Applications of graphical models in signal processing

Brain-Style Information Systems Research Group, RIKEN BSI, Saitama, Japan

Justin Dauwels
dauwels@isi.ee.ethz.ch

Signal and Information Processing Laboratory
Department of Information Technology and Electrical Engineering
ETH Zurich, Switzerland
Electromyography

Motor Unit (MU)

- Dendrites
- Axon
- Motor neuron
- Muscle fibres

Motor Unit Action Potential (MUAP)
EMG Signal Decomposition

Signal from MU 0:

Signal from MU 1:

EMG signal:
$(\sigma_{\text{noise}} = 20)$
Estimation Theory

Optimal estimators

- **Block MAP (Maximum a posteriori)**
  \[ \hat{x} = \arg\max_x p(x|y) = \arg\max_x p(x, y) \]

- **Symbol MAP**
  \[ \hat{x}_k = \arg\max_{x_k} p(x_k|y) = \arg\max_{x_k} p(x_k, y) = \arg\max_{x_k} \sum_{\sim x_k} p(x, y) \]

Key observations

- Summation and maximization are the involved operations.
- The probability function \( p(x, y) \) factorizes in most practical systems . . .
- ... which leads to more efficient computations.
- Despite this fact, optimal estimators are often infeasible . . .
- ... and hence one typically resorts to approximations.
Our methodology

Our approach rests on two pillars

1. A graphical representation of \( p(x, y) \), a so-called factor graph.

2. An algorithm (sum-product algorithm) that computes marginals \( p(x_k) \) and operates on a factor graph that represents the system at hand.

Within this framework

- the factorization of \( p(x, y) \) is intensively exploited and, as a consequence, marginals \( p(x_k) \) are computed efficiently.
- approximations can be designed in a systematic fashion.
Motivation

• Many algorithms in coding, signal processing, and machine learning may be viewed as instances of the sum-product algorithm.
Motivation

• Many algorithms in coding, signal processing, and machine learning may be viewed as instances of the sum-product algorithm.

• Specific instances of such algorithms include Kalman filtering and smoothing, the forward-backward algorithm for hidden Markov models, Viterbi algorithm, probability propagation in Bayesian networks, decoding algorithms for error correcting codes (in particular iterative decoding of turbo codes and low-density parity check codes), ...
Motivation

• Many algorithms in coding, signal processing, and machine learning may be viewed as instances of the sum-product algorithm.

• Specific instances of such algorithms include Kalman filtering and smoothing, the forward-backward algorithm for hidden Markov models, Viterbi algorithm, probability propagation in Bayesian networks, decoding algorithms for error correcting codes (in particular iterative decoding of turbo codes and low-density parity check codes), . . .

• Applications at ISI: Decomposition of EMG signals, signal processing for hearing aids, synchronization in digital communications, decoding, estimation/detection with analog circuits, computation of capacities of communication channels, smart fire detection, body-state estimation, seismosomnography, . . .
Motivation

• Many algorithms in coding, signal processing, and machine learning may be viewed as instances of the sum-product algorithm.

• Specific instances of such algorithms include Kalman filtering and smoothing, the forward-backward algorithm for hidden Markov models, Viterbi algorithm, probability propagation in Bayesian networks, decoding algorithms for error correcting codes (in particular iterative decoding of turbo codes and low-density parity check codes), . . .

• Applications at ISI: Decomposition of EMG signals, signal processing for hearing aids, synchronization in digital communications, decoding, estimation/detection with analog circuits, computation of capacities of communication channels, smart fire detection, body-state estimation, seismosomnography, . . .

