Gaussian BP

Introduction to Graphical Models and Inference for Communications

UC3M

March 3, 2018

Today

- One particular scenario: we want to perform inference over Gaussian probability distributions that factorize according to a graphical model
- Factor Graphs.
- Belief Propagation naturally extends to this scenario by replacing summations to integrals (*Gaussian Belief Propagation*).
- GaBP is exact for Gaussian tree factor graphs.
- GaBP computes the exact mean for graphs with cycles, but approximated covariance matrix.

Index

Multivariate Gaussian distribution

Let $\boldsymbol{X}$ be a Gaussian random vector ($\boldsymbol{x} \in \mathbb{R}^n$).

- **Covariance form:** the probability density function is

$$\mu(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m})\right\}$$

  denoted as $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{m} = \mathbb{E}[\boldsymbol{x}]$ and covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}[(\boldsymbol{x}-\boldsymbol{m})^T(\boldsymbol{x}-\boldsymbol{m})]$.

- **Natural form:** the probability density function is

$$\mu(\boldsymbol{x}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{J}\boldsymbol{x} + \boldsymbol{h}^T\boldsymbol{x}\right\}$$

  denoted as $\boldsymbol{x} \sim \mathcal{N}^{-1}(\boldsymbol{h}, \boldsymbol{J})$ with potential vector $\boldsymbol{h}$ and *precision matrix* $\boldsymbol{J}$.

- Note $\boldsymbol{J} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{h} = \boldsymbol{J}\boldsymbol{m}$.

Gaussian graphical models typically describe Gaussian joint probability density functions in **natural form**.

Product of two Gaussians

$$
\begin{aligned}
&\mathcal{N}(\boldsymbol{m}_a, \boldsymbol{\Sigma}_a)\mathcal{N}(\boldsymbol{m}_b, \boldsymbol{\Sigma}_b) \\
=&\mathcal{N}^{-1}(\boldsymbol{h}_a, \boldsymbol{J}_a)\mathcal{N}^{-1}(\boldsymbol{h}_b, \boldsymbol{J}_b) \\
=&\mathcal{N}^{-1}(\boldsymbol{h}_a + \boldsymbol{h}_b, \boldsymbol{J}_a + \boldsymbol{J}_b) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})
\end{aligned}
$$

where

$$
\boldsymbol{\Sigma} = (\boldsymbol{J}_a + \boldsymbol{J}_b)^{-1} \qquad \boldsymbol{m} = \boldsymbol{\Sigma}(\boldsymbol{h}_a + \boldsymbol{h}_b)
$$

Marginalization via algebraic manipulation

- Let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}$, where $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}_x, \boldsymbol{\Sigma}_x) = \mathcal{N}^{-1}(\boldsymbol{h}_x, \boldsymbol{J}_x)$ and $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{m}_v, \boldsymbol{\Sigma}_v) = \mathcal{N}^{-1}(\boldsymbol{h}_v, \boldsymbol{J}_v)$, then

$$\mu(\boldsymbol{y}) = \int_{-\infty}^{\infty} \mu(\boldsymbol{y}|\boldsymbol{x})\mu(\boldsymbol{x})d\boldsymbol{x}?$$

$$\begin{aligned}
\mu(\boldsymbol{y}|\boldsymbol{x})\mu(\boldsymbol{x}) &\propto \exp\left\{\frac{-1}{2}\boldsymbol{y}^T\boldsymbol{J}_v\boldsymbol{y} + (\boldsymbol{J}_v\boldsymbol{m}_v)^T\boldsymbol{y} + (\boldsymbol{J}_v\boldsymbol{A}\boldsymbol{x})^T\boldsymbol{y} - \frac{-1}{2}\boldsymbol{x}^T\boldsymbol{J}_x\boldsymbol{x} + (\boldsymbol{J}_x\boldsymbol{m}_x)^T\boldsymbol{x}\right\} \\
&\propto \left\{\frac{-1}{2}\begin{bmatrix}\boldsymbol{y}\\\boldsymbol{x}\end{bmatrix}^T \boldsymbol{J}_* \begin{bmatrix}\boldsymbol{y}\\\boldsymbol{x}\end{bmatrix} + \boldsymbol{h}_*^T \begin{bmatrix}\boldsymbol{y}\\\boldsymbol{x}\end{bmatrix}\right\}
\end{aligned}$$

