概率图模型理论及应用

Theory and Applications of Probabilistic Graphical Models
(Lesson 6)

欧智坚

清华大学电子工程系

Addr: 罗姆楼 6-104

Tel: 62796193

Email: ozj@tsinghua.edu.cn

课程章节

- 第一章 引言（1）
- 第二章 图模型的表示理论（3）
  - **DGM-UGM**
  - **Semantics**
  - **HMM-CRF**
- 第三章 图模型的推理理论（6）
  - 精确推理：**variable-elimination**，cluster-tree，triangulate
  - 连续变量：Kalman
  - 采样近似：sampling
  - 变分近似：variational
- 第四章 图模型的学习理论（3）
  - 参数学习：**MaxlikelihoodEstimate**，BayesEstimate
  - 结构学习：StructureLearning
- 第五章 一个综合例子（1）

第三章 图模型的推理理论

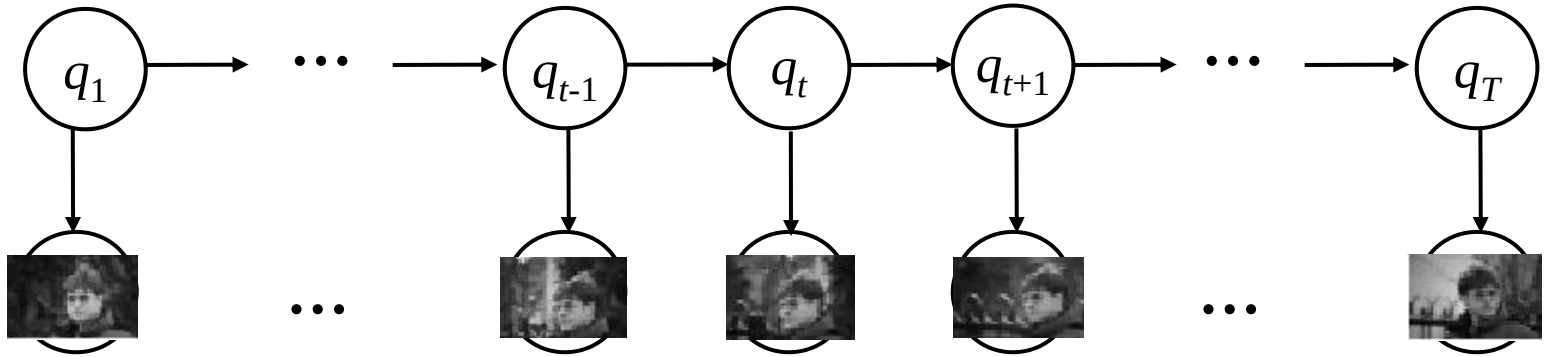
推理计算

- **定义**　假设联合分布已知

  给定观测量取值 $X_E=x_E$，计算其它一些未知量 X_Q 的条件分布

  - 即计算 $p(x_Q \mid X_E=x_E)$　$e.g.\ p(q_t \mid y_{1:T})=?$

query nodes　evidence



$$p(x_Q \mid x_E) \propto p(x_Q, x_E) = \sum_{x_{V\setminus(Q,E)}} p(x_Q, x_E, x_{V\setminus(Q,E)})$$

$$p(q_t \mid y_{1:T}) \propto p(q_t, y_{1:T}) = \sum_{q_{1:T\setminus\{t\}}} p(q_{1:T}, y_{1:T}) = \sum_{q_{1:T\setminus\{t\}}} p(q_1) \cdot \prod_{t=1}^{T-1} p(q_{t+1} \mid q_t) \cdot \prod_{t=1}^{T} p(y_t \mid q_t)$$

Sum-product: 对连乘积求和　　$(\pi,\ A,\ B)$

推理计算的三种具体形式

- 信度更新（Belief updating）

$$p(x_Q \mid x_E) \propto \sum_{x_{V\setminus(Q,E)}} p(x_V) \quad e.g.\ p(q_t \mid y_{1:T}), p(q_t, q_{t+1} \mid y_{1:T})$$ Sum-product

- 计算最有可能取值（most probable explanation, MPE）

$$x_Q^* = \arg\max_{x_Q} p(x_Q \mid x_E), \text{ where } Q \cup E = V$$

$$= \arg\max_{x_Q} p(x_V) \qquad e.g.\ \max_{q_{1:T}} p(q_{1:T} \mid y_{1:T})$$ Max-product

