Structured Prediction of Network Response

Hongyu Su, Aristides Gionis, Juho Rousu

{hongyu.su, aristides.gionis, juho.rousu}@aalto.fi

Helsinki Institute for Information Technology, Aalto University, Finland

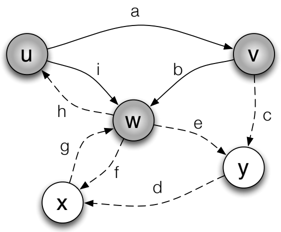
Aalto University
School of Science

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

I. Research Question

Given a complex network and an action, predict the subnetwork that responds to action, that is which nodes perform the actions and which directed edges rely the action to adjacent node.

II. An Example



$\mathbf{a} \xrightarrow{\phi} \phi(\mathbf{a})$

$(u, \mathbf{a}, t = 1)$
$(v, \mathbf{a}, t = 2)$
$(w, \mathbf{a}, t = 2)$

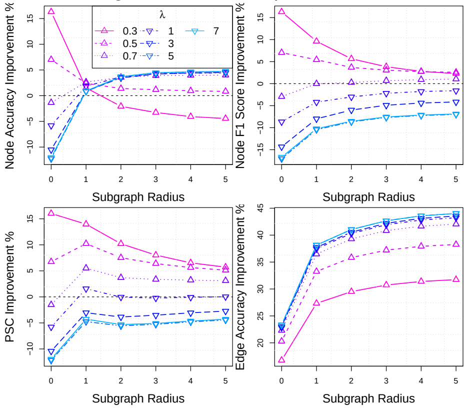
$G_{\mathbf{a}} \rightarrow \psi(G_{\mathbf{a}})$

$$\psi(G_{\mathbf{a}}) = (a_{\text{pp}}, a_{\text{pn}}, a_{\text{nn}}, b_{\text{pp}}, b_{\text{pn}}, b_{\text{nn}}, c_{\text{pp}}, c_{\text{pn}}, c_{\text{nn}}, \cdots)$$
$$= (1 \;\; 0, \;\; 0, \;\; 1, \;\; 0, \;\; 0, \;\; 0, \;\; 0 \;\; 1, \cdots)$$
$$\gamma_G(u) = (\gamma(u;a), \gamma(u;b), \gamma(u;c), \gamma(u;d), \cdots)$$
$$= (1, \quad \lambda, \quad \lambda, \quad \lambda^2, \quad \cdots)$$

1. G gets exposed to an action a, a *response network* $G_{\mathbf{a}} = (V_a, E_a) \subseteq G$ gets activated.
2. Feature map $A \xrightarrow{\phi} \mathcal{F}_A$ maps each action a to a k-dimensional vector $\phi(a)$ (e.g. bag-of-word).
3. Feature map $G \xrightarrow{\psi} \mathcal{F}_G$ encodes activated network $G_{\mathbf{a}}$ as $\psi(G_{\mathbf{a}})$ (a set of edges and labels).
4. Training data $\{(a_i, G_{\mathbf{a}_i})\}_{i=1}^m$ is given.
5. $\gamma(u)$ is scaling function (a set of edge weights).

V. Effect of γ_G

Figure 1: The improvement of performance for different scaling factor λ with respect to SVM.



III. Model For Network Response

Structured Prediction Model

Embed input a and output $G_{\mathbf{a}}$ in to a joint feature space $\varphi(\mathbf{a}, G_{\mathbf{a}}) = \phi(a) \otimes \psi(G_{\mathbf{a}})$. Learn a *compatibility score* on action a and response network $G_{\mathbf{a}}$

$$F(\mathbf{a}, G_{\mathbf{a}}; \mathbf{w}) = \langle \mathbf{w}, \varphi(\mathbf{a}, G_{\mathbf{a}}) \rangle,$$

where $\mathbf{w}$ is feature weights to be learned. Action with correct response network $(\mathbf{a}, G_{\mathbf{a}})$ will have higher compatibility score than any incorrect response network $(\mathbf{a}, G'_{\mathbf{a}})$.

Maximum Margin Learning

The feature weights $\mathbf{w}$ are learned by solving the following structured output learning problem (in primal form)

$$\min_{\mathbf{w},\xi} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{m}\xi_i$$
$$\text{s.t.} \quad F(a_i, G_{\mathbf{a}_i}; \mathbf{w}) > \operatorname*{argmax}_{G'_{\mathbf{a}_i} \in \mathcal{H}(G)} (F(a_i, G'_{\mathbf{a}_i}, \mathbf{w})$$
$$+ \ell_G(G_{\mathbf{a}_i}, G'_{\mathbf{a}_i})) - \xi_i, \xi_i \geq 0, \forall i \in \{1, \cdots, m\},$$

where C is the slack parameter that controls the amount of regularization, and $\mathcal{H}(G)$ is the collection of DAG in G.

