Nonlinear Label Spreading on Hypergraphs

Konstantin Prokopchik

Gran Sasso Science Institute (GSSI), L'Aquila, Italy
konstantin.prokopchik@gssi.it

joint work with: Austin R Benson (Cornell Univ.) and Francesco Tudisco (GSSI)

Label Spreading (LS) and Semi-Supervised Learning (SSL)

- Given the dataset made out of c classes, In **SSL** the task is to assign unknown labels based on a small portion of known input labels
- In **LS** unknown labels are inferred by "spreading" the known labels following the edges of a graph
- Data should be represented as a graph that could be either a point cloud or a relational network

Higher-order notation

- $H = (V, \mathcal{E}, \omega)$: $\mathcal{E} = \{e_1, \dots, e_m\}$ and $w(e) > 0$ is a positive weight
- Every edge can contain an arbitrary number of nodes.
- $D = \mathrm{Diag}(\delta_1, \dots, \delta_n)$, where $\delta_i = \sum_{e: i \in e} w(e)$ - the (hyper)degree of node i
- We assume that $\delta_i > 0$ for all i, i.e. that hypergraph has no isolated nodes

Higher-order notation

Incidence matrix:

$$K_{i,e} = \begin{cases} 1 & i \in e \\ 0 & \text{otherwise.} \end{cases}$$

- $W = (w(e_1), \dots, w(e_m))$ - weight matrix
- $X \in \mathbb{R}^{n \times d}$, where row $x_i = X_{i,:} \in \mathbb{R}^d$ is the feature vector of $i \in V$
- Suppose each node i belongs to one of c classes $\{1, \dots, c\}$ and we know the label of a (small) subset $\mathcal{T} \subset V$
- $Y \in \mathbb{R}^{n \times c}$ the input-labels matrix of the nodes, in which $Y_{ij} = 1$ if node i belongs to class j, and $Y_{ij} = 0$ otherwise.

Laplacian regularization

- $\min_F \ell_\Omega := \|F - Y\|^2 + \lambda\,\Omega(F)$ - regularized square loss function
- $\Omega_{L^2}(F) = \sum_{e\in E}\sum_{i,j\in e}\frac{w(e)}{|e|}\left\|\frac{f_i}{\sqrt{\delta_i}} - \frac{f_j}{\sqrt{\delta_j}}\right\|^2$ - clique expansion approach [Zhou et al., 2007].
- $\Omega_{TV}(F) = \sum_{e\in E} w(e)\max_{i,j\in e}\|f_i - f_j\|^2$ - total variation on hypregraph regularizer [Hein et al., 2013]

Label Spreading

- For Ω_{L^2} we can use the power method, as $\nabla \ell_{\Omega_{L^2}}$ is linear:

  $$F^{(k+1)} = \alpha \bar{A}_H F^{(k)} + (1 - \alpha) Y,$$

  where $\alpha = \lambda/(1 + \lambda)$ and $\bar{A}_H$ is the normalized adjacency matrix of the clique-expanded graph of H.

  We call this method **"Higher Order Label Spreading"**
- For Ω_{TV} we have to use more complex approaches, as it is not easily interpreted as a label diffusion

Hyperedge variance regularization

We introduce a new hypergraph regularization term that aims at reducing the variance across the hyperedge nodes:

- $\Omega_\mu(F) = \sum_{e\in E} \sum_{i\in e} w(e) \Big\| \frac{f_i}{\sqrt{\delta_i}} - \mu\Big(\Big\{ \frac{f_j}{\sqrt{\delta_j}} : j \in e \Big\}\Big) \Big\|^2$
- When μ is the mean $\mu(\{z_j : j \in e\}) = \frac{1}{|e|} \sum_{j\in e} z_j$, we obtain the variance of $f_i/\sqrt{\delta_i}$ on the hyperedge e

Hyperedge variance regularization

In this presentation we consider:

$$\mu(\{\frac{f_i}{\sqrt{\delta_i}}, i \in e\}) = \mathrm{mean}_p\{\frac{f_i}{\sqrt{\delta_i}} : i \in e\} = \Big(\frac{1}{|e|}\sum_{i \in e}(\frac{f_i}{\sqrt{\delta_i}})^p\Big)^{1/p}$$

With this family of μ functions the embedding F minimizes the variation of each node embedding f_i from the p-power mean of the embeddings of the nodes in each hyperedge i participates in.