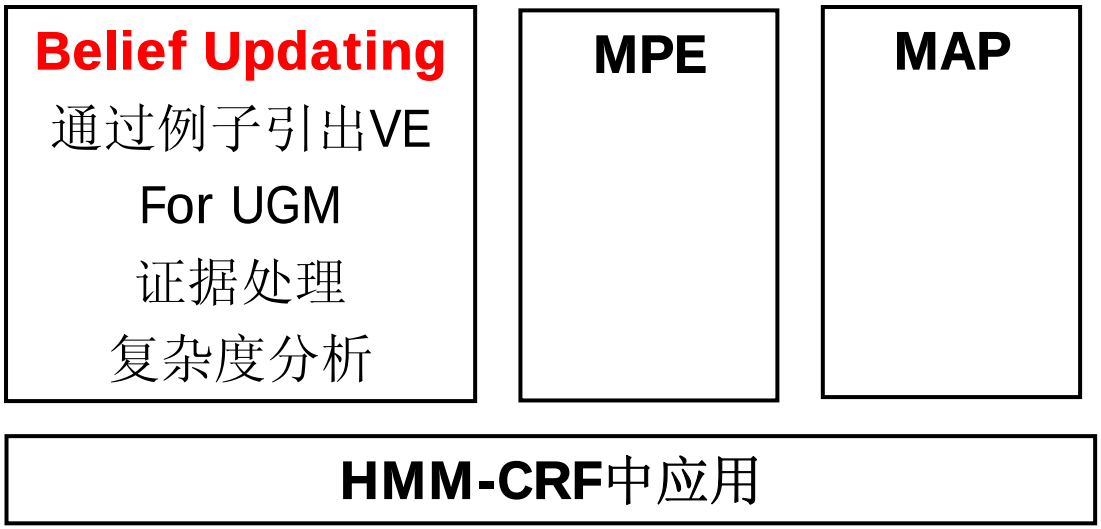
- 计算最大后验假设（maximum aposteriori hypothesis, MAP）

$$x_Q^* = \arg\max_{x_Q} p(x_Q \mid x_E), \text{ where } Q \cup E \subset V$$

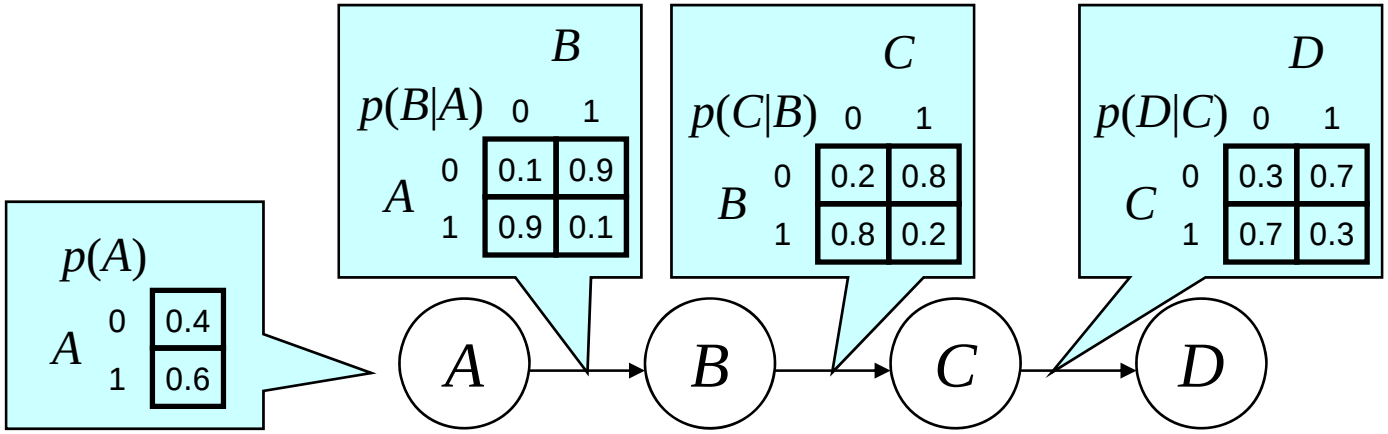
$$= \arg\max_{x_Q} \sum_{x_{V\setminus(Q,E)}} p(x_V)$$ Max-sum-product

