods based on the idea of back-reference: while the textfile is scanned, substrings or *phrases* are identified and stored in a *dictionary*, and whenever, later in the process, a phrase or concatenation of phrases is encountered again, this is compactly encoded by suitable pointers or indices [8,10,11]. In view of Theorem 1.5 and of the good performance of motif based off-line compression [4], it is natural to inquire into the structure of ZL and ZLW parses which would use these patterns in lieu of exact strings. Possible schemes along these lines include, e.g., adaptations of those in [9], or more radical schemes in which the innovative add-on inherent to ZLW phrase growth is represented not by one symbol alone, but rather by that symbol plus the longest match with the substring that follows

Table 1
Lossy compression of gray-scale images (1 pixel = 1 byte).

file	file len	GZip len [%compr]	$Codec_2$ [%compr]	$Codec_1$ [%compr]	%Diff gzip	%loss	'.'/ char
bridge	66336	$61657_{[7.05]}$	$60987_{[8.06]}$	$57655_{[13.08]}$	6.49	0.42	1/4
			$60987_{[8,06]}$	$50656_{[23.63]}$	17.84	14.29	1/3
camera	66336	$48750_{[26.51]}$	$47842_{[27.88]}$	$46192_{[30.36]}$	5.25	0.74	1/6
			$48044_{[27.57]}$	$45882_{[30.83]}$	5.88	2.17	1/5
			$47316_{[28.67]}$	$43096_{[35.03]}$	11.60	9.09	1/4
lena	262944	$234543_{[12.10]}$	$226844_{[13.73]}$	$210786_{[19.83]}$	10.13	4.17	1/4
			$186359_{[29.13]}$	$175126_{[33.39]}$	25.33	20.00	1/3
peppers	262944	$232334_{[11.64]}$	$218175_{[17.03]}$	$199605_{[23.85]}$	14.09	6.25	1/4
			$180783_{[31.25]}$	$173561_{[33,99]}$	25.30	20.00	1/3

some previous occurrence of the phrase. In other words, the task of vocabulary build-up is assigned to the growth of (candidate), perhaps irredundant, 2-motifs.

We test the power of ZLW encoding on the motifs produced in greedy off-line schemata such as above. Despite the apparent superiority of such greedy off-line approaches in capturing long range repetitions, one drawback is in the encoding of references, which are bi-directional and thus inherently more expensive than those in ZLW.This requires building a small dictionary that needs to be sent over to the decoder together with the encoded string.

With the dictionary in place, the parse phase of the algorithm proceeds in much the same way as in the original scheme, with the proviso that once a motif is chosen, then all of its occurrences are to be deployed. Decoding is easier. The recovery follows closely the standard ZLW, except for initialization of the dictionary. The only difference is thus that now the decoder receives, as part of the encoding, also an initial dictionary containing all motifs utilized, which are used to initialize the trie. The table below summarize results obtained on gray-scale images (Table 2, 1 pixel = 1 byte), for each case, the compression is reported first for lossy encoding with various don't care densities, then also for the respective lossless completions.

CONCLUSION: Irredundant motifs seem to provide an excellent repertoire of codewords for grammar based compression and syntactic inference of

Table 2
Lossy/Lossless compression of gray-scale images using LZW-like encoding.

File	File len	GZip len	LZW-like lossy	% Diff GZip	% Loss	LZW-like lossless	% Diff GZip	'.'/ car
bridge	66.336	61.657	38.562	37.46	0.29	38.715	37.21	1/4
			38.366	37.78	5.35	42.288	31.41	1/3
camera	66.336	48.750	34.321	29.60	0.00	34.321	29.60	1/6
			34.321	29.60	0.06	34.321	29.60	1/5
			32.887	32.54	6.16	35.179	27.84	1/4
lena	262.944	234.543	120.308	48.71	1.36	123.278	47.44	1/4
			123.182	47.48	7.32	135.306	42.31	1/3
peppers	262.944	232.334	117.958	49.23	1.75	121.398	47.75	1/4
			119.257	48.67	4.45	129.012	44.47	1/3

documents of various kinds. Various completion strategies and possible extensions (e.g., to nested descriptors) and generalizations (notably, to higher dimensions) suggest that the notions explored here can develop in a versatile arsenal of data compression methods capable of bridging lossless and lossy textual substitution in a way that is both aesthetically pleasant and practically advantageous. Algorithms for efficient motif extraction as well as for their efficient deployment in compression are highly desirable from this perspective. In particular, algorithms for computing the statistics for maximal sets of *non-overlapping* occurrences for each motif should be set up for use in gain estimations, along the lines of the constructions given in [6] for solid motifs. Progress in these directions seems not out of reach.

References

[1] A. Apostolico and A. Fraenkel, Robust transmission of unbounded strings using Fibonacci representations, IEEE Trans. Inform. Theory, Vol. 33, No. 2, 238–245, 1987.

[2] A. Apostolico and S. Lonardi, Off-line compression by greedy textual substitution, Proceedings of the IEEE, Vol. 88, No. 11, 1733–1744, 2000.

[3] A. Apostolico and L. Parida, Incremental paradigms of motif discovery, Journal

of Computational Biology, Vol. 11, No. 1, 15-25, 2004.

[4] A. Apostolico and L. Parida, Compression and the wheel of fortune, Proceedings of IEEE DCC Data Compression Conference, Computer Society Press, 143–152, 2003.

[5] A. Apostolico, M. Comin, and L. Parida, Motifs in Ziv-Lempel-Welch clef, Proceedings of IEEE DCC Data Compression Conference, 72–81, Computer Society Press, 72–81, 2004.

[6] A. Apostolico and F.P. Preparata, Data structures and algorithms for the string statistics problem, Algorithmica, 15, 481–494, 1996.

[7] S. DeAgostino and J.A. Storer, On-line versus off-line computation in dynamic text compression, Inform. Process. Lett., Vol. 59, No. 3, 169–174, 1996.

[8] A. Lempel and J. Ziv, On the complexity of finite sequences, IEEE Trans. Inform. Theory, Vol. 22, 75–81, 1976.

[9] I. Sadeh, On approximate string matching, Proceedings of DCC 1993, IEEE Computer Society Press, 148–157, 1993.

[10] J. Ziv and A. Lempel, A universal algorithm for sequential data compression, IEEE Trans. Inform. Theory, Vol. 23, No. 3, 337-343, 1977.

[11] J. Ziv and A. Lempel, Compression of individual sequences via variable-rate coding, IEEE Trans. Inform. Theory, Vol. 24, No. 5, 530-536, 1978.