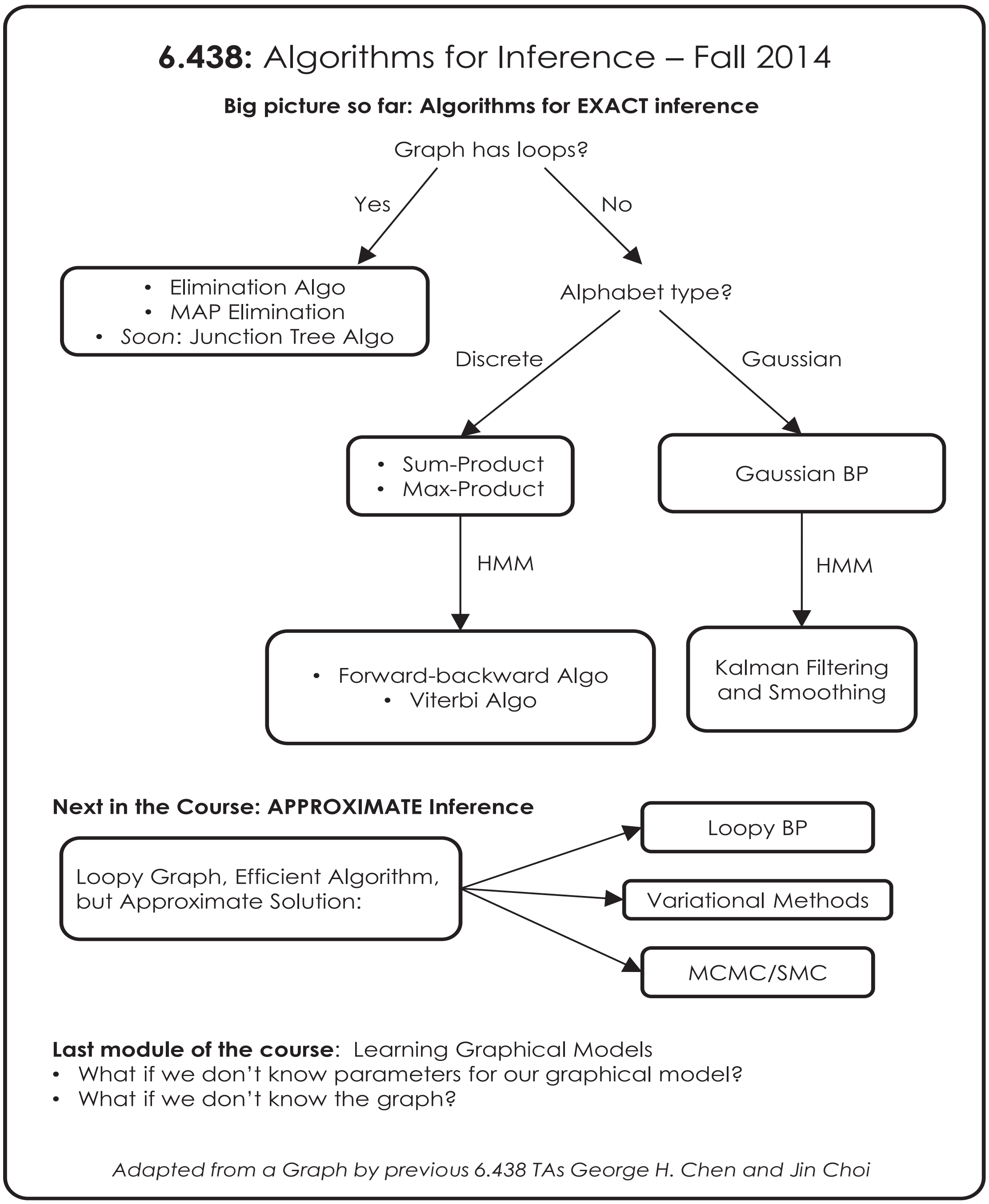
Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.438 Algorithms For Inference
Fall 2014

Recitation 7

1 Big Picture So Far



2 Gaussian BP

Exercise 1: Warm up

(a) If $\exp(-\frac{1}{2}\mathbf{x}_0^T J_0 \mathbf{x}_0 - \frac{1}{2}\mathbf{x}_1^T J_1 \mathbf{x}_1 - \mathbf{x}_1^T L \mathbf{x}_0 + h_0^T \mathbf{x}_0) \propto \mathcal{N}^{-1}(h, J)$, what is h and J?

