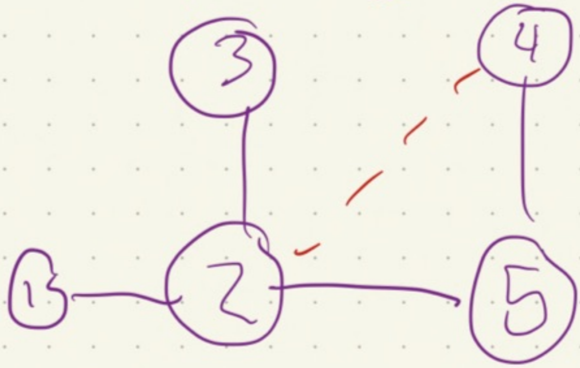
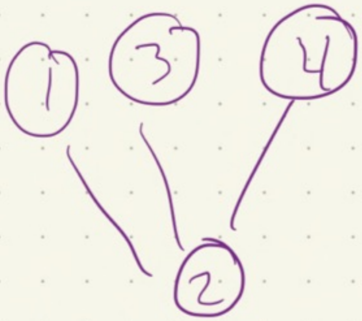
Elimination on a tree ⇒ Sum-Product algorithm
(loopy) belief prop

Def A *tree* is a graph $G = (V, E)$ which
- Is connected ($\exists$ path from i to j $\forall i,j$)
- No cycles

$n-1$ edges on n vertices ⇐



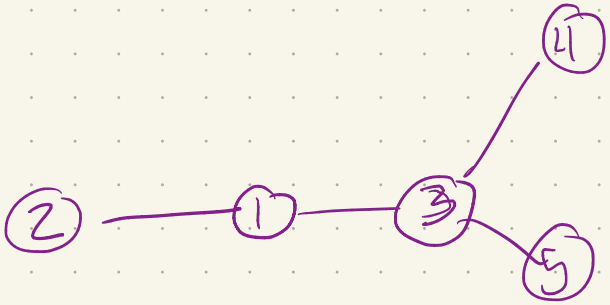
Wrong Elim ordering:
(5, 4, 3, 2, 1)



⇒ $\tilde{E}$ contains K_3

"Obvious" elimination ordering: eliminate leaves! (nodes w degree 1)

~~4, 5, 3, 2~~, 1



Elim ordering
(4, 5, 3, 2, 1)

// belief from 4→3

$\boxed{m_{4\to3}(x_3)}$

$$P(x_1) \propto \sum_{x_2,x_3,x_5} f_{12} f_{13} f_{35} \sum_{x_4} f_{34}(x_3, x_4)$$

If 

→ were whole graph

$m_{4\to3}(x_3) \propto p(x_3)$

$$\propto \sum_{x_2,x_3} f_{12} f_{13} m_{4\to3}(x_3) \sum_{x_5} f_{35}(x_3, x_5)$$

$m_{5\to3}(x_3)$

$$\propto \sum_{x_2} f_{12} \sum_{x_3} f_{13} m_{4\to3}(x_3) m_{5\to3}(x_3)$$

$m_{3\to1}(x_1)$, depends on
$m_{4\to3}$
$m_{5\to3}$

$$\propto m_{3\to1}(x_1) \sum_{x_2} f_{12}(x_1, x_2)$$

$m_{2\to1}(x_1)$

$$\propto m_{3\to1}(x_1) m_{2\to1}(x_1)$$

$m_1(x_1)$

So, $P(X_1 = x_1) = \dfrac{m_1(x_1)}{\sum_{x_1'} m_1(x_1')}$

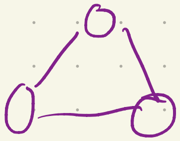
Message Passing on a graph

- Applied to any G
- Only works well for Pairwise MRF
- Exact if G is a tree. O/w, it's an appx (HW2)

$$P(x) = \frac{1}{Z} \prod_{(i,j)\in E} f_{ij}(x_i, x_j)$$

[General MRFs aren't like this]

