Towards Combining Statistical Relational Learning and Graph Neural Networks for Reasoning

Jian Tang

Mila-Quebec AI Institute

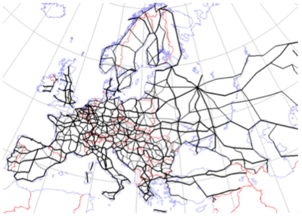
HEC Montreal

CIFAR AI Research Chair

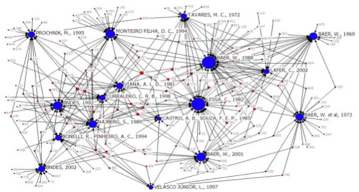
Relational Data/Graphs are Ubiquitous

- Graphs: a general and flexible data structure to encode the relations between objects

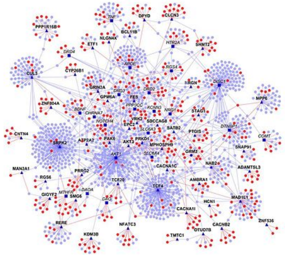
Social Graph



Road Graph

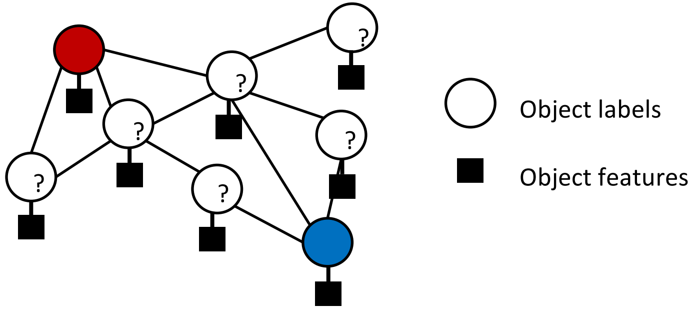


Citation Graph

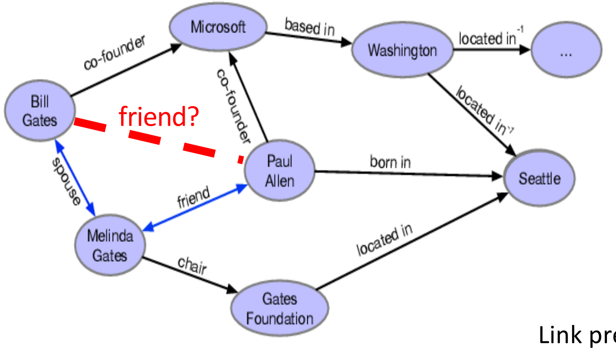


Protein-protein Interaction Graph

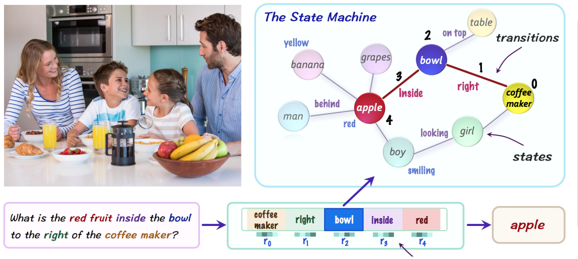
Relational Prediction and Reasoning



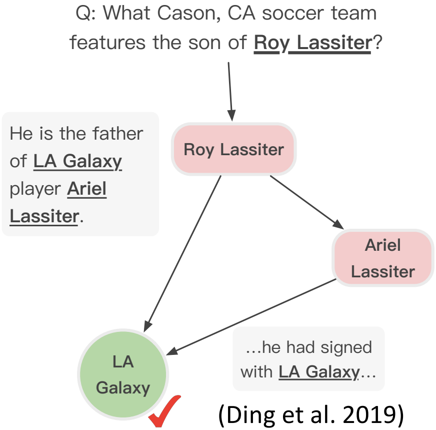
Node classification



Link prediction on knowledge graphs



Visual relational reasoning (Hudson et al. 2019)

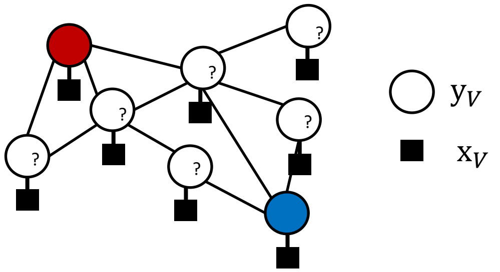


(Ding et al. 2019)