Solution: $h = \begin{pmatrix} h_0 \\ \mathbf{0} \end{pmatrix}$, $J = \begin{pmatrix} J_0 & L^T \\ L & J_1 \end{pmatrix}$

(b) If $\phi_1(\mathbf{x}) \propto \mathcal{N}^{-1}(h_1, J_1)$ and $\phi_2(\mathbf{x}) \propto \mathcal{N}^{-1}(h_2, J_2)$, and $\phi_1(\mathbf{x})\phi_2(\mathbf{x}) \propto \mathcal{N}^{-1}(h, J)$. What is h and J?

Solution: $h = h_1 + h_2$, $J = J_1 + J_2$

(c) If $\mathbf{x} \sim \mathcal{N}(\mu_\mathbf{x}, \Sigma_\mathbf{x})$ and $\mathbf{y} \sim \mathcal{N}(\mu_\mathbf{y}, \Sigma_\mathbf{y})$, is it true that $\mathbf{x} + \mathbf{y} \sim \mathcal{N}(\mu_\mathbf{x} + \mu_\mathbf{y}, \Sigma_\mathbf{x} + \Sigma_\mathbf{y})$?

Solution: False. If $\mathbf{x}$ and $\mathbf{y}$ are not independent, $\mathbf{x} + \mathbf{y}$ does not necessarily have a Gaussian distribution. For example, consider $\mathbf{y} = -\mathbf{x}$, then the sum will be a deterministic value, not a variable.
However, if $\mathbf{x}$ is independent of $\mathbf{y}$, the statement is true because:

$$\begin{aligned}
&\mathbb{E}[\mathbf{x} + \mathbf{y}] = \mathbb{E}[\mathbf{x}] + \mathbb{E}[\mathbf{y}] = \mu_\mathbf{x} + \mu_\mathbf{y} \\
&\mathbb{E}[(\mathbf{x} + \mathbf{y} - \mu_\mathbf{x} - \mu_\mathbf{y})(\mathbf{x} + \mathbf{y} - \mu_\mathbf{x} - \mu_\mathbf{y})^T] \\
&= \mathbb{E}[(\mathbf{x} - \mu_\mathbf{x})(\mathbf{x} - \mu_\mathbf{x})^T] + \mathbb{E}[(\mathbf{x} - \mu_\mathbf{x})(\mathbf{y} - \mu_\mathbf{y})^T] + \mathbb{E}[(\mathbf{y} - \mu_\mathbf{y})(\mathbf{x} - \mu_\mathbf{x})^T] + \mathbb{E}[(\mathbf{y} - \mu_\mathbf{y})(\mathbf{y} - \mu_\mathbf{y})^T] \\
&= \Sigma_\mathbf{x} + \mathbb{E}[(\mathbf{x} - \mu_\mathbf{x})]\mathbb{E}[(\mathbf{y} - \mu_\mathbf{y})^T] + \mathbb{E}[(\mathbf{y} - \mu_\mathbf{y})]\mathbb{E}[(\mathbf{x} - \mu_\mathbf{x})^T] + \Sigma_\mathbf{y} \\
&= \Sigma_\mathbf{x} + \Sigma_\mathbf{y}
\end{aligned}$$

(d) $\int_\mathbf{y} \mathcal{N}^{-1}(\begin{pmatrix} h_\mathbf{x} \\ h_\mathbf{y} \end{pmatrix}, \begin{pmatrix} J_{\mathbf{xx}} & J_{\mathbf{xy}} \\ J_{\mathbf{yx}} & J_{\mathbf{yy}} \end{pmatrix}) d\mathbf{y} \propto \mathcal{N}^{-1}(h, J)$, What is h and J?