- Still need an ordering for elim
- Messages can be re-used over $P(x_1), P(x_2)$ e.g.
- Lots of parallelization opportunities



Sum-Product Alg

Input: $G = (V, E)$, $f_{ij}(x_i, x_j)$ $\mathcal{X}$, T = # iterations

Output: $\{\hat{P}(x_i)\}_{i=1}^{n}$

- Initialize messages $\{m_{i\to j}(x_j), m_{j\to i}(x_i)\}_{(i,j)\in E}$ Random
- For $t = 1 \ldots T$, $\forall (i,j)\in E$ \& $(j,i)\in E$
  - Update $m_{i\to j}(x_j) = \sum_{x_i} f_{ij}(x_i, x_j) \prod_{\ell\in\partial i\setminus\{j\}} m_{\ell\to i}(x_i)$
  - $\forall i \in V$

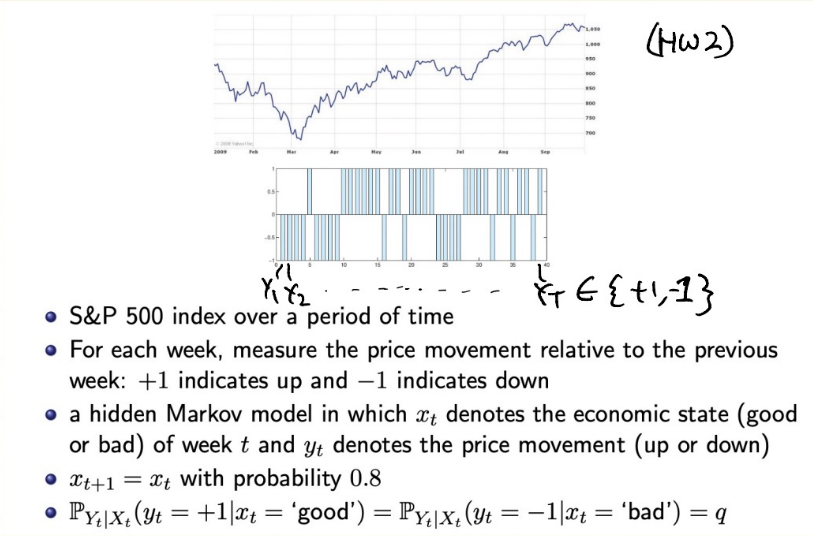
$$m_i(x_i) = \prod_{\ell\in\partial i} m_{\ell\to i}(x_i)$$

$$\hat{P}_i(x_i) = \frac{m_i(x_i)}{\sum m_i(x_i')}$$

If G is a tree of diameter d, $T \geq d \Rightarrow \tilde{P}_i(x_i) = P_i(x_i)$

\# of longestpath

HMMs


(HW2)

$Y_1 Y_2 \dots Y_T \in \{+1,-1\}$

- S&P 500 index over a period of time
- For each week, measure the price movement relative to the previous week: +1 indicates up and −1 indicates down
- a hidden Markov model in which x_t denotes the economic state (good or bad) of week t and y_t denotes the price movement (up or down)
- $x_{t+1} = x_t$ with probability 0.8
- $\mathbb{P}_{Y_t|X_t}(y_t = +1 | x_t = \text{'good'}) = \mathbb{P}_{Y_t|X_t}(y_t = -1 | x_t = \text{'bad'}) = q$

