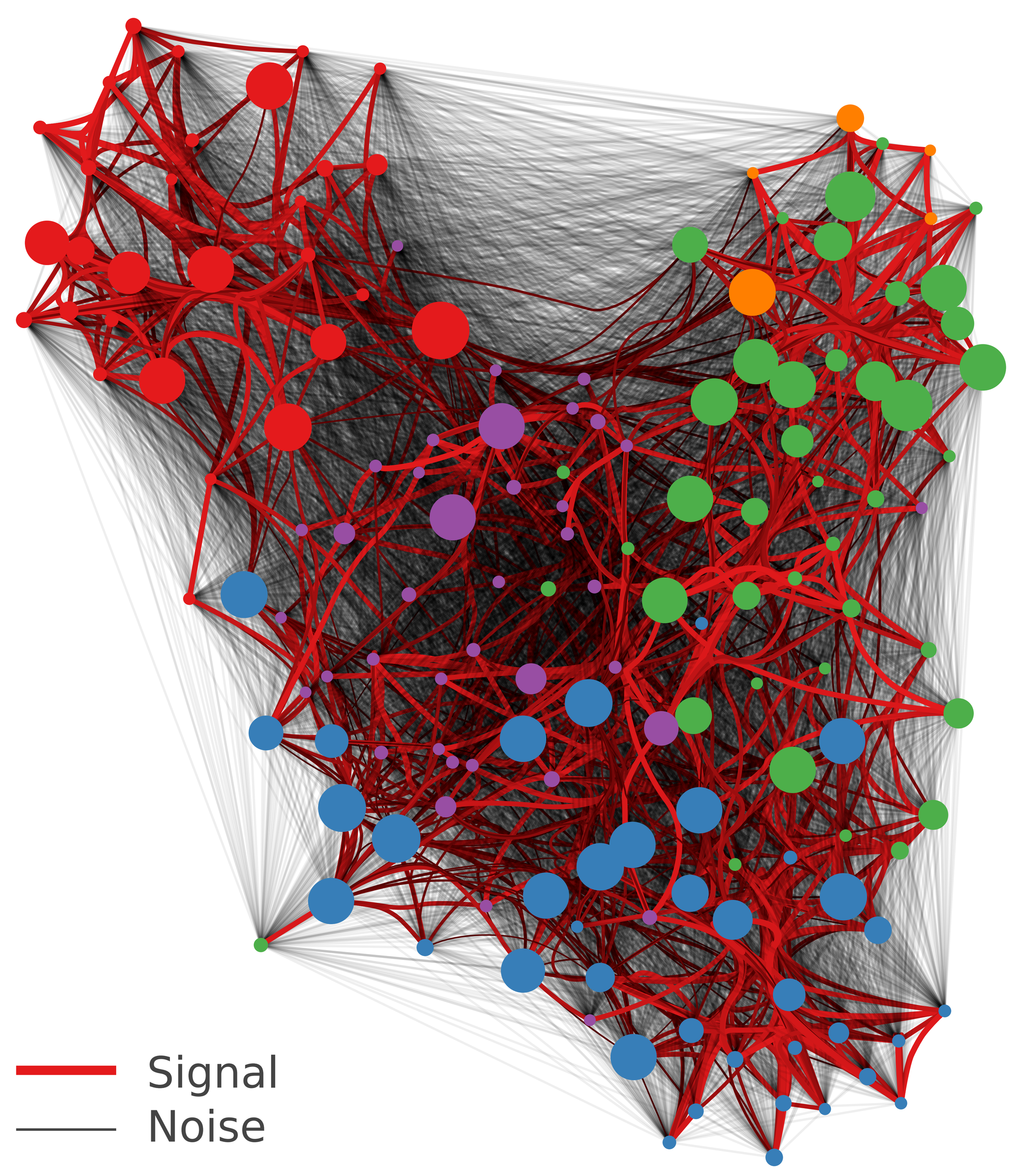
Network Backboning with Noisy Data

Michele Coscia and Frank Neffke

Center for international Development, Harvard University

michele_coscia@hks.harvard.edu frank_neffke@hks.harvard.edu

Motivation



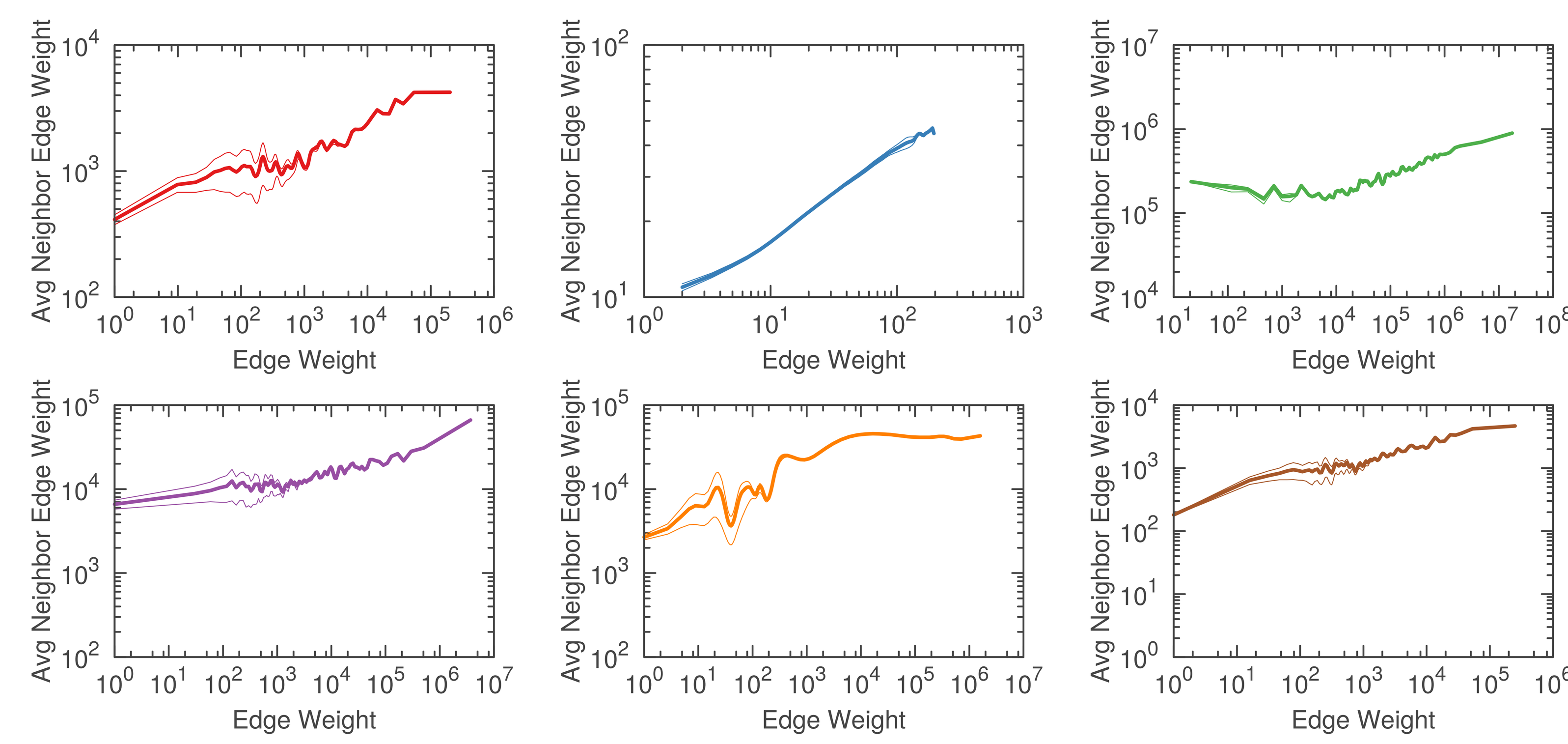
Network Backboning:

The detection of the largest possible subset of nodes and the smallest possible subset of salient connections of a network.

Current data gathering allows for very granular output.

When everything connects with everything else, how do we distinguish between true connections and noise-driven ones?