Multi-hop Question answering

Statistical Relational Learning

- Probabilistic graphical models for relational data
  - Markov Networks (Ross et al. 1980)
  - Conditional Random Fields (Lafferty et al. 2001)
  - Markov Logic Networks (Richardson and Domingos, 2006)
- Pros:
  - Captures uncertainty and domain knowledge
  - Collective inference
- Cons:
  - Limited model capacity
  - Inference is difficult

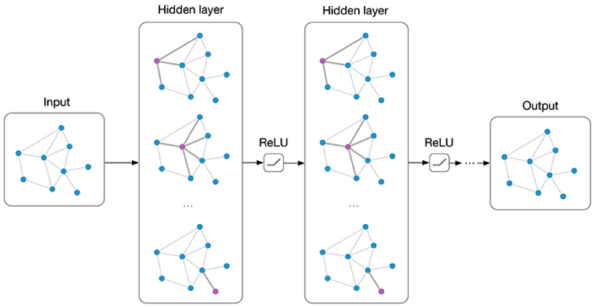


$$p(\mathbf{y}_V|\mathbf{x}_V) = \frac{1}{Z(\mathbf{x}_V)} \prod_{(i,j)\in E} \psi_{i,j}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{x}_V)$$

Figure: Conditional Random Fields

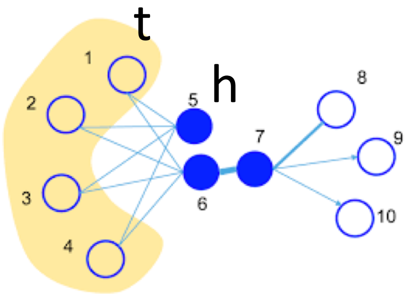
Graph Representation Learning

- Graph Neural Networks
  - Graph convolutional Networks (Kipf et al. 2016)
  - Graph attention networks (Veličković et al. 2017)
  - Neural message passing (Gilmer et al. 2017)

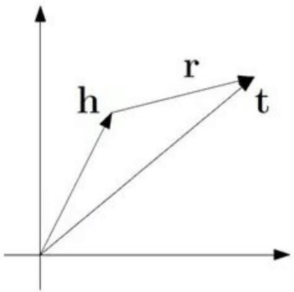


Graph convolutional Networks
(Kipf et al. 2016)

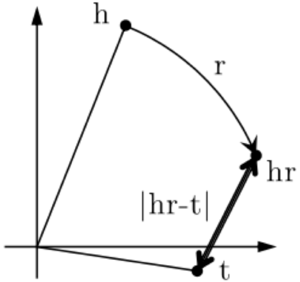
- Node Embedding and Knowledge Graph Embedding



DeepWalk, LINE, node2vec
(Perozzi et al. 2014, Tang et al. 2015, Grover et al. 2016)



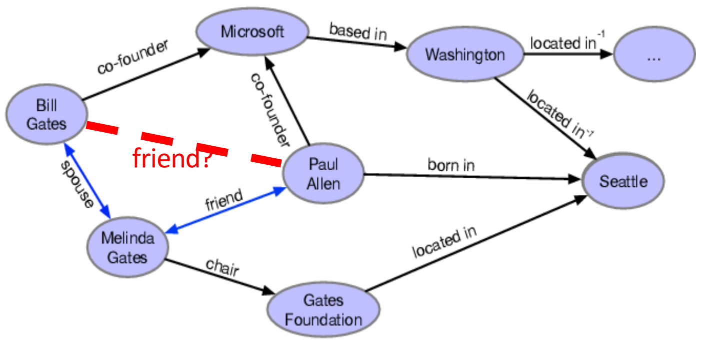
TransE
(Bordes et al. 2013)



RotatE
(Sun et al. 2019)

Link Prediction on Knowledge Graphs

- A set of facts $\mathrm{KG} = \{(h, r, t)\}$ represented as triplets
  - (Bill_Gates, Co_Founder, Microsoft)
- A variety of applications
  - Question answering
  - Search
  - Recommender Systems
  - Natural language understanding
  - …



- A fundamental problem: **predicting the missing facts by reasoning with existing facts**

