Structured Prediction of Network Response

Hongyu Su, Aristides Gionis, and Juho Rousu

Helsinki Institute for Information Technology (HIIT)
Department of Information and Computer Science, Aalto university

We introduce the *network response* problem (Su et al. 2014): given a network $G = (V, E)$ and an action $\mathbf{a}$, predict the subnetwork $G_{\mathbf{a}} = (V_{\mathbf{a}}, E_{\mathbf{a}})$ that responses to the action, that is, which nodes $v \in V_{\mathbf{a}}$ perform the action, and which directed edges $e = (v, u) \in E_{\mathbf{a}}$ relay the action from v to an adjacency node u. We assume that G is directed, and any undirected network can be seen as a special case.

We assume each action $\mathbf{a}$ is represented by a feature map $\phi(\mathbf{a})$ (e.g., bag-of-words). We use output feature map $\psi(G_{\mathbf{a}})$ (e.g., vector of edges and labels) to encode the response graph $G_{\mathbf{a}}$. Our model is based on embedding $\mathbf{a}$ and $G_{\mathbf{a}}$ into a joint feature space and learn from on that space a compatibility score function

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

where $\varphi(\mathbf{a}, G_{\mathbf{a}}) = \phi(\mathbf{a}) \otimes \psi(G_{\mathbf{a}})$ denotes the joint feature map. The intuition behind this formulation is that the action with correct response graph $G_{\mathbf{a}}$ will achieve higher compatibility score than the action with an incorrect network $G'_{\mathbf{a}}$.

The feature weight parameter $\mathbf{w}$ of the compatibility score function F is learned via a regularized structured-output learning problem, where we have to solve similar inference problem both in training and in prediction. In prediction, given feature weights $\mathbf{w}$ and a network $G = (V, E)$, the prediction for a new input action $\mathbf{a}$ is the maximally-scoring response graph $H^* = (V^H, E^H)$

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

where we have substituted $s_{\mathbf{y}_e}(e, \mathbf{a}, \mathbf{w}) = \sum_i w_{i,e,\mathbf{y}_e} \phi_i(\mathbf{a})$ and $\mathcal{H}(G)$ is a set of DAG of G. Depending on the values $\mathbf{y}_e$ can take, the inference problem diverges into two modes. **Activation mode**, we only consider the activated part of the network G by setting $\mathbf{y}_e \in \{\mathtt{pp}, \mathtt{pn}\}$. **Negative-feed mode**, we also consider the inactivated part of the network G by setting $\mathbf{y}_e \in \{\mathtt{pp}, \mathtt{pn}, \mathtt{nn}\}$.

We have proved the inference is $\mathcal{NP}$-hard, and propose two inference algorithms. The SDP-INFERENCE algorithm formulates the inference problem as a quadratic program and solves the *QP* by SDP relaxation. The GREEDY-INFERENCE algorithm is a greedy scheme that solves the problem by iteratively maximizing $H^*(\mathbf{a}) = \mathbf{argmax}_{H \in \mathcal{H}(G)} \sum_{v_i \in V_{\mathrm{p}}^H} F_m(v_i)$, where F_m is the marginal gain function of adding v_i into activated vertex set.

We evaluate the proposed method, which we call SPIN, by comparing it with state-of-the-art network-inference methods on *DBLP* and *Memetracker* datasets. We use two popular metrics *accuracy* and F_1 *score*. In addition, we compute *Predicted Subgraph Coverage* (PSC) defined as $PSC = \frac{1}{mn} \sum_{i=1}^{m} \sum_{v \in V_i} |G_v|$,

