10-418/10-618 Machine Learning for Structured Data

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Coordinate Ascent Variational Inference

Matt Gormley
Lecture 17
Nov. 2, 2022

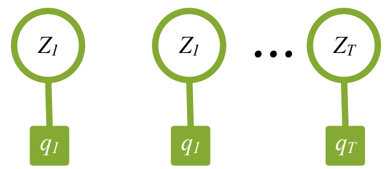
Reminders

- **Lecture on Friday, Recitation on Monday**
- **Exam Rubrics and Exam Viewings**
- **Homework 4: MCMC**
  - **Out: Mon, Oct 24**
  - **Due: Fri, Nov ~~3~~ 4 at 11:59pm**
- **Homework 5: Variational Inference**
  - **Out: Fri, Nov ~~3~~ 4**
  - **Due: Wed, Nov 16 at 11:59pm**

MEAN FIELD WITH GRADIENT ASCENT

Mean Field V.I. Overview

1. *Goal*: estimate $p_\alpha(\mathbf{z} \mid \mathbf{x})$
   we assume this is intractable to compute exactly
2. *Idea*: approximate with another distribution $q_\theta(\mathbf{z}) \approx p_\alpha(\mathbf{z} \mid \mathbf{x})$ for each $\mathbf{x}$
3. *Mean Field*: assume $q_\theta(\mathbf{z}) = \prod_t q_t(z_t; \theta)$
   i.e., we decompose over variables
   other choices for the decomposition of $q_\theta(\mathbf{z})$ give rise to "structured mean field"



4. *Optimization Problem*: pick the q that minimizes KL(q || p)

$$\hat{q}(\mathbf{z}) = \underset{q(\mathbf{z}) \in \mathcal{Q}}{\text{argmin}}\, \text{KL}(q(\mathbf{z}) || p(\mathbf{z} \mid \mathbf{x}))$$

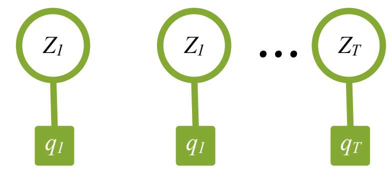
$$\hat{\theta} = \underset{\theta \in \Theta}{\text{argmin}}\, \text{KL}(q_\theta(\mathbf{z}) || p_\alpha(\mathbf{z} \mid \mathbf{x}))$$

equivalent

5. *Optimization Algorithm*: pick your favorite {coordinate descent, gradient descent, etc.}

Mean Field V.I. Overview

1. *Goal*: estimate $p_\alpha(\mathbf{z} \mid \mathbf{x})$
   we assume this is intractable to compute exactly
2. *Idea*: approximate with another distribution $q_\theta(\mathbf{z}) \approx p_\alpha(\mathbf{z} \mid \mathbf{x})$ for each $\mathbf{x}$



3. *Mean Field*: assume $q_\theta(\mathbf{z}) = \prod_t q_t(z_t; \theta)$
   i.e., we decompose over variables
   other choices for the decomposition of $q_\theta(\mathbf{z})$ give rise to "structured mean field"
4. *Optimization Problem*: pick the q that minimizes KL(q || p)

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \, \mathrm{KL}(q_\theta(\mathbf{z}) \parallel p_\alpha(\mathbf{z} \mid \mathbf{x})) = \underset{\theta}{\operatorname{argmax}} \, \mathrm{ELBO}(q_\theta)$$

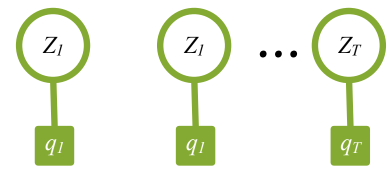
$$\mathrm{ELBO}(q_\theta) = E_{q_\theta(\mathbf{z})}\left[\log p_\alpha(\mathbf{x}, \mathbf{z})\right] - E_{q_\theta(\mathbf{z})}\left[\log q_\theta(\mathbf{z})\right]$$

$$\mathrm{ELBO}(q_\theta) = E_{q_\theta(\mathbf{z})}\left[\log \tilde{p}_\alpha(\mathbf{z} \mid \mathbf{x})\right] - E_{q_\theta(\mathbf{z})}\left[\log q_\theta(\mathbf{z})\right]$$