Traditional Symbolic Logic-Rule based approaches

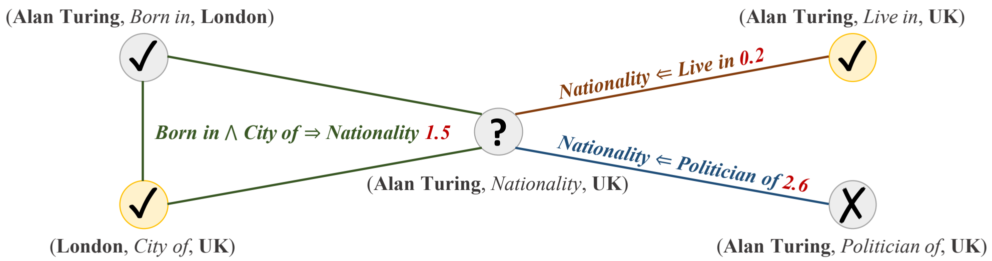
- Expert systems: hard logic rules
  - E.g., $\forall X, Y, \mathrm{Husband}(\mathrm{X}, Y) => \mathrm{Wife}(\mathrm{Y}, \mathrm{X})$
  - $\forall X, Y$, Live(X, Y) => Nationality(X, Y)

- Problematic as logic rules can be imperfect or contradictory
- We must handle the uncertainty of logic rules

Markov Logic Networks (Richardson and Domingos, 2006)

- Combines first-order logic and probabilistic graphical models

0.2	Live(X, Y) => Nationality (X, Y)
2.6	Politician_of(X, Y) => Nationality (X, Y)
1.5	Born(X,Y)∧City_of (Y,Z) => Nationality(X, Z)



$$p(\mathbf{v}_O, \mathbf{v}_H) = \frac{1}{Z} \exp\left(\sum_{l \in L} w_l \sum_{g \in G_l} \mathbb{1}\{g \text{ is true}\}\right) = \frac{1}{Z} \exp\left(\sum_{l \in L} w_l n_l(\mathbf{v}_O, \mathbf{v}_H)\right)$$

V_O: observed facts

V_H: unobserved/hidden facts

w_l: weight of logic rule l

$n_l(V_O, V_H)$: number of true grounds of the logic rule l

Pros and Cons of Markov Logic Networks

- Pros
  - Effectively leverage domain knowledge with logic rules
  - Handle the uncertainty
- Limitation
  - Inference is difficult due to complicated graph structures
  - Recall is low since many facts are not covered by any logic rules

Knowledge Graph Embeddings

- Learning the entity and relation embeddings for predicting the missing facts (e.g., TransE, ComplEx, DisMult, RotatE)
- Defining the joint distribution of all the facts

$$p(\mathbf{v}_O, \mathbf{v}_H) = \prod_{(h,r,t) \in O \cup H} \mathrm{Ber}(\mathbf{v}_{(h,r,t)} | f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t)),$$

An example:

$\mathrm{Ber}(\mathbf{v}_{(h,r,t)} | f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t))$ = $\sigma(\gamma - ||\mathbf{x}_h + \mathbf{x}_r - \mathbf{x}_t||)$ σ is the sigmoid function, γ is a fixed margin

- Trained by treating V_O as positive facts and V_H as negative facts

Pros and Cons

- Pros
  - Can be effectively and efficiently trained by SGD
  - High recall of missing link prediction with entity and relation embeddings
- Cons
  - Hard to leverage domain knowledge (logic rules)

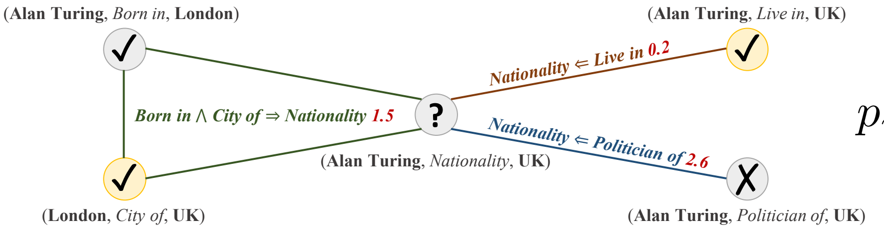
Probabilistic Logic Neural Networks for Reasoning (Qu and Tang, NeurIPS'19.)