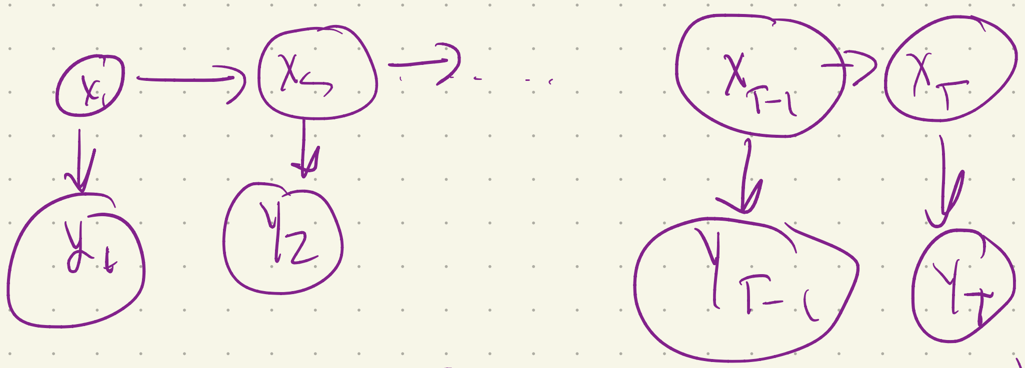
$X_t \in \{Good, Bad\}$

	G	B
$P(X_{t+1} \mid X_t = G$	.8	.2
$X_t = B$	.2	.8

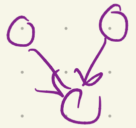
$P_{Y_t|X_t} =$

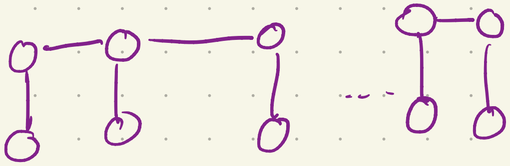
Y_t \ X_t	+1	-1
+1	q	1-q
-1	1-q	q

BN



① Moralize BN to get MRF (no v's!


)



② Write Factor

$$P(x) \propto \prod_{i \in T-1} f_{i,i+1}(x_i, x_{i+1}) = \begin{cases} P(x_{i+1}|x_i)\,P(y_i|x_i) & i < T-1 \\ P(x_T|x_{T-1})\,P(y_T|x_T)\,P(y_{T-1}|x_{T-1}) & i = T-1 \end{cases}$$

③ Write BP update

$$m_{i\to i+1}(x_{i+1}) = \sum_{x_i} f_{i,i+1}(x_i, x_{i+1})\, m_{i-1\to i}(x_i)$$

$$m_{i+1\to i}(x_i) = \sum_{x_{i+1}} f_{i,i+1}(x_i, x_{i+1})\, m_{i+2\to i+1}(x_{i+1})$$

$$m_i(x_i) = m_{i-1\to i}(x_i)\, m_{i+1\to i}(x_i)$$

Can do for marginals (Sum-Prod)
or $\max_x P(x)$ (Max - Prod)

$$\tilde{P}_i(x_i) \triangleq \max_{x_{-i}} P(x)$$

Update rule : $m^*_{i\to j}(x_j) = \max_{x_i} f_{ij}(x_i, x_j) \prod_{l\in\partial i\setminus\{j\}} m^*_{l\to i}(x_i)$

$$x^*_{i\to j}(x_j) = \operatorname{argmax}_{x_i} f_{ij}(x_i\, x_j) \prod_{l\in\partial i\setminus\{j\}} m^*_{l\to i}(x_i)$$

Max-marg $m^*_i(x_i) = \prod_{l\in\partial_i} m^*_{l\to i}(x_i)$

$$\tilde{P}(x_i) = \frac{m^*_i(x_i)}{\sum_{x_i'} m^*_i(x_i')}$$

To get x^*, fix x_i^* & backtrack.

Exact if G is tree
T > diam

Inference Recap

Marginalization	Maximization	E/Apx	Which graphs	C.C.
Elim Algo		Exact	any G	$\Theta(\lvert X\rvert^{\text{treew}})$ if opt ordering but opt ordering is NP-Hard
Sum-Prod	Max-prod	Apx	Pairwise MRFs	$O(\lvert X\rvert^2)$
Sum-Prod on Factor Graphs	Max-prod on Factor Graphs	Apx	Any G	$O(\lvert X\rvert^{\text{max-deg}})$
EA on junction trees →		Exact	Junction trees	$O(\lvert X\rvert^{Tw})$

* Belief Propagation for factor graphs.