5. *Optimization Algorithm*: pick your favorite {coordinate ascent, gradient ascent, etc.}

Mean Field V.I. Overview

1. *Goal*: estimate $p_\alpha(\mathbf{z} \mid \mathbf{x})$
   we assume this is intractable to compute exactly
2. *Idea*: approximate with another distribution $q_\theta(\mathbf{z}) \approx p_\alpha(\mathbf{z} \mid \mathbf{x})$ for each $\mathbf{x}$



3. *Mean Field*: assume $q_\theta(\mathbf{z}) = \prod_t q_t(z_t; \theta)$
   i.e., we decompose over variables
   other choices for the decomposition of $q_\theta(\mathbf{z})$ give rise to "structured mean field"
4. *Optimization Problem*: pick the q that minimizes KL(q || p)

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \, \mathsf{KL}(q_\theta(\mathbf{z}) \parallel p_\alpha(\mathbf{z} \mid \mathbf{x})) = \underset{\theta}{\operatorname{argmax}} \, \mathsf{ELBO}(q_\theta)$$

$$\mathsf{ELBO}(q_\theta) = E_{q_\theta(\mathbf{z})}\left[\log p_\alpha(\mathbf{x}, \mathbf{z})\right] - E_{q_\theta(\mathbf{z})}\left[\log q_\theta(\mathbf{z})\right]$$

$$\mathsf{ELBO}(q_\theta) = E_{q_\theta(\mathbf{z})}\left[\log \tilde{p}_\alpha(\mathbf{z} \mid \mathbf{x})\right] - E_{q_\theta(\mathbf{z})}\left[\log q_\theta(\mathbf{z})\right]$$

5. *Optimization Algorithm*: gradient ascent

Mean Field w/Gradient Ascent

- **Note:** GA does local maximization, but ELBO is generally non-convex
- **Algorithm:**
  - Initialize θ
  - while not converged:

$$\theta \leftarrow \theta + \gamma \nabla_\theta \text{ELBO}(q_\theta)$$

- **Gradient of ELBO:**

$$\begin{aligned}\nabla_\theta \text{ELBO}(q_\theta) &= \nabla_\theta \mathbb{E}_{q_\theta}[\log p_\alpha(x, z)] - \nabla_\theta \mathbb{E}_{q_\theta}[\log q_\theta(z)] \\ &= \cdots \\ &= \cdots \\ &= \text{easy b/c of a simple } q_\theta\end{aligned}$$

HW5?