where

$$\boldsymbol{J}_* = \begin{bmatrix}\boldsymbol{J}_v & -\boldsymbol{J}_v\boldsymbol{A}\\ -(\boldsymbol{J}_v\boldsymbol{A})^T & \boldsymbol{J}_x\end{bmatrix} \qquad \boldsymbol{h}_* = \begin{bmatrix}\boldsymbol{h}_v\\\boldsymbol{h}_x\end{bmatrix}$$

$$\boldsymbol{J}_* = \begin{bmatrix} \boldsymbol{J}_x & -\boldsymbol{J}_v\boldsymbol{A} \\ -(\boldsymbol{J}_v\boldsymbol{A})^T & \boldsymbol{J}_v \end{bmatrix} \qquad \boldsymbol{h}_* = \begin{bmatrix} \boldsymbol{h}_v \\ \boldsymbol{h}_x \end{bmatrix}$$

Marginalization its easy if we obtain the covariance form of the joint distribution. Applying the matrix inversion lemma:

$$\begin{bmatrix} \boldsymbol{J}_v & -\boldsymbol{J}_v\boldsymbol{A} \\ -(\boldsymbol{J}_v\boldsymbol{A})^T & \boldsymbol{J}_x \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{S}^{-1} & \boldsymbol{S}^{-1}\boldsymbol{J}_v\boldsymbol{A}\boldsymbol{J}_x^{-1} \\ \boldsymbol{J}_x^{-1}(\boldsymbol{J}_v\boldsymbol{A})^T\boldsymbol{S}^{-1} & \boldsymbol{J}_x^{-1} + \boldsymbol{J}_x^{-1}(\boldsymbol{J}_v\boldsymbol{A})^T\boldsymbol{S}^{-1}\boldsymbol{J}_v\boldsymbol{A}\boldsymbol{J}_x^{-1} \end{bmatrix}$$

where $\boldsymbol{S} = \boldsymbol{J}_v - \boldsymbol{J}_v\boldsymbol{A}\boldsymbol{J}_x^{-1}(\boldsymbol{J}_v\boldsymbol{A})^T$.

Therefore

$$p(\boldsymbol{y}) = \mathcal{N}^{-1}(\boldsymbol{h}_y, \boldsymbol{J}_y)$$

where

$$\boldsymbol{J}_y = \boldsymbol{J}_v - \boldsymbol{J}_v\boldsymbol{A}\boldsymbol{\Sigma}_x\boldsymbol{A}^T\boldsymbol{J}_v^T \qquad \boldsymbol{h}_y = \boldsymbol{h}_v + \boldsymbol{J}_v\boldsymbol{A}\boldsymbol{\Sigma}_x\boldsymbol{h}_x$$

Index

Let $X_i \in \mathcal{X}$ $i = 1, \dots, 5$ be a collection of real R.V. with joint distribution of the form

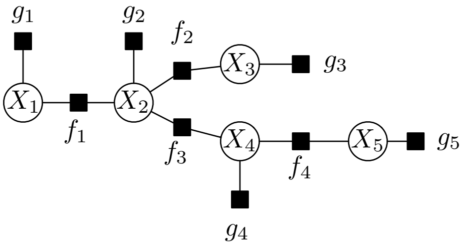
$$\mu(\boldsymbol{x}) = \frac{1}{Z} f_1(x_1, x_2) f_2(x_2, x_3) f_3(x_2, x_4) f_4(x_4, x_5) \prod_{i=1}^{5} g(x_i)$$

where

$$f_1(x_1, x_2) = \exp\{F_1 x_1 x_2\} \qquad f_2(x_2, x_3) = \exp\{F_2 x_2 x_3\}$$
$$f_3(x_2, x_4) = \exp\{F_3 x_2 x_4\} \qquad f_4(x_4, x_5) = \exp\{F_4 x_4 x_5\}$$

and

$$g(x_i) = \exp\{b_i x_i - \frac{1}{2}\pi_i x_i^2\}$$



- $\mu(\boldsymbol{x})$ is a Gaussian distribution defined in natural form!

$$\boldsymbol{J} = \begin{bmatrix} \pi_1 & -F_1 & 0 & 0 & 0 \\ -F_1 & \pi_2 & -F_2 & -F_3 & 0 \\ 0 & -F_2 & \pi_3 & 0 & 0 \\ 0 & -F_3 & 0 & \pi_4 & -F_4 \\ 0 & 0 & 0 & -F_4 & \pi_5 \end{bmatrix}$$

$$\boldsymbol{h} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix}^T$$

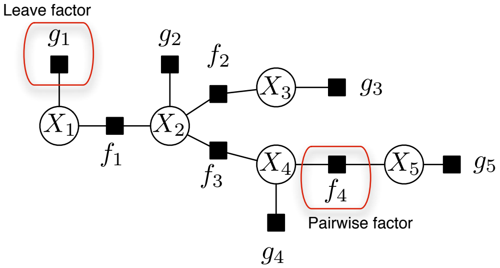
Computing the mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$ requires computing $\boldsymbol{\Sigma} = \boldsymbol{J}^{-1}$ and $\boldsymbol{m} = \boldsymbol{J}^{-1}\boldsymbol{h}$. $O(n^3)$ cost!

