6.962: Week 4 Summary of Discussion

Presenter: Albert Chan
Date: October 4, 2000
Topic: Codes on Graphs

1 A Brief History of Codes on Graphs

1962 Gallager invents low density parity-check (LDPC) codes [1][2] and what is now known as the sum-product iterative decoding algorithm for *a posteriori* probability (APP) decoding.

1981 Tanner founds the field of codes on graphs, introducing the bipartite graphical model now widely popular referred to as a "Tanner graph" and the decoding algorithm now called the "min-sum" (or "max-product") algorithm. Tanner generalizes the single parity-check local constraints of LDPC codes to arbitrary linear code constraints.

1993 Turbo codes are invented, and the gap to capacity is essentially closed. However, the decoding algorithm remains to be analyzed, and the subsequent development of turbo codes occurs largely independently of codes on graphs.

1994-1996 Spielman's thesis introduces LDPC codes based on expander graphs [5][6], which have linear-time encoding and decoding algorithms but still decent performance. These results are quickly adapted by Alon and Luby [7] to construct low-complexity, near-optimal coding schemes, the so-called "tornado codes," for reconstructing large Internet files in the presence of packet erasures.

Meanwhile, MacKay and Neal use simulations to show that long LDPC codes can perform as well as turbo codes [8][9], shocking the coding world.

Independently, Wiberg produces his thesis "Codes and decoding on general graphs" [10], which Prof. Forney calls "a triumph of unification that established the intellectual foundations of the field." In his thesis, Wiberg extends Tanner graphs to include state variables as well as symbol variables, permitting unification of turbo codes, LDPC codes, and trellis codes together in a common framework.

1998–present Kschischang, Frey, and Loeliger generalize Wiberg-type graphs and introduce "factor graphs" [11], natural graphical models of a global function that can be factored into a product of local functions.

Forney introduces "normal graphs" [12], defined as Wiberg-type graphs in which all symbol variables have degree 1 and state variables have degree 2, leading to a clean separation of functions in sum-product decoding.

Divsalar, Jin, and McEliece introduce "repeat-accumulate" (RA) codes [13]–simple "turbo-like" codes combine simple repetition codes, a 2-state rate-1/1 convolutional "accumulator" code, and a large pseudo-random interleaver. These codes are within 1.5 dB of the Shannon limit.

Tornado codes become the first LDPC codes to be commercialized [14]. However, LDPC codes (even to this day) remain well behind turbo codes in terms of real-world applications.

Richardson, Urbanke, and Shokrollahi design long irregular LDPC codes which clearly beat turbo codes for block lengths of 10^5 or more [15][16].

Using long irregular LDPC codes and an improved density evolution algorithm, Chung, Forney, Richardson, and Urbanke design rate 1/2 codes with performance within a few hundredths of a decibel of the Shannon limit [17].

2 Factor Graphs and the Sum-Product Algorithm

Many well-known algorithms that deal with complicated global functions of many variables derive their computational efficiency by factoring the global function into simpler, local functions. Such factorizations can be visualized using factor graphs [11], which are bipartite graphs that express which variables are arguments of which local functions. An example is shown in Fig. 1.

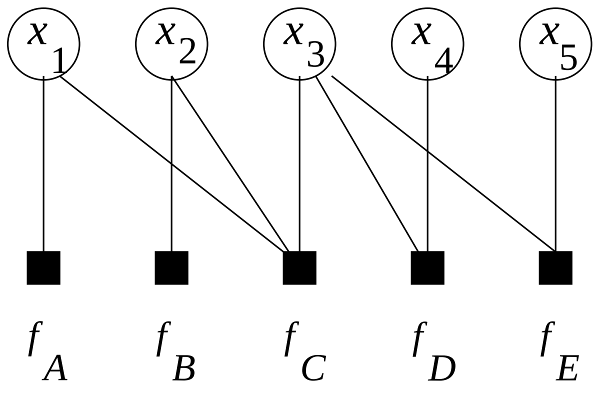

Figure 1: A factor graph for the product $f_A(x_1)f_B(x_2)f_C(x_1,x_2,x_3)f_D(x_3,x_4)f_E(x_3,x_5)$.

The sum-product algorithm is a generic message-passing algorithm that operates in a factor graph and attempts to compute various marginal functions associated with the global function. The sum-product algorithm follows one simple rule:

The Sum-Product Update Rule: The message sent from a node v on an edge e is the product of the local function at v (or the unit function if v is a variable node) with all messages received at v on edges *other* than e, "summarized" for the variable assiciated with e.