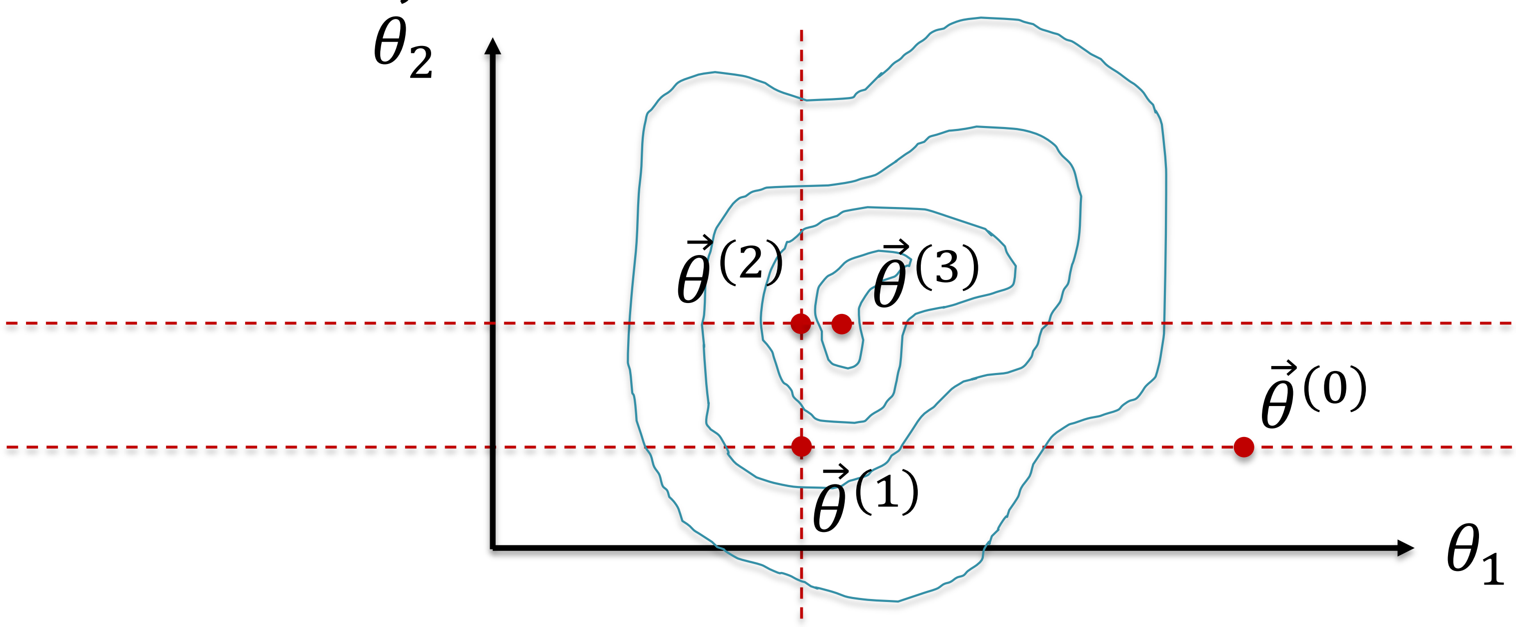
BACKGROUND: BLOCK COORDINATE DESCENT

Coordinate Descent

- Goal: minimize some objective

$$\vec{\theta}^* = \underset{\vec{\theta}}{\operatorname{argmin}}\, J(\vec{\theta})$$

- Idea: iteratively pick one variable and minimize the objective w.r.t. just that one variable, *keeping all the others fixed.*



Block Coordinate Descent

- Goal: minimize some objective (with 2 blocks)

$$\vec{\alpha}^*, \vec{\beta}^* = \underset{\vec{\alpha}, \vec{\beta}}{\operatorname{argmin}} J(\vec{\alpha}, \vec{\beta})$$

- Idea: iteratively pick one *block* of variables ($\vec{\alpha}$ or $\vec{\beta}$) and minimize the objective w.r.t. that block, keeping the other(s) fixed.

Init α, β

while not converged:

$$\vec{\alpha} = \underset{\vec{\alpha}}{\operatorname{argmin}} J(\vec{\alpha}, \vec{\beta})$$

$$\vec{\beta} = \underset{\vec{\beta}}{\operatorname{argmin}} J(\vec{\alpha}, \vec{\beta})$$

Block Coordinate Descent

- Goal: minimize some objective (with T blocks)

$$\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T = \underset{\boldsymbol{\alpha}_1}{\text{argmin}} \cdots \underset{\boldsymbol{\alpha}_T}{\text{argmin}}\, J(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T)$$

- Idea: iteratively pick one *block* of variables (e.g. the vector $\boldsymbol{\alpha}_t$) and minimize the objective w.r.t. that block, keeping the other(s) fixed.