BP provides a way to exploit graph structure to perform this computation in $\mathcal{O}(n)$ time instead of $O(n^3)$. It is only exact when the graph is a tree!

If the Gaussian graphical model has cycles, GBP still computes the exact $\boldsymbol{m}$, but only an estimation to $\boldsymbol{\Sigma}$.

Gaussian BP algorithm

- It is described as a *message-passing algorithm*.
- Messages represent *local computations* at each node of the graph.
- The FG has leave factors (only connected to one variable) and pairwise factors (connected to two variables).
- Integrals instead of sums. The rest of the algorithm remains unaltered.



Iteration 0

- Messages send by variable nodes are initialized by their leave factors

E.g.

$$m^0_{x_2 \to f_2}(x_2) = g(x_2) = \exp\{b_2 x_2 - \frac{1}{2}\pi_2 x_2^2\} \sim \mathcal{N}^{-1}(b_2, \pi_2) \quad \text{Gaussian message!}$$

- Messages send by the f_j factors are computed as usual but replacing sums by integrals.

E.g.

$$\begin{aligned} m^0_{f_1 \to x_1}(x_1) &= \int \exp\{F_1 x_1 x_2\} \exp\{b_2 x_2 - \frac{1}{2}\pi_2 x_2^2\} dx_2 \\ &= \int \exp\Big\{ -\frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0 & -F_1 \\ -F_1 & \pi_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Big\} dx_2 \\ &\sim \mathcal{N}^{-1}(F_1 \pi_2^{-1} b_2, F_1^2 \pi_2^{-1}) = \mathcal{N}^{-1}(h^0_{f_1 \to x_1}, J^0_{f_1 \to x_1}) \quad \text{Gaussian message!} \end{aligned}$$

At iteration ℓ

- **Step 1:** Variable nodes multiply their incoming messages to send a new message to factors.

E.g.

$$\begin{aligned} m^{\ell}_{x_2\to f_3}(x_2) &= \mathcal{N}^{-1}(h^{\ell-1}_{f_1\to x_2}, J^{\ell-1}_{f_1\to x_2})\mathcal{N}^{-1}(h^{\ell-1}_{f_2\to x_2}, J^{\ell-1}_{f_2\to x_2}) \\ &= \mathcal{N}^{-1}(h^{\ell-1}_{f_1\to x_2} + h^{\ell-1}_{f_2\to x_2}, J^{\ell-1}_{f_1\to x_2} + J^{\ell-1}_{f_2\to x_2}) \\ &= \mathcal{N}^{-1}(h^{\ell}_{x_2\to f_3}, J^{\ell}_{x_2\to f_3}). \end{aligned}$$

At iteration ℓ

- **Step 2:** Factor nodes compute the messages to variable nodes by marginalization.

E.g.

$$m^{\ell}_{f_3\to x_4}(x_4) = \int \exp\{F_3 x_2 x_4\}\mathcal{N}^{-1}(h^{\ell}_{x_2\to f_3}, J^{\ell}_{x_2\to f_3})dx_2$$

$$= \int \exp\left\{ -\frac{1}{2}\begin{bmatrix} x_2 \\ x_4 \end{bmatrix}^T \begin{bmatrix} 0 & -F_3 \\ -F_3 & J^{\ell}_{x_2\to f_3} \end{bmatrix}\begin{bmatrix} x_2 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ h^{\ell}_{x_2\to f_3} \end{bmatrix}^T \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}\right\} dx_2$$

$$\sim \mathcal{N}^{-1}\left(\frac{F_3 h^{\ell}_{x_2\to f_3}}{J^{\ell}_{x_2\to f_3}}, \frac{F_3^2}{J^{\ell}_{x_2\to f_3}}\right) = \mathcal{N}^{-1}(h^{\ell}_{f_3\to x_4}, J^{\ell}_{f_3\to x_4}) \quad \text{Gaussian message!}$$

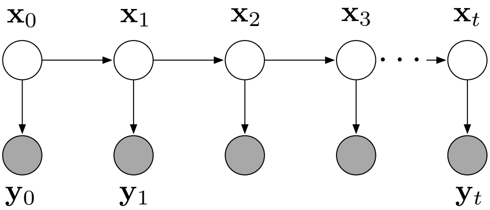
All messages are Gaussian! The parameters of the Gaussian messages are computed in closed form without doing any integral!