Nonlinear diffusion method

Recall that each node $i \in V$ has a label-encoding vector y_i and a feature vector x_i, hence the initial embedding is $(c+d)$-dimensional and forms an input matrix $U = [Y \ X]$

$$\begin{cases} F^{(k+1)} = \alpha\,\Phi(F^{(k)}) + (1-\alpha)\,U \\ \Phi(F) = D^{-1/2} K W \sigma(K^\top \varrho(D^{-1/2} F)) \end{cases}$$

- $\varrho(Z_1) := Z_1^p,\ \sigma(Z_2) := (D_E^{-1} Z_2)^{1/p}$
- We will show that the limit point of the diffusion process $F_\star = [Y_\star \, X_\star] \in \mathbb{R}^{n \times (c+d)}$ exists, is unique and minimizes a normalized version of the SSL regularized loss $\ell_{\Omega_{\mu_{\sigma,\varrho}}}$.
- We will then use $F_\star$ to train a logistic multi-class classifier based on the known labels $i \in \mathcal{T}$

Relation with HOLS

Looking at the iterative processes again:

$$F^{(k+1)} = \alpha \bar{A}_H F^{(k)} + (1-\alpha) Y$$
$$F^{(k+1)} = \alpha \, \Phi(F^{(k)}) + (1-\alpha) \, U$$

- Our diffusion process propagates both input node label and feature embeddings through the hypergraph in a manner similar to the case with Ω_{L^2}, but allowing for nonlinear activations, which increases the modeling power.
- $\Phi(F) = \bar{A}_H$ when σ and ϱ are linear

Related nonlinear diffusion models

$$\Phi(x) = K\sigma(K^T(\varrho(x))$$

Different choices of σ and ϱ are used in different settings:

- If $\varrho = \mathrm{id}$ and $\sigma(x) = |x|^{p-1}\mathrm{sign}(x)$ .... graph p-Laplacian [Saito et al., 2018]
- *exp* and *log* .... chemical reactions and consensus dynamics [Schaft et al., 2016] [Neuhäuser et al., 2021]
- Trigonometric functions .... network oscillators [Battiston et al., 2021] [Schaub et al., 2016]
- Polynomials .... semi-supervised learning [Arya et al., 2021] [Ibrachim & Gleich, 2021] [Tudisco et al., 2021]

Main theorem

Theorem

Let Φ and μ be defined as before. Define the real-valued function:

$$\varphi(F) = 2\sqrt{\sum_{e\in E} w(e)\Big\|\mu\Big(\Big\{\frac{f_j}{\sqrt{\delta_j}}, j \in e\Big\}\Big)\Big\|^2}$$

Then, for any starting point $F^{(0)} \geq 0$, the sequence

$$\begin{cases} \tilde{F}^{(k)} = \alpha\, \Phi(F^{(k)}) + (1-\alpha)\, U \\ F^{(k+1)} = \tilde{F}^{(k)}/\varphi(\tilde{F}^{(k)}) \end{cases} \quad \to F_\star$$

such that $\varphi(F_\star) = 1$, $F_\star > 0$. Moreover, $F_\star$ is the solution of

$$\begin{cases} \min_F \ \Big\|F - \dfrac{U}{\varphi(U)}\Big\|^2 + \lambda\, \Omega_\mu(F) \\ \text{subject to } \ F \geq 0,\ \varphi(F) = 1, \end{cases} \quad \textit{where } \lambda = \alpha/(1-\alpha)$$

Datasets

We use five co-citation and co-authorship hypergraphs: Cora co-authorship, Cora co-citation, Citeseer, Pubmed [Sen et al., 2008] and DBLP [Rossi & Ahmed et al., 2015]. All nodes in the datasets are documents, features are given by the content of the abstract and hyperedge connections are based on either co-citation or co-authorship. The task for each dataset is to predict the topic to which a document belongs. We also consider a foodweb hypergraph, where the nodes are organisms and hyperedges represent directed carbon exchange in the Florida bay **foodweb**. Here we predict the role of the nodes in the food chain.

		DBLP co-authorship	**Pubmed** co-citation	**Cora** co-authorship	**Cora** co-citation	**Citeseer** co-citation	**Foodweb** carbon-exchange
$\lvert V \rvert$	(#nodes)	43413	19717	2708	2708	3312	122
$\lvert E \rvert$	(#hyperedges)	22535	7963	1072	1579	1079	141233
d	(#features)	1425	500	1433	1433	3703	0
c	(#labels)	6	3	7	7	6	3

Competitors

- **HGNN** - hypergraph neural network model that uses the clique-expansion Laplacian for the hypergraph convolutional filter [Feng et al., 2019]
- **HyperGCN** - hypergraph convolutional network model with regularization similar to the total variation [Yadati et al., 2019]
- **HTV** - confidence-interval subgradient-based method that minimizes the Ω_{TV} loss. [Hein et al., NeurIPS, 2013]
- **APPNP** - graph convolutional network model combined with PageRank [Klicpera et al., 2019]
- **SGC** - graph convolutional network model without nonlinearities [Wu et al., 2017]
- **SCE** - graph convolutional network model inspired by a sparset-cut problem, where unsupervised network embedding is learned only using negative samples for training.. [Zhang et al., ICML, 2020]

Method comparison

Setup: For HyperND and HTV we run 5-fold CV with label-balanced 50/50 splits to choose α from $\{0.1, 0.2, \ldots, 0.9\}$ and p from $\{1, 2, 3, 5, 10\}$. For the network-based models we use 2 layers and 200 epochs.