Init. $\alpha_1, \ldots, \alpha_T$

while not converged:

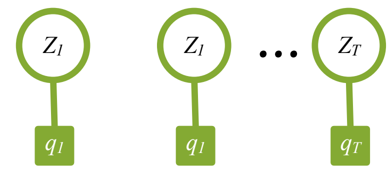
for $t = 1, \ldots, T$:

$$\boldsymbol{\alpha}_t = \underset{\boldsymbol{\alpha}_t}{\text{argmin}}\, J(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T)$$

COORDINATE ASCENT VARIATIONAL INFERENCE (CAVI)

Mean Field V.I. Overview

1. *Goal*: estimate $p_\alpha(\mathbf{z} \mid \mathbf{x})$
   we assume this is intractable to compute exactly
2. *Idea*: approximate with another distribution $q_\theta(\mathbf{z}) \approx p_\alpha(\mathbf{z} \mid \mathbf{x})$ for each $\mathbf{x}$



3. *Mean Field*: assume $q_\theta(\mathbf{z}) = \prod_t q_t(z_t; \theta)$
   i.e., we decompose over variables
   other choices for the decomposition of $q_\theta(\mathbf{z})$ give rise to "structured mean field"
4. *Optimization Problem*: pick the q that minimizes KL(q || p)

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \, \mathsf{KL}(q_\theta(\mathbf{z}) \parallel p_\alpha(\mathbf{z} \mid \mathbf{x})) = \underset{\theta}{\operatorname{argmax}} \, \mathsf{ELBO}(q_\theta)$$

$$\mathsf{ELBO}(q_\theta) = E_{q_\theta(\mathbf{z})}[\log p_\alpha(\mathbf{x}, \mathbf{z})] - E_{q_\theta(\mathbf{z})}[\log q_\theta(\mathbf{z})]$$

$$\mathsf{ELBO}(q_\theta) = E_{q_\theta(\mathbf{z})}[\log \tilde{p}_\alpha(\mathbf{z} \mid \mathbf{x})] - E_{q_\theta(\mathbf{z})}[\log q_\theta(\mathbf{z})]$$

5. *Optimization Algorithm*: coordinate ascent
   i.e. pick the best $q_t(z_t)$ based on the other $\{ q_s(z_s) \}_{s \neq t}$ being fixed

Choosing coordinate descent here yields the Coordinate Ascent Variational Inference (CAVI) algorithm

CAVI Algorithm

Coordinate Ascent Variational Inference (CAVI)

- here we assume a **mean field** approximation
- application of **coordinate ascent** to maximization of ELBO
- converges to a **local optimum** of the **nonconvex** ELBO objective

$q_{\neg t}(z_{\neg t}) = \prod_{s:s\neq t} q_s(z_s)$

```
procedure CAVI(p_α)
    Let q_θ(z) = ∏_{t=1}^{T} q_t(z_t)                        ▷ Mean field approx.
    while ELBO(q_θ) has not converged do
        for t ∈ {1, ..., T} do                              ▷ For each variable
            Set q_t(z_t) ∝ exp(E_{q_{¬t}}[log p_α(z_t | z_{¬t}, x)])
            while keeping all {q_s(·)}_{s≠t} fixed
        Compute ELBO(q_θ) = E_{q_θ(z)}[log p_α(x, z)] − E_{q_θ(z)}[log q_θ(z)]
    return q_θ
```

CAVI Algorit

Coordinate … e (CAVI)

– here … tion
– appl … imization
– conv … **nconvex**

Similar to Belief Propagation: can be viewed as **message passing** where we update our **variable beliefs** based on what **neighbors** think it should be

Like Gibbs Sampling, we compute a variable specific quantity at each step conditioned on the **Markov boundary**

procedure CAVI(p_α)
  Let $q_\theta(\mathbf{z}) = \prod_{t=1}^{T} q_t(z_t)$ ▷ Mean field approx.
  while ELBO(q_θ) has not converged **do**
    for $t \in \{1, \ldots, T\}$ **do** ▷ For each variable
      Set $q_t(z_t) \propto \exp(E_{q_{\neg t}}[\log p_\alpha(z_t \mid z_{\neg t}, x)])$
      while keeping all $\{q_s(\cdot)\}_{s \neq t}$ fixed
    Compute ELBO$(q_\theta) = E_{q_\theta(\mathbf{z})}[\log p_\alpha(\mathbf{x}, \mathbf{z})] - E_{q_\theta(\mathbf{z})}[\log q_\theta(\mathbf{z})]$
  return q_θ