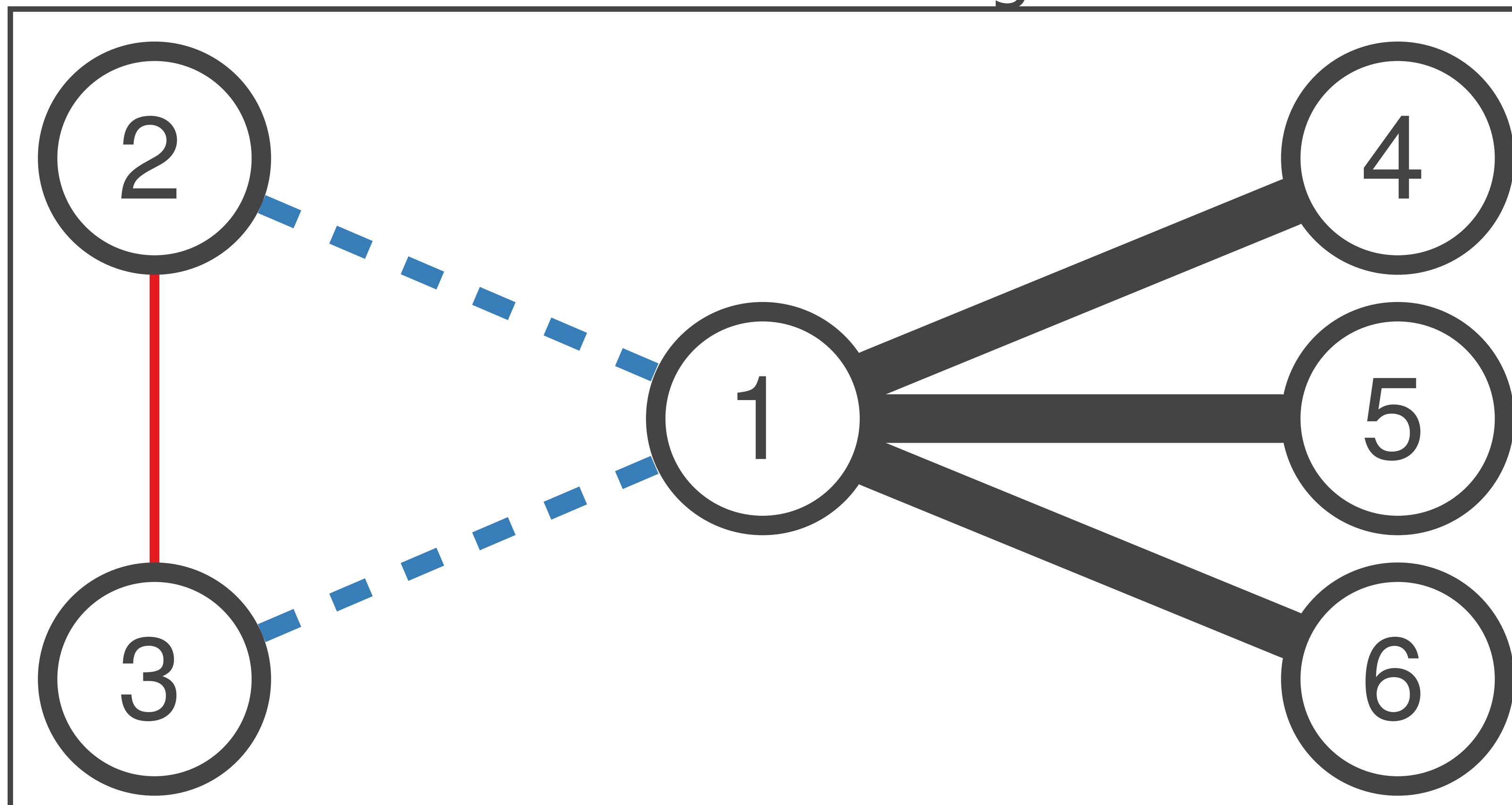
Empirical Issues



In many real world networks, edge weights are broadly distributed, and they lack a meaningful characteristic.

Moreover, they are nontrivially coupled with topology: an edge weight is correlated with the weights of the edges sharing a node with it. If we set an arbitrary threshold, we prune from the network weakly connected, but significant, portions of it.

Noise-Corrected Backboning



State of the art method checks if an edge is significant for a node's average weight. Blue edges are stronger than red, so they are kept.

Noise-Corrected backboning sees an edge as a **collaboration** between **both** nodes. The hub usually connects with stronger weights, so the blue edges are unimportant, and the red one is kept.

Implementation

We transform the count edge weights to control for unexpectedness.

$$E[N_{ij}] = \hat{N}_{i.} \frac{\hat{N}_{.j}}{\hat{N}_{..}} \qquad L_{ij} = \frac{\hat{N}_{ij}}{E[N_{ij}]}$$

All terms depend on N_{ij}, estimating its variance with the delta method gives us the variance of the correction.

$$V\left[\tilde{L}_{ij}\right] = V[\hat{N}_{ij}] \left(\frac{2\left(\kappa + \hat{N}_{ij}\frac{d\kappa}{d\hat{N}_{ij}}\right)}{\left(\kappa \hat{N}_{ij} + 1\right)^2} \right)^2$$

