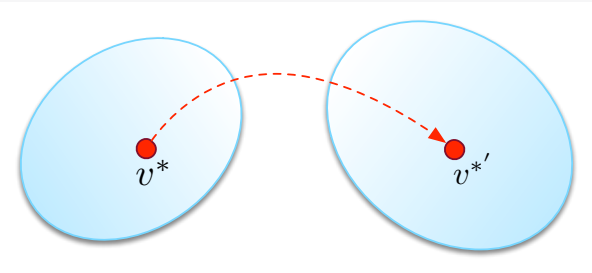
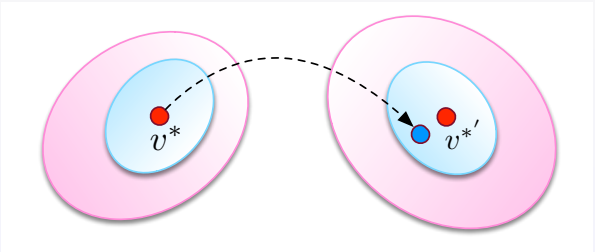
Vertex Nomination

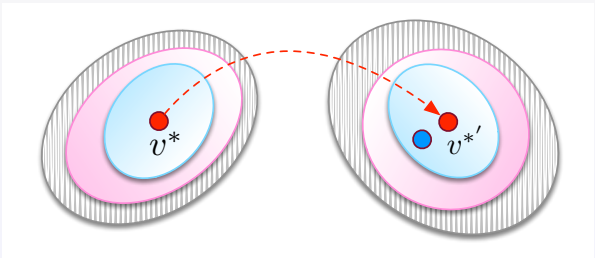
Carey E. Priebe
Johns Hopkins University

December 10, 2018
Seattle

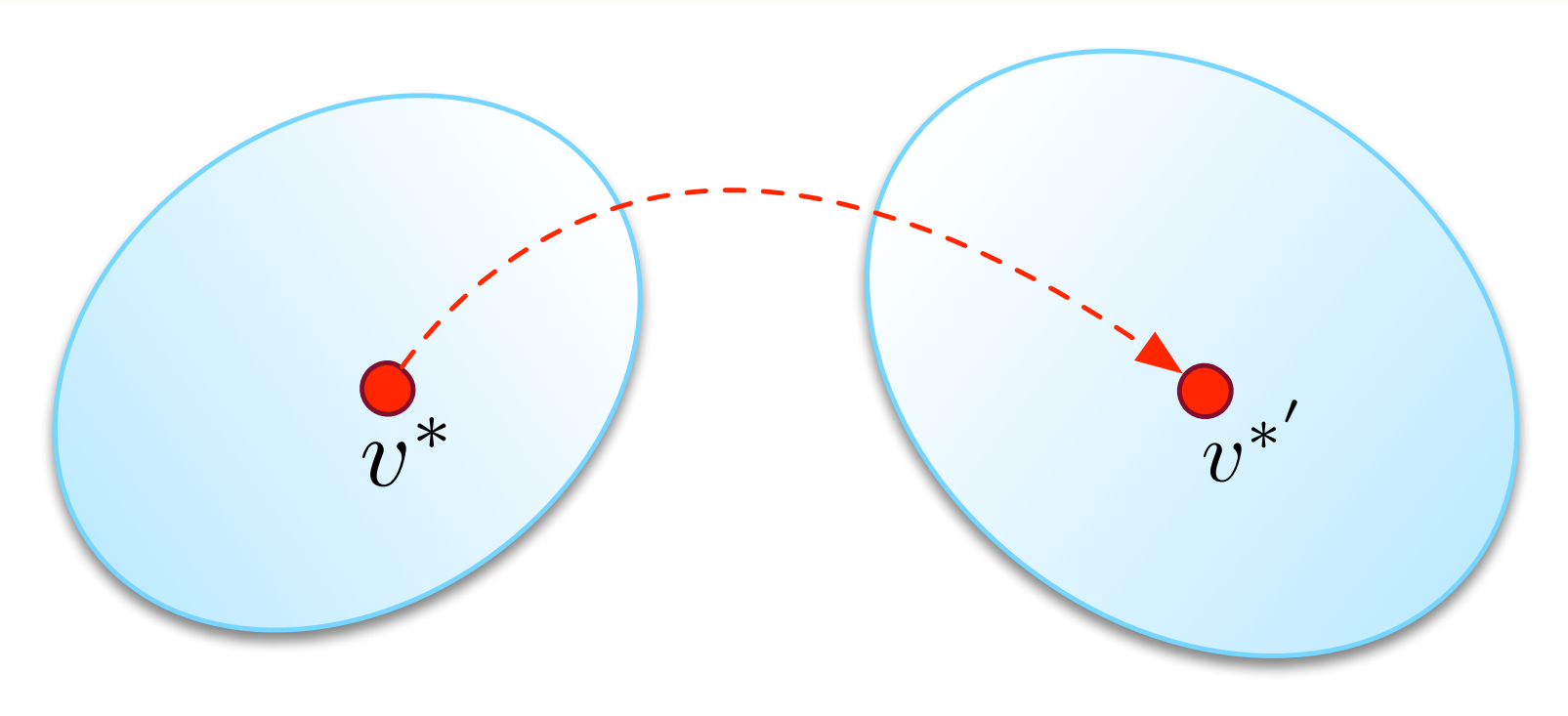
GTA^3 2.0:
The 2nd workshop on
Graph Techniques for Adversarial Activity Analytics



