Mean-field theory of graph neural networks in graph partitioning

Tatsuro Kawamoto, Masashi Tsubaki (AIST, Japan), Tomoyuki Obuchi (Tokyo Tech., Japan)

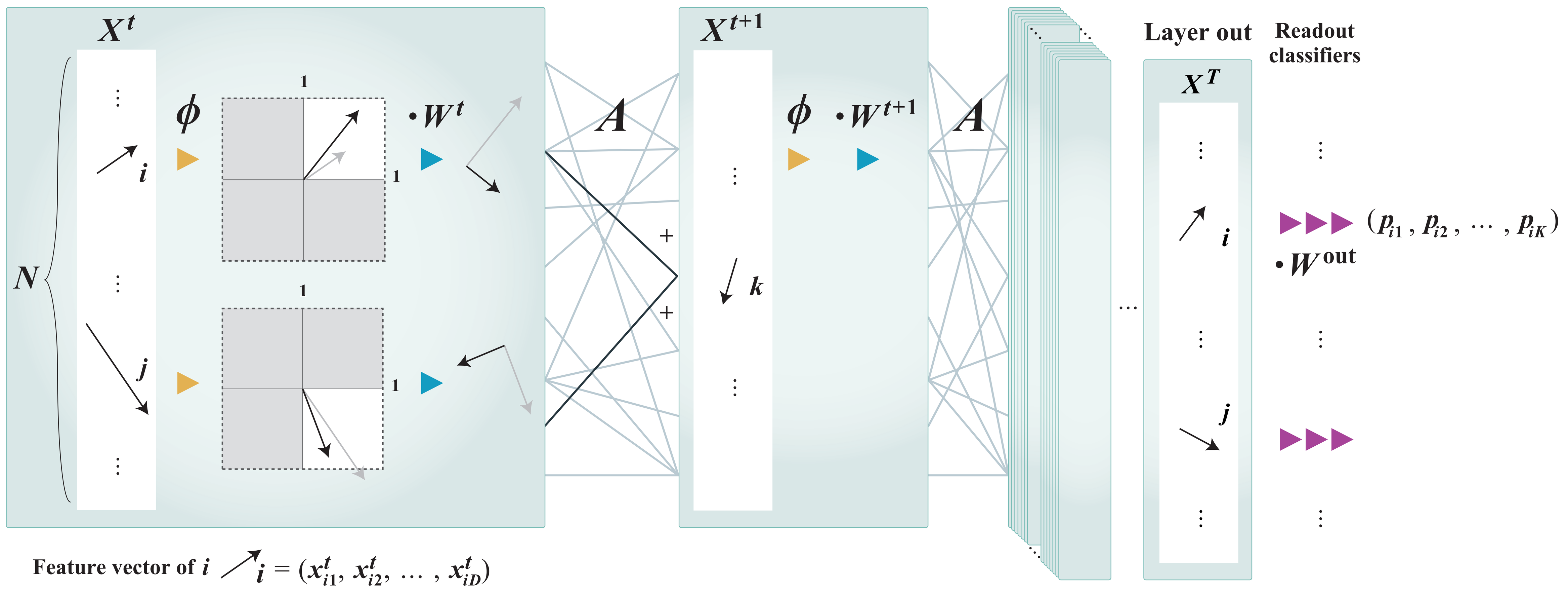
Background

When Graph Neural Networks (GNN) perform well, is it thanks to

- learning of model parameters (e.g., backprop.)? or
- architecture of the model (nonlinear structure + input graph) itself?

Question

For a simple graph partitioning problem, does a GNN already perform well without training?
Can we evaluate the performance analytically?