We cannot reliably estimate the variance when evidence on nodes is sparse.

$$V\left[N_{ij}\right] = N_{..} P_{ij} \left(1 - P_{ij}\right) \qquad \hat{P}_{ij} = \frac{\hat{N}_{ij}}{\hat{N}_{..}}$$

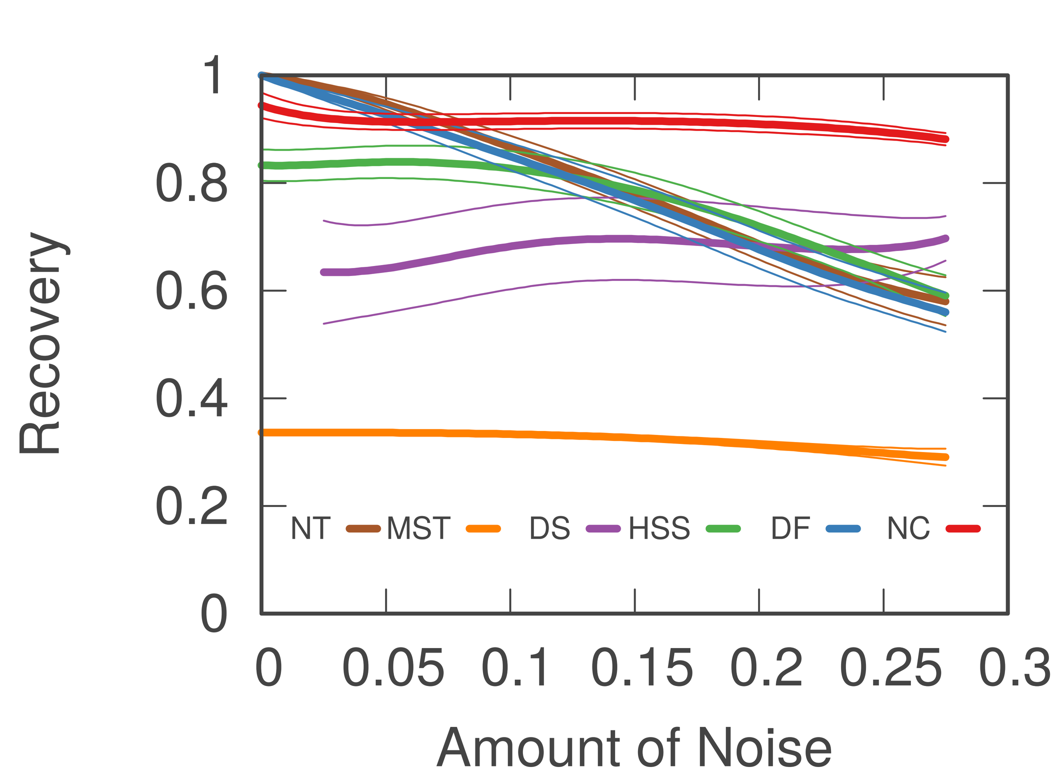
To account for this, we make a Bayesian correction. We define a threshold on the variance for different noise tolerances.

$$Pr\left[N_{ij} = n_{ij} \,|N_{..} = n_{..}, P_{ij} = p_{ij}\right] = \begin{pmatrix} n_{..} \\ n_{ij} \end{pmatrix} p_{ij}^{n_{ij}} \left(1 - p_{ij}\right)^{n_{..} - n_{ij}}$$

$$E\left[P_{ij}\right] = E\left[\frac{N_{ij}}{N_{..}}\right] = \frac{1}{N_{..}} E\left[N_{ij}\right] := \frac{1}{N_{..}} \frac{N_{i.}N_{.j}}{N_{..}}$$

$$V\left[P_{ij}\right] = \frac{1}{N_{..}^2} V\left[N_{ij}\right] := \frac{1}{N_{..}^2} \frac{N_{i.}N_{.j}\left(N_{..} - N_{i.}\right)\left(N_{..} - N_{.j}\right)}{N_{..}^2\left(N_{..} - 1\right)}.$$