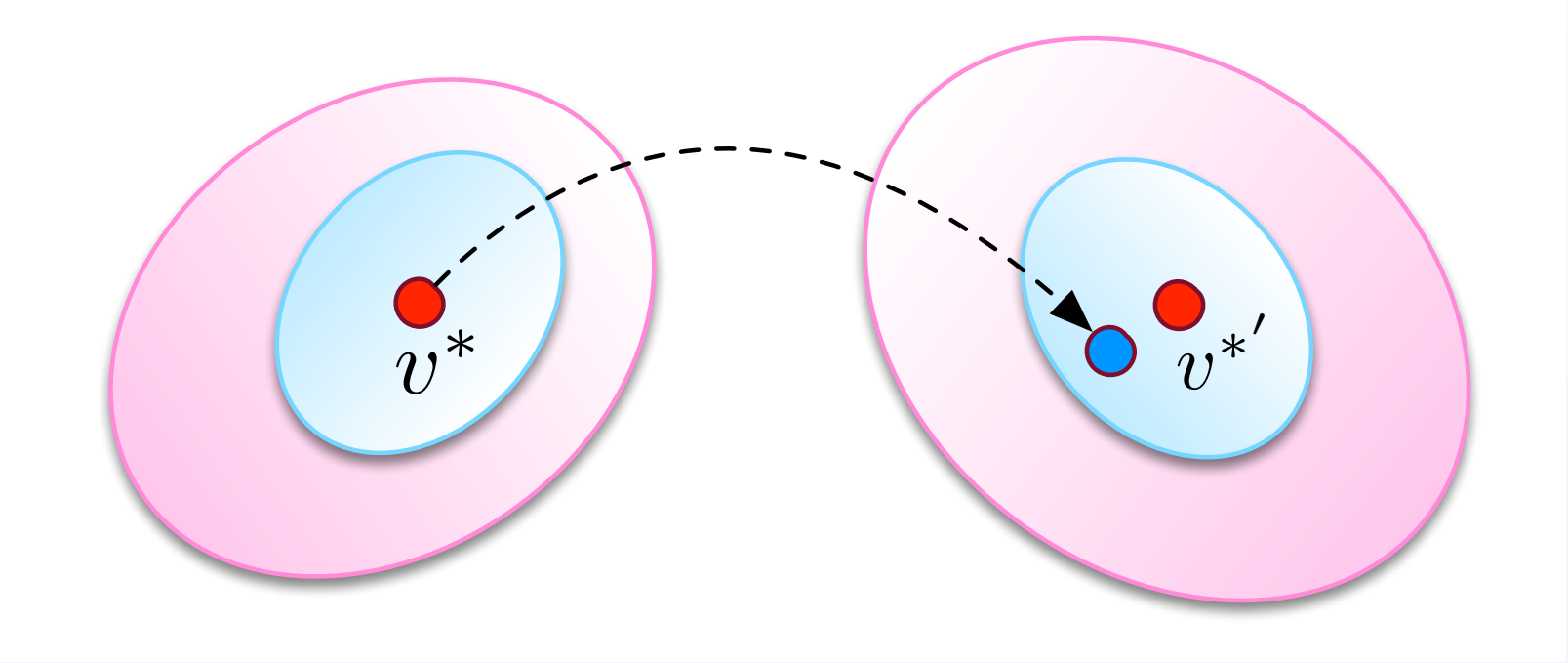




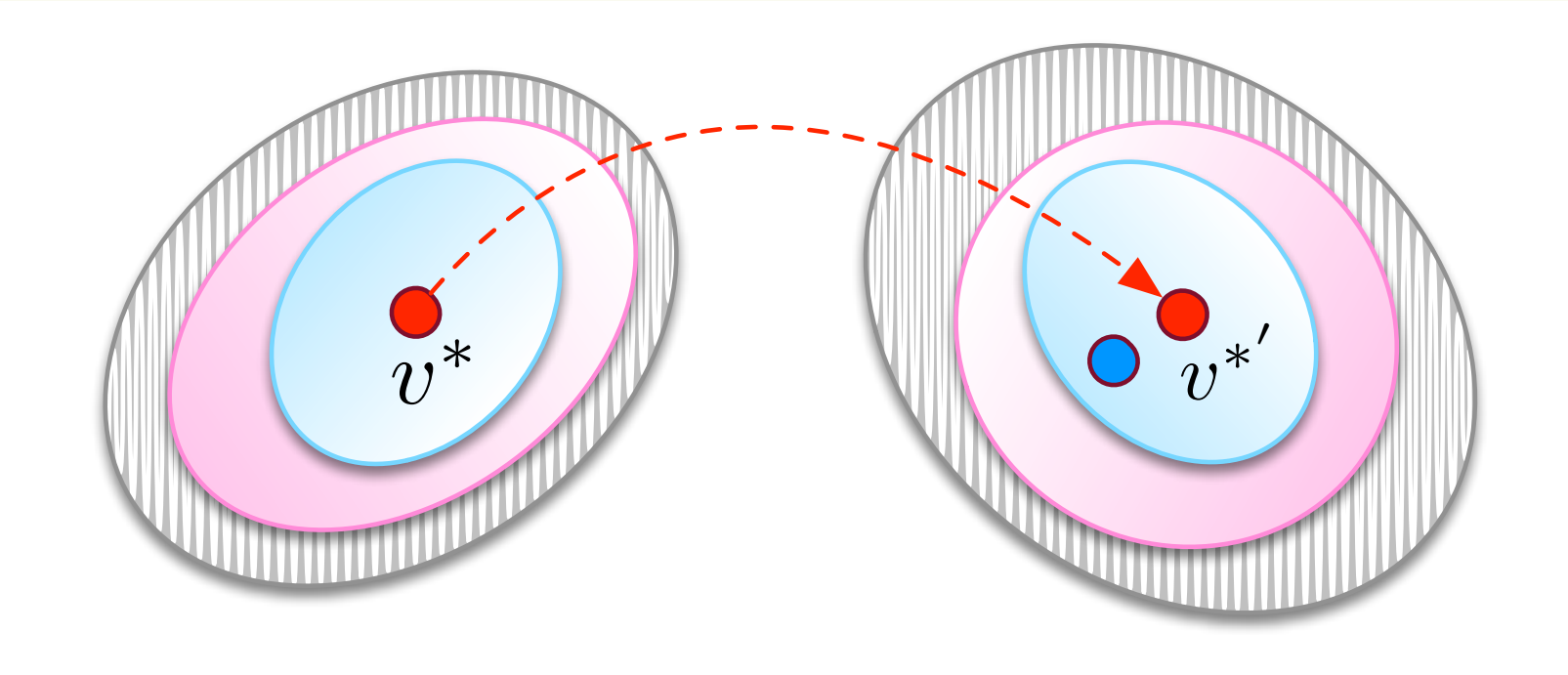
Vertex Nomination



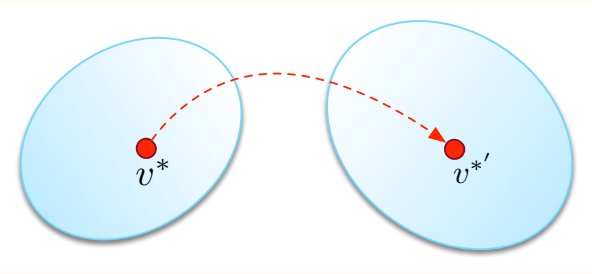
Vertex Nomination and an Adversary

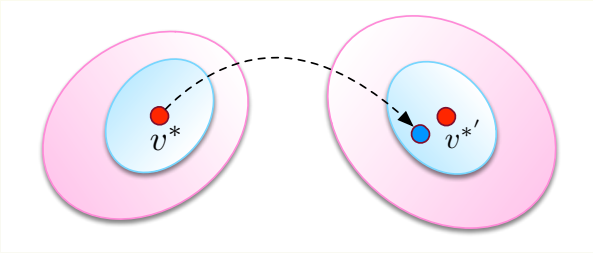