- Towards combining Markov Logic Networks and knowledge graph embedding
  - Leverage logic rules and handling their uncertainty
  - Effective and efficient inference
- Define the joint distribution of facts with Markov Logic Network
- Optimization with variational EM
  - Parametrize the variational distribution with knowledge graph embedding methods

Meng Qu and Jian Tang. "Probabilistic Logic Neural Networks for Reasoning." To appear in NeurIPS'2019.

pLogicNet

- Define the joint distribution of facts with an MLN



$$p_w(\mathbf{v}_O, \mathbf{v}_H) = \frac{1}{Z} \exp\left(\sum_l w_l n_l(\mathbf{v}_O, \mathbf{v}_H)\right)$$

- Learning by maximizing the variational lower-bound of the log-likelihood of observed facts

$$\log p_w(\mathbf{v}_O) \geq \mathcal{L}(q_\theta, p_w) = \mathbb{E}_{q_\theta(\mathbf{v}_H)}[\log p_w(\mathbf{v}_O, \mathbf{v}_H) - \log q_\theta(\mathbf{v}_H)]$$

Inference

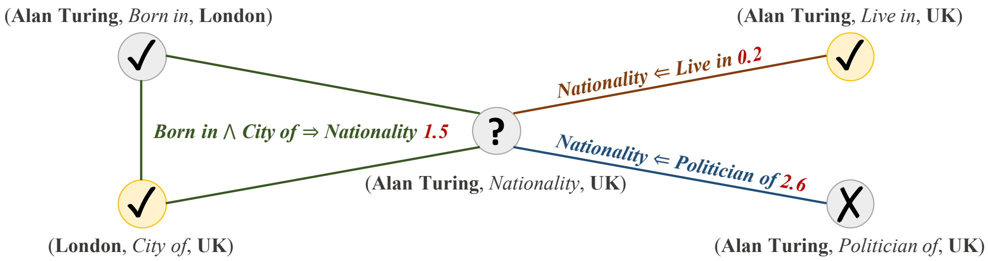
- Amortized mean-field variational inference
  - Use knowledge graph embedding model to parameterize the variational distribution

$$q_\theta(\mathbf{v}_H) = \prod_{(h,r,t)\in H} q_\theta(\mathbf{v}_{(h,r,t)}) = \prod_{(h,r,t)\in H} \mathrm{Ber}(\mathbf{v}_{(h,r,t)} | f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t)),$$

Learning

- Optimize pseudo-likelihood function
  - Update the weights of logic rules

$$\ell_{PL}(w) \triangleq \mathbb{E}_{q_\theta(\mathbf{v}_H)}[\sum_{h,r,t} \log p_w(\mathbf{v}_{(h,r,t)}|\mathbf{v}_{O\cup H\setminus(h,r,t)})] = \mathbb{E}_{q_\theta(\mathbf{v}_H)}[\sum_{h,r,t} \log p_w(\mathbf{v}_{(h,r,t)}|\mathbf{v}_{\text{MB}(h,r,t)})],$$



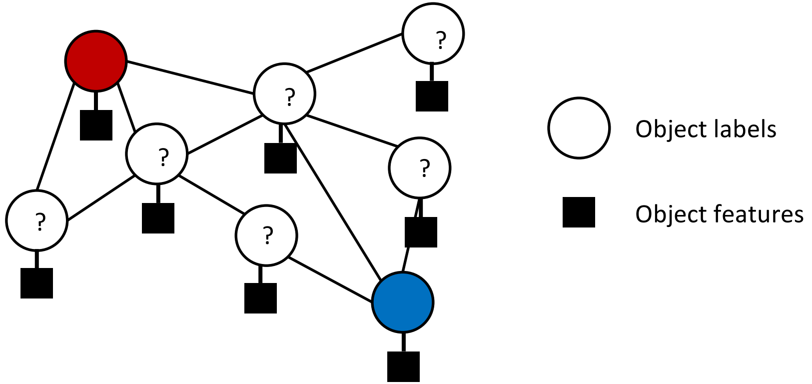
Performance of Link Prediction

- **Datasets:** benchmark knowledge graphs
  - FB15K, WN18, FB15K-237, WN18-RR
- Logic rules:
  - Composition rules (e.g., Father of Father is GrandFather)
  - Inverse rules (e.g., Husband and Wife)
  - Symmetric rules (e.g., Similar)
  - Subrelation rules (e.g., Man => Person)