Unlike Gibbs Sampling:
- we compute an entire distribution (instead of sampling a value)
- we condition on variable marginals (instead of on variable assignment)

Variational Inference

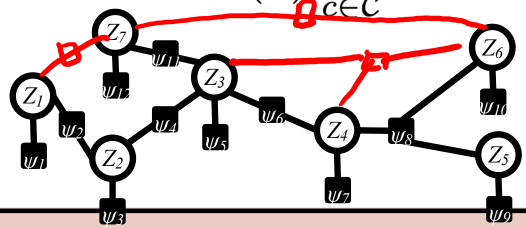
Whiteboard

– Computing marginals from a trained mean field approximation

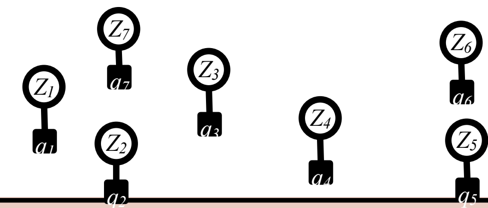
EXAMPLE: CAVI FOR DISCRETE FACTOR GRAPH

CAVI for a Discrete Factor Graph

$$p_\alpha(\mathbf{z} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{z}_c, \mathbf{x})$$



$$q_\theta(\mathbf{z}) = \prod_{t=1}^{T} q_t(z_t)$$



procedure CAVI(p_α)

  Let $q_\theta(\mathbf{z}) = \prod_{t=1}^{T} q_t(z_t)$ ▷ Mean field approx.

  while ELBO(q_θ) has not converged **do**

    for $t \in \{1, \ldots, T\}$ **do** ▷ For each variable

      Set $q_t(z_t) \propto \exp(E_{q_{\neg t}}[\log p_\alpha(z_t \mid z_{\neg t}, x)])$

      while keeping all $\{q_s(\cdot)\}_{s \neq t}$ fixed

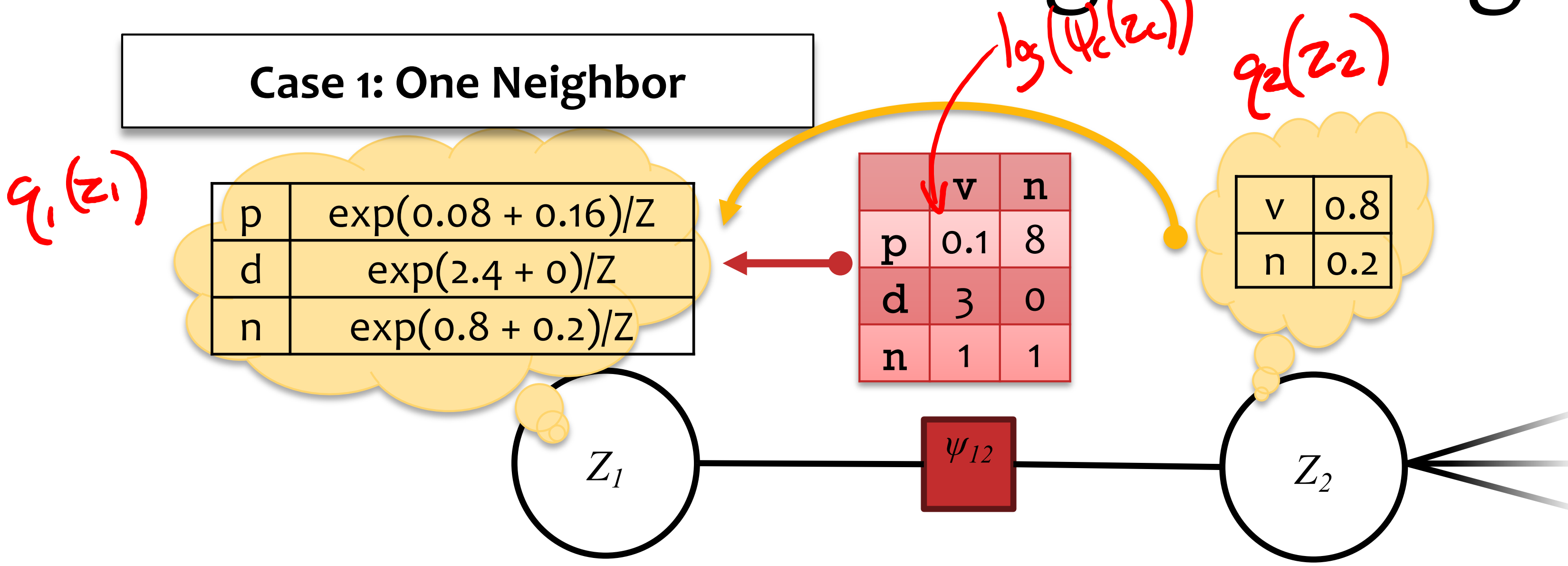
    Compute ELBO$(q_\theta) = E_{q_\theta(\mathbf{z})}[\log p_\alpha(\mathbf{x}, \mathbf{z})] - E_{q_\theta(\mathbf{z})}[\log q_\theta(\mathbf{z})]$