Exponential scaling. Mistakes $\ell(G_{\mathbf{a}}, G'_{\mathbf{a}})$ are penalized by λ weighted according to the shortest-path distance to the focal point.

$$\gamma_G(v_k; v_i, v_j) = \begin{cases} 1 & \text{if } i = 0 \\ \lambda^{D(k,i)} & \text{if } i \neq 0 \text{ and } D(k,i) \leq R \\ \lambda^{(R+1)} & \text{if } D(k,i) > R \end{cases}$$

Diffusion scaling. Mistakes $\ell(G_{\mathbf{a}}, G'_{\mathbf{a}})$ are penalized by values from diffusion kernel K.

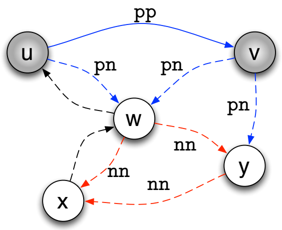
$$\gamma_G(v_k; v_i, v_j) = \begin{cases} 1 & \text{if } i = 0, \\ K(v_k, v_i) & \text{otherwise.} \end{cases}$$

Inference Problem

We have to solve the inference problem both in training and in prediction. Given feature weights $\mathbf{w}$ and the network G, the prediction for a new action a is the maximally-scoring response graph $H^* = (V_H, E_H)$

$$H^*(a) = \operatorname*{argmax}_{H \in \mathcal{H}(G)} F(\mathbf{a}, H; \mathbf{w})$$
$$= \operatorname*{argmax}_{H \in \mathcal{H}(G)} \langle \mathbf{w}, \phi(a) \otimes \psi(H) \rangle$$
$$= \operatorname*{argmax}_{H \in \mathcal{H}(G)} \sum_{e \in E^H} s_{\mathbf{y}_e}(e, a, \mathbf{w}), \tag{1}$$

where $\mathbf{y}_e$ is the possible label of edge e. Given $\mathbf{w}$ and a, $s_{\mathbf{y}_e}(e, a, \mathbf{w})$ will assign a score for labeling e as $\mathbf{y}_e$.



Activation mode. We assume $\mathbf{y}_e \in \{\text{pp}, \text{pn}\}$ and only consider the activated network.

Negative-feed mode. We assume $\mathbf{y}_e \in \{\text{pp}, \text{pn}, \text{nn}\}$ and consider both the activated network and the inactivated network.

Lemma 1. *Finding the graph that maximizes Eq. 1 is an $\mathcal{NP}$-hard problem.*

Inference Algorithm

The SDP inference. We introduce a variable $x_u \in \{-1, +1\}$ for each node $u \in V$, a special variable $x_0 \in \{-1, +1\}$ to distinguish activated node. Eq. 1 can be formulated as *quadratic programming*

$$\max \frac{1}{4} \sum_{(u,v) \in E} [s_{\text{pn}}(u,v)(1 + x_0 x_u - x_0 x_v - x_u x_v)$$
$$+ s_{\text{nn}}(u,v)(1 - x_0 x_u - x_0 x_v + x_u x_v)$$
$$+ s_{\text{pp}}(u,v)(1 + x_0 x_u + x_0 x_v + x_u x_v)]$$
$$\text{s.t.} \quad x_0, x_u, x_v \in \{-1, +1\}, \text{ for all } u, v \in V,$$

which can be solved by SDP relaxation as

$$\max \frac{1}{4} \sum_{u,v=1}^{k} [s_{\text{pn}}(u,v)(1 + y_{0,u} - y_{0,v} - y_{u,v})$$
$$+ s_{\text{nn}}(u,v)(1 - y_{0,u} - y_{0,v} + y_{u,v})$$
$$+ s_{\text{pp}}(u,v)(1 + y_{0,u} + y_{0,v} + y_{u,v})]$$
$$\text{s.t.} \quad Y \succeq 0.$$

The GREEDY inference. Eq. 1 can be formulated equivalently as a a function of activated vertices

$$H^*(\mathbf{a}) = \operatorname*{argmax}_{H \in \mathcal{H}(G)} \sum_{v_i \in V_{\text{p}}^H} F_m(v_i), \tag{2}$$

where F_m is the marginal gain function of adding v_i into activated vertex set defined as

$$F_m(v_i) = \sum_{v_p \in \text{parents}(v_i)} [s_{\text{pp}}(v_p, v_i) - s_{\text{pn}}(v_p, v_i)]$$
$$+ \sum_{v_c \in \text{children}(v_i)} [s_{\text{pn}}(v_i, v_c) - s_{\text{nn}}(v_i, v_c)].$$

The GREEDY algorithm, in $\Theta(|E|\log|V|)$, iteratively maximize Eq. 2 by adding vertex into activated vertex set.