第三章 图模型的推理理论

推理计算的三大形式

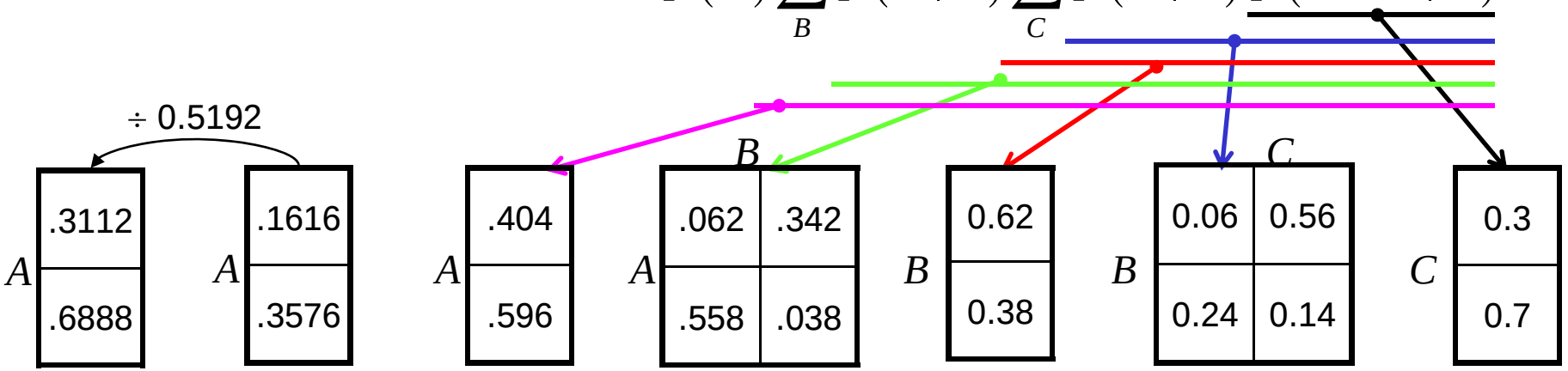


Inference example



$$p(A \mid D=0) \propto p(A, D=0) = \sum_B \sum_C p(A)\, p(B \mid A)\, p(C \mid B)\, p(D=0 \mid C)$$

$$= p(A) \sum_B p(B \mid A) \sum_C p(C \mid B)\, p(D=0 \mid C)$$



Two basic operations: product, marginalization

Comment

- Exponential reduction in computation !
  - 对特定变量的求和尽可能往早挪，直到求和号里的乘积项均包含这个变量
  - 缓存中间结果（Cache intermediate results）

Variable elimination (bucket elimination):

对上面操作的一个系统的算法描述

R. Dechter, "Bucket Elimination: A Unifying Framework for Probabilistic Inference", 1996

Bucket elimination *elim-bel* (Dechter 1996)

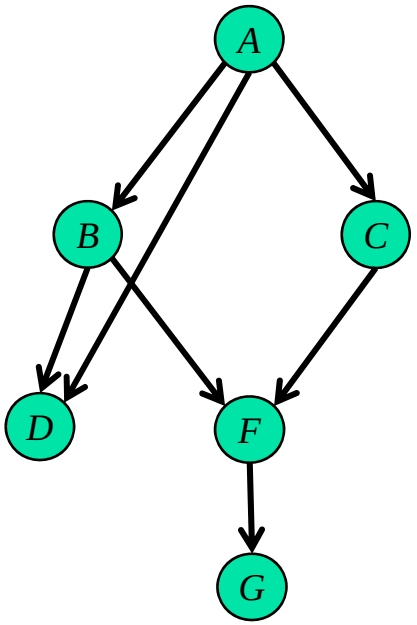
$$p(a \mid g=1) \propto \sum_{c}\sum_{b}\sum_{f}\sum_{d}\sum_{g} p(a)\,p(c \mid a)\,p(b \mid a)\,p(f \mid b,c)\,p(d \mid a,b)\,p(g \mid f)\,\delta(g=1)$$

给定变量消除排序（例如 a, c, b, f, d, g），询问结点居首

①**初始化：**将局部函数放入桶中。规则：最先被消除的自变量所标识的桶

②**依次消除变量：**桶内各函数相乘，消除掉本变量，消息函数放入新桶

③**返回：**询问结点标识的桶中所有函数相乘



bucket_G : $p(g \mid f)$ $\delta(g=1)$

bucket_D : $p(d \mid a,b)$

bucket_F : $p(f \mid b,c)$ $\lambda_G(f) = p(g=1 \mid f)$

bucket_B : $p(b \mid a)$ $\lambda_D(a,b)$ $\lambda_F(b,c)$

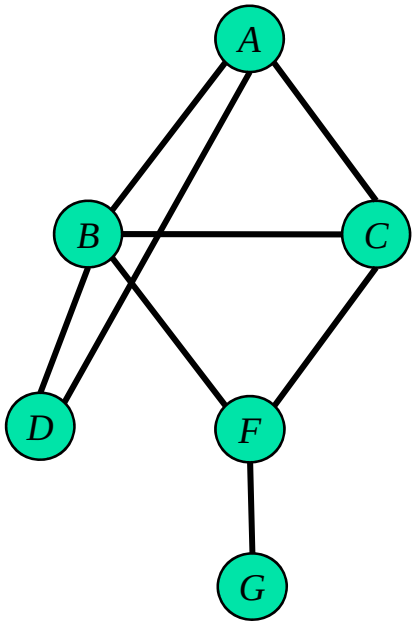
bucket_C : $p(c \mid a)$ $\lambda_B(a,c)$

bucket_A : $p(a)$ $\lambda_C(a)$

Bucket elimination *for UGM*

$$p(a,b,c,d,f,g) = \frac{1}{Z}\phi(a,b,c)\phi(b,c,f)\phi(a,b,d)\phi(f,g)$$

$$p(a \mid g=1) \propto \sum_c \sum_b \sum_f \sum_d \sum_g \phi(a,b,c)\phi(b,c,f)\phi(a,b,d)\phi(f,g)\delta(g=1)$$



