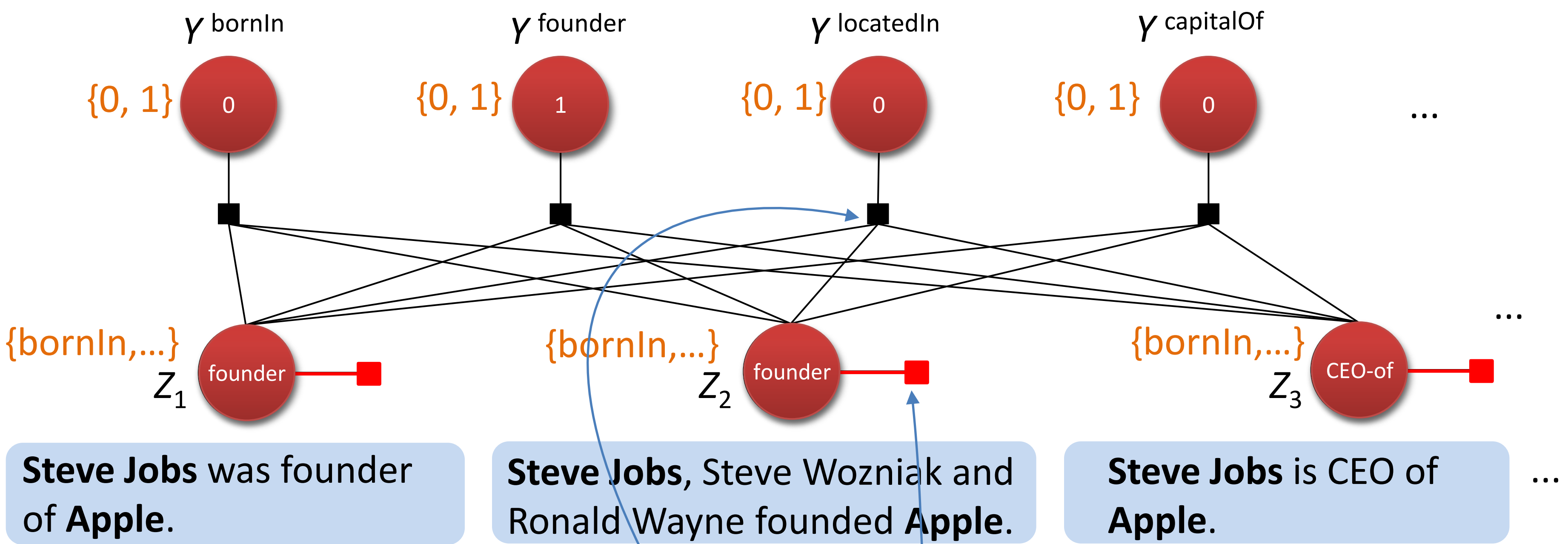
# Model

Steve Jobs, Apple:



