# Lecture 8 and 9: Message passing on Factor Graphs

Machine Learning 4F13, Michaelmas 2015

Zoubin Ghahramani

Department of Engineering
University of Cambridge

http://mlg.eng.cam.ac.uk/teaching/4f13/

# Key Concepts

- Factor graphs are a class of graphical model
- A factor graph represents the product structure of a function, and contains factor nodes and variable nodes
- We can compute marginals and conditionals efficiently by passing messages on the factor graph, this is called the sum-product algorithm (a.k.a. belief propagation or factor-graph propagation)
- We can apply this to the True Skill graph
- But certain messages need to be approximated
- One approximation method based on moment matching is called Expectation Propagation (EP)

# Factor Graphs

Factor graphs are a type of *probabilistic graphical model* (others are directed graphs, a.k.a. Bayesian networks, and undirected graphs, a.k.a. Markov networks)

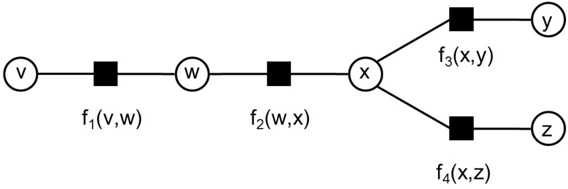
Factor graphs allow to represent the product structure of a function.

Example: consider the factorising probability distribution:

$$p(v, w, x, y, z) = f_1(v, w)f_2(w, x)f_3(x, y)f_4(x, z)$$

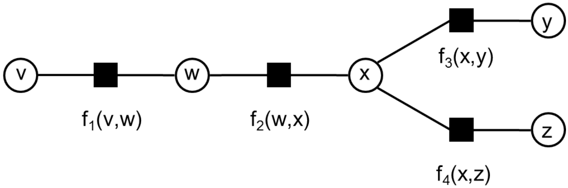
A factor graph is a bipartite graph with two types of nodes:

- Factor node: ■ Variable node: ○
- Edges represent the dependency of factors on variables.



# Factor Graphs

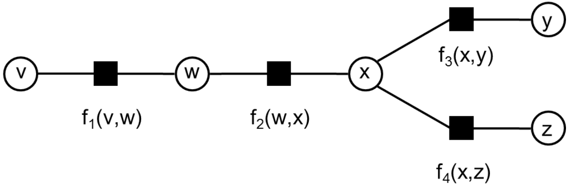
$$p(v, w, x, y, z) = f_1(v, w)f_2(w, x)f_3(x, y)f_4(x, z)$$




- What are the marginal distributions of the individual variables?
- What is $p(w)$?
- How do we compute conditional distributions, e.g. $p(w|y)$?

For now, we will focus on *tree-structured* factor graphs.

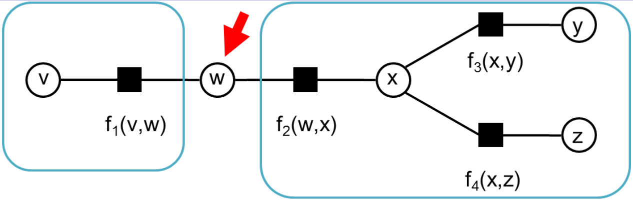
# Factor trees: separation (1)




$$p(w) = \sum_v \sum_x \sum_y \sum_z f_1(v,w) f_2(w,x) f_3(x,y) f_4(x,z)$$

- If $w$, $v$, $x$, $y$ and $z$ take $K$ values each, we have $\approx 3K^4$ products and $\approx K^4$ sums, for each value of $w$, i.e. total $\mathcal{O}(K^5)$.
- Multiplication is distributive: $ca + cb = c(a+b)$.
  The right hand side is more efficient!