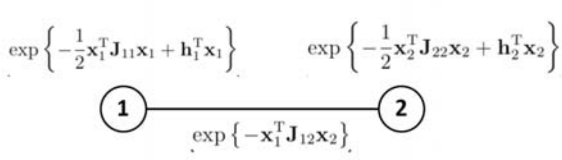
Solution: This is just marginalizing out $\mathbf{y}$.

$$h = h_\mathbf{x} - J_{\mathbf{xy}} J_{\mathbf{yy}}^{-1} h_\mathbf{y}$$

$$J = J_{\mathbf{xx}} - J_{\mathbf{xy}} J_{\mathbf{yy}}^{-1} J_{\mathbf{yx}}$$

Gaussian BP Equations

2 Nodes Case



Message

$$m_{2\to1}(\mathbf{x}_1) \propto \mathtt{N}^{-1}(\mathbf{x}_1; \mathbf{h}_{2\to1}, \mathbf{J}_{2\to1}),$$

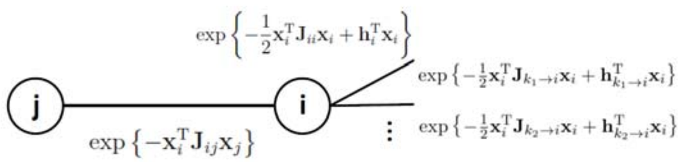
where

$$\mathbf{h}_{2\to 1} \triangleq -\mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{h}_2 \quad \text{and} \quad \mathbf{J}_{2\to 1} \triangleq -\mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{J}_{21}.$$

Marginal

$$\begin{aligned} p_{\mathbf{x}_1}(\mathbf{x}_1) &\propto \phi_1(\mathbf{x}_1)\, m_{2\to 1}(\mathbf{x}_1) \\ &\propto \mathtt{N}^{-1}(\mathbf{x}_1; \mathbf{h}_1 + \mathbf{h}_{2\to 1}, \mathbf{J}_{11} + \mathbf{J}_{2\to 1}). \end{aligned}$$

General Case: Undirected Tree



Message

$$m_{i\to j}(\mathbf{x}_j) \propto \mathtt{N}^{-1}(\mathbf{x}_j; \mathbf{h}_{i\to j}, \mathbf{J}_{i\to j}),$$

where

$$\begin{aligned} \mathbf{h}_{i\to j} &= -\mathbf{J}_{ji}\left(\mathbf{J}_{ii} + \sum_{k\in N(i)\setminus j}\mathbf{J}_{k\to i}\right)^{-1}\left(\mathbf{h}_i + \sum_{k\in N(i)\setminus j}\mathbf{h}_{k\to i}\right) \\ \mathbf{J}_{i\to j} &= -\mathbf{J}_{ji}\left(\mathbf{J}_{ii} + \sum_{k\in N(i)\setminus j}\mathbf{J}_{k\to i}\right)^{-1}\mathbf{J}_{ij}. \end{aligned}$$

Marginal

$$\begin{aligned} p_{\mathbf{x}_i}(\mathbf{x}_i) &\propto \phi_i(\mathbf{x}_i)\prod_{k\in N(i)} m_{k\to i}(\mathbf{x}_i) \\ &= \exp\left\{-\frac{1}{2}\mathbf{x}_i^{\mathrm{T}}\left(\mathbf{J}_{ii} + \sum_{k\in N(i)}\mathbf{J}_{k\to i}\right)\mathbf{x}_i + \left(\mathbf{h}_i + \sum_{k\in N(i)}\mathbf{h}_{k\to i}\right)^{\mathrm{T}}\mathbf{x}_i\right\} \\ &\propto \mathtt{N}^{-1}(\mathbf{h}_i + \sum_{k\in N(i)}\mathbf{h}_{k\to i}, \mathbf{J}_{ii} + \sum_{k\in N(i)}\mathbf{J}_{k\to i}) \end{aligned}$$

Gaussian BP and Gaussian Elimination

Recall in linear algebra, one standard method for solving system of linear equations of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ is *Gaussian elimination*. Row manipulations are performed to transform the matrix $\mathbf{A}$ into an upper triangular matrix, so the value of last element in $\mathbf{x}$ can be solved directly. Then back substitution is used to solve for the other elements of $\mathbf{x}$.