Category	Algorithm	FB15k					WN18				
		MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
KGE	TransE [3]	40	0.730	64.5	79.3	86.4	272	0.772	70.1	80.8	92.0
	DistMult [17]	42	0.798	-	-	89.3	655	0.797	-	-	94.6
	HolE [26]	-	0.524	40.2	61.3	73.9	-	0.938	93.0	94.5	94.9
	ComplEx [41]	-	0.692	59.9	75.9	84.0	-	0.941	93.6	94.5	94.7
	ConvE [8]	51	0.657	55.8	72.3	83.1	374	0.943	93.5	94.6	95.6
Rule-based	BLP [7]	415	0.242	15.1	26.9	42.4	736	0.643	53.7	71.7	83.0
	MLN [32]	352	0.321	21.0	37.0	55.0	717	0.657	55.4	73.1	83.9
Hybrid	RUGE [15]	-	0.768	70.3	81.5	86.5	-	-	-	-	-
	NNE-AER [9]	-	0.803	76.1	83.1	87.4	-	0.943	**94.0**	94.5	94.8
Ours	pLogicNet	**33**	0.792	71.4	85.7	90.1	255	0.832	71.6	94.4	95.7
	pLogicNet*	**33**	**0.844**	**81.2**	**86.2**	**90.2**	**254**	**0.945**	93.9	**94.7**	**95.8**

Semi-supervised Object Classification

- Given G= (V, E, $\mathbf{x}_V$)
  - $V = V_L \cup V_U$: objects/nodes
  - E : edges
  - $\mathbf{x}_V$: object features



- Give some labeled objects V_L, we want to infer the labels of the rest of objects V_U

GMNN: Graph Markov Neural Networks (Qu, Bengio, and Tang, ICML'19)

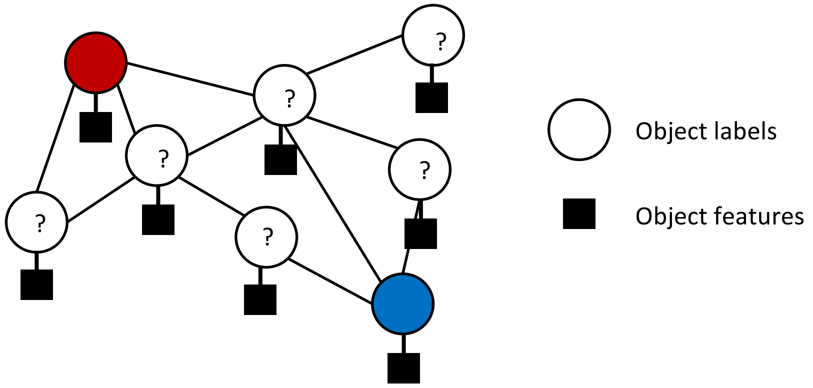
- Combining conditional random fields and graph neural networks
  - Learning effective node representations
  - Modeling the label dependencies of nodes
- Model the joint distribution of object labels $\mathbf{y}_V$ conditioned on object attributes $\mathbf{x}_V$, i.e., $p_\phi(\mathbf{y}_V|\mathbf{x}_V)$ with CRFs
  - Optimization with Pseudolikelihood Variational-EM

$$\log p_\phi(\mathbf{y}_L|\mathbf{x}_V) \geq$$
$$\mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\log p_\phi(\mathbf{y}_L, \mathbf{y}_U|\mathbf{x}_V) - \log q_\theta(\mathbf{y}_U|\mathbf{x}_V)]$$

Meng Qu, Yoshua Bengio, **Jian Tang**. "GMNN: Graph Markov Neural Networks". In ICML'19.

Overall Optimization Procedure

- Two Graph Neural Networks Collaborate with each other
  - p_ϕ: learning network, modeling the label dependency
  - q_θ: inference network, learning the object representations
- q_θ infer the labels of unlabeled objects trained with supervision from p_ϕ and labeled objects
- p_ϕ is trained with a fully labeled graph, where the unlabeled objects are labeled by q_θ



Take Away

- Relational reasoning is important to a variety of applications
  - Node classification, link prediction on knowledge graphs, question answering
- Towards combining two learning frameworks
  - Statistical Relational Learning
  - Graph Representation Learning
- Looking forward
  - Combining deep learning and symbolic reasoning systems
  - Incorporating common sense knowledge, handling uncertainty, and maybe automatically learn the logic rules.

Questions?