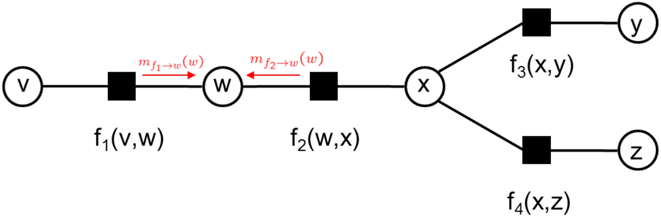
# Factor trees: separation (2)




$$
\begin{aligned}
p(w) &= \sum_v \sum_x \sum_y \sum_z f_1(v,w) f_2(w,x) f_3(x,y) f_4(x,z) \\
&= \left[\sum_v f_1(v,w)\right] \cdot \left[\sum_x \sum_y \sum_z f_2(w,x) f_3(x,y) f_4(x,z)\right]
\end{aligned}
$$

- In a tree, each node separates the graph into disjoint parts.
- Grouping terms, we go from sums of products to products of sums.
- The complexity is now $\mathcal{O}(K^4)$.

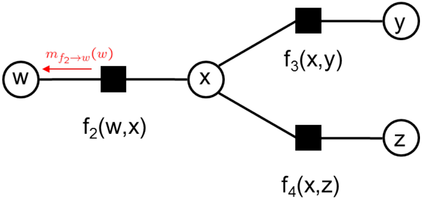
# Factor trees: separation (3)




$$p(w) = \underbrace{\left[\sum_v f_1(v, w)\right]}_{m_{f_1 \to w}(w)} \cdot \underbrace{\left[\sum_x \sum_y \sum_z f_2(w, x) f_3(x, y) f_4(x, z)\right]}_{m_{f_2 \to w}(w)}$$

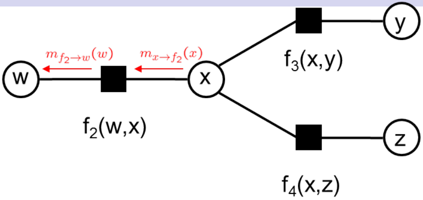
- Sums of products becomes products of sums of all messages from neighbouring factors to variable.

# Messages: from factors to variables (1)




$$m_{f_2 \to w}(w) = \sum_x \sum_y \sum_z f_2(w,x) f_3(x,y) f_4(x,z)$$

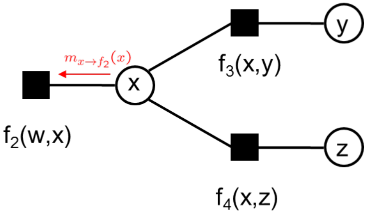
# Messages: from factors to variables (2)




$$
\begin{aligned}
m_{f_2\to w}(w) &= \sum_x \sum_y \sum_z f_2(w,x) f_3(x,y) f_4(x,z) \\
&= \sum_x f_2(w,x) \cdot \underbrace{\left[\sum_y \sum_z f_3(x,y) f_4(x,z)\right]}_{m_{x\to f_2}(x)}
\end{aligned}
$$

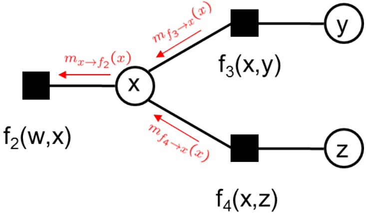
- Factors only need to sum out all their local variables.

# Messages: from variables to factors (1)




$$m_{x\to f_2}(x) = \sum_y \sum_z f_3(x,y) f_4(x,z)$$

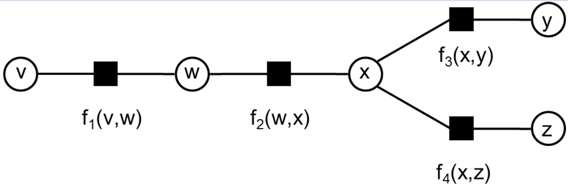
# Messages: from variables to factors (2)




$$
\begin{aligned}
m_{x\to f_2}(x) &= \sum_y \sum_z f_3(x,y) f_4(x,z) \\
&= \underbrace{\left[\sum_y f_3(x,y)\right]}_{m_{f_3\to x}(x)} \cdot \underbrace{\left[\sum_z f_4(x,z)\right]}_{m_{f_4\to x}(x)}
\end{aligned}
$$

- Variables pass on the product of all incoming messages.