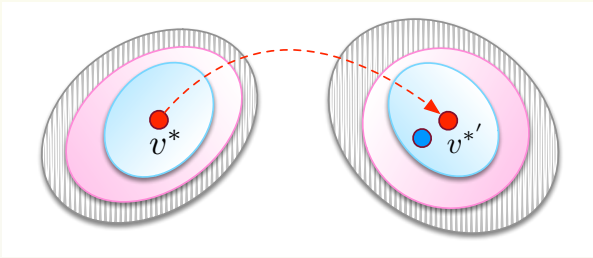
Vertex Nomination and an Adversary and Countering that Adversary



Artist's Conception

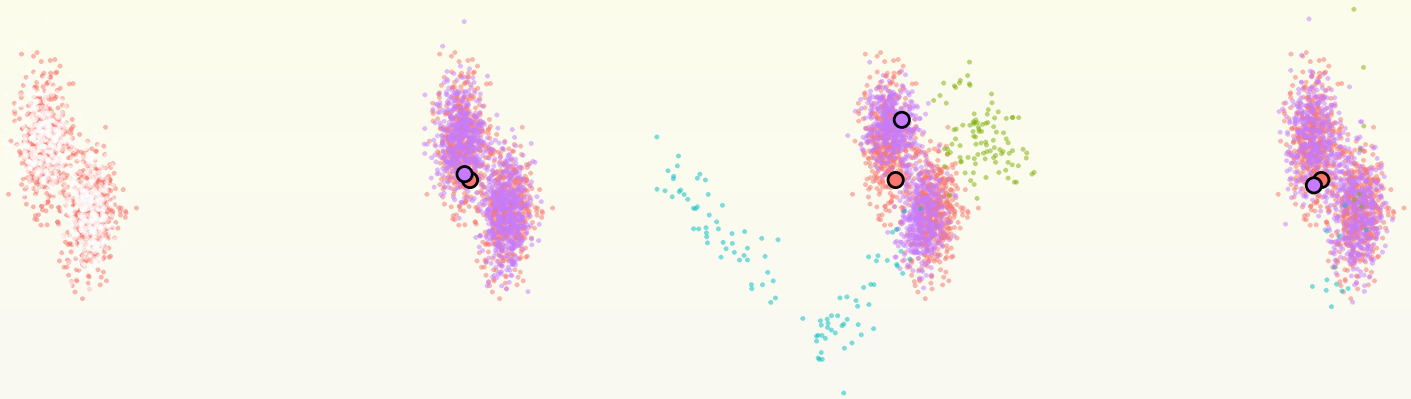




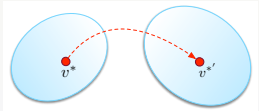


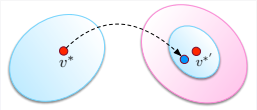
(a) G_1 & G_2: VN succeeds
(b) G_1^c & G_2^c: VN fails under contamination
(c) G_1^{cr} & G_2^{cr}: VN succeeds after regularization