  return q_θ

$$\Rightarrow q_t(z_t) \propto \exp\left(\sum_{\mathbf{z}_{\text{MB}(z_t)}} \prod_{s \in \text{MB}(z_t)} q_s(z_s) \log \prod_{c \in N(z_t)} \psi_c(\mathbf{z}_c) \right)$$

efficiently computed assuming number of neighbors $N(z_t)$ is not too large

CAVI as Message Passing



CAVI message passing differs from BP in several ways:

- the beliefs are normalized (i.e. beliefs = marginals)
- no messages **to** factors (i.e. all messages are directly to a variable)
- matrix-vector product is exponentiated and normalized

$$\Rightarrow q_t(z_t) \propto \exp\left(\sum_{\mathbf{z}_{\text{MB}(z_t)}} \prod_{s \in \text{MB}(z_t)} q_s(z_s) \log \prod_{c \in N(z_t)} \psi_c(\mathbf{z}_c) \right)$$

Recall…

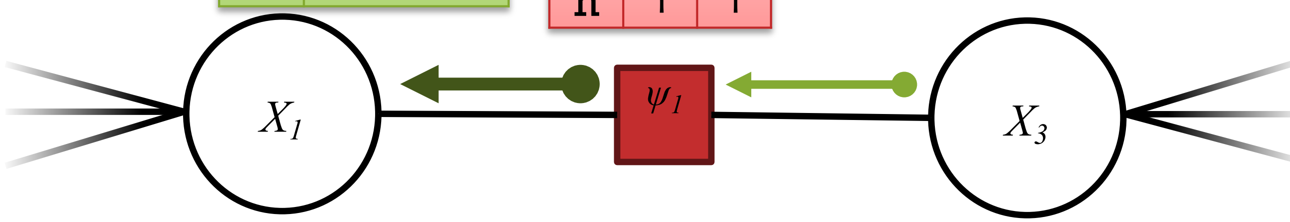
Sum-Product Belief Propagation

Factor Message

p	0.8 + 0.16
d	24 + 0
n	8 + 0.2

	v	n
p	0.1	8
d	3	0
n	1	1

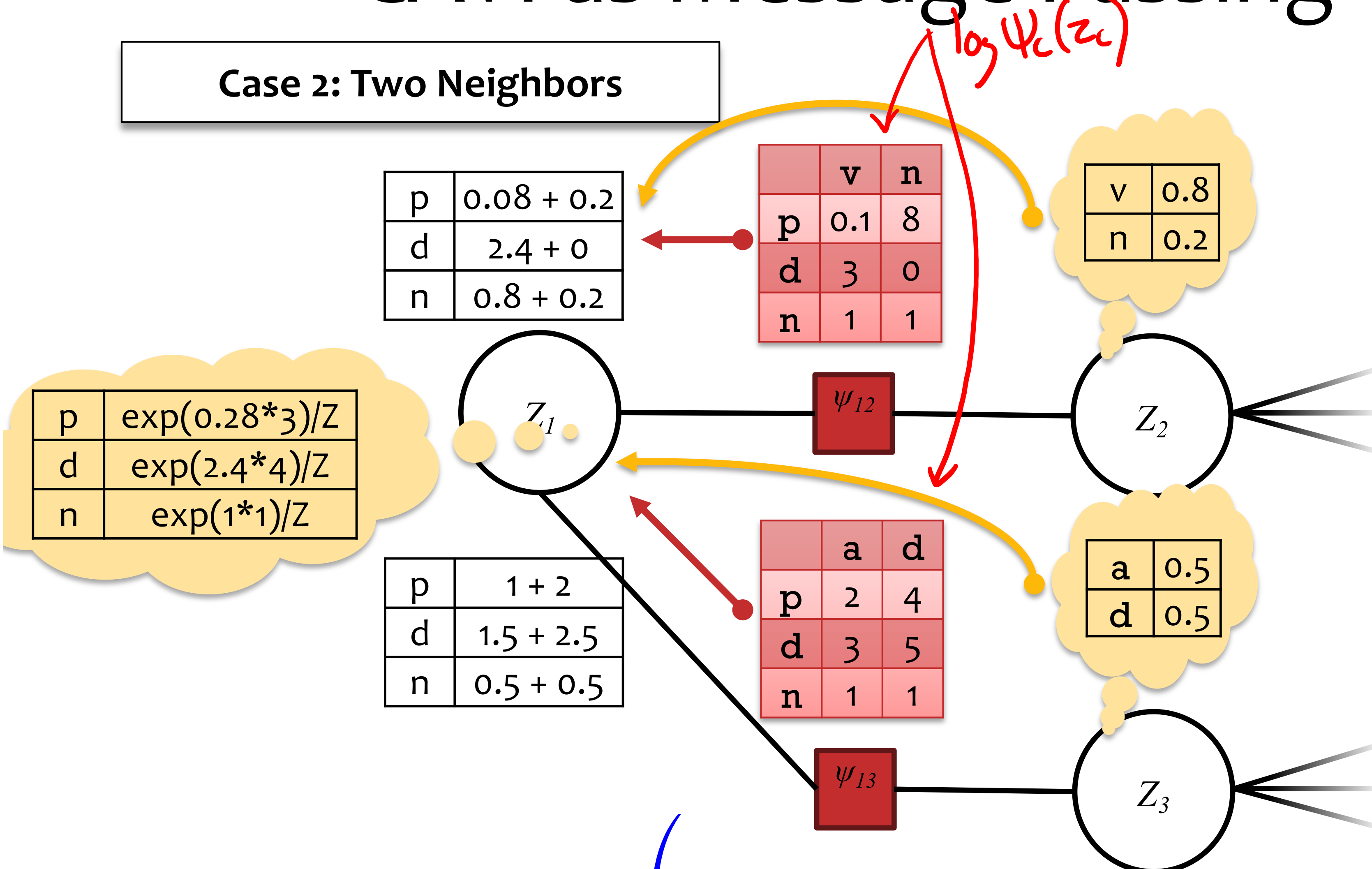
v	8
n	0.2



$$\mu_{\alpha \to i}(x_i) = \sum_{\boldsymbol{x}_{\boldsymbol{\alpha}} : \boldsymbol{x}_{\boldsymbol{\alpha}}[i] = x_i} \psi_\alpha(\boldsymbol{x}_{\boldsymbol{\alpha}}) \prod_{j \in \mathcal{N}(\alpha) \setminus i} \mu_{j \to \alpha}(\boldsymbol{x}_{\boldsymbol{\alpha}}[i])$$