# Factor graph marginalisation: summary




$$
\begin{aligned}
p(w) &= \sum_v \sum_x \sum_y \sum_z f_1(v,w) f_2(w,x) f_3(x,y) f_4(x,z) \\
&= \underbrace{\Big[\sum_v f_1(v,w)\Big]}_{m_{f_1 \to w}(w)} \cdot \underbrace{\Big[\sum_x f_2(w,x) \cdot \underbrace{\Big[\underbrace{\Big[\sum_y f_3(x,y)\Big]}_{m_{f_3 \to x}(x)} \cdot \underbrace{\Big[\sum_z f_4(x,z)\Big]}_{m_{f_4 \to x}(x)}\Big]}_{m_{x \to f_2}(x)}\Big]}_{m_{f_2 \to w}(w)}
\end{aligned}
$$

- The complexity is reduced from $\mathcal{O}(K^5)$ (naïve implementation) to $\mathcal{O}(K^2)$.

# The sum-product algorithm

Three update equations:

- Marginals are the product of all incoming messages from neighbour factors

$$p(t) = \prod_{f \in F_t} m_{f \to t}(t)$$

- Messages from factors sum out all variables except the receiving one

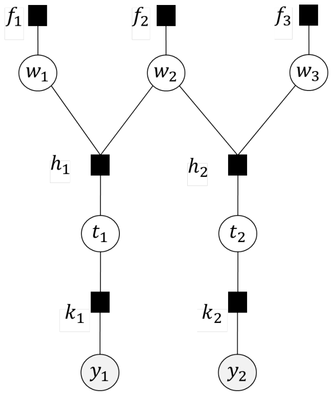
$$m_{f \to t_1}(t_1) = \sum_{t_2} \sum_{t_3} \dots \sum_{t_n} f(t_1, t_2, \dots, t_n) \prod_{i \neq 1} m_{t_i \to f}(t_i)$$

- Messages from variables are the product of all incoming messages except the message from the receiving factor

$$m_{t \to f}(t) = \prod_{f_j \in F_t \setminus \{f\}} m_{f_j \to t}(t) = \frac{p(t)}{m_{f \to t}(t)}$$

Messages are results of partial computations. Computations are localised.

# The full TrueSkill graph




Prior factors: $f_i(w_i) = \mathcal{N}(w_i; \mu_0, \sigma_0^2)$

"Game" factors:

$$h_g(w_{I_g}, w_{J_g}, t_g) = \mathcal{N}(t_g; w_{I_g} - w_{J_g}, 1)$$

($I_g$ and $J_g$ are the players in game $g$)

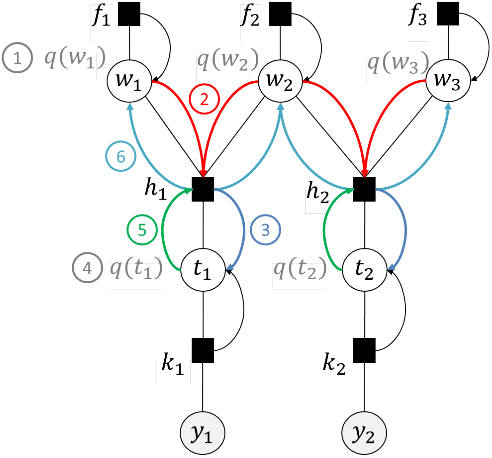
Outcome factors:

$$k_g(t_g, y_g) = \delta\big(y_g - \text{sign}(t_g)\big)$$

We are interested in the marginal distributions of the skills $w_i$.

- What shape do these distributions have?
- We need to make some approximations.
- We will also pretend the structure is a tree (ignore loops).

# Expectation Propagation in the full TrueSkill graph




Iterate

(1) Update skill marginals.

(2) Compute skill to game messages.

(3) Compute game to performance messages.

(4) Approximate performance marginals.

(5) Compute performance to game messages.

(6) Compute game to skill messages.