• Factor graphs and message-passing algorithm have been proven to be very powerful in decoding and signal processing.
Overview

• Factor graphs
• Sum-product algorithm
• Application: decomposition of EMG-signals
• Conclusions
Factor Graphs (FGs)

- Factor graphs represent the factorization of a function.
- Example

\[ f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3) f_B(x_3, x_4, x_5) f_C(x_4). \]
Factor Graphs (FGs)

- Factor graphs represent the factorization of a function.
- Example

\[ f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3)f_B(x_3, x_4, x_5)f_C(x_4). \]

- \(x_1, x_2, x_3, x_4,\) and \(x_5\) are the variables.
- \(f\) is called the global function.
- \(f_A, f_B,\) and \(f_C\) are called local functions.
Factor Graphs (FGs)

- Factor graphs represent the factorization of a function.
- Example

\[ f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3) f_B(x_3, x_4, x_5) f_C(x_4). \]

- Rules for drawing a factor graph
  - A node for every factor
  - An edge for every variable
  - Node \( g \) is connected to edge \( x \) iff variable \( x \) appears in factor \( g \)
Equality constraint node

$$f_{=} (x, y, z) \triangleq \begin{cases} 1, & \text{if } x = y = z \\ 0, & \text{else} \end{cases}$$

$$= \delta[x - y] \cdot \delta[x - z]$$
• Block diagram: $X = g(U, W), \ Z = h(X, Y)$
• Block diagram: \( X = g(U, W), Z = h(X, Y) \)

• Factor graph: \( f_G = \delta(x - g(u, w)), f_H = \delta(z - h(x, y)) \)

• Global function \( f(u, w, x, y, z) = \delta(x - g(u, w)) \cdot \delta(z - h(x, y)) \)

\( f \neq 0 \) (configuration valid) iff the configuration is consistent with the functions \( g \) and \( h \) of the block diagram.
Joint Probability Distribution

- Factor graphs are used mostly to represent probabilistic models.
- Given is a joint probability distribution

\[ p_{XY}(x, y) \]

of the two discrete random variables \( X \) and \( Y \).
Factorization of Joint Probability Distribution

- Chain rule

\[ p_{XY}(x, y) = p_X(x)p_{Y|X}(y|x) \]

- Factor graph
Independent Random Variables

• If the two random variables $X$ and $Y$ are independent, then
$$p_{Y|X}(y|x) = p_Y(y)$$
and we get the factorization
$$p_{XY}(x, y) = p_X(x)p_Y(y).$$

• Factor graph

![Factor Graph]

• If two components are unconnected, then every random variable of one component is independent of every random variable in the other component.
Markov Chain

- Markov chain of the form

\[ p_{XYZ}(x, y, z) = p_X(x) \cdot p_{Y|X}(y|x) \cdot p_{Z|Y}(z|y) \]

- Factor graph
Summary: Factor Graphs

- Factor graphs
- Equal node
- Block diagrams and factor graphs
- Probabilistic models represented by factor graph
Overview

• Factor graphs
• Sum-product algorithm
• Application: decomposition of EMG-signals
• Conclusions
Our approach rests on two pillars

1. A graphical representation of $p(x, y)$, a so-called factor graph.

2. An algorithm (sum-product algorithm) that computes marginals $p(x_k)$ and operates on a factor graph that represents the system at hand.

Within this framework

- the factorization of $p(x, y)$ is intensively exploited and, as a consequence, marginals $p(x_k)$ are computed efficiently.

- approximations can be designed in a systematic fashion.
Computing marginals

- **Given:** Discrete probability mass function
  \[ f(x_1, \ldots, x_8) = \left( f_1(x_1)f_2(x_2)f_3(x_1, x_2, x_3, x_4) \right) \cdot \left( f_4(x_4, x_5, x_6)f_5(x_5)(f_6(x_6, x_7, x_8)f_7(x_7)) \right) \]

- **Wanted:** Marginal probability
  \[ p(x_4) = \sum_{x_1, x_2, x_3, x_5, x_6, x_7, x_8} f(x_1, \ldots, x_8) \]