bucket_G : $\phi(f,g)$ $\delta(g=1)$

bucket_D : $\phi(a,b,d)$

bucket_F : $\phi(b,c,f)$ $\lambda_G(f) = \phi(f,g=1)$

bucket_B : $\phi(a,b,c)$ $\lambda_D(a,b)$ $\lambda_F(b,c)$

bucket_C : $\lambda_B(a,c)$

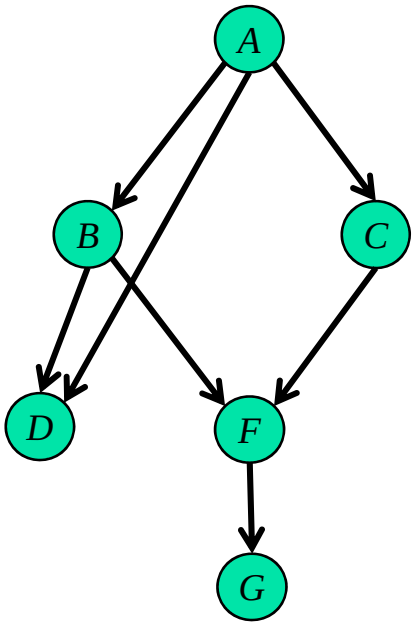
bucket_A : $\lambda_C(a)$

尽早代入证据

$$p(a \mid b=1) \propto \sum_c \sum_b \sum_f \sum_d \sum_g p(a)\, p(c \mid a)\, p(b \mid a)\, p(f \mid b,c)\, p(d \mid a,b)\, p(g \mid f)\, \delta(b=1)$$

最先消去证据变量，或者说，直接代入

$$p(a \mid b=1) \propto \sum_c \sum_f \sum_d \sum_g p(a)\, p(c \mid a)\, p(b=1 \mid a)\, p(f \mid b=1,c)\, p(d \mid a,b=1)\, p(g \mid f)$$



bucket_G : $p(g \mid f)$

bucket_D : $p(d \mid a,b)$

bucket_F : $p(f \mid b,c)$ $\lambda_G(f)$

bucket_B : $p(b \mid a)$ $\lambda_D(a,b)$ $\lambda_F(b,c)$ $\delta(b=1)$

bucket_C : $p(c \mid a)$ $\lambda_B(a,c)$

bucket_A : $p(a)$ $\lambda_C(a)$

复杂度分析

- Complexities (time and space) vary greatly for different elimination ordering.
- Complexity of processing a bucket

$$\lambda_i(\cdot)=\sum_{x_i}\prod_{h\in \text{bucket}_i} h \qquad scope(\lambda_i)=\left[\bigcup_{h\in \text{bucket}_i} scope(h)\right]-\{X_i\}$$

$$\lambda_B(a,c)=\sum_b p(b\mid a)\lambda_D(a,b)\lambda_F(b,c)$$

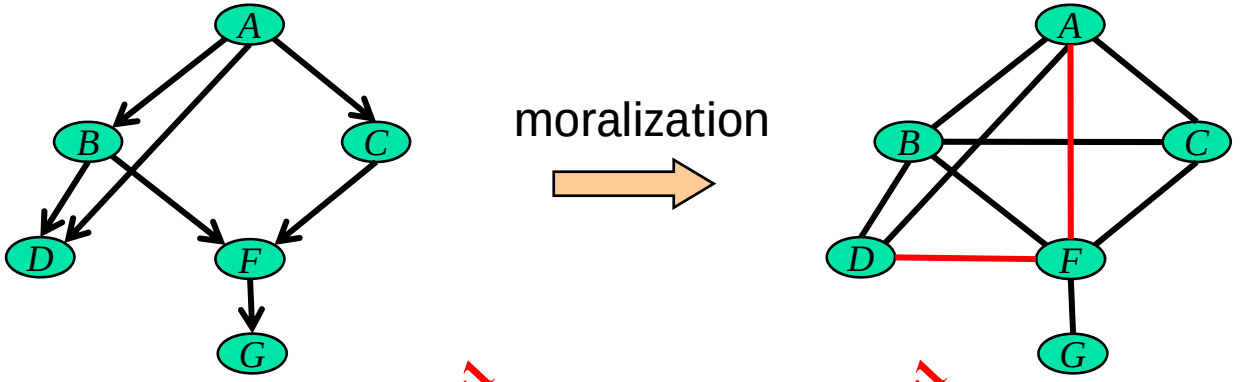
  - Time complexity for computing message functions $r^{|scope(\lambda_i)|+1}$ （消息函数的自变量集的大小）
  - Space complexity for saving message functions $r^{|scope(\lambda_i)|}$

To be determined by graph-theoretic analysis ?