# Message passing for TrueSkill

$$
\begin{aligned}
m^{\tau=0}_{h_g \to w_{I_g}}(w_{I_g}) &= 1, \quad m^{\tau=0}_{h_g \to w_{J_g}}(w_{J_g}) = 1, \quad \forall\, g, \\
q^{\tau}(w_i) &= f(w_i) \prod_{g=1}^{N} m^{\tau}_{h_g \to w_i}(w_i) \;\sim\; \mathcal{N}(\mu_i, \sigma_i^2), \\
m^{\tau}_{w_{I_g} \to h_g}(w_{I_g}) &= \frac{q^{\tau}(w_{I_g})}{m^{\tau}_{h_g \to w_{I_g}}(w_{I_g})}, \quad m^{\tau}_{w_{J_g} \to h_g}(w_{J_g}) = \frac{q^{\tau}(w_{J_g})}{m^{\tau}_{h_g \to w_{J_g}}(w_{J_g})}, \\
m^{\tau}_{h_g \to t_g}(t_g) &= \iint h_g(t_g, w_{I_g}, w_{J_g})\, m^{\tau}_{w_{I_g} \to h_g}(w_{I_g})\, m^{\tau}_{w_{J_g} \to h_g}(w_{J_g})\, dw_{I_g}\, dw_{J_g}, \\
q^{\tau+1}(t_g) &= \mathrm{Approx}\big(m^{\tau}_{h_g \to t_g}(t_g)\, m_{k_g \to t_g}(t_g)\big), \\
m^{\tau+1}_{t_g \to h_g}(t_g) &= \frac{q^{\tau+1}(t_g)}{m^{\tau}_{h_g \to t_g}(t_g)}, \\
m^{\tau+1}_{h_g \to w_{I_g}}(w_{I_g}) &= \iint h_g(t_g, w_{I_g}, w_{J_g})\, m^{\tau+1}_{t_g \to h_g}(t_g)\, m^{\tau}_{w_{J_g} \to h_g}(w_{J_g})\, dt_g\, dw_{J_g}, \\
m^{\tau+1}_{h_g \to w_{J_g}}(w_{J_g}) &= \iint h_g(t_g, w_{J_g}, w_{J_g})\, m^{\tau+1}_{t_g \to h_g}(t_g)\, m^{\tau}_{w_{I_g} \to h_g}(w_{I_g})\, dt_g\, dw_{I_g}.
\end{aligned}
$$

# In a little more detail

At iteration $\tau$ messages $m$ and marginals $q$ are Gaussian, with *means* $\mu$, *standard deviations* $\sigma$, *variances* $\nu = \sigma^2$, *precisions* $r = \nu^{-1}$ and *natural means* $\lambda = r\mu$.

**Step 0** Initialise incoming skill messages:

$$\left.\begin{array}{rcl} r^{\tau=0}_{h_g \to w_i} & = & 0 \\ \mu^{\tau=0}_{h_g \to w_i} & = & 0 \end{array}\right\} m^{\tau=0}_{h_g \to w_i}(w_i)$$

**Step 1** Compute marginal skills:

$$\left.\begin{array}{rcl} r^{\tau}_{i} & = & r_0 + \sum_g r^{\tau}_{h_g \to w_i} \\ \lambda^{\tau}_{i} & = & \lambda_0 + \sum_g \lambda^{\tau}_{h_g \to w_i} \end{array}\right\} q^{\tau}(w_i)$$

**Step 2** Compute skill to game messages:

$$\left.\begin{array}{rcl} r^{\tau}_{w_i \to h_g} & = & r^{\tau}_{i} - r^{\tau}_{h_g \to w_i} \\ \lambda^{\tau}_{w_i \to h_g} & = & \lambda^{\tau}_{i} - \lambda^{\tau}_{h_g \to w_i} \end{array}\right\} m^{\tau}_{w_i \to h_g}(w_i)$$