A *minimal* GNN (untrained)

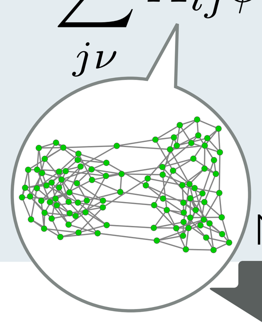
Random N×D matrix
$X^0 \equiv [x^0_{i\mu}]$

tanh(x)

$$x^{t+1}_{i\mu} = \sum_{j\nu} A_{ij}\phi\left(x^t_{j\nu}\right) W^t_{\nu\mu}$$

untrained GNN

Gaussian random matrix
$W^t_{\nu\mu} \sim \mathcal{N}(0, 1/D)$



N×N adjacency matrix

$X^T = [x^T_{i\mu}]$
Classify using the k-means

Classification of D-dim vectors

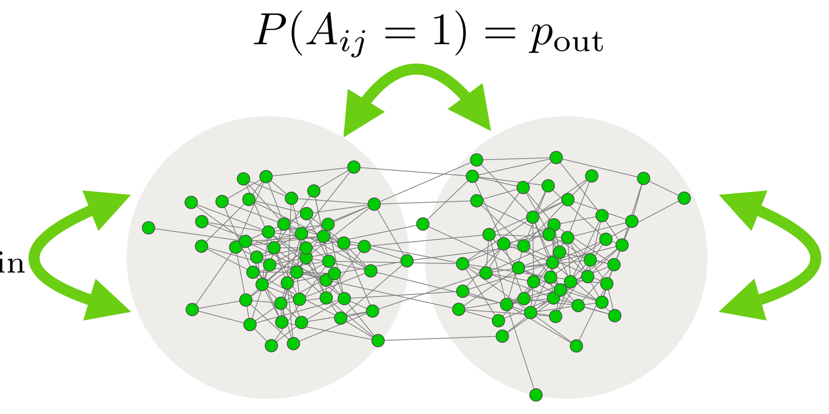
Stochastic block model inference

Graph partitioning

$|V| = N$

$P(A_{ij} = 1) = p_{\mathrm{out}}$

$P(A_{ij} = 1) = p_{\mathrm{in}}$



$P(A_{ij} = 1) = p_{\mathrm{in}}$

(equal-size groups)

c = average degree

$$\epsilon \equiv \frac{p_{\mathrm{out}}}{p_{\mathrm{in}}}$$

GNN is related to many other algorithms

$$x^{t+1}_{i\mu} = \sum_j M_{ij}\,\varphi\left(\sum_\nu \phi(x^t_{j\nu})W^t_{\nu\mu}\right) + b^t_{i\mu}$$

algorithm	domain	M	$\phi(x)$	$\varphi(x)$	W^t	b^t	$\{W^t, b^t\}$ update
untrained GNN	V	A	tanh	$I(x)$	random	omitted	not trained
trained GNN [KipfWelling2016]	V	$I-L$	ReLu	$I(x)$	trained	omitted	trained via backprop.
Spectral method	V	L	$I(x)$	$I(x)$	QR	/	updated at each layer
EM + BP	E	B	softmax	$\log(x)$	learned	learned	learned via M-step

Mean-field（Martin-Siggia-Rose formalism）

[A Crisanti, HJ Sommers, and H Sompolinsky, (1990)]

We introduce a group-wise (macroscopic) state variable.

$$\mathrm{x}^t_{\sigma\mu} \equiv (\gamma_\sigma N)^{-1} \sum_{i\in V_\sigma} x^t_{i\mu}$$

$\mathrm{x}^t = [\mathrm{x}^t_{\sigma\mu}]$ (2×D matrix)

$\sigma \in \{1, 2\}$ group label

The state equation

$$P(\mathrm{x}^{t+1}) = \left\langle \prod_{\sigma\mu} \delta\left(\mathrm{x}^{t+1}_{\sigma\mu} - \frac{1}{\gamma_\sigma N}\sum_{i\in V_\sigma}\sum_{j\nu} A_{ij}\phi(x^t_{j\nu})W^t_{\nu\mu}\right)\right\rangle_{A, W^t, X^t}$$

Averages w.r.t.