连乘积的素图表示(primal graph)

- 一个连乘积的素图表示如下定义:
  - 以变量为结点；
  - 属于同一个局部函数的变量，两两之间添加（无向）边。
- 举例
  - 一个无向图的分布的素图表示是：无向图自身
  - 一个有向图的分布的素图表示是：有向图的moral graph
  - 代入证据后的连乘积（e.g. c=1）
  - 连乘积求和

$$p(a,b,c,d,f,g) \propto \phi(a,b,c)\phi(b,c,f)\phi(a,b,d)\phi(f,g)$$



$$p(a,b,c,d,f,g) = p(a)\,p(c|a)\,p(b|a)\,p(f|b,c)\,p(d|a,b)\,p(g|f)$$

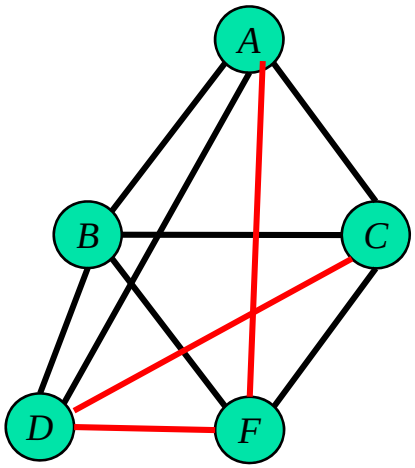
Graphic effect of variable elimination

连乘积消除一个变量，仍是一个（新的）连乘积

消除变量x_i：图上消除结点 x_i 且将 x_i 的邻居结点两两间加上边

消除 X_i 时，X_i的邻居数目 = $\left|scope(\lambda_i)\right|$

$$p(a\,|\,g=1) \propto \sum_f \sum_d \sum_c \sum_b p(a)\,p(c\,|\,a)\,p(b\,|\,a)\,p(f\,|\,b,c)\,p(d\,|\,a,b)\,p(g=1|\,f)$$



bucket_B : $p(d\,|\,a,b)$ $p(f\,|\,b,c)$ $p(b\,|\,a)$ $\left|scope(\lambda_i)\right|$

bucket_C : $p(c\,|\,a)$ $\lambda_B(a,c,d,f)$ 4

bucket_D : $\lambda_C(a,d,f)$ 3

bucket_F : $p(g=1|\,f)$ $\lambda_F(a,f)$ 2

bucket_A : $p(a)$ $\lambda_F(a)$ 1

Induced-graph, induced-width

- 一个素图 G 在结点排序 $d=(x_1,\ldots,x_N)$ 下的诱导图 G_d 如下定义:
  - 从后往前依次消除各结点;
  - 消除结点 x_i 时，将此时图中 x_i 的邻居两两间添加边。
  - 称此时图中 x_i 的邻居数目为：结点x_i在排序d下的诱导宽度
  - G_d = G 加上 上述过程多添加的边。 = 消除 x_i 得到的消息函数的自变量集大小
- 图G在排序d下的诱导宽度 $w(G_d)$
  = 图G的各结点在排序d下的诱导宽度的最大者
- 图G的诱导宽度 $w(G)$
  = 图G在所有可能排序下的诱导宽度的最小者

Complexity theorem

- Given an ordering $d=(X_1,\ldots,X_N)$ of a primal graph G, the time complexity of *elim-bel* is $O(N \cdot r^{w(G_d)+1})$ and space complexity is $O(N \cdot r^{w(G_d)})$.

  where r is the maximum domain size of any variable.

- Finding an ordering with the smallest induced-width is NP-hard.
  - Useful greedy heuristics are available.
  - 人为在图上找出好的变量排序。

Comment

- Exponential reduction in computation !
  - 对特定变量的求和尽可能往早挪，直到求和号里的乘积项均包含这个变量？
  - 缓存中间结果（Cache intermediate results）

Variable elimination (bucket elimination):

对上面操作的一个系统的算法描述

R. Dechter, "Bucket Elimination: A Unifying Framework for Probabilistic Inference", 1996

S. M. Aji and R. J. McEliece, “The generalized distributive law,” IEEE Trans. Information Theory, vol.46, pp.325-343, Mar. 2000.

第三章 图模型的推理理论

推理计算的三大形式

Belief Updating	**MPE**	**MAP**
通过例子引出VE For UGM 证据处理 复杂度分析		

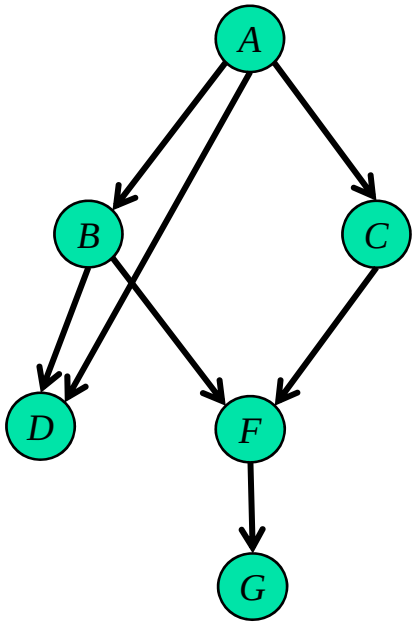
HMM-CRF中应用

Finding MPE *elim-mpe* (Dechter 1996)

$$MPE = \arg\max_{a,c,b,f,d} p(a,c,b,f,d \mid g=1)$$

$$\max_{a,c,b,f,d} p(a)\,p(c \mid a)\,p(b \mid a)\,p(f \mid b,c)\,p(d \mid a,b)\,p(g=1 \mid f)$$