- This factorization can be represented by a factor graph.
Elimination of Variables

\[ f(x_1, \ldots, x_8) = \left( f_1(x_1) f_2(x_2) f_3(x_1, x_2, x_3, x_4) \right) \cdot \left( f_4(x_4, x_5, x_6) f_5(x_5) (f_6(x_6, x_7, x_8) f_7(x_7)) \right) \]
Elimination of Variables

\[ p(x_4) = \left( \sum_{x_1} \sum_{x_2} \sum_{x_3} f_3(x_1, x_2, x_3, x_4) f_1(x_1) f_2(x_2) \right) \cdot \]

\[ \mu_{f_3 \rightarrow x_4} \]

\[ \sum_{x_5} \sum_{x_6} f_4(x_4, x_5, x_6) f_5(x_5) \left( \sum_{x_7} \sum_{x_8} f_6(x_6, x_7, x_8) f_7(x_7) \right) \cdot \]

\[ \mu_{f_6 \rightarrow x_6} \]

\[ \mu_{f_4 \rightarrow x_4} \]
Sum-product Algorithm

Sum-product update rule

$$
\mu_{X_1 \rightarrow h(x_1)} \\
\vdots \\
\mu_{X_N \rightarrow h(x_N)} \\
\mu_{h \rightarrow Y(y)} \\
\mu_{h \rightarrow Y(y)} = \sum_{x_1, \ldots, x_N} h(x_1, \ldots, x_N, y) \mu_{X_1 \rightarrow h(x_1)} \cdots \mu_{X_N \rightarrow h(x_N)}.
$$

Computing marginals

$$
\gamma \mu_{f \rightarrow x}(x) \mu_{g \rightarrow x}(x)
$$

$$
p(X) = \gamma \mu_{f \rightarrow x}(x) \mu_{g \rightarrow x}(x)
$$
Sum-product Algorithm: Equal Node

\[ f(x_1, x_2, y) \triangleq \delta[x_1 - x_2] \cdot \delta[x_1 - y] \]

For binary variables \( X_1, X_2, \) and \( Y \) we get:

\[
\mu_{\text{out}}(y) = \sum_{x_1} \sum_{x_2} \delta[x_1 - x_2] \delta[x_1 - y] \mu_1(x_1) \mu_2(x_2)
\]

\[
= \mu_1(y) \mu_2(y)
\]

\[
\mu_{\text{out}}(0) = \mu_1(0) \mu_2(0)
\]

\[
\mu_{\text{out}}(1) = \mu_1(1) \mu_2(1)
\]
Sum-product Algorithm

- Factor graph without loops
  - The sum-product algorithm computes all marginals simultaneously and exactly.

- Factor graph with loops
  - The sum-product rule can be applied repeatedly: iterative algorithm.
  - The iterative algorithm might not converge.
    - Then another factor graph (factorization) can be tried.
    - There are several factor graph design choices.
  - The algorithm computes only an approximation of the marginals $p(x_k)$.
    - This approximation is often (but not always!) good enough.
    - In some very demanding applications the performance is nearly optimal.
    - The algorithm is suboptimal - but it can handle very large systems.
Intermediate Summary

• A factor graph is a graphical representation of a mathematical model.
Intermediate Summary

• A factor graph is a graphical representation of a mathematical model.
• The sum-product algorithm
  • runs on a factor graph and calculates messages that are sent along the edges of the graph and
  • is suboptimal on graphs with cycles.
Intermediate Summary

- A factor graph is a graphical representation of a mathematical model.
- The sum-product algorithm
  - runs on a factor graph and calculates messages that are sent along the edges of the graph and
  - is suboptimal on graphs with cycles.
- However, iterative algorithms on graphs with cycles have been proven to be very powerful.
Intermediate Summary

• A factor graph is a graphical representation of a mathematical model.

• The sum-product algorithm
  • runs on a factor graph and calculates messages that are sent along the edges of the graph and
  • is suboptimal on graphs with cycles.

• However, iterative algorithms on graphs with cycles have been proven to be very powerful.