We claim that if matrix **A** has some special properties, we can turn to Gaussian BP algorithm to find an optimal ordering of the row manipulations such that the number of manipulations needed is as few as possible. To be more precise, if **A** is symmetric and semi-positive definite, and the undirected graph[1] corresponding to **A** is a tree, we will take **A** as the information matrix **J** and **b** as the potential vector **h**, and consider the message-passing schedule of running Gaussian BP algorithm on this graph. If we translate each message into a row manipulation, we will obtain the optimal sequence of row manipulations. Furthermore, the solution $\mathbf{x}$ is the mean vector μ in the Gaussian graphical model.

Moreover, notice that we only consider the ordering of row manipulations, not the actual values. After a few more thoughts, you should realize that we can relax the requirement on **A** to 'the non-zero pattern of **A** should be symmetric'[2]. But in this case, the solution $\mathbf{x}$ no longer has the interpretation of mean vector of the Gaussian GM.

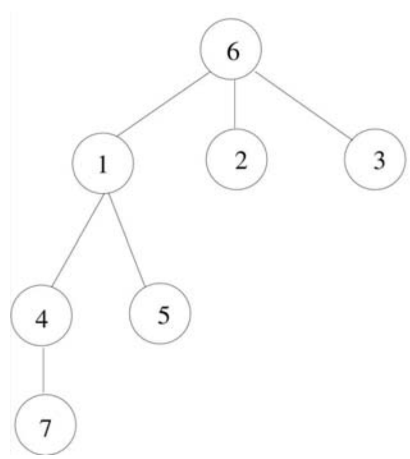
Exercise 2 illustrates the ideas in the above discussion.

Exercise 2: Gaussian BP and Gaussian Elimination

Solve the following system of equations. You should solve this system by hand rather than using Matlab. Explain why this is related to Gaussian belief propagation.

$$\begin{bmatrix} 1 & 0 & 0 & -4 & 1 & -3 & 0 \\ 0 & 4 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 3 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & -3 & 0 & 0 & 6 \end{bmatrix} \mathbf{x} = \begin{bmatrix} -32 \\ 32 \\ 8 \\ 24 \\ 5 \\ 12 \\ 12 \end{bmatrix}$$

Solution:



[1]This is the undirected graph obtained by adding an edge between node i and j if and only if $\ni_{ij} \neq 0$.
[2]All the diagonal entries need to be non-zero.

Notice that in our particular system, the sparsity pattern, i.e., the set of nonzero entries, does correspond to a tree. Thus, we will perform Gaussian elimination steps in the order suggested by the tree.[3]

We will work from the leaves in towards the root, and then back out to the leaves. For brevity, in what follows, for each row reduction step we will only write out the row of the matrix that actually changes.

Step 1: We use row three to eliminate an entry in row 6, i.e., replace row 6 with row $6 + \frac{1}{2}\cdot$ row 3. The new row 6 is

$$[1 \quad -1 \quad 0 \quad 0 \quad 0 \quad \tfrac{11}{2} \quad 0 \quad | \quad 16] \ .$$

The entry after the | represents the right hand side of the equation.

Step 2: We use row two to eliminate an entry in row 6, i.e., replace row 6 with row 6 + $\frac{1}{4}\cdot$ row 2. The new row 6 is

$$[1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \tfrac{23}{4} \quad 0 \quad | \quad 24] \ .$$

Step 3: We use row 7 to eliminate an entry in row 4, i.e., replace row 4 with row 4 - $\frac{1}{6}\cdot$ row 7. The new row 4 is

$$[1 \quad 0 \quad 0 \quad \tfrac{7}{2} \quad 0 \quad 0 \quad 0 \quad | \quad 22] \ .$$

Step 4: We use row 4 to eliminate an entry in row 1, i.e., replace row 1 with row 1 + $\frac{8}{7}\cdot$ row 4. The new row 1 is

$$[\tfrac{15}{7} \quad 0 \quad 0 \quad 0 \quad 1 \quad -3 \quad 0 \quad | \quad -\tfrac{48}{7}] \ .$$

Step 5: We use row 5 to eliminate an entry in row 1, i.e., replace row 1 with row 1 - row 5. The new row 1 is

$$[\tfrac{1}{7} \quad 0 \quad 0 \quad 0 \quad 0 \quad -3 \quad 0 \quad | \quad -\tfrac{83}{7}] \ .$$

Step 6: We use row 1 to eliminate an entry in row 6, i.e., replace row 6 with row 6 - 7$\cdot$ row 1. The new row 6 is

$$[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \tfrac{107}{4} \quad 0 \quad | \quad 107] \ .$$

We can now solve to get $x_6 = 4$.