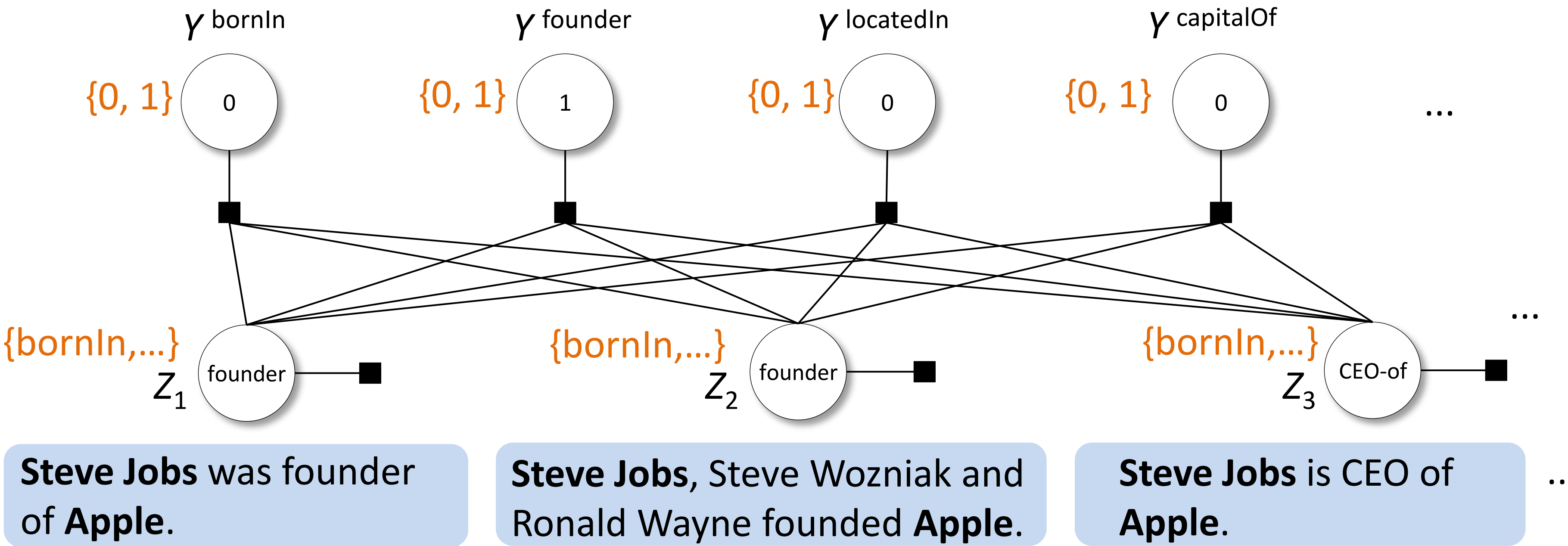

$$p(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z}|\mathbf{x};\theta) \overset{\text{def}}{=} \frac{1}{Z_x}\prod_r \Phi^{\text{join}}(y^r,\mathbf{z})\prod_i \Phi^{\text{extract}}(z_i,x_i)$$

$$\Phi^{\text{join}}(y^r,\mathbf{z}) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } y^r = true \wedge \exists i : z_i = r \\ 0 & \text{otherwise} \end{cases}$$

All features at sentence-level

(join factors are deterministic ORs)

# Model

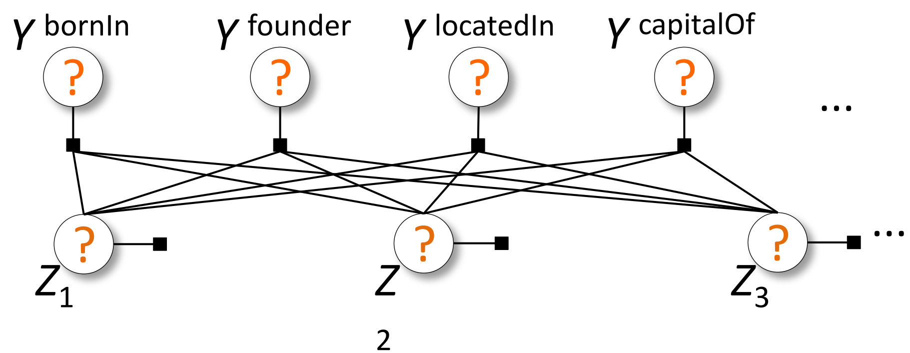



- Extraction almost entirely driven by sentence-level reasoning
- Tying of facts $Y_r$ and sentence-level extractions $Z_i$ still allows us to model weak supervision for training

# Inference

Need:

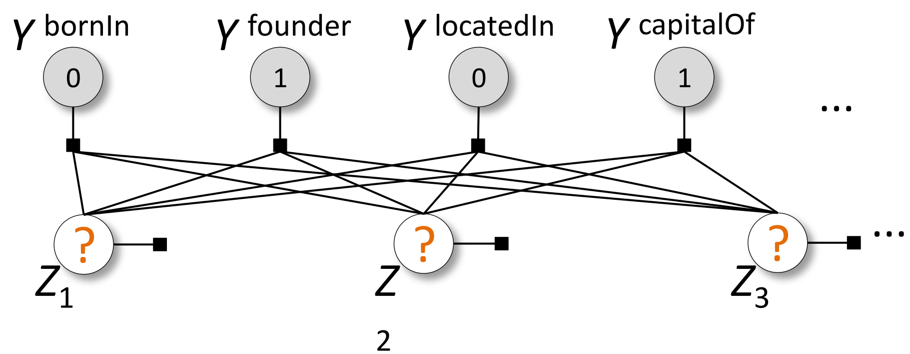
- Most likely sentence labels:




$$\arg\max_{\mathbf{y},\mathbf{z}} p(\mathbf{y}, \mathbf{z} | \mathbf{x}; \theta)$$

**Easy**

- Most likely sentence labels *given* facts:

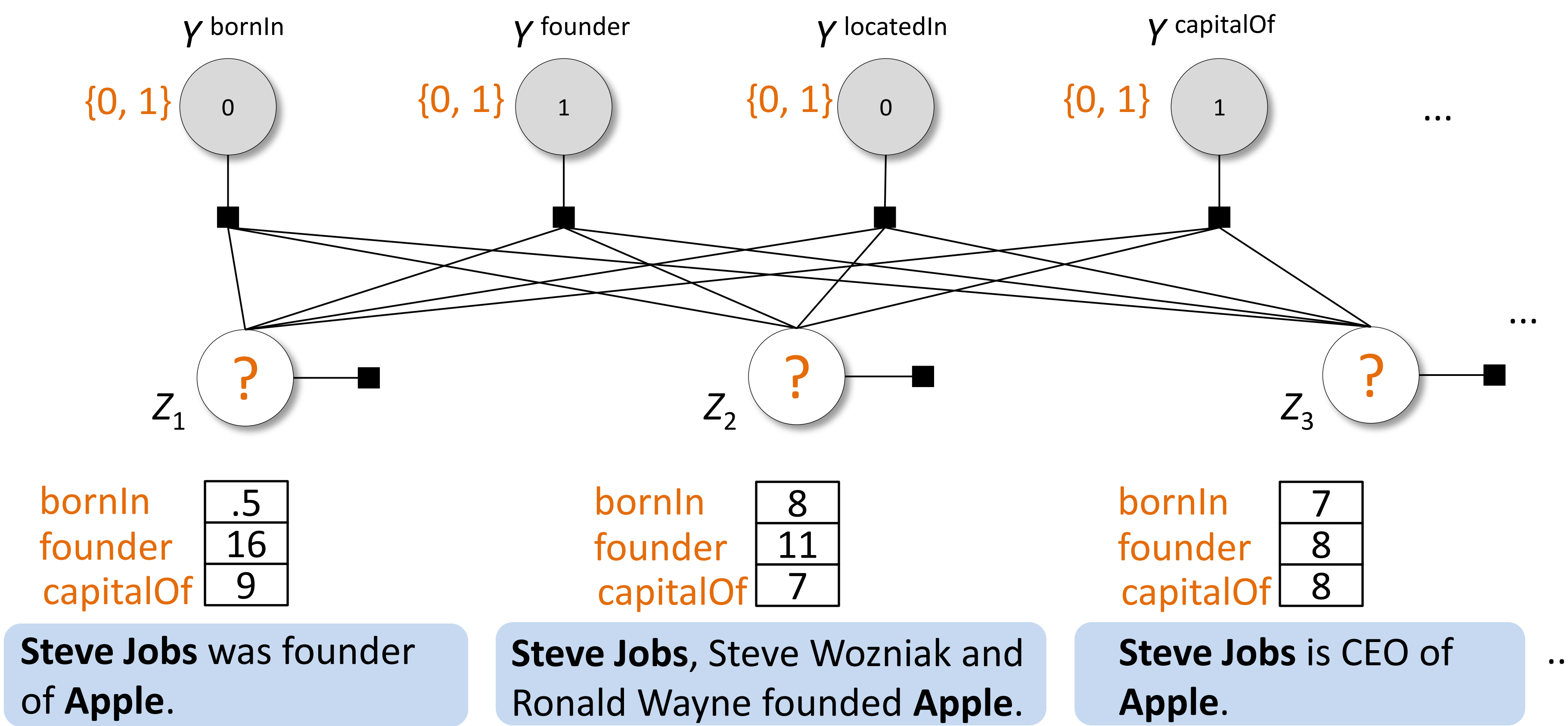



$$\arg\max_{\mathbf{z}} p(\mathbf{z} | \mathbf{x}, \mathbf{y}; \theta)$$

**Challenging**

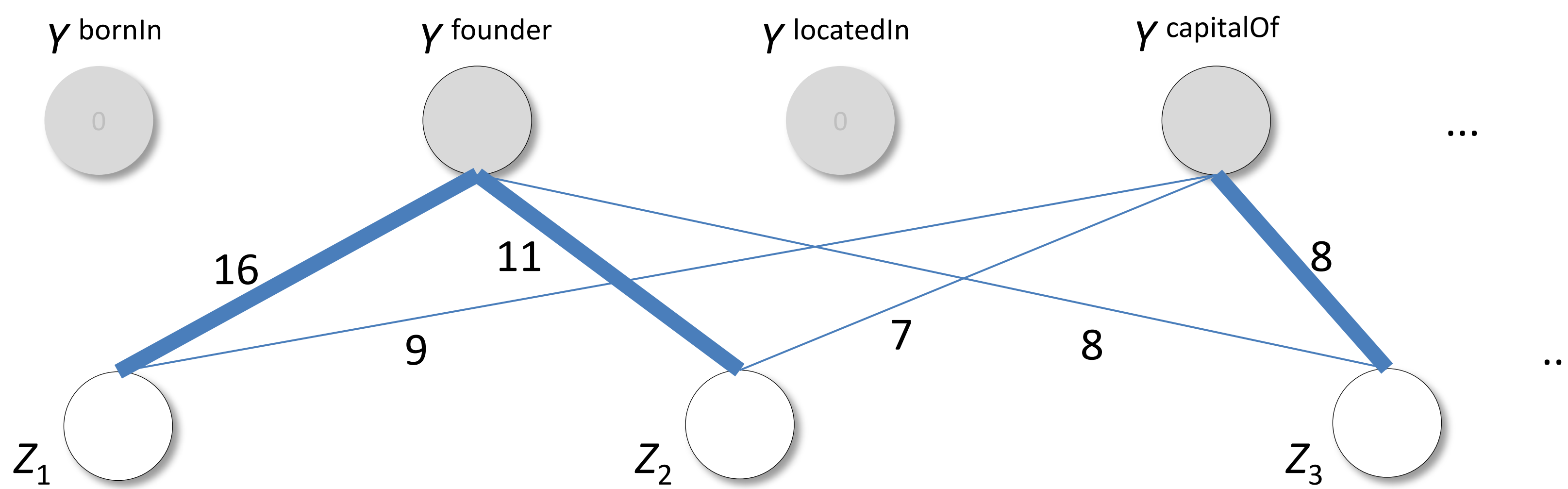
# Inference

- Computing $\arg\max_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \mathbf{y}; \theta)$ :



# Inference

- Variant of the weighted, edge-cover problem:




| | |
|---|---|
| bornIn | .5 |
| founder | 16 |
| capitalOf | 9 |

**Steve Jobs** was founder of **Apple**.

| | |
|---|---|
| bornIn | 8 |
| founder | 11 |
| capitalOf | 7 |

**Steve Jobs**, Steve Wozniak and Ronald Wayne founded **Apple**.

| | |
|---|---|
| bornIn | 7 |
| founder | 8 |
| capitalOf | 8 |

**Steve Jobs** is CEO of **Apple**.

...

# Learning

- Training set $\{(\mathbf{x}_i, \mathbf{y}_i) | i = 1 \dots n\}$ , where
  - $i$ corresponds to a particular entity pair
  - $\mathbf{x}_i$ contains all sentences with mentions of pair
  - $\mathbf{y}_i$ bit vector of facts about pair from database
- Maximize Likelihood

$$O(\theta) = \prod_i p(\mathbf{y}_i | \mathbf{x}_i; \theta) = \prod_i \sum_{\mathbf{z}} p(\mathbf{y}_i, \mathbf{z} | \mathbf{x}_i; \theta)$$

# Learning

- Scalability: Perceptron-style additive updates
- Requires two approximations:
  1. Online learning