- graph(stochastic block model)
- $W^t_{\nu\mu} \sim \mathcal{N}(0, 1/D)$
- state in the previous layer *t*

Covariance matrix of x^t in the limit of T = ∞

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \qquad B = \frac{N}{4}\begin{pmatrix} p_{\mathrm{in}} & p_{\mathrm{out}} \\ p_{\mathrm{out}} & p_{\mathrm{in}} \end{pmatrix}$$

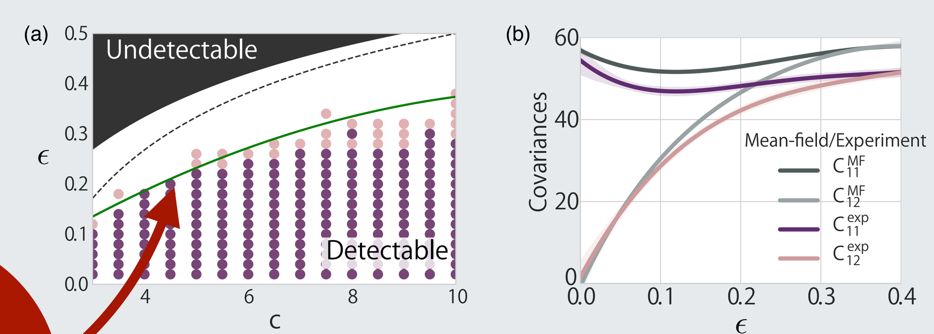
Self-consistent equation w. r. t. C

$$C_{\sigma\sigma'} = \frac{1}{\gamma_\sigma\gamma_{\sigma'}} \sum_{\tilde{\sigma}\tilde{\sigma}'} B_{\sigma\tilde{\sigma}}B_{\sigma'\tilde{\sigma}'} \int \frac{d\mathrm{x}\, \mathrm{e}^{-\frac{1}{2}\mathrm{x}^\top C^{-1}\mathrm{x}}}{(2\pi)^{\frac{N}{2}}\sqrt{\det C}}\, \phi(\mathrm{x}_{\tilde{\sigma}})\phi(\mathrm{x}_{\tilde{\sigma}'})$$

Numerical experiments

Comparison of the (*algorithmic*) *detectability limit estimates* between the mean-field result & numerical experiments

Behaviors of $C = [C_{\sigma\sigma'}]$



(algorithmic) detectability limit
The limit such that an algorithm cannot identify statistically significant groups.

$C_{11} \simeq C_{12}$
overlap ~ 0.5

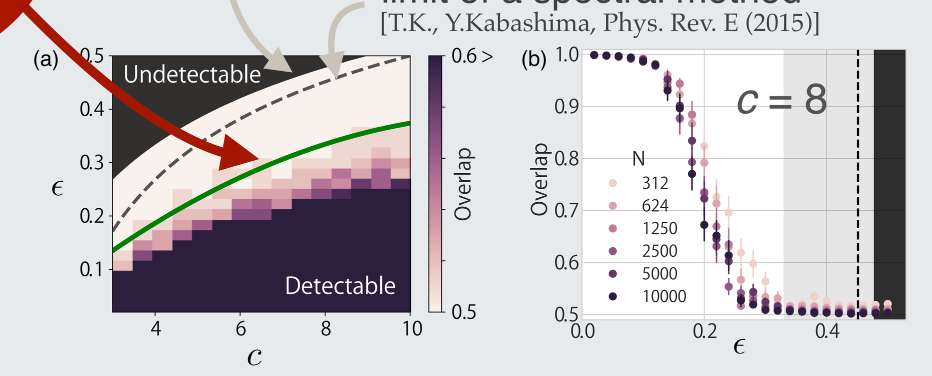
Mean-field estimate

information-theoretic limit
[E.Mossel, J.Neeman, and A.Sly, Probab. Theory Relat. Fields (2015)]

limit of a spectral method
[T.K., Y.Kabashima, Phys. Rev. E (2015)]

Behaviors of the overlaps

overlap = fraction of correctly classified vertices



Even a minimal GNN performs well, indeed.