Convergence is guaranteed and achieved in a finite number ℓ_* of iterations. The overall complexity is $\mathcal{O}(\ell_* n)$.

Index

Gaussian Hidden Markov Models

- Wireless communication channels.
- Speech processing.
- Tracking applications.



Gaussian Hidden Markov Models

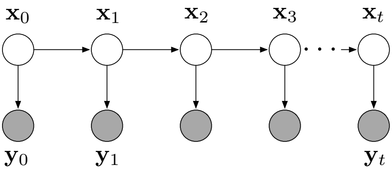
- States $\boldsymbol{x}_t \in \mathbb{R}^d$.
- State transition matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$.
- Process noise $\boldsymbol{v}_t \in \mathbb{R}^p$ and $\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_v)$ for some $\boldsymbol{\Sigma}_v \in \mathbb{R}^{p \times p}$.
- Dynamic equations:

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{v}_t$$
$$\boldsymbol{x}_0 \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x^0)$$

- Noisy observation $\boldsymbol{y}_t \in \mathbb{R}^{d'}$:

$$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{w}_t$$

where $\boldsymbol{C} \in \mathbb{R}^{d' \times d}$ and $\boldsymbol{w}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma}_w)$.



Gaussian Hidden Markov Models

- In summary, for $\boldsymbol{\Sigma}_h = \boldsymbol{B}\boldsymbol{\Sigma}_v\boldsymbol{B}^T$ we have

$$\begin{aligned}
\boldsymbol{x}_0 &\sim \mathcal{N}(0, \boldsymbol{\Sigma}_x^0) \\
\boldsymbol{x}_{t+1}|\boldsymbol{x}_t &\sim \mathcal{N}(\boldsymbol{A}\boldsymbol{x}_t, \boldsymbol{\Sigma}_h) \\
\boldsymbol{y}_t|\boldsymbol{x}_t &\sim \mathcal{N}(\boldsymbol{C}\boldsymbol{x}_t, \boldsymbol{\Sigma}_w)
\end{aligned}$$

- Factorization

$$\mu(\boldsymbol{x}, \boldsymbol{y}) = \mu(x_0)\mu(\boldsymbol{y}_0|x_0)\mu(\boldsymbol{x}_1|\boldsymbol{x}_0)\mu(\boldsymbol{y}_1|x_1)\mu(\boldsymbol{x}_2|\boldsymbol{x}_1)\mu(\boldsymbol{y}_2|x_2)\dots$$

- Gaussian graphical model with no cycles!! GaBP is exact and cheap!
- This factorization can be expanded as the product of leave factors and pairwise factors.

Inference over Gaussian HMMs

Forward/Backward algorithm

Given $\boldsymbol{y}_0, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_L$, use GaBP to compute the (Gaussian) marginal for each state $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L$.

Kalman filter

Given the actual observation $\boldsymbol{y}_t$ and the observations in the past $\boldsymbol{y}_0, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t - 1$, use GaBP to compute the mean of $\boldsymbol{x}_t$, $\mathbb{E}[\boldsymbol{x}_t]$.

Index

BP over any graphical models

- So far, we have applied BP to discrete and Gaussian graphical models.
- In the Gaussian case, we can avoid integration by algebraic manipulation.
- The same update rules can be applied to any Graphical model. However, integrals of the form

$$\int_{\boldsymbol{x}_j \sim x_i} f_j(\boldsymbol{x}_j) \prod_{u \in \mathcal{N}(f_j)} \boldsymbol{m}_{x_u \to f_j}(x_u) d(\boldsymbol{x}_j \sim x_i)$$

are intractable in general! we cannot solve the integrals!!

Approximate message passing (APM)

Approximate BP messages by Gaussian distributions:

- Compressed sensing `http://people.ee.duke.edu/~lcarin/AMP1.pdf`
- Efficient multiuser detection in CDMA systems `http://arxiv.org/pdf/0810.1729.pdf`
- Communications over Fading ISI channels: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04907469`
- ...

Alternatives

Instead of approximating the BP messages, construct directly a tractable Gaussian approximation to the joint pdf.

- Variational Inference and Mean Field.
- Expectation Propagation.