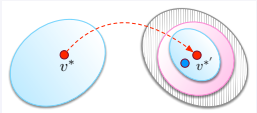
Illustrative Simulation Example



(a) G_1

(b) G_1 & G_2: VN succeeds 

(c) G_1 & G_2^c: VN fails under contamination 

(d) G_1 & G_2^{cr}: VN succeeds after regularization 

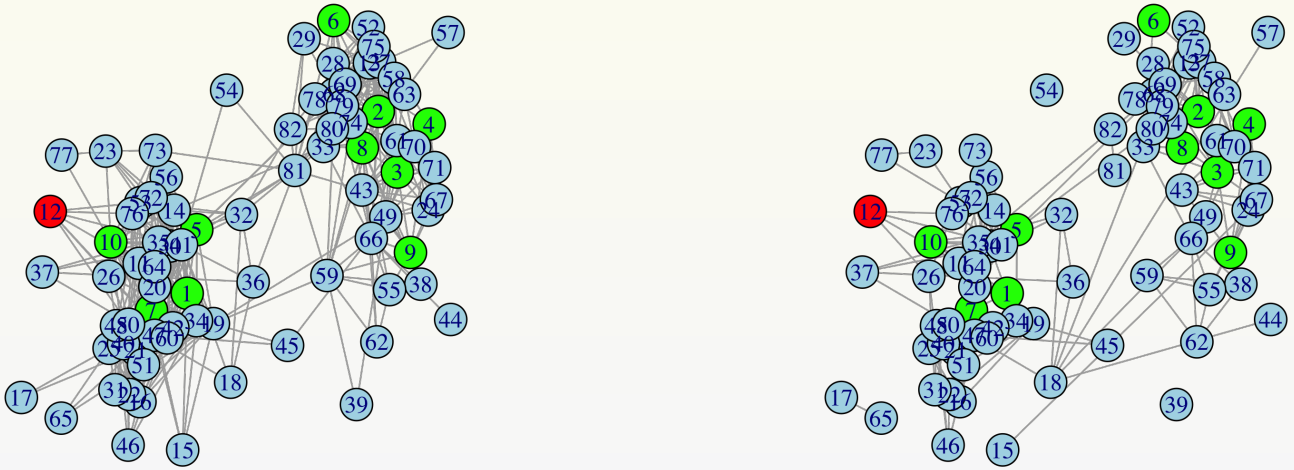
Illustrative Real Data Example: Facebook and Friendship Survey Networks

R Mastrandrea, J Fournet, and A Barrat,
Contact Patterns in a High School:
A Comparison between Data Collected Using
Wearable Sensors, Contact Diaries and Friendship Surveys,
PLoS ONE, 2015.

HG Patsolic, Y Park, V Lyzinski, CE Priebe,
Vertex Nomination Via Local Neighborhood Matching,
https://arxiv.org/abs/1705.00674

Illustrative Real Data Example

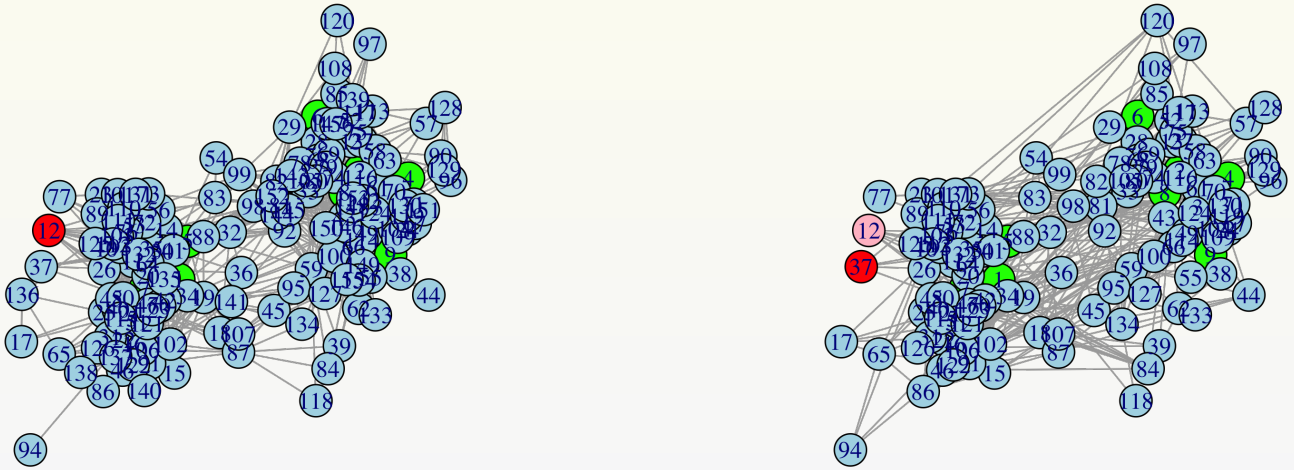
Core:



VN succeeds.