A trained GNN

(This is not the only way, and probably not the best way.)

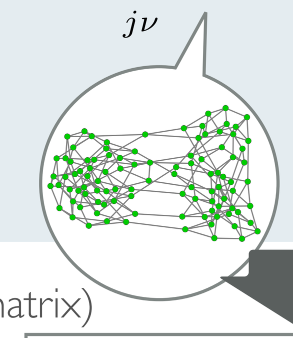
random N×D matrix
$X^0 \equiv [x^0_{i\mu}]$

tanh(x)

$$x^{t+1}_{i\mu} = \sum_{j\nu} A_{ij}\phi\left(x^t_{j\nu}\right) W^t_{\nu\mu}$$

trained GNN

Back prop.



N×N adjacency matrix

(D×2 matrix)

Read out classifier W^{out}

error function

learn from the “correct” label

normalized mutual information
error function

$$a_{i\sigma} = \sum_\mu x_{i\mu}W^{\mathrm{out}}_{\mu\sigma} \qquad p_{i\sigma} \equiv \mathrm{softmax}(a_{i\sigma})$$

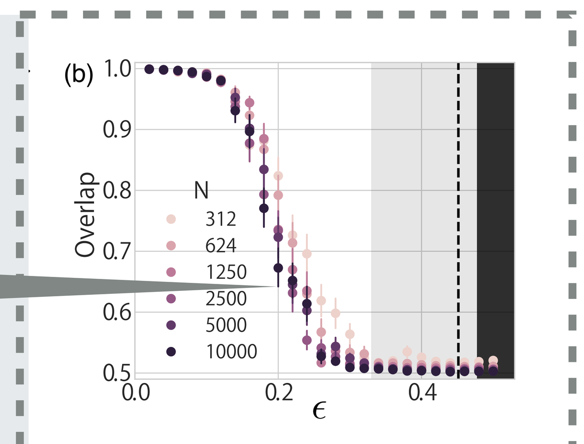
$$P_{\sigma\hat{\sigma}} = \frac{1}{N}\sum_{i=1}^N P(i\in V_\sigma, i\in V_{\hat{\sigma}}) = \frac{1}{N}\sum_{i\in V_\sigma} p_{i\hat{\sigma}},$$

$$P_\sigma = \sum_{\hat{\sigma}} P_{\sigma\hat{\sigma}} = \gamma_\sigma,$$

$$\mathrm{NMI}\left([P_{\sigma\hat{\sigma}}]\right) = 2\left(1 - \frac{\sum_{\sigma\hat{\sigma}} P_{\sigma\hat{\sigma}}\log P_{\sigma\hat{\sigma}}}{\sum_\sigma \gamma_\sigma\log\gamma_\sigma + \sum_{\sigma\hat{\sigma}} P_{\sigma\hat{\sigma}}\log\sum_\sigma P_{\sigma\hat{\sigma}}}\right)$$

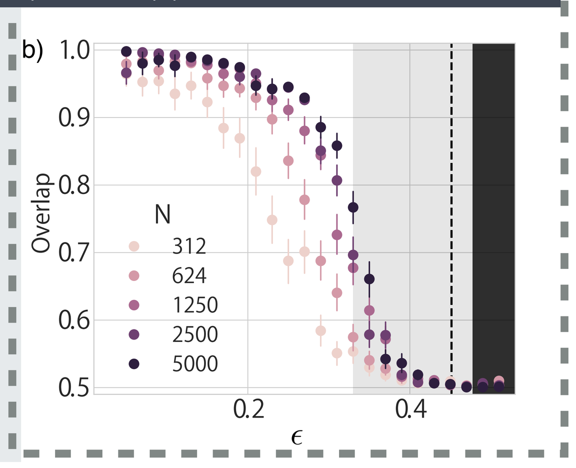
untrained GNN

Performance deteriorates because of the k-means.



The transition points appear to be the same…!

trained GNN



Chainer
+ Batch normalization
+ ResNet