	Method	**HyperND**	**APPNP**	**HGNN**	**HyperGCN**	**SGC**	**SCE**	**HTV**
Data	% labeled							
Citeseer	4.2%	**72.13** ±1.00	63.51 ±1.39	61.78 ±3.46	50.94 ±8.27	52.66 ±2.18	61.28 ±1.61	29.63±0.3
Cora-author	5.2%	**77.33** ±1.51	71.34 ±1.60	63.11 ±2.73	61.27 ±1.06	30.46 ±0.22	71.96 ±2.18	44.55±0.6
Cora-cit	5.2%	**83.13** ±1.11	82.08 ±1.61	62.88 ±2.26	62.78 ±2.73	29.08 ±0.25	79.85 ±1.91	35.60±0.8
DBLP	4.0%	**89.63** ±0.12	88.94 ±0.07	73.82 ±0.71	70.02 ±0.10	43.61 ±0.17	87.50 ±0.19	45.19±0.9
Foodweb	5.0%	64.09 ±5.94	**69.12** ±3.30	57.09 ±2.33	56.14 ±3.85	57.45 ±0.47	63.50 ±4.78	57.23±0.9
Pubmed	0.8%	**82.81** ±2.16	81.50 ±1.18	72.57 ±1.03	78.11 ±0.99	54.30 ±1.11	77.57 ±2.34	47.04±0.8

Table: Accuracy (mean ± standard deviation) over five random samples of the training nodes $\mathcal{T}$. We compare HyperND and the six baseline methods (APPNP, HGNN, HyperGCN, SGC, SCE, HTV). Overall, HyperND is more accurate than the baselines.

Time comparison

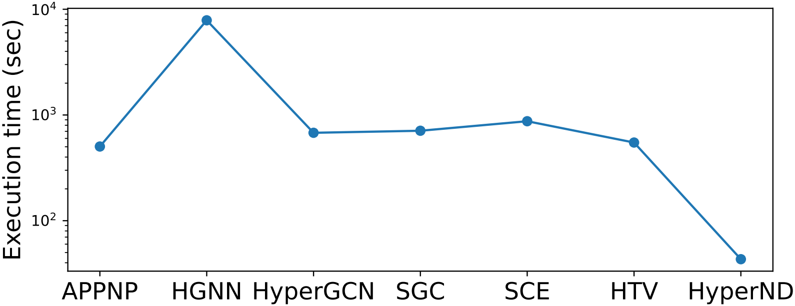

Figure: Execution time on the largest dataset DBLP (for one hyper-parameter setting in each case). All methods are comparable on small datasets.

Papers

This presentation was based on the works of two papers:

- F. Tudisco, A. R. Benson, K. Prokopchik, Nonlinear Higher-Order Label Spreading [WWW 2021]
- F. Tudisco, K. Prokopchik, A. R. Benson, A nonlinear diffusion method for semi-supervised learning on hypergraphs, arXiv:2103.14867

Thank You!

Parameter dependence

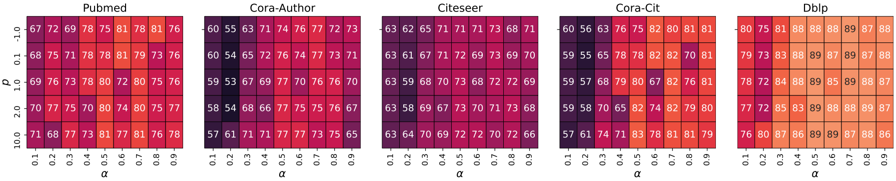

Figure: Performance of the proposed HyperND for varying p and α parameters.

Embedding comparison

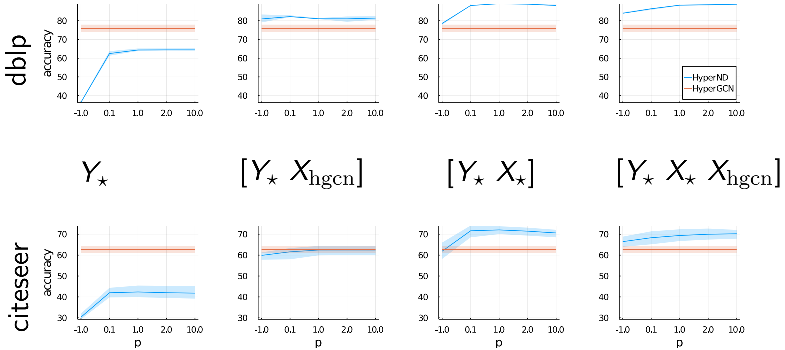

Figure: Accuracy (mean and standard deviation) of multinomial logistic regression classifier, using different combinations of features obtained from embeddings.