$\sum$ is replaced by max



$D: \max_d p(d \mid a,b)$

$F: \quad p(f \mid b,c) \quad p(g=1 \mid f)$

$B: \quad p(b \mid a) \quad \lambda_D(a,b) \quad \lambda_F(b,c)$

$C: \quad p(c \mid a) \quad \lambda_B(a,c)$

$A: \quad p(a) \quad \lambda_C(a)$

MPE

$$D^*(a,b) = \arg\max_d p(d \mid a,b)$$

$$F^*(b,c) = \arg\max_f p(f \mid b,c)\,p(g=1 \mid f)$$

$$B^*(a,c) = \arg\max_b p(b \mid a)\,\lambda_D(a,b)\,\lambda_F(b,c)$$

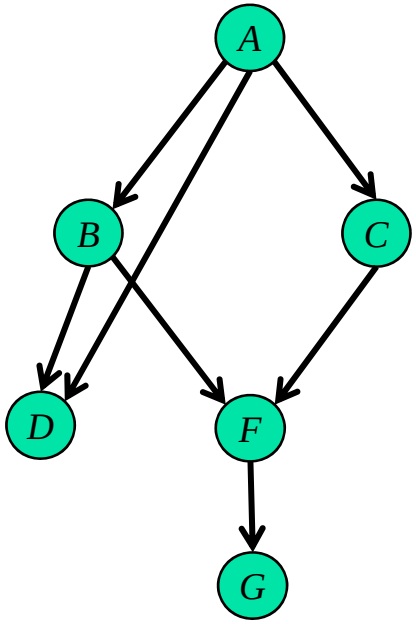
$$C^*(a) = \arg\max_c p(c \mid a)\,\lambda_B(a,c)$$

$$A^* = \arg\max_a p(a)\,\lambda_C(a)$$

Finding MAP *elim-map* (Dechter 1996)

Elimination operation: either summation or maximization
Restriction on variable ordering: summation before maximization

$$\max_{a,c} p(a,c \mid g=1) = \max_{a,c} \sum_{b,f,d} p(a)\,p(c \mid a)\,p(b \mid a)\,p(f \mid b,c)\,p(d \mid a,b)\,p(g=1 \mid f)$$



$D:\ p(d \mid a,b)$

$F:\ p(f \mid b,c)\quad p(g=1 \mid f)$

$B:\ p(b \mid a)\ \lambda_D(a,b)\ \lambda_F(b,c)$

$C:\ p(c \mid a)\ \lambda_B(a,c)$

$A:\ p(a)\ \lambda_C(a)$

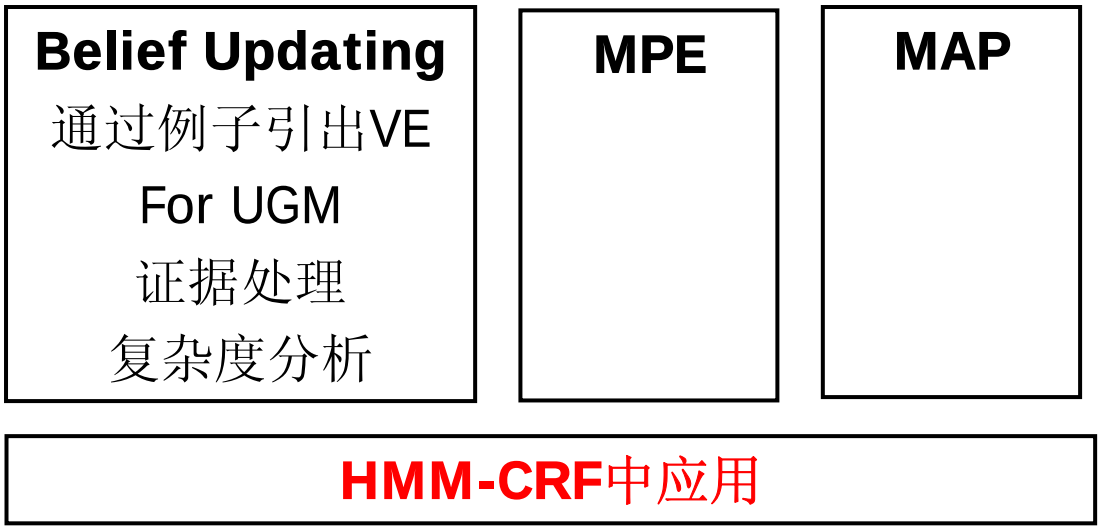
MAP

$$C^*(a) = \arg\max_c p(c \mid a)\,\lambda_B(a,c)$$

$$A^* = \arg\max_a p(a)\,\lambda_C(a)$$

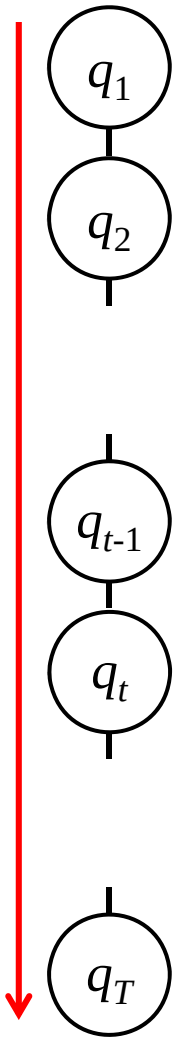
第三章 图模型的推理理论

推理计算的三大形式



Viterbi algorithm: MPE

$$\max_{q_{1:T}} p(q_{1:T} \mid y_{1:T}) \propto \max_{q_T \cdots q_1} p(q_1) \cdot \prod_{t=1}^{T-1} p(q_{t+1} \mid q_t) \cdot \prod_{t=1}^{T} p(y_t \mid q_t)$$