Let $\mu_{x\to f}(x)$ denote the message sent from node x to node f in one operation of the sum-product algorithm, let $\mu_{f\to x}(x)$ denote the message sent from node f to node x, and let $n(v)$

denote the set of neighbors of a given node v in a factor graph. Then, the messages passed between nodes can be computed as follows:

variable to local function:

$$\mu_{x \to f}(x) = \prod_{h \in n(x), h \neq f} \mu_{h \to x}(x) \tag{1}$$

local to variable function:

$$\mu_{f \to x}(x) = \sum_{\text{over all variables except } x} \left(f(X) \prod_{y \in n(f), y \neq x} \mu_{y \to f}(y) \right) \tag{2}$$

Since many well-known algorithms essentially solve the "MPF" (marginalize product-of-functions) problem [18], factor graphs, together with the sum-product algorithm, are a simple way to bring together many seemingly different algorithms in computer science and engineering into one common framework. Such algorithms include:

- forward/backward algorithm, a.k.a. BCJR, APP, or MAP algorithm
- Viterbi algorithm
- iterative turbo decoding algorithm
- Pearl's belief propagation algorithm for Bayesian networks
- Kalman filter
- certain fast Fourier transform algorithms

When a factor graph is cycle-free, the structure of the graph not only represents the way a given function factors, but the structure also encodes arithmetic expressions via the sum-product algorithm, by which the various marginal functions associated with the given function may be computed. When a factor graph has cycles, such as those associated with turbo codes and low-density parity-check codes, algorithms with no natural termination result. Note that a given function may be factored in many different ways, and thus may be represented by various factor graphs with or without cycles, with varying degrees of structural complexity.

References

[1] R. G. Gallager, "Low-density partiy-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[2] R. G. Gallager, *Low-Density Partiy-Check Codes.* Cambridge, MA: MIT Press, 1963.

[3] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes (1)," *Proc. ICC '93*, Geneva, Switzerland, pp. 1064–1070, June 1993.

[5] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710–1722, Nov. 1996. Also *Proc. 35th Symp. Found. Comp. Sci.*, pp. 566–576, 1994.

[6] D. A. Spielman, "Linear-time encodable and decodable error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1723–1731, Nov. 1996.

[7] N. Alon and M. Luby, "A linear-time erasure-resilient code with nearly optimal recovery," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1732–1736, Nov. 1996.

[8] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999. Also *Proc. 5th IMA Conf. Crypto. Coding*, pp. 100-111, 1995.

[9] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity check codes," *Elect. Lett.*, vol. 32, pp. 1645–1546, Aug. 1996. Reprinted *Elect. Lett.*, vol. 33, pp. 457–458, Mar. 1997.

[10] N. Wiberg, "Approaches to neural-network decoding of error-correcting codes," Lic. Thesis, U. Linköping, Sweden, 1994.

[11] F. R. Kschischang, B. J. Frey and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," submitted to *IEEE Trans. Inform. Theory*, July 1998.

[12] G. D. Forney, Jr., "Codes on graphs: Normal realizations," submitted to *IEEE Trans. Inform. Theory*, Dec. 1998.

[13] D. Divsalar, H. Jin and R. J. McEliece, "Coding theorems for 'turbo-like' codes," *Proc. 1998 Allerton Conf.*, Allerton, IL, pp. 201–210, Sept. 1998.

[14] J. W. Byers, M. Luby, M. Mitzenmacher and A. Rege, "A digital fountian approach to reliable distribution fo bulk data," *Proc. ACM SIG-COMM '98*, Vancouver, Canada, 1998.

[15] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," submitted to *IEEE Trans. Inform. Theory*, Nov. 1998.

[16] T. J. Richardson, A. Shokrollahi and R. Urbanke, "Design of provably good low-density parity-check codes," submitted to *IEEE Trans. Inform. Theory*, May. 1999.

[17] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0057 dB from the Shannon limit," submitted to *IEEE Comm. Letters*, May 2000.

[18] S. M. Aji and R. J. McEliece, "A general algorithm for distributing information on a graph," in *Proc. 1997 IEEE Int. Symp. on Inform. Theory*, Ulm, Germany, p. 6, July 1997.