• Applications of iterative algorithms on graphs are, e.g.,
  • Decoding of error correcting codes,
  • Signal processing.
Overview

• Factor graphs
• Sum-product algorithm
• Application: decomposition of EMG-signals
• Conclusions
EMG Signal Decomposition

Signal from MU 0:

Signal from MU 1:

EMG signal:
\( \sigma_{\text{noise}} = 20 \)
Lab History and Objective

• Our lab: 8 PhD students in the field of EMG since 1975

• 1992: Richard Gut (Prof. Moschytz)

• 2000: Peter Wellig (Prof. Moschytz)

• 200?: Volker Koch (Prof. Loeliger)
  • Decomposition algorithm based on a
    • message-passing algorithm (belief propagation) in a
    • graphical model (factor graph)
  • Preliminary result: decomposition of more than 8 overlapping MUAPs
  • Information on the web
    • Volker’s Webpage: http://www.volker-koch.de/
    • EMG forum: http://www.emg.ethz.ch
Block Diagram $\rightarrow$ Factor Graph

Block Diagram

$$Y_{0,k} = \sum_{i=0}^{1} \sum_{\ell=0}^{M} X_{i,k-\ell} \cdot h_{i,0,\ell} + W_{0,k}$$

$$Y_0 \rightarrow X_0, X_1$$
Block Diagram → Factor Graph

Block Diagram

\[ \begin{align*}
X_0 & \xrightarrow{FIR} T_0 \xrightarrow{+} W_0 \xrightarrow{+} Y_0 \\
X_1 & \xrightarrow{FIR} T_1 \xrightarrow{+} X_0, X_1
\end{align*} \]

Factor Graph

\[ Y_{0,k} = \sum_{i=0}^{1} \sum_{\ell=0}^{M} X_{i,k-\ell} \cdot h_{i,0,\ell} + W_{0,k} \]

\[ Y_0 \rightarrow X_0, X_1 \]

28/39
Estimating Motor Unit Firing Times

\[
S_{0,k} \quad X_{0,k} \quad S_{1,k} \quad X_{1,k} \quad S_{0,k+1}
\]

\[
S_{1,k} \quad C \quad S_{1,k+1}
\]

\[
NW_{0,k} \quad + \quad Y_{0,k}
\]
Estimating Motor Unit Firing Times

\[ X_{0,k}, S_{0,k}, S_{0,k+1}, X_{1,k}, S_{1,k}, S_{1,k+1}, C, C, W_{0,k}, Y_{0,k} \]
Factor Graph for Two Channels

\[ X_{0,k}, S_{0,k}, X_{1,k}, S_{0,k+1}, S_{1,k}, S_{1,k+1} \]

\[ X_{0,k}, X_{1,k}, S_{0,k}, S_{0,k+1}, S_{1,k}, S_{1,k+1} \]

\[ W_{0,k}, W_{1,k}, Y_{0,k}, Y_{1,k} \]
Preliminary Results

- Simulated signals
  - Decomposition of superpositions of more than 8 overlapping MUAPs

- Measured signals
  - First tests done

- Changing MUAP shapes in simulated long-term recordings
  - First tests done

- MUAP shapes
  - Need to be given to our algorithm

- Noise model
  - AWGN, tried others
Results: 1 Simulated Signal, 6 MU

- = actual firing
- = detected firing
Results: 3000 Simulated Signals, 6 MU

<table>
<thead>
<tr>
<th>$\sigma_{\text{noise}}$</th>
<th>Number of completely correct decomposed signals / total number of signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>899/1000</td>
</tr>
<tr>
<td>5</td>
<td>836/1000</td>
</tr>
<tr>
<td>10</td>
<td>720/1000</td>
</tr>
</tbody>
</table>
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Simulated Signal, 8 MU
Results: 1 Measured Signal, 9 MU

Thanks to Kevin McGill for providing the EMG signal!
Overview

- Factor graphs
- Sum-product algorithm
- Application: decomposition of EMG-signals
- Conclusions
Conclusions

- EMG signal decomposition algorithm
- Based on belief propagation in a graph
- Aim: Resolving difficult superpositions without much a-priori information
  - e.g., no firing statistics
  - e.g., dynamic contractions
  - e.g., doublets
Applications of graphical models in signal processing

Brain-Style Information Systems Research Group, RIKEN BSI, Saitama, Japan

Justin Dauwels
dauwels@isi.ee.ethz.ch

Signal and Information Processing Laboratory
Department of Information Technology and Electrical Engineering
ETH Zurich, Switzerland