$p(q_1)\, p(q_2 \mid q_1)\, p(y_1 \mid q_1)$

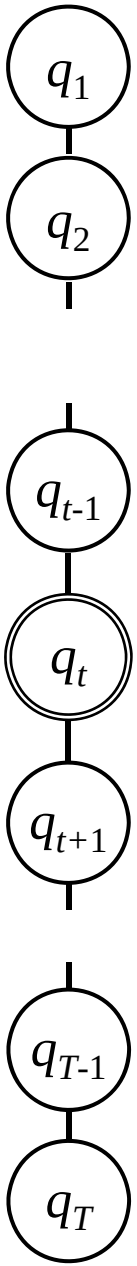
$p(q_t \mid q_{t-1})\, p(y_{t-1} \mid q_{t-1})\, \lambda(q_{t-1})$

$$\lambda(q_t) = \max_{q_{t-1}} p(q_t \mid q_{t-1})\, \underbrace{p(y_{t-1} \mid q_{t-1})\, \lambda(q_{t-1})}$$

$$\delta(q_t) = p(y_t \mid q_t) \max_{q_{t-1}} p(q_t \mid q_{t-1})\, \delta(q_{t-1})$$

$p(y_T \mid q_T)\, \lambda(q_T)$

Forward-backward algorithm



$p(q_1)\,p(q_2 \mid q_1)\,p(y_1 \mid q_1)$

$$p(q_t \mid y_{1:T}) \propto \sum_{q_{1:T} \setminus q_t} p(q_1) \cdot \prod_{t=1}^{T-1} p(q_{t+1} \mid q_t) \cdot \prod_{t=1}^{T} p(y_t \mid q_t)$$

$$p(q_t \mid y_{1:T}) \propto \underbrace{p(y_t \mid q_t)\,\lambda(q_t)}_{\alpha(q_t)} \cdot \beta(q_t)$$

$p(q_t \mid q_{t-1})\,p(y_{t-1} \mid q_{t-1})\,\lambda(q_{t-1})$

$p(y_t \mid q_t)\,\lambda(q_t)$

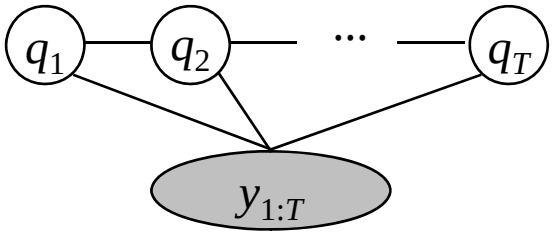
$$\beta(q_t) = \sum_{q_{t+1}} p(q_{t+1} \mid q_t)\,p(y_{t+1} \mid q_{t+1})\,\beta(q_{t+1})$$

$p(q_{t+1} \mid q_t)\,p(y_{t+1} \mid q_{t+1})\,\beta(q_{t+1})$

$p(q_{T-1} \mid q_{T-2})\,p(y_{T-1} \mid q_{T-1})\,\beta(q_{T-1})$

$p(q_T \mid q_{T-1})\,p(y_T \mid q_T)$

Linear-chain CRF



q_t, q_{t+1}组成的簇上的特征函数 $\psi_t(q_t, q_{t+1}, y)$

q_t 组成的簇上的特征函数 $\psi_t(q_t, y)$

$$p(q_{1:T} \mid y) \propto \exp\left\{\sum_{t=1}^{T-1}\sum_i \lambda_i f_i(q_t, q_{t+1}, y, t) + \sum_{t=1}^{T}\sum_j \mu_j f_j(q_t, y, t)\right\}$$