Email: jian.tang@hec.ca

Results on FB15k-237 and WN18RR

Category	Algorithm	FB15k-237					WN18RR				
		MR	**MRR**	**H@1**	**H@3**	**H@10**	**MR**	**MRR**	**H@1**	**H@3**	**H@10**
KGE	TransE [3]	181	0.326	22.9	36.3	52.1	3410	0.223	1.3	40.1	53.1
	DistMult [17]	254	0.241	15.5	26.3	41.9	5110	0.43	39	44	49
	ComplEx [41]	339	0.247	15.8	27.5	42.8	5261	**0.44**	**41**	**46**	51
	ConvE [8]	244	0.325	**23.7**	35.6	50.1	4187	0.43	40	44	52
Rule-based	BLP [7]	1985	0.092	6.2	9.8	15.0	12051	0.254	18.7	31.3	35.8
	MLN [32]	1980	0.098	6.7	10.3	16.0	11549	0.259	19.1	32.2	36.1
Ours	pLogicNet	**173**	0.330	23.1	**36.9**	**52.8**	3436	0.230	1.5	41.1	53.1
	pLogicNet*	**173**	**0.332**	**23.7**	36.7	52.4	**3408**	**0.441**	39.8	44.6	**53.7**

GMNN: Graph Markov Neural Networks

- Model the joint distribution of object labels $\mathbf{y}_V$ conditioned on object attributes $\mathbf{x}_V$, i.e., $\mathrm{p}_\phi(\mathbf{y}_V|\mathbf{x}_V)$
- Learning the model parameters ϕ by maximizing the lower-bound of log-likelihood of the observed data, $\log \mathrm{p}_\phi(\mathbf{y}_L|\mathbf{x}_V)$

$$\log p_\phi(\mathbf{y}_L|\mathbf{x}_V) \geq$$
$$\mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\log p_\phi(\mathbf{y}_L, \mathbf{y}_U|\mathbf{x}_V) - \log q_\theta(\mathbf{y}_U|\mathbf{x}_V)]$$

Optimization with Pseudolikelihood Variational-EM

- E-step: fix p_ϕ and update the variational distribution $q_\theta(\mathbf{y}_U|\mathbf{x}_V)$ to approximate the true posterior distribution $p_\phi(\mathbf{y}_U|\mathbf{y}_L, \mathbf{x}_V)$.
- M-step: fix q_θ and update p_ϕ to maximize the lower bound

$$\ell(\phi) = \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\log p_\phi(\mathbf{y}_L, \mathbf{y}_U|\mathbf{x}_V)]$$

- Directly optimize the joint likelihood is difficult due to the partition function in p_ϕ, instead we optimize the pseudolikelihood function

$$\ell_{PL}(\phi) \triangleq \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n \in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{V \setminus n}, \mathbf{x}_V)]$$
$$= \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n \in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x}_V)]$$

Inference/E-step: approximate $p_\phi(\mathbf{y}_U | \mathbf{y}_L, \mathbf{x}_V)$

- Approximate it with variational distribution $q_\theta(\mathbf{y}_U | \mathbf{x}_V)$. Specifically we use mean-field method:

$$q_\theta(\mathbf{y}_U | \mathbf{x}_V) = \prod_{n \in U} q_\theta(\mathbf{y}_n | \mathbf{x}_V)$$

- We parametrize each variational distribution with a Graph Neural Network

$$q_\theta(\mathbf{y}_n | \mathbf{x}_V) = \mathrm{Cat}(\mathbf{y}_n | \mathrm{softmax}(W_\theta \mathbf{h}_{\theta,n}))$$

Object representations learned by GNN

Learning/M-step:

- The log-pseudo likelihood:

$$\ell_{PL}(\phi) \triangleq \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n\in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{V\setminus n}, \mathbf{x}_V)]$$

$$= \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n\in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x}_V)]$$

- According to the inference, only the $p_\phi(\mathbf{y_n}|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x}_\mathrm{V})$ is required
- Parametrize $p_\phi(\mathbf{y_n}|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x}_\mathrm{V})$ with another GCN

$$p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB}(n)}, \mathbf{x}_V) = \mathrm{Cat}(\mathbf{y}_n|\mathrm{softmax}(W_\phi \mathbf{h}_{\phi,n}))$$

Overall Optimization Procedure

- Two Graph Neural Networks Collaborate with each other
  - p_ϕ: learning network, modeling the label dependency
  - q_θ: inference network, learning the object representations
- q_θ infer the labels of unlabeled objects trained with supervision from p_ϕ and labeled objects
- p_ϕ is trained with a fully labeled graph, where the unlabeled objects are labeled by q_θ