Minos Garofalakis

Wavelet-Based Approximation Techniques in Database Systems

A growing number of database applications require online, interactive access to very large volumes of data to perform a variety of data analysis tasks. As an example, large Internet service providers (ISPs) typically collect and store terabytes of detailed usage information (NetFlow/SNMP flow statistics and packet-header information) from the underlying network to satisfy the requirements of various network-management tasks, including billing, fraud/anomaly detection, and strategic planning. This data gives rise to massive, multidimensional relational data tables typically stored and queried/analyzed using commercial database engines (e.g., Oracle, SQL Server, DB2). To handle the huge data volumes, high query complexities, and interactive response-time requirements characterizing these modern data analysis applications, the idea of effective, easy-to-compute approximate query answers over precomputed, compact data synopses has recently emerged as a viable solution. Due to the exploratory nature of most target applications, there are a number of scenarios in which a reasonably accurate, fast approximate answer over a small-footprint summary of the database is actually preferable over an exact answer that takes hours or days to compute. For example, during a drill-down query sequence in ad-hoc data mining, initial queries in the sequence frequently have the sole purpose of determining the truly interesting queries and regions of the database. Providing fast approximate answers to these initial queries gives users the ability to focus their explorations quickly and effectively, without consuming inordinate amounts of valuable system resources. Of course, the key behind such approximate techniques for dealing with massive data sets lies in the use of appropriate data-reduction techniques for constructing compact synopses that can accurately approximate the important features of the underlying data distribution.

Wavelets are a useful mathematical tool for hierarchically decomposing functions in ways that are both efficient and theoretically sound. Broadly speaking, the wavelet transform of a function consists of a coarse overall approximation together with detail coefficients that influence the function at various scales. The wavelet transform has a long history of successful applications in signal and image processing [9], [10]. Several recent studies have also demonstrated the effectiveness of the wavelet transform (and Haar wavelets, in particular) as a tool for approximate query processing over massive relational tables [2], [5], [6] and continuous data streams [3], [7]. Briefly, the idea is to apply wavelet transform to the input relation to obtain a compact data synopsis that comprises a select small collection of wavelet coefficients. The excellent energy compaction and decorrelation properties of the wavelet transform allow for concise and effective approximate representations that exploit the structure of the data. Furthermore, wavelet transforms can generally be computed in linear time, thus allowing for very efficient algorithms. In this column, we provide a brief overview of recent work and results on wavelet-based approximation techniques for relational database systems.

HAAR WAVELET BASICS

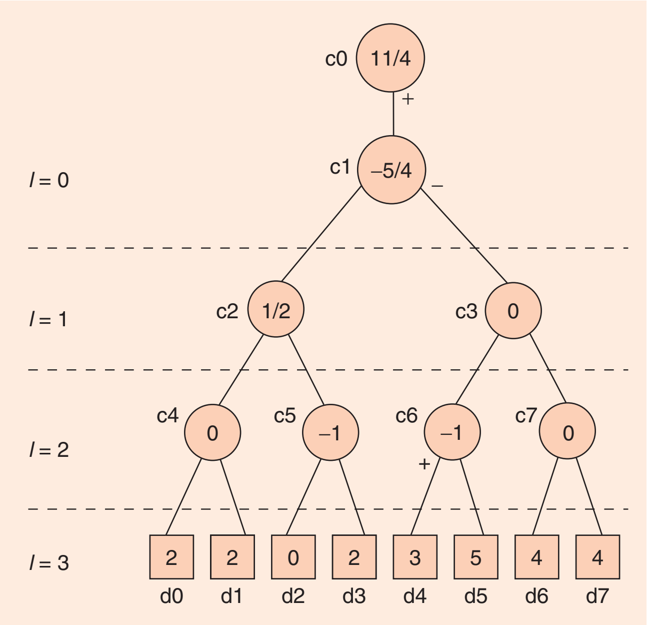
Consider a one-dimensional (1-D) data vector A containing the $N = 8$ data values $A = [2, 2, 0, 2, 3, 5, 4, 4]$. The Haar wavelet transform of A can be computed as follows. We first average the values together pairwise to get a new lower-resolution representation of the data with the average values $[2, 1, 4, 4]$. To be able to restore the original values of the data array, we need to store some detail coefficients, which capture the information lost due to this averaging. In Haar wavelets, these detail coefficients are simply the differences of the second of the averaged values from the computed pairwise average, i.e., $[2 - 2, 1 - 2, 4 - 5, 4 - 4] = [0, -1, -1, 0]$. No information has been lost in this process—it is simple to reconstruct the eight values of the original data array from the lower-resolution array containing the four averages and the four detail coefficients. Recursively applying the above pairwise averaging and differencing process on the lower-resolution array containing the averages, we get the following full transform:

Resolution	Averages	Detail Coefficients
3	[2, 2, 0, 2, 3, 5, 4, 4]	—
2	[2, 1, 4, 4]	[0, –1, –1, 0]
1	[3/2, 4]	[1/2, 0]
0	[11/4]	[–5/4]

The wavelet transform W_A of A is the single coefficient representing the overall average of the data values followed by the detail coefficients in the order of increasing resolution, i.e., $WA = [11/4, -5/4, 1/2, 0, 0, -1, -1, 0]$ (each entry is called a wavelet coefficient). For vectors containing similar values, most of the detail coefficients tend to be very small; thus, eliminating them from the wavelet transform (i.e., treating them as zeros) introduces only small errors when reconstructing the original data, resulting in a very effective form of lossy data compression [10].

1053-5888/06/$20.00©2006IEEE

A helpful tool for conceptualizing the recursive Haar wavelet transform process is the error tree structure (shown in Figure 1 for our example array A). Each internal node $c_i (i = 0, \ldots, 7)$ is associated with a wavelet coefficient value, and each leaf $d_i (i = 0, \ldots, 7)$ is associated with a value in the original data array; in both cases, the index i denotes the positions in the data or wavelet transform array. For instance, c_0 corresponds to the overall average of A. The resolution levels l for the coefficients (corresponding to levels in the tree) are also depicted.



[FIG1] **Error-tree structure for our example data array A ($N = 8$).**

Given an error tree T and an internal node t of T, $t \neq c_0$, we let $\mathtt{leftleaves}(t)$ ($\mathtt{rightleaves}(t)$) denote the set of leaf (i.e., data) nodes in the subtree rooted at t's left (respectively, right) child. Also, given any internal or leaf node u, we let $\mathtt{path}(u)$ be the set of all internal nodes in T that are proper ancestors of u (i.e., the nodes on the path from u to the root of T, including the root but not u) with nonzero coefficients. Finally, for any two leaf nodes d_l and d_h, we let $d(l : h)$ denote the range sum $\sum_{i=l}^{h} d_i$. Using the error tree representation T, we can outline the following important reconstruction properties of the Haar wavelet transform.

- (P1) The reconstruction of any data value d_i depends only on the values of the nodes in $\mathtt{path}(d_i)$. More specifically, we have $d_i = \sum_{c_j \in \mathtt{path}(d_i)} \delta_{ij} \cdot c_j$, where $\delta_{ij} = +1$ if $d_i \in \mathtt{leftleaves}(c_j)$ or $j = 0$, and $\delta_{ij} = -1$ otherwise; for example, $d_4 = c_0 - c_1 + c_6 = (11/4) - (-5/4) + (-1) = 3$.
- (P2) An internal node c_j contributes to the range sum $d(l : h)$ only if $c_j \in \mathtt{path}(d_l) \cup \mathtt{path}(d_h)$. More specifically, $d(l : h) = \sum_{c_j \in \mathtt{path}(d_l) \cup \mathtt{path}(d_h)} x_j$, where

$$x_j = \begin{cases} (h - l + 1) \cdot c_j, & \text{if } j = 0 \\ (|\text{leftleaves}(c_j, l : h)| \\ \quad -|\text{rightleaves}(c_j, l : h)|) \\ \quad \times c_j, & \text{otherwise}. \end{cases}$$