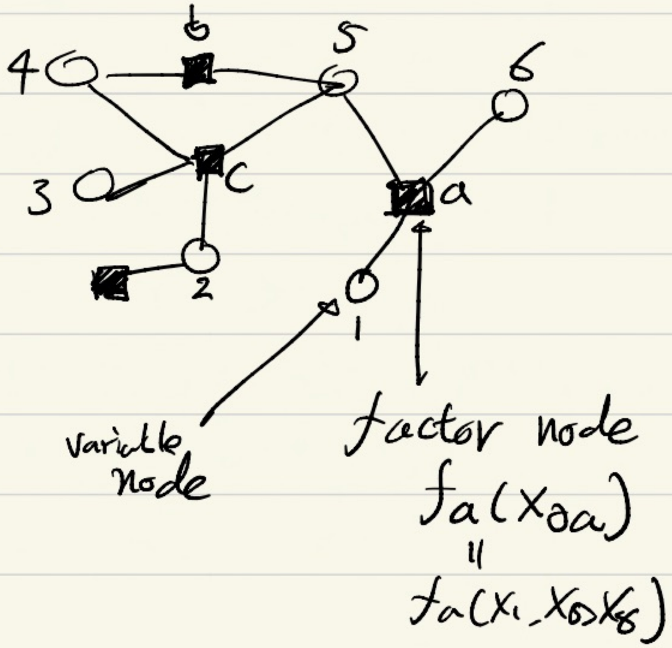
Def. factor graph $G(V \cup F, E)$

V: variable nodes

W: factor nodes

a probability distribution factorizes according to G if:

$$P(x) = \frac{1}{Z} \prod_{a \in F} f_a(X_{\partial a})$$



- factor graphs are strict generalizations of MRFs & can encode more detailed factorizations.
  fine grained

- FGs have the same Markov property as MRFs.

A-B-C separated in $G = (V \cup F, E)$, then $X_A \perp\!\!\!\perp X_C \mid X_B$.

Belief Propagation

- Sum-Product Algorithm on factor graphs

recall: $E = \{(\underset{\text{variable}}{i}, \underset{\text{factor}}{a})\}$

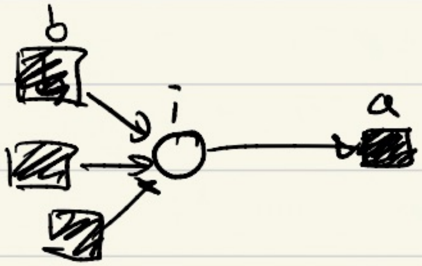
Input: $G = (V \cup F, E)$, T,

Output: $\{\hat{P}(X_i)\}_{i \in V}$

Initialize: $\{(\underbrace{m_{i \to a}, \quad m_{a \to i}}_{[\quad]\}|\mathcal{X}|})\}_{(i,a) \in E} = \vec{\mathbb{1}} \frac{1}{|\mathcal{X}|}$ or RANDOM

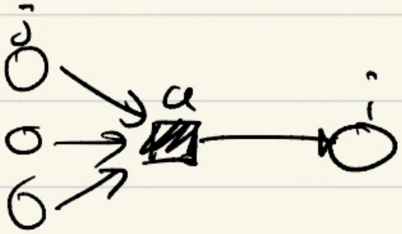
Repeat $t = 1 \dots T$

Update $m_{i \to a}$'s:



$$m_{i \to a}(X_i) = \prod_{b \in \partial i \setminus \{a\}} \tilde{m}_{b \to i}(X_i)$$