**Step 3** Game to performance messages:

$$\left.\begin{aligned} v^{\tau}_{h_g \to t_g} &= 1 + v^{\tau}_{w_{I_g} \to h_g} + v^{\tau}_{w_{J_g} \to h_g} \\ \mu^{\tau}_{h_g \to t_g} &= \mu^{\tau}_{I_g \to h_g} - \mu^{\tau}_{J_g \to h_g} \end{aligned}\right\} m^{\tau}_{h_g \to t_g}(t_g)$$

**Step 4** Compute marginal performances:

$$\begin{aligned} p(t_g) &\propto \mathcal{N}(\mu^{\tau}_{h_g \to t_g}, v^{\tau}_{h_g \to t_g})\mathbb{I}\big(y - \text{sign}(t)\big) \\ &\simeq \mathcal{N}(\tilde{\mu}^{\tau+1}_g, \tilde{v}^{\tau+1}_g) \ = \ q^{\tau+1}(t_g) \end{aligned}$$

We find the parameters of $q$ by *moment matching*

$$\left.\begin{aligned} \tilde{v}^{\tau+1}_g &= v^{\tau}_{h_g \to t_g}\Big(1 - \Lambda\Big(\frac{\mu^{\tau}_{h_g \to t_g}}{\sigma^{\tau}_{h_g \to t_g}}\Big)\Big) \\ \tilde{\mu}^{\tau+1}_g &= \mu^{\tau}_{h_g \to t_g} + \sigma^{\tau}_{h_g \to t_g}\Psi\Big(\frac{\mu^{\tau}_{h_g \to t_g}}{\sigma^{\tau}_{h_g \to t_g}}\Big) \end{aligned}\right\} q^{\tau+1}(t_g)$$

where we have defined $\Psi(x) = \mathcal{N}(x)/\Phi(x)$ and $\Lambda(x) = \Psi(x)(\Psi(x) + x)$.

**Step 5** Performance to game message:

$$\left.\begin{aligned} r^{\tau+1}_{t_g \to h_g} &= \tilde{r}^{\tau+1}_g - r^{\tau}_{h_g \to t_g} \\ \lambda^{\tau+1}_{t_g \to h_g} &= \tilde{\lambda}^{\tau+1}_g - \lambda^{\tau}_{h_g \to t_g} \end{aligned}\right\} m^{\tau+1}_{t_g \to h_g}(t_g)$$

**Step 6** Game to skill message:
For player 1 (the winner):

$$\left.\begin{aligned} \nu^{\tau+1}_{h_g \to w_{I_g}} &= 1 + \nu^{\tau+1}_{t_g \to h_g} + \nu^{\tau}_{w_{J_g} \to h_g} \\ \mu^{\tau+1}_{h_g \to w_{I_g}} &= \mu^{\tau}_{w_{J_g} \to h_g} + \mu^{\tau+1}_{t_g \to h_g} \end{aligned}\right\} m^{\tau+1}_{h_g \to w_{I_g}}(w_{I_g})$$

and for player 2 (the looser):

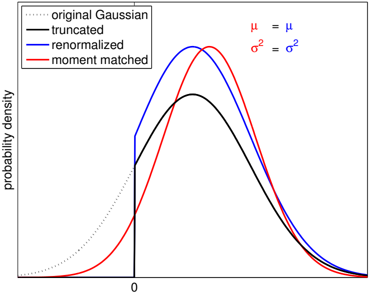
$$\left.\begin{aligned} \nu^{\tau+1}_{h_g \to w_{J_g}} &= 1 + \nu^{\tau+1}_{t_g \to h_g} + \nu^{\tau}_{w_{I_g} \to h_g} \\ \mu^{\tau+1}_{h_g \to w_{J_g}} &= \mu^{\tau}_{w_{I_g} \to h_g} - \mu^{\tau+1}_{t_g \to h_g} \end{aligned}\right\} m^{\tau+1}_{h_g \to w_{J_g}}(w_{J_g})$$

Go back to **Step 1** with $\tau := \tau + 1$ (or stop).