Dataset	**Node Accuracy**			**Node F_1 Score**			**Edge Acc**		*PSC*		
	SVM	MMCRF	SPIN	SVM	MMCRF	SPIN	SVM	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**
memeM	**82.1**	79.0	*81.5*	29.1	*30.1*	**38.0**	61.1	**68.8**	18.6	*18.8*	**28.3**
memeL	**89.9**	88.3	*89.8*	26.7	*27.1*	**35.0**	45.5	**80.0**	17.7	*18.9*	**27.6**
M100	71.2	*73.6*	**76.7**	49.3	*50.8*	**54.3**	33.3	**61.7**	33.3	**35.6**	*34.6*
M500	89.0	*91.4*	**92.0**	**18.8**	13.5	*14.6*	28.2	**92.6**	*29.3*	26.4	**29.5**
M700	91.9	**94.1**	*92.1*	*13.8*	7.3	**14.2**	26.3	**93.0**	*29.4*	23.9	**34.4**
M1k	94.1	**95.8**	*94.2*	**10.9**	3.5	*9.3*	26.6	**94.7**	*33.7*	16.6	**35.2**
M2k	*96.8*	**97.6**	96.7	**6.2**	1.4	*3.4*	25.3	**97.6**	**34.6**	9.6	*14.7*
L100	69.4	*72.2*	**75.7**	51.1	*53.1*	**57.4**	31.6	**62.3**	30.9	*31.7*	**33.4**
L500	85.9	**89.1**	*86.8*	*21.7*	15.1	**24.7**	27.9	**87.9**	*14.2*	11.2	**19.7**
L700	89.7	**92.4**	*89.7*	*16.2*	9.4	**17.3**	26.5	**90.4**	*9.5*	6.7	**12.5**
L1k	*92.4*	**94.4**	91.5	*12.4*	6.4	**13.9**	26.4	**92.3**	*6.1*	4.4	**8.4**
L2k	*92.5*	**94.5**	91.9	*12.3*	5.4	**12.7**	26.5	**93.2**	*6.0*	2.9	**7.2**
Geom.	85.5	*86.4*	**86.6**	*19.8*	12.6	**20.3**	32.6	**79.7**	*18.9*	14.2	**21.7**

Table 1. Comparison of prediction performance on global context.

Dataset	**Model**	**T** (10^3s)	Precision @ K									
			10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
memeS	SPIN	*5.50*	**82.9**	**81.0**	**76.0**	**74.0**	**74.0**	**70.0**	**69.8**	**67.9**	**66.7**	**64.7**
	ICM-EM	**0.01**	60.3	63.5	65.1	62.0	62.0	61.5	62.2	60.4	60.7	*61.9*
	NETRATE	5.83	*76.2*	*73.8*	*70.4*	*68.7*	*68.7*	*66.8*	*64.9*	*63.4*	*62.9*	*61.9*
memeM	SPIN	*5.52*	**82.7**	**72.1**	**70.5**	**69.2**	**69.2**	**67.9**	**66.2**	**65.6**	**64.3**	**64.2**
	ICM-EM	**0.02**	56.3	55.3	56.8	57.4	57.4	56.3	57.5	57.8	58.3	*58.5*
	NETRATE	13.93	*61.2*	*64.6*	*62.9*	*62.5*	*62.5*	*62.4*	*61.2*	*60.1*	*58.7*	*58.5*
memeL	SPIN	*4.75*	**82.2**	**73.6**	**69.1**	**66.7**	**66.7**	**65.9**	**66.1**	**65.9**	**63.9**	**63.6**
	ICM-EM	**0.01**	52.1	55.7	54.2	56.5	56.5	56.7	57.4	58.0	*57.6*	*57.0*
	NETRATE	12.63	*56.5*	*57.8*	*60.0*	*59.3*	*59.3*	*59.4*	*58.9*	*58.4*	57.5	*57.0*

Table 2. Model performance in context-free influence network prediction.

where V_i is the set of focal node given action $\mathbf{a}$. PSC expresses the relative size of correctly predicted subgraph in terms of node labels.

For context-aware prediction, we assume the action feature is known and the task is to predict the response network given an action. The prediction performance against SVM and MMCRF is listed in Table 1. We observe that SPIN can dramatically boost the performance and is 2-3 times faster than other models.

For context-free prediction, the task is to predict the network skeleton given only a collection of the response subnetworks. The measure of success is *Precision@K*, where we ask for top-K edge predictions from each model and compute the precision. The result, shown in Table 2, indicates that SPIN outperforms NETRATE and ICM-EM in all K spectrum.

The results demonstrate that SPIN can be successfully used in influence network prediction problems, achieving superior performance compared to other advanced models.

Reference

Hongyu Su, Aristides Gionis, Juho Rousu, Structured Prediction of Network Response. ICML-2014, to appear