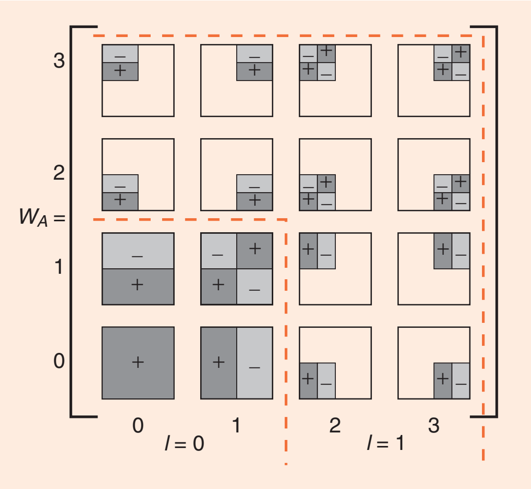
where $\mathtt{leftleaves}(c_j, l : h) = \mathtt{leftleaves}(c_j) \cap \{d_l, d_{l+1}, \ldots, d_h\}$ (i.e., the intersection of $\mathtt{leftleaves}(c_j)$ with the summation range) and $\mathtt{rightleaves}(c_j, l : h)$ is defined similarly. (Clearly, coefficients whose subtree is completely contained within the summation range have a net contribution of zero, and can be safely ignored.) For example, $d(2 : 6) = 5c_0 + (2 - 3)c_1 - 2c_2 = 5 \times (11/4) - (-5/4) - 1 = 14$. Thus, reconstructing a single data value involves summing at most $\log N + 1$ coefficients and reconstructing a range sum involves summing at most $2 \log N + 1$ coefficients, regardless of the width of the range. The support region for a coefficient c_i is defined as the set of (contiguous) data values that c_i is used to reconstruct.

The Haar wavelet transform can be naturally extended to multidimensional data arrays using two distinct methods, namely the standard and nonstandard Haar transform [10]. As in the 1-D case, the Haar transform of a d-dimensional data array A results in a d-dimensional wavelet-coefficient array W_A with the same dimension ranges and number of entries. Consider a d-dimensional wavelet coefficient W in the (standard or nonstandard) wavelet-coefficient array W_A. W contributes to the reconstruction of a d-dimensional rectangular region of cells in the original data array A (i.e., W's support region). Further, the sign of W's contribution ($+W$ or $-W$) can vary along the quadrants of W's support region in A.

As an example, Figure 2 depicts the support regions and signs of the 16 nonstandard, two-dimensional (2-D) Haar coefficients in the corresponding locations of a 4×4 wavelet-coefficient array W_A. The blank areas for each coefficient correspond to regions of A whose reconstruction is independent of the coefficient, i.e., the coefficient's contribution is 0. Thus, $W_A[0, 0]$ is the overall average that contributes positively (i.e., $+W_A[0, 0]$) to the reconstruction of all values in A, whereas $W_A[3, 3]$ is a detail coefficient that contributes (with the signs shown) only to values in A's upper-right quadrant. Each data cell in A can be accurately reconstructed by adding up the contributions (with the appropriate signs) of those coefficients whose support regions include the cell. Error-tree structures for d-dimensional Haar coefficients are essentially d-dimensional quadtrees, where each internal node t corresponds to a set of (at most) $2^d - 1$ Haar coefficients, and has 2^d children corresponding to the quadrants of the common support region of all coefficients in t; furthermore, properties (P1) and (P2) can also be naturally extended to the multidimensional case [2], [5], [6].