     For example i (entity pair), define
     $$\phi(\mathbf{x}, \mathbf{z}) = \sum_j \phi(x_j, z_j)$$
     Use gradient of local log likelihood for example i:
     $$\frac{\partial \log O_i(\theta)}{\partial \theta_j} = E_{p(\mathbf{z}|\mathbf{x}_i,\mathbf{y}_i;\theta)}[\phi_j(\mathbf{x}_i, \mathbf{z})] - E_{p(\mathbf{y},\mathbf{z}|\mathbf{x}_i;\theta)}[\phi_j(\mathbf{x}_i, \mathbf{z})]$$
  2. Replace expectations with maximizations

# Learning: Hidden-Variable Perceptron

passes over dataset

for each entity pair *i*

most likely sentence labels and inferred facts (ignoring DB facts)

most likely sentence labels given DB facts

**initialize** parameter vector $\mathbf{\Theta} \leftarrow \mathbf{0}$
**for** $t = 1...T$ **do**
  **for** $i = 1...n$ **do**
    $(\mathbf{y}', \mathbf{z}') \leftarrow \arg\max_{\mathbf{y},\mathbf{z}} p(\mathbf{y}, \mathbf{z} | \mathbf{x_i}; \theta)$
    **if** $\mathbf{y}' \neq \mathbf{y_i}$ **then**
      $\mathbf{z}^* \leftarrow \arg\max_{\mathbf{z}} p(\mathbf{z} | \mathbf{x_i}, \mathbf{y_i}; \theta)$
      $\mathbf{\Theta} \leftarrow \mathbf{\Theta} + \phi(\mathbf{x_i}, \mathbf{z}^*) - \phi(\mathbf{x_i}, \mathbf{z}')$
    **end if**
  **end for**
**end for**
**Return** $\Theta$