Experiments

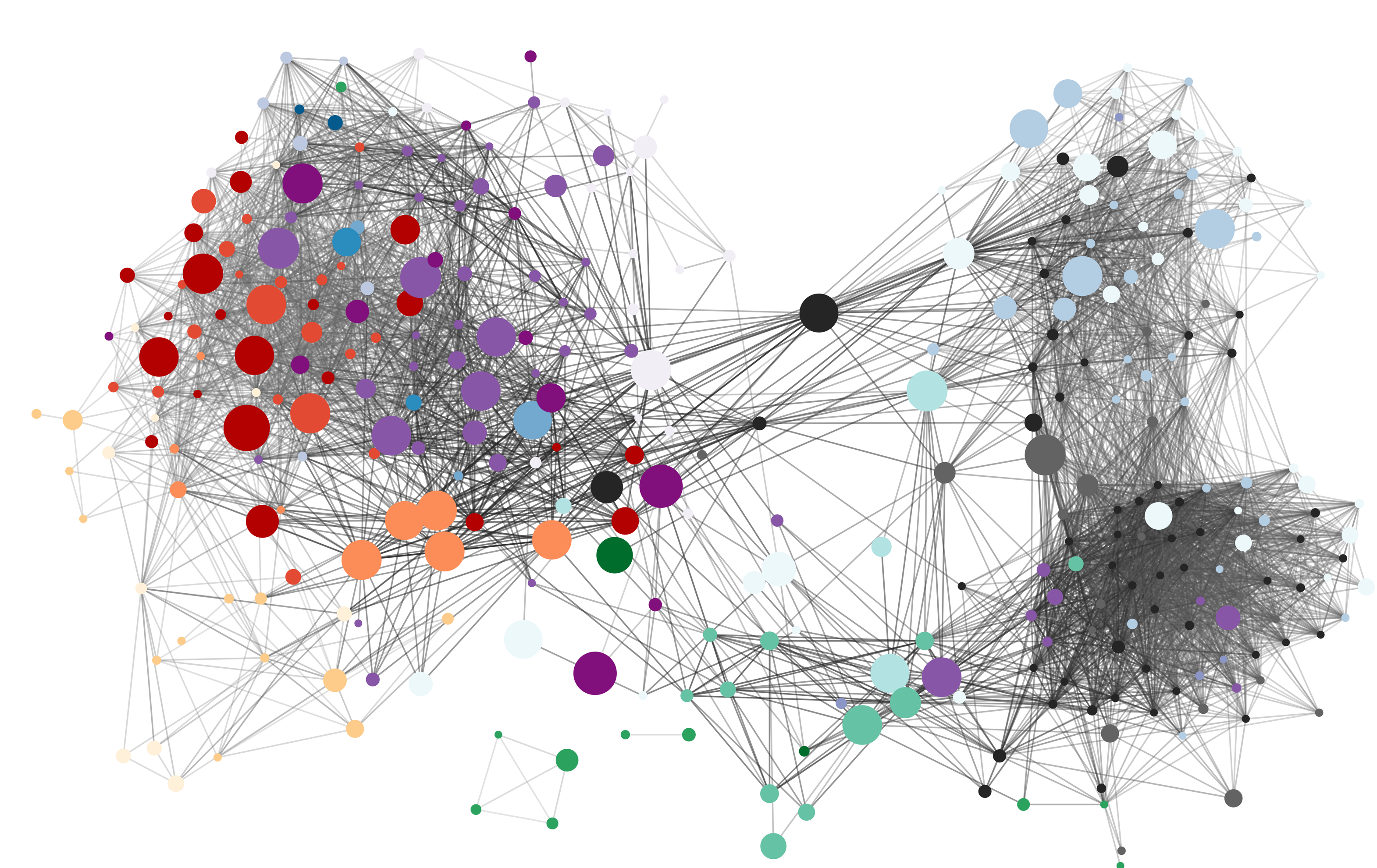


The NC backbone outperforms when the edges have significant noise (left, from synthetic networks).

In all real-world networks tested, NC provided the highest predictability boost (bottom).

Method	Business	Country Space	Flight	Migration	Ownership	Trade
Doubly Stochastic	n/a	2.0975	n/a	1.5153	n/a	0.9287
Naive Threshold	0.7766	0.6834	0.5196	1.1616	1.2384	0.3935
Disparity Filter	0.9315	1.4082	0.8569	2.0715	0.5374	0.9024
High Sal. Skelet.	1.1341	1.6549	0.9447	1.2597	0.9744	0.8662
Max. Spanning Tree	1.1183	1.9180	0.7981	1.0036	0.9288	0.9532
Noise-Corrected	**1.1767**	**2.2437**	**1.4676**	**2.1493**	**1.4165**	**1.1037**

Case Study: Predicting Job Switches



We create an occupation similarity network based on skills & tasks. When using it to predict job switchers we obtain r = .454, 30% higher gain than the current state of the art.