Now comes the back substition phase. Of course, back substitution can be thought of as more row reduction steps to make the matrix diagonal, but instead of writing the back

[3] The root of the tree can be chosen arbitrarily. Here we assume node 6 is chosen as the root.

substition steps this way we will simply write the relevant equation that is solved in each step.

Step 7: Since x_6 is known, we get $x_2 = \frac{32-x_6}{4} = 7$.

Step 8: Since x_6 is known, we get $x_3 = \frac{8-x_6}{2} = 2$.

Step 9: Since x_6 is known and we eliminated the other parts of row 1, we get $x_1 = 21x_6 - 83 = 1$.

Step 10: Since x_1 is known, we get $x_5 = 5 - 2x_1 = 3$.

Step 11: Since x_1 is known and we eliminated the other parts of row 4, we get $x_4 = \frac{22-x_1}{\frac{7}{2}} = 6$.

Step 12: Since x_4 is known, we get $x_7 = \frac{12+3x_4}{6} = 5$.

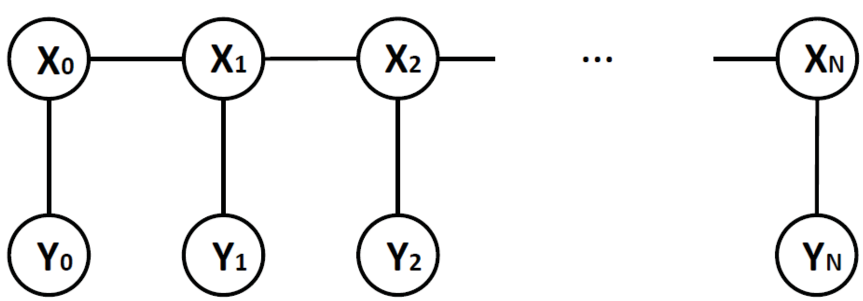
In summary,

$$\mathbf{x} = \begin{bmatrix} 1 & 7 & 2 & 6 & 3 & 4 & 5 \end{bmatrix}^T.$$

While it is arguable how much work was saved by doing Gaussian elimination in this order, hopefully this problem is enough to convince you that for large systems, this method will be much faster than doing Gaussian elimination in the standard order, i.e., where we would first zero out terms below the main diagonal and then do back substition.

3 Kalman Filtering and Smoothing

Setup



$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{G}\mathbf{v}_t, \text{ where } \mathbf{v}_t \sim \mathtt{N}(\mathbf{0}, \mathbf{Q}), \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{w}_t, \text{ where } \mathbf{w}_t \sim \mathtt{N}(\mathbf{0}, \mathbf{R}) \\ \mathbf{x}_0 &\sim \mathtt{N}(0, \Sigma_0) \end{aligned}$$

Some more notations:

$$\mu_{t|s} = \mathbb{E}[\mathbf{x}_t|\mathbf{y}_0, ..., \mathbf{y}_s]$$
$$\Sigma_{t|s} = \mathbb{E}[(\mathbf{x}_t - \mu_{t|s})(\mathbf{x}_t - \mu_{t|s})^T|\mathbf{y}_0, ..., \mathbf{y}_s]$$

Equations

Kalman filter was first introduced by Rudolf Kalman in the 1960s. Viewing it through the lens of inference algorithms, it is nothing more than Gaussian BP algorithm applied to Hidden Markov Models(HMM), with some regrouping of the computations.

Kalman filtering and smoothing, similar to other inference algorithms for HMM, contains two 'passes'. The 'forward pass', known as the *filtering step*, starts with some initial values $\mu_{0|0}$ and $\Sigma_{0|0}$, and then recursively compute $\mu_{t+1|t+1}$ and $\Sigma_{t+1|t+1}$ using $\mu_{t|t}$ and $\Sigma_{t|t}$. The 'backward pass', i.e. *smoothing*, on the other hand, starts with $\mu_{N|N}$ and $\Sigma_{N|N}$[4], and then recursively compute $\mu_{t|N}$ and $\Sigma_{t|N}$ using $\mu_{t+1|N}$ and $\Sigma_{t+1|N}$.