IV. Experimental Results

Table 1: Context free prediction, where action feature is unknown and the task is to predict network skeleton. The measure of success is *Precision@K*, where we ask for top-K percent edge predictions from each model and compute the precision.

Dataset	Model	T (10^3s)	Precision @ K							
			10%	20%	30%	40%	50%	60%	70%	80%
memeS	SPIN	*5.50*	**82.9**	**81.0**	**76.0**	**74.0**	**74.0**	**70.0**	**69.8**	**67.9**
	ICM-EM	**0.01**	60.3	63.5	65.1	62.0	62.0	61.5	62.2	60.4
	NETRATE	5.83	*76.2*	*73.8*	*70.4*	*68.7*	*68.7*	*66.8*	*64.9*	*63.4*
memeM	SPIN	*5.52*	**82.7**	**72.1**	**70.5**	**69.2**	**69.2**	**67.9**	**66.2**	**65.6**
	ICM-EM	**0.02**	56.3	55.3	56.8	57.4	57.4	56.3	57.5	57.8
	NETRATE	13.93	*61.2*	*64.6*	*62.9*	*62.5*	*62.5*	*62.4*	*61.2*	*60.1*
memeL	SPIN	*4.75*	**82.2**	**73.6**	**69.1**	**66.7**	**66.7**	**65.9**	**66.1**	**65.9**
	ICM-EM	**0.01**	52.1	55.7	54.2	56.5	56.5	56.7	57.4	58.0
	NETRATE	12.63	*56.5*	*57.8*	*60.0*	*59.3*	*59.3*	*59.4*	*58.9*	*58.4*

Table 2: Context sensitive prediction, where action feature is assumed to be known and the task is to predict the response network $G_{\mathbf{a}}$ given an action a. *Predicted Subgraph Coverage* (PSC) is defined as $\text{PSC} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{v \in V_i}|G_v|$.

Dataset	Node Accuracy			Node F_1 Score			Edge Acc		*PSC*			Time (10^3s)		
	SVM	MMCRF	SPIN	SVM	MMCRF	SPIN	SVM	SPIN	SVM	MMCRF	SPIN	SVM	MMCRF	SPIN
memeS	**73.4**	68.0	*72.2*	39.0	*39.8*	**47.1**	**62.7**	45.6	23.4	*25.3*	**33.6**	6.6	**2.9**	*4.1*
memeM	**82.1**	79.0	*81.5*	29.1	*30.1*	**38.0**	61.1	**68.8**	18.6	*18.8*	**28.3**	13.7	**3.2**	*7.3*
memeL	**89.9**	88.3	*89.8*	26.7	*27.1*	**35.0**	45.5	**80.0**	17.7	*18.9*	**27.6**	19.9	**5.9**	*11.8*
M700	91.9	**94.1**	*92.1*	*13.8*	7.3	**14.2**	26.3	**93.0**	*29.4*	23.9	**34.4**	18.5	*8.3*	**4.4**
M1k	94.1	**95.8**	*94.2*	**10.9**	3.5	*9.3*	26.6	**94.7**	*33.7*	16.6	**35.2**	42.2	*14.7*	**10.4**
M2k	*96.8*	**97.6**	96.7	**6.2**	1.4	*3.4*	25.3	**97.6**	**34.6**	9.6	*14.7*	165.0	*88.4*	**54.1**
L700	89.7	**92.4**	*89.7*	*16.2*	9.4	**17.3**	26.5	**90.4**	*9.5*	6.7	**12.5**	16.0	*7.8*	**5.3**
L1k	*92.4*	**94.4**	91.5	*12.4*	6.4	**13.9**	26.4	**92.3**	*6.1*	4.4	**8.4**	40.3	*13.7*	**10.4**
L2k	*92.5*	**94.5**	91.9	*12.3*	5.4	**12.7**	26.5	**93.2**	*6.0*	2.9	**7.2**	41.9	*21.9*	**13.1**
Geom.	85.5	*86.4*	**86.6**	*19.8*	12.6	**20.3**	32.6	**79.7**	*18.9*	14.2	**21.7**	9.4	*4.6*	**4.3**