RELATIONAL DATA REDUCTION AND APPROXIMATE QUERY PROCESSING

Consider a relational table R with d data attributes $X_1, X_2, \ldots X_d$. We can represent the information in R as a d-dimensional array A_R, whose jth dimension is indexed by the values of attribute X_j and whose cells contain the count of tuples in R having the corresponding combination of attribute values. A_R is essentially the joint frequency distribution of all the data attributes of R. Given a limited



[FIG2] Support regions and signs for the 16 nonstandard 2-D Haar basis functions.

amount of storage for building a wavelet synopsis of an input relation R, a thresholding procedure retains a certain number $B << N$ of the coefficients in the wavelet transform of A_R as a highly compressed approximate representation of the original data (the remaining coefficients are implicitly set to 0). (The full details as well as efficient transform algorithms can be found in [2], [11].) The goal of coefficient thresholding is to determine the best subset of B coefficients to retain, so that some overall error measure in the approximation is minimized—in the next subsection, we discuss different thresholding strategies proposed in the database literature.

The construction of wavelet synopses typically takes place during the statistics collection process, whose goal is to create concise statistical approximations for the value distributions of either individual attributes or combinations of attributes in the relations of a database management system (DBMS). Once created, a wavelet synopsis is typically stored (as a collection of B wavelet coefficients) as part of the DBMS-catalog information, and can be exploited for several different purposes. The primary (and more conventional) use of such summaries is as a tool for enabling effective (compile-time) estimates of the result sizes of relational operators for the purpose of cost-based query optimization. (Accurate estimates of such result sizes play a critical role in choosing an effective physical execution plan for an input SQL query.) For instance, estimating the number of data tuples that satisfy a range-predicate selection like $l \leq X \leq h$ is equivalent to estimating the range summation $f(l:h) = \sum_{i=l}^{h} f_i$, where f is the frequency distribution array for attribute X. As mentioned earlier, given a B-coefficient synopsis of the f array, computing $f(l:h)$ only involves retained coefficients in `path`(f_l)∪ `path`(f_h) and, thus, can be estimated by summing only $\min\{B, 2\log N + 1\}$ synopsis coefficients [11]. A B-coefficient wavelet synopsis can also be easily expanded (in $O(B)$ time) into an $O(B)$-bucket histogram (i.e., piecewise-constant) approximation of the underlying data distribution with several possible uses (e.g., as a data visualization/approximation tool).

More generally, wavelet synopses can enable very fast and accurate approximate query answers during interactive data-exploration sessions. As demonstrated in [2], an approximate query processing algebra (which includes all conventional aggregate and nonaggregate SQL operators, such as `select`, `project`, `join`, `sumz`, and `average`) can operate directly over the wavelet synopses of relations, while guaranteeing the correct relational operator semantics. Query processing algorithms for these operators work entirely in the wavelet-coefficient domain. This allows for extremely fast response times, since the approximate query execution engine can do the bulk of its processing over compact wavelet synopses, essentially postponing the (expensive) expansion step into relational tuples until the end-result of the query.

CONVENTIONAL AND ADVANCED WAVELET THRESHOLDING SCHEMES

Recall that coefficient thresholding achieves data reduction by retaining $B << N$ of the coefficients in the wavelet transform of A_R as a highly compressed, lossy representation of the original relational data. The goal, of course, is to minimize the amount of loss quantified through some overall approximation error metric. Conventional wavelet thresholding (the method of choice for most studies on wavelet-based data reduction) greedily retains the B largest Haar-wavelet coefficients in absolute value after a simple normalization step. It is a well-known fact that this thresholding method is in fact provably optimal with respect to minimizing the overall root-mean-squared error (i.e., L_2-norm average error) in the data compression [10]. More formally, letting $\hat{d}_i$ denote the approximate reconstructed data value for cell i, retaining the B largest normalized coefficients implies that the resulting synopsis minimizes $\sqrt{\frac{1}{N}\sum_i(\hat{d}_i - d_i)^2}$ for the given amount of space B.