# Moments of a truncated Gaussian density (1)

Consider the truncated Gaussian density function

$$p(t) \;=\; \frac{1}{Z_t}\delta\big(y - \mathrm{sign}(t)\big)\mathcal{N}(t;\mu,\sigma^2) \;\text{ where }\; y \in \{-1, 1\} \;\text{ and }\; \delta(x) = 1 \;\text{ iff }\; x = 0.$$




We want to *approximate* $p(t)$ by a Gaussian density function $q(t)$ with mean and variance equal to the first and second central moments of $p(t)$. We need:

- First moment: $\mathbb{E}[t] = \langle t \rangle_{p(t)}$
- Second central moment: $\mathbb{V}[t] = \langle t^2 \rangle_{p(t)} - \langle t \rangle^2_{p(t)}$

# Moments of a truncated Gaussian density (2)

We have seen that the normalisation constant is $Z_t = \Phi\big(\frac{y\mu}{\sigma}\big)$.

**First moment.** We take the derivative of $Z_t$ wrt. $\mu$:

$$\begin{aligned}\frac{\partial Z_t}{\partial \mu} &= \frac{\partial}{\partial \mu}\int_0^{+\infty} N(t; y\mu, \sigma^2)dt = \int_0^{+\infty} \frac{\partial}{\partial \mu} N(t; y\mu, \sigma^2)dt \\ &= \int_0^{+\infty} y\sigma^{-2}(t - y\mu)N(t; y\mu, \sigma^2)dt = yZ_t\sigma^{-2}\int_{-\infty}^{+\infty}(t - y\mu)p(t)dt \\ &= yZ_t\sigma^{-2}\langle t - y\mu\rangle_{p(t)} = yZ_t\sigma^{-2}\langle t\rangle_{p(t)} - \mu Z_t\sigma^{-2}\end{aligned}$$

where $\langle t\rangle_{p(t)}$ is the expectation of $t$ under $p(t)$. We can also write:

$$\frac{\partial Z_t}{\partial \mu} = \frac{\partial}{\partial \mu}\Phi\big(\frac{y\mu}{\sigma}\big) = y\mathcal{N}(y\mu; 0, \sigma^2)$$

Combining both expressions for $\frac{\partial Z_t}{\partial \mu}$ we obtain

$$\langle t\rangle_{p(t)} = y\mu + \sigma^2\frac{\mathcal{N}(y\mu; 0, \sigma^2)}{\Phi(\frac{y\mu}{\sigma})} = y\mu + \sigma\frac{\mathcal{N}(\frac{y\mu}{\sigma}; 0, 1)}{\Phi(\frac{y\mu}{\sigma})} = y\mu + \sigma\Psi\big(\frac{y\mu}{\sigma}\big)$$

where use $\mathcal{N}(y\mu; 0, \sigma^2) = \sigma^{-1}\mathcal{N}(\frac{y\mu}{\sigma}; 0, 1)$ and define $\Psi(z) = \frac{N(z;0,1)}{\Phi(z)}$.

# Moments of a truncated Gaussian density (3)

**Second moment.** We take the second derivative of $Z_t$ wrt. $\mu$:

$$\frac{\partial^2 Z_t}{\partial \mu^2} = \frac{\partial}{\partial \mu} \int_0^{+\infty} y\sigma^{-2}(t - y\mu)N(t; y\mu, \sigma^2)dt$$
$$= \Phi\left(\frac{y\mu}{\sigma}\right)\langle -\sigma^{-2} + \sigma^{-4}(t - y\mu)^2 \rangle_{p(t)}$$

We can also write

$$\frac{\partial^2 Z_t}{\partial \mu^2} = \frac{\partial}{\partial \mu} y\mathcal{N}(y\mu; 0, \sigma^2) = -\sigma^{-2} y\mu\mathcal{N}(y\mu; 0, \sigma^2)$$

Combining both we obtain

$$\mathbb{V}[t] = \sigma^2\left(1 - \Lambda\left(\frac{y\mu}{\sigma}\right)\right)$$

where we define $\Lambda(z) = \Psi(z)\big(\Psi(z) + z\big)$.