Update $\tilde{m}_{a \to i}$'s:



$$\underbrace{\tilde{m}_{a \to i}(X_i)}_{P(X_i)} = \sum_{\partial a \setminus \{i\}} \underbrace{f_a(X_{\partial a})}_{\text{cond dist}} \underbrace{\prod_{j \in \partial a \setminus \{i\}} m_{j \to a}(X_j)}_{\text{Prior on } X_j\text{'s}}$$

has the interpretation of

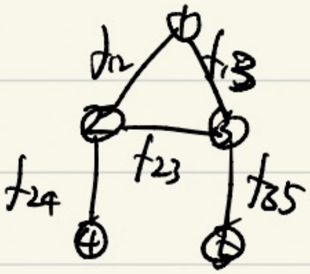
$P(X_i | X_{j_1}, X_{j_2}, X_{j_3})$ $\quad P(X_{j_1}) P(X_{j_2}) P(X_{j_3})$

Compute marginal:

$$m_i(x_i) = \prod_{a \in \partial i} \tilde{m}_{a \to i}(x_i)$$

Claim: this marginal is exact if FG is a tree

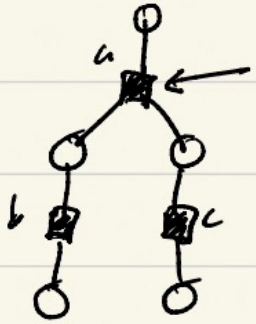
Pairwise MRF



BP is approximate
as there is a loop.

Complexity $O(|\mathcal{X}|^2)$

F.G.



$f_a = f_{12} \cdot f_{13} \cdot f_{23}$

BP is exact
as it is a tree

Complexity $O(|\mathcal{X}|_p^3)$

max degree on graph

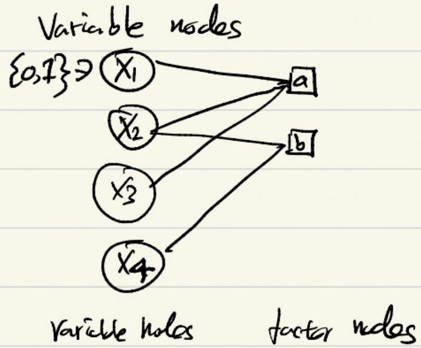
Example > Decoding LDPC codes. - one of most impactful successful application of B.P.

Def. LDPC (Low Density Parity Check) codes are a family of codes defined as follows

$$G = (V \cup F, E)$$

with factors for parity checking:

$$\underset{X_a}{\underset{\shortparallel}{f_a(X_{\partial a})}} = \mathbb{I}(X_1 \underset{\overset{\uparrow}{\text{XOR}}}{\oplus} X_2 \oplus X_3 = 0)$$



Def. $\text{Codebook} = \{X \in \mathbb{Z}_2^n \mid \text{satisfy all parities}\} = \left\{\begin{matrix} 0000 \\ 0111 \\ 1010 \\ 1101 \end{matrix}\right\}$

$$|\text{Codebook}| = 2^{\underbrace{|V| - |F|}_{K}} = 2^K$$

At transmission, one of the codeword from codebook is sent over a noisy channel, defined by $P(Y_i \mid X_i)$ and we need to recover X from observed Y's.

$$\text{e.g. } P(Y_i \mid X_i) = \begin{array}{c|cc|c} & \multicolumn{2}{c}{X_i} & \\ & 0 & 1 & \\ \hline & .7 & .3 & 0 \\ & .3 & .7 & 1 \end{array} \; Y_i$$

Strategy: Use Belief Propagation to estimate $P(X_i \mid Y_1 \cdots Y_n)$, $\forall i \in [n]$ and out put $\left(\text{sign}\left(\log \frac{P(X_i = 1 \mid Y)}{P(X_i = 0 \mid Y)}\right) + 1\right)\frac{1}{2} \in \{0, 1\}$

* How to start with a BN formulation and get Factor Graph. Seamlessly.

* How to include observed variables in inference/Modeling.