Likelihood calculation——Forward-backward算法

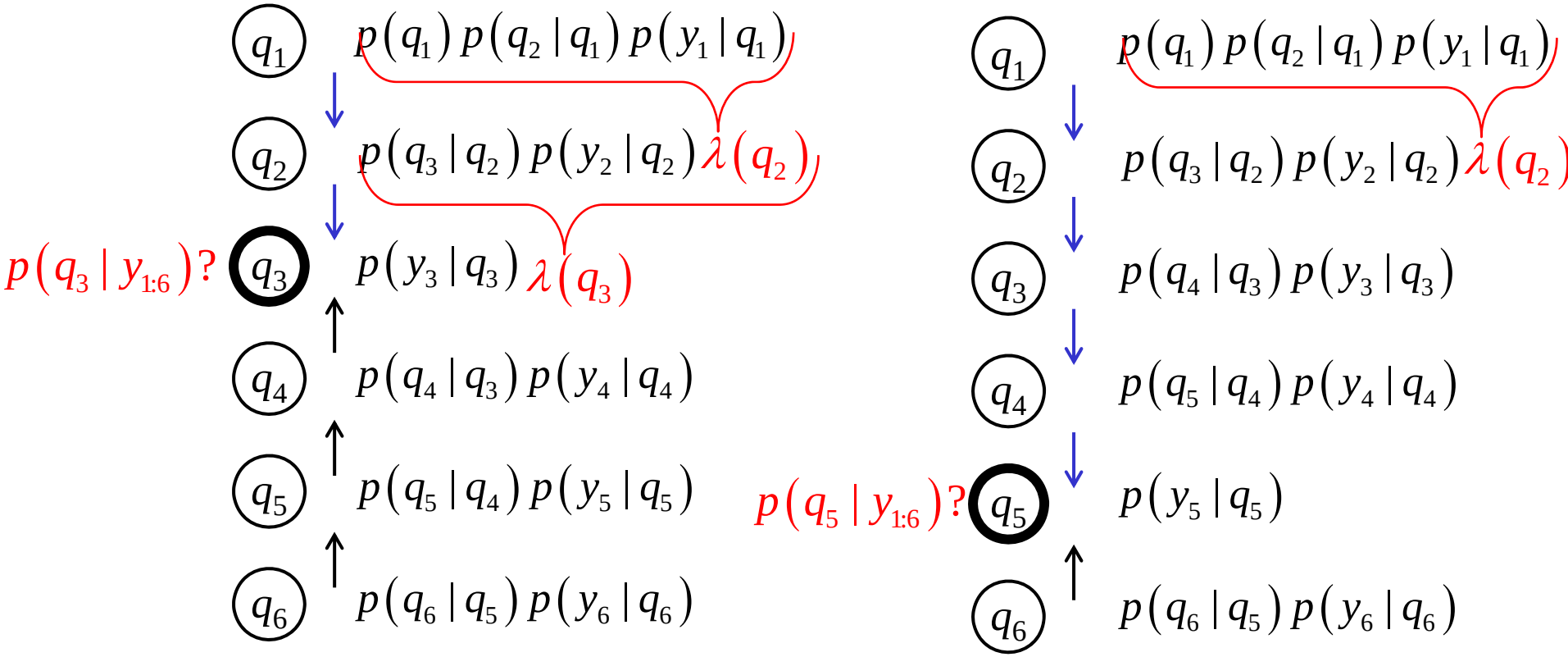
$$p(q_t \mid y) \propto \sum_{q_{1:T} \setminus q_t} \exp\left\{\sum_{t=1}^{T-1}\psi_t(q_t, q_{t+1}, y) + \sum_{t=1}^{T}\psi_t(q_t, y)\right\}$$

Decoding (recognition)——Viterbi算法

$$\max_{q_{1:T}} p(q_{1:T} \mid y) \propto \max_{q_T \cdots q_1} \exp\left\{\sum_{t=1}^{T-1}\psi_t(q_t, q_{t+1}, y) + \sum_{t=1}^{T}\psi_t(q_t, y)\right\}$$

多遍运行Variable elimination

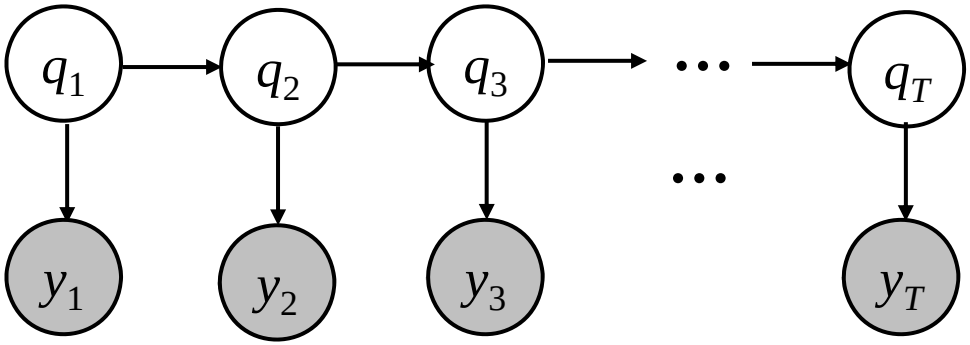
- 求多个隐变量的后验分布？ $p(q_t \mid y_{1:T}),\ \forall t$



5次α计算、5次β计算 VS 5次消息计算/遍 * 6遍ve

Limitations of VE

- How to avoid redundant computation for multiple runs of elimination?



Consider for the above graph, we need to compute
$p(q_1 \mid y_1, y_2, \ldots, y_T)$, $p(q_2 \mid y_1, y_2, \ldots, y_T)$, $p(q_3 \mid y_1, y_2, \ldots, y_T)$, ...

Go to cluster-tree algorithm

- If $w(G)$ is too large ?

The variable-elimination algorithm is of no help here; the complexity is intrinsic to the graph.

Go to approximate algorithm