Conventional wavelet synopses optimized for overall L_2 error may not always be the best choice for approximate query processing systems. The quality of the approximate answers such synopses provide can vary widely, and users have no way of knowing the accuracy of any particular answer. Even for the simplest case of approximating a value in the original data set, the absolute and relative errors can show wide variation. Consider the example depicted in Table 1. The first line shows the 16 original data values (the exact answer), whereas the second line shows the 16 approximate answers returned when using conventional wavelet synopses and storing eight coefficients. Although the first half of the values is basically a mirror image of the second half, all the approximate answers for the first half are 65, whereas all the approximate answers for the second half are exact! Similar data values have widely different approximations, e.g., 30 and 31 have approximations 30 and 65, respectively. The approximate answers make the first half appear as a uniform distribution with widely different values, e.g., 3 and 127, having the same approximate answer 65. Moreover, the results do not improve when one considers the presumably easier problem of approximating the sum over a range of values—for all possible ranges

[TABLE 1] ERRORS WITH CONVENTIONAL WAVELET SYNOPSES.

ORIGINAL DATA VALUES	127	71	87	31	59	3	43	99	100	42	0	58	30	88	72	130
WAVELET ANSWERS	65	65	65	65	65	65	65	65	100	42	0	58	30	88	72	130

within the first half involving $x = 2$ to 7 of the values, the approximate answer will be $65 \cdot x$, while the actual answers vary widely. For example, for both the range d_0 to d_2 and the range d_3 to d_5, the approximate answer is 195, while the actual answer is 285 and 93, respectively. On the other hand, exact answers are provided for all possible ranges within the second half.

Our simple example illustrates that conventional wavelet synopses suffer from several important problems, including the introduction of severe bias in the data reconstruction and wide variance in the quality of the data approximation, as well as the lack of nontrivial guarantees for individual approximate answers. To address these shortcomings, recent work has proposed novel thresholding schemes for building wavelet synopses that try to minimize different approximation-error metrics, such as the maximum relative error (with an appropriate sanity bound $\mathtt{s}$) in the approximation of individual data values based on the synopsis; i.e., minimize

$$\max_i \left\{ \frac{|\hat{d}_i - d_i|}{\max\{|d_i|, s\}} \right\}.$$

Such relative-error metrics are arguably the most important quality measures for approximate query answers. Note that the role of the sanity bound is to ensure that relative-error numbers are not unduly dominated by small data values.

More specifically, [5] introduces probabilistic thresholding schemes based on ideas from randomized rounding, that probabilistically round coefficients either up to a larger rounding value (to be retained in the synopsis) or down to zero. Intuitively, their probabilistic schemes assign each nonzero coefficient fractional storage $y \in (0, 1)$ equal to its retention probability, and then flip independent, appropriately biased coins to construct the synopsis. Their thresholding algorithms are based on dynamic-programming (DP) formulations that explicitly minimize appropriate probabilistic metrics (such as the maximum normalized standard error or the maximum normalized bias) in the randomized synopsis construction; these formulations are then combined with a quantization of the potential fractional-storage allotments to give combinatorial techniques [5]. In more recent work, [6] shows that the pitfalls of randomization can be avoided by introducing efficient schemes for deterministic wavelet thresholding with the objective of optimizing a general class of error metrics (e.g., maximum or mean relative error). Their optimal and approximate thresholding algorithms are based on novel DP techniques that take advantage of the Haar transform error-tree structure, and can handle a broad, natural class of distributive error metrics; this class includes several useful error measures for approximate query answers, such as maximum or mean weighted relative error and weighted L_p-norm error [6].