The filtering step is further divided into two smaller steps. First, a 'prediction' step aims to estimate the state $\mathbf{x}_{t+1}$ given observations $\mathbf{y}_0$, $\mathbf{y}_1$, ..., $\mathbf{y}_t$:

$$\mu_{t+1|t} = \mathbf{A}\mu_{t|t}$$
$$\Sigma_{t+1|t} = \mathbf{A}\Sigma_{t|t}A^T + \mathbf{GQG}^T$$

Notice these two equations follows directly from $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{G}\mathbf{v}_t$.

Secondly, to compute $\mu_{t+1|t+1}$ and $\Sigma_{t+1|t+1}$ from $\mu_{t+1|t}$ and $\Sigma_{t+1|t}$, we perform an 'adjustment' step:

$$\mu_{t+1|t+1} = \mu_{t+1|t} + \Sigma_{t+1|t}\mathbf{C}^T(\mathbf{C}\Sigma_{t+1|t}\mathbf{C}^T + \mathbf{R})^{-1}(\mathbf{y}_{t+1} - \mathbf{C}\mu_{t+1|t})$$
$$\Sigma_{t+1|t+1} = \Sigma_{t+1|t} - \Sigma_{t+1|t}\mathbf{C}^T(\mathbf{C}\Sigma_{t+1|t}\mathbf{C}^T + \mathbf{R})^{-1}\mathbf{C}\Sigma_{t+1|t}$$

Notice the mean $\mu_{t+1|t+1}$ is $\mu_{t+1|t}$ plus a term proportional to the 'error of prediction' $\mathbf{y}_{t+1} - \mathbf{C}\mu_{t+1|t}$, while $\Sigma_{t+1|t+1}$ equals $\Sigma_{t+1|t}$ minus some positive definite matrix. We do expect the covariance $\Sigma_{t+1|t+1}$ to decrease because we now have one extra observation $\mathbf{y}_{t+1}$.

The equations for the smoothing step are as follows:

$$\mu_{t|T} = \mu_{t|t} + \mathbf{L}_t(\mu_{t+1|N} - \mu_{t+1|t})$$
$$\Sigma_{t|T} = \Sigma_{t|t} + \mathbf{L}_t(\Sigma_{t+1|N} - \Sigma_{t+1|t})\mathbf{L}_t^T$$

where $\mathbf{L}_t = \Sigma_{t|t}\mathbf{A}^T\Sigma_{t+1|t}^{-1}$. Notice that terms $\mu_{t|t}$, $\mu_{t+1|t}$, $\Sigma_{t|t}$, and $\Sigma_{t+1|t}$ are already computed in the filtering step.

For those interested in how these equations are derived, a rigorous treatment can be found

[4] $\mu_{N|N}$ and $\Sigma_{N|N}$ are available after the filtering step.

in chapter 15 of Jordan's notes.

Summary: Inference Algorithms on HMM

The following table summarizes the variants of inference algorithms that compute marginal distributions on an HMM[5].

Discrete Variables	Gaussian Variables
Sum-Product: $m_{i\to i+1}(x_{i+1})$ $m_{i\to i-1}(x_{i-1})$	**Gaussian BP:** $m_{i\to i+1}(x_{i+1})$ $m_{i\to i-1}(x_{i-1})$
Forward-Backward α-β: $\alpha_i(x_i) = \mathbb{P}(y_1, ..., y_i, x_i)$ $\beta_i(x_i) = \mathbb{P}(y_{i+1}, ..., y_N \vert x_i)$	**'2 Filter Smoother':** $\mathbb{P}(x_i \vert y_0, ..., y_i)$ $\mathbb{P}(x_i \vert y_{i+1}, ..., y_N)$
α-γ Algorithm: $\alpha_i(x_i) = \mathbb{P}(y_1, ..., y_i, x_i)$ $\gamma_i(x_i) = \mathbb{P}(x_i \vert y_1, ..., y_N)$	**Kalman Filtering and Smoothing:** $\mathbb{P}(x_i \vert y_0, ..., y_i)$ $\mathbb{P}(x_i \vert y_0, ..., y_N)$

[5] Details about '2-filter smoother' can be found in Jordan notes section 15.7.2.

6.438 Algorithms for Inference
Fall 2014