Illustrative Real Data Example

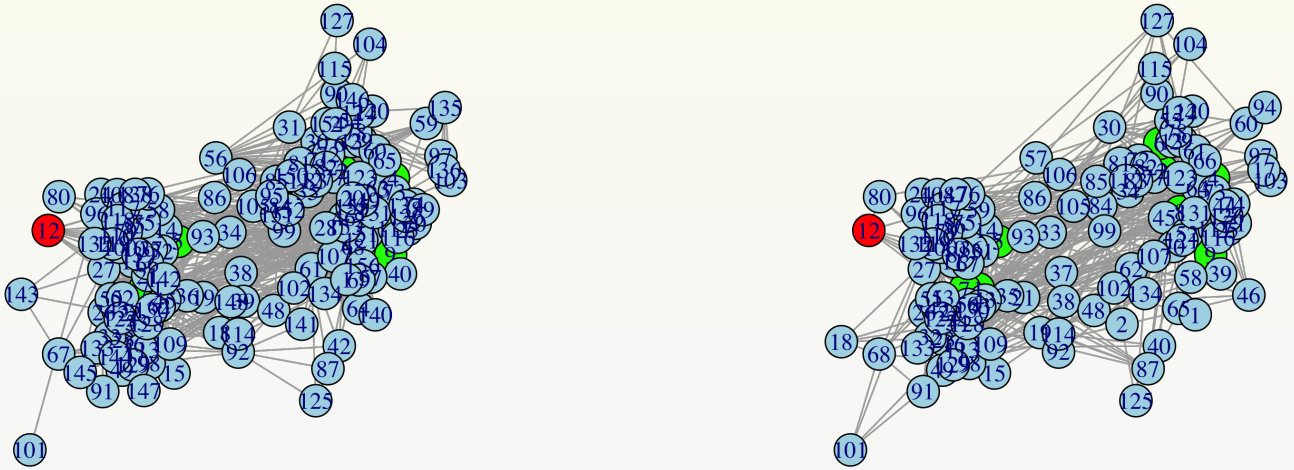
Full:



VN fails.

Illustrative Real Data Example

Regularized:



VN succeeds.

Model

Definition

Consider $X = [X_1, \ldots, X_n]^\top \in \mathbb{R}^{n \times d}$ satisfying $XX^\top \in [0,1]^{n \times n}$.
The bivariate graph valued random variables (G, G')
with respective adjacency matrices A and A'
are said to be distributed as a pair of ρ-correlated random dot product graphs with parameter X (abbreviated $(G, G') \sim \rho\text{-}RDPG(X)$) if

1. Marginally, $G, G' \sim RDPG(X)$, and
2. $\{A_{i,j}, A'_{k,l}\}_{\{i,j\},\{k,l\} \in \binom{V}{2}}$ are collectively independent except that for each $\{i,j\} \in \binom{V}{2}$,

$$\text{correlation}(A_{i,j}, A'_{i,j}) = \rho.$$

HG Patsolic, Y Park, V Lyzinski, CE Priebe,
Vertex Nomination Via Local Neighborhood Matching,
https://arxiv.org/abs/1705.00674

Model

The ρ-$RDPG$ model defines what we mean by "corresponding vertices," in which we generate two graphs G and G' from the same model in such a way that the nodes in G correspond to the nodes of G' via an identity mapping.

If we wish to also capture a relabeling of the vertices, we can apply a random permuation to the vertices of G'; however, this step, while more practical from a real-data perspective, is unnecessary for theoretical purposes.

Model

The ρ-$RDPG$ model defines what we mean by "corresponding vertices," in which we generate two graphs G and G' from the same model in such a way that the nodes in G correspond to the nodes of G' via an identity mapping.