EXTENDED AND STREAMING WAVELET SYNOPSES

Complex tabular data sets with multiple measures (multiple numeric entries for each table cell) introduce interesting challenges for wavelet-based data reduction. Such massive, multimeasure tables arise naturally in several application domains, including online analytical processing (OLAP) environments and time-series analysis/correlation systems. As an example, a corporate sales database may tabulate, for each available product, 1) the number of items sold, 2) revenue and profit numbers for the product, and 3) costs associated with the product, such as shipping and storage costs. Similarly, real-life applications that monitor continuous time-series typically have to deal with several readings (measures) that evolve over time; for example, a network-traffic monitoring system takes readings on each time-tick from a number of distinct elements, such as routers and switches, in the underlying network and typically several measures of interest need to be monitored (e.g., input/output traffic numbers for each router or switch interface) even for a fixed network element. [4] shows that obvious approaches for building wavelet synopses for such multimeasure data can lead to poor synopsis-storage utilization and suboptimal solutions even in very simple cases. Instead, their proposed solution is based on 1) extended wavelet coefficients, the first adaptive, efficient storage scheme for multimeasure wavelet coefficients and 2) novel algorithms for selecting the optimal subset of extended coefficients to retain for minimizing the weighted sum of L_2 errors across all measures under a given storage constraint.

Traditional database systems and approximation techniques are typically based on the ability to make multiple passes over persistent data sets, that are stored reliably in stable storage. For several emerging application domains, however, data arrives at high rates and needs to be processed on a continuous (24 × 7) basis, without the benefit of several passes over a static, persistent data image. Such continuous data streams arise naturally, for example, in the network installations of large Telecom and ISPs where detailed usage information [call-detail-records (CDRs) and SNMP/RMON packet-flow data] from different parts of the underlying network needs to be continuously collected and monitored for interesting trends and phenomena (e.g., fraud or denial-of-service attacks). Efficiently tracking an accurate wavelet synopsis over such massive streaming data, using only small space and time (per streaming update), poses a host of new challenges. Recently proposed solutions [3], [7] rely on maintaining small-space, pseudorandom sketches (essentially, random linear projections) over the input data stream [1]. These sketches can then be queried to efficiently recover the topmost wavelet coefficients of the underlying data distribution within provable error guarantees [3].

CONCLUSIONS AND FUTURE DIRECTIONS

Approximate query processing over concise, precomputed synopses is slowly emerging as an essential tool for numerous data-intensive applications requiring interactive response times. Recent database research efforts have clearly shown the effectiveness of the wavelet transform as a data-reduction tool that enables fast and accurate approximate query answers over complex relational data. In addition, novel algorithmic tools have been proposed for wavelet thresholding under a variety of different error metrics, handling multimeasure data sets, and maintaining wavelet summaries over massive continuous data streams.

An important open question concerns the general suitability of the Haar wavelet transform as a data-summarization and approximate query processing tool when it comes to error metrics other than L_2 norms. In fact, recent work [8] shows that considering the unrestricted version of the problem (where one is allowed to store any number as a synopsis coefficient instead of the standard Haar coefficients), can result in significant accuracy benefits. Thus, the question is, are there other (existing or new) wavelet bases that are better suited for optimizing different error metrics (e.g, mean weighted relative-error) in the data approximation?

AUTHOR

Minos Garofalakis (minos.garofalakis@intel.com) is a senior research scientist with Intel Research Berkeley. He obtained his Ph.D. from the University of Wisconsin-Madison in 1998 and joined Intel Research in July 2005, after spending six and one-half years as a member of technical staff with Bell Labs in Murray Hill, New Jersey. His current research interests include data streaming, approximate query processing, and XML databases.

REFERENCES

[1] N. Alon, Y. Matias, and M. Szegedy, "The space complexity of approximating the frequency moments," in *Proc. 28th Annual ACM Symp. Theory Computing*, Philadelphia, PA, May 1996, pp. 20–29.

[2] K. Chakrabarti, M.N. Garofalakis, R. Rastogi, and K. Shim, "Approximate query processing using wavelets," *VLDB J.*, vol. 10, no. 2–3, pp. 199–223, Sept. 2001.

[3] G. Cormode, M. Garofalakis, and D. Sacharidis, "Fast approximate wavelet tracking on streams," in *Proc. 10th Int. Conf. Extending Database Technology (EDBT'2006)*, Munich, Germany, Mar. 2006, pp. 4–22.