CAVI as Message Passing

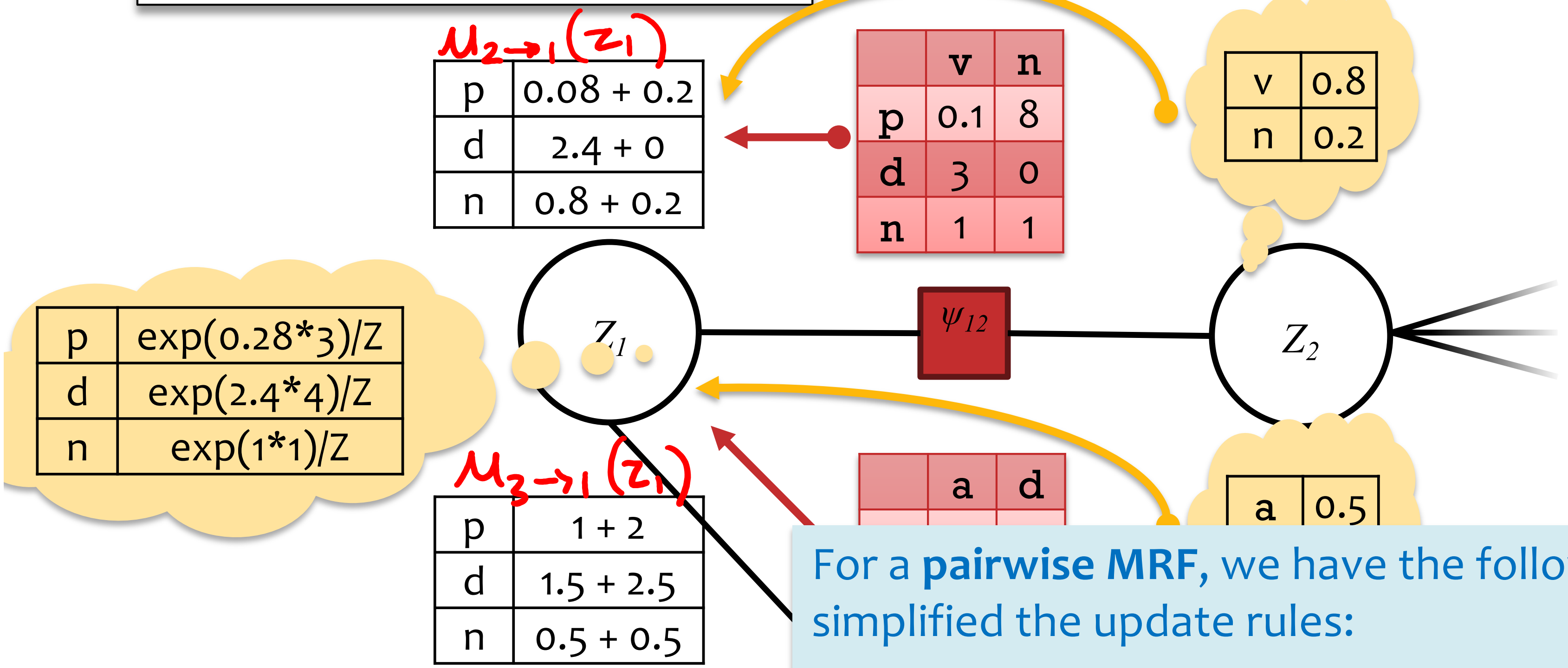
Case 2: Two Neighbors



$$\Rightarrow q_t(z_t) \propto \exp\left(\sum_{\mathbf{z}_{\mathrm{MB}(z_t)}} \prod_{s\in \mathrm{MB}(z_t)} q_s(z_s) \log \prod_{c\in N(z_t)} \psi_c(\mathbf{z}_c)\right)$$

CAVI as Message Passing

Case 2: Two Neighbors



For a **pairwise MRF**, we have the following simplified the update rules:

$$\mu_{s\to t}(z_t) = \sum_{z_s} q_s(z_s)\psi_{s,t}(z_s, z_t)$$

$$q_t(z_t) \propto \exp\left(\prod_{s\in \mathrm{MB}(z_t)} \mu_{s\to t}(z_t)\right)$$

Variational Inference

Whiteboard

- Computing the CAVI update
  - Multinomial full conditionals
- Example: two variable factor graph
  - Joint distribution
  - Mean Field Variational Inference
  - Gibbs Sampling

Q2: what questions do you have?

Q&A