[4] A. Deligiannakis and N. Roussopoulos, "Extended wavelets for multiple measures," in *Proc. 2003 ACM SIGMOD Int. Conf. Management Data*, San Diego, CA, June 2003, pp. 229–240.

[5] M. Garofalakis and P.B. Gibbons, "Probabilistic wavelet synopses," *ACM Trans. Database Syst.*, vol. 29, no. 1, pp. 43–90, Mar. 2004.

[6] M. Garofalakis and A. Kumar, "Wavelet synopses for general error metrics," *ACM Trans. Database Syst.*, vol. 30, no. 4, pp. 888–928, Dec. 2005.

[7] A.C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M.J. Strauss, "One-pass wavelet decomposition of data streams," *IEEE Trans. Knowledge Data Eng.*, vol. 15, no. 3, pp. 541–554, May 2003.

[8] S. Guha and B. Harb, "Wavelet synopsis for data streams: minimizing non-Euclidean error," in *Proc. 11th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, Chicago, IL, Aug. 2005, pp. 88–97.

[9] B. Jawerth and W. Sweldens, "An overview of wavelet based multiresolution analyses," *SIAM Rev.*, vol. 36, no. 3, pp. 377–412, 1994.

[10] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin, *Wavelets for computer graphics—Theory and applications*. San Francisco, CA: Morgan Kaufmann, 1996.

[11] J.S. Vitter and M. Wang, "Approximate computation of multidimensional aggregates of sparse data using wavelets," in *Proc. 1999 ACM SIGMOD Int. Conf. Management Data*, Philadelphia, PA, May 1999, pp. 193–204.

[applications CORNER] continued from page 53

CONCLUSIONS

The Darbellay-Vajda algorithm reviewed here develops a skeletonized approximation to a joint probability density of sampled data. The approximation is presented as a collection of nonoverlapping multidimensional cuboids, having varying sizes, locations, and probabilities in sample space. It is already known that a mutual information (or marginal redundancy) value can be extracted from this collection. This article demonstrates that the joint density has a far wider range of application in exploring Bayesian and conditional probability distributions among the observations. The examples have shown only autonomous data modeling, but categorical data is easily input as an additional independent variable for supervised training purposes. Though the mathematical fundamentals of the algorithm are hardly straightforward, the associated computation load is low, and the overall flexibility of the technique points to the possibility of attractive new algorithms for statistical signal processing in numerous areas such as machine learning, pattern recognition, and nonlinear filtering.

AUTHOR

John E. Hudson (john-hudson@lycosmax.co.uk) was with Nortel Networks, United Kingdom, from 1988 until recently retiring. Prior to being with Nortel, he held a teaching position at the University of Loughborough. His work has focused on adaptive signal processing for sonar and radio. He is a Member of the IEEE and IEE and a Fellow of the Institute of Mathematics and Applications. He has a visiting professorship at the University of Newcastle. He has authored or coauthored 15 journal papers, wrote an early book on adaptive antennas, and holds 12 patents.

REFERENCES

[1] A.G. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Trans. Inform. Theory*, vol. 45, no. 4, pp. 1315–1321, May 1999.

[2] A.G. Darbellay and P. Tichavsky, "Independent component analysis through direct estimation of the mutual information," in *Proc. ICA2000: 2nd Int. Workshop Independent Component Analysis Blind Separation*, Helsinki, 19–22 June 2000, pp. 69–75.

[3] A. Webb, *Statistical Pattern Recognition*, 2nd ed. New York: Wiley, 2002.

[4] T. Cover and J.A. Thomas, *Elements of Information Theory*. New York: Wiley, 2002.

[5] M. Menendez, P.D. Morales, and L. Pardo, "Maximum entropy principle and statistical inference on condensed ordered data," *Statist. Probability Lett*. vol. 34, no. 1, pp. 85–93, May 1997.

[6] D.R. Brillinger, "Some data analyses using mutual information," *Brazilian J. Probability Statist.*, vol. 18, pp. 163–183, 2004.