If we wish to also capture a relabeling of the vertices, we can apply a random permuation to the vertices of G'; however, this step, while more practical from a real-data perspective, is unnecessary for theoretical purposes.

Model

Our framework posits $(H, H') \sim \rho\text{-}RDPG(X)$ for a latent position matrix $X \in \mathbb{R}^{(1+s+n)\times d}$.
In order to generate the full graphs G and G' which also have unshared vertices, we generate $G \sim RDPG([X, Y])$ and $G' \sim RDPG([X, Y'])$, so that the induced subgraphs $(H, H') \sim \rho\text{-}RDPG(X)$ and the remaining edges of G and G' are formed independently as in the case for the general RDPG. Thus, the first $1+s+n$ vertices in the two graphs correspond to one another via the identity map and the remaining m and m' vertices of G and G', respectively, represent the unshared vertices. Here, $Y \in \mathbb{R}^{m\times d}$ and $Y' \in \mathbb{R}^{m'\times d}$ represent the respective latent positions for the unshared vertices in G and G'.

For ease of notation, we will write $(G, G') \sim \rho\text{-}RDPG(X, Y, Y')$, where (G, G') is realized as two graphs: G on $\eta = 1+s+n+m$ vertices $\{x\} \cup S \cup W \cup J$ and G' on $\eta' = 1+s+n+m'$ vertices $\{x'\} \cup S' \cup W' \cup J'$.

Model

Our framework posits $(H, H') \sim \rho\text{-}RDPG(X)$ for a latent position matrix $X \in \mathbb{R}^{(1+s+n)\times d}$.
In order to generate the full graphs G and G' which also have unshared vertices, we generate $G \sim RDPG([X, Y])$ and $G' \sim RDPG([X, Y'])$, so that the induced subgraphs $(H, H') \sim \rho\text{-}RDPG(X)$ and the remaining edges of G and G' are formed independently as in the case for the general RDPG.
Thus, the first $1+s+n$ vertices in the two graphs correspond to one another via the identity map and the remaining m and m' vertices of G and G', respectively, represent the unshared vertices.
Here, $Y \in \mathbb{R}^{m\times d}$ and $Y' \in \mathbb{R}^{m'\times d}$ represent the respective latent positions for the unshared vertices in G and G'.

For ease of notation, we will write $(G, G') \sim \rho\text{-}RDPG(X, Y, Y')$, *where* (G, G') *is realized as two graphs:* G *on* $\eta = 1+s+n+m$ *vertices* $\{x\} \cup S \cup W \cup J$ *and* G' *on* $\eta' = 1+s+n+m'$ *vertices* $\{x'\} \cup S' \cup W' \cup J'$.

On consistent vertex nomination schemes

https://arxiv.org/abs/1711.05610

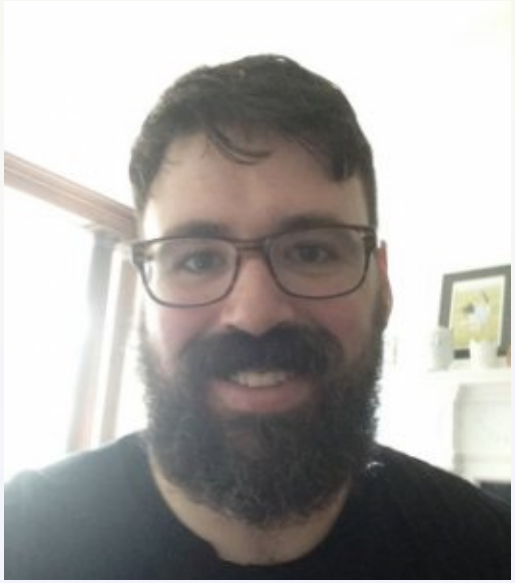
(*Journal of Machine Learning Research*)

Vince Lyzinski[†] Keith Levin[‡] Carey E. Priebe[*]

[†]Department of Math & Statistics, University of Massachusetts Amherst

[‡]Department of Statistics, University of Michigan

[*]Department of Applied Math & Statistics, Johns Hopkins University



Vertex Nomination: a general formulation

given a collection of vertices of interest,
find more with similar structural & functional role

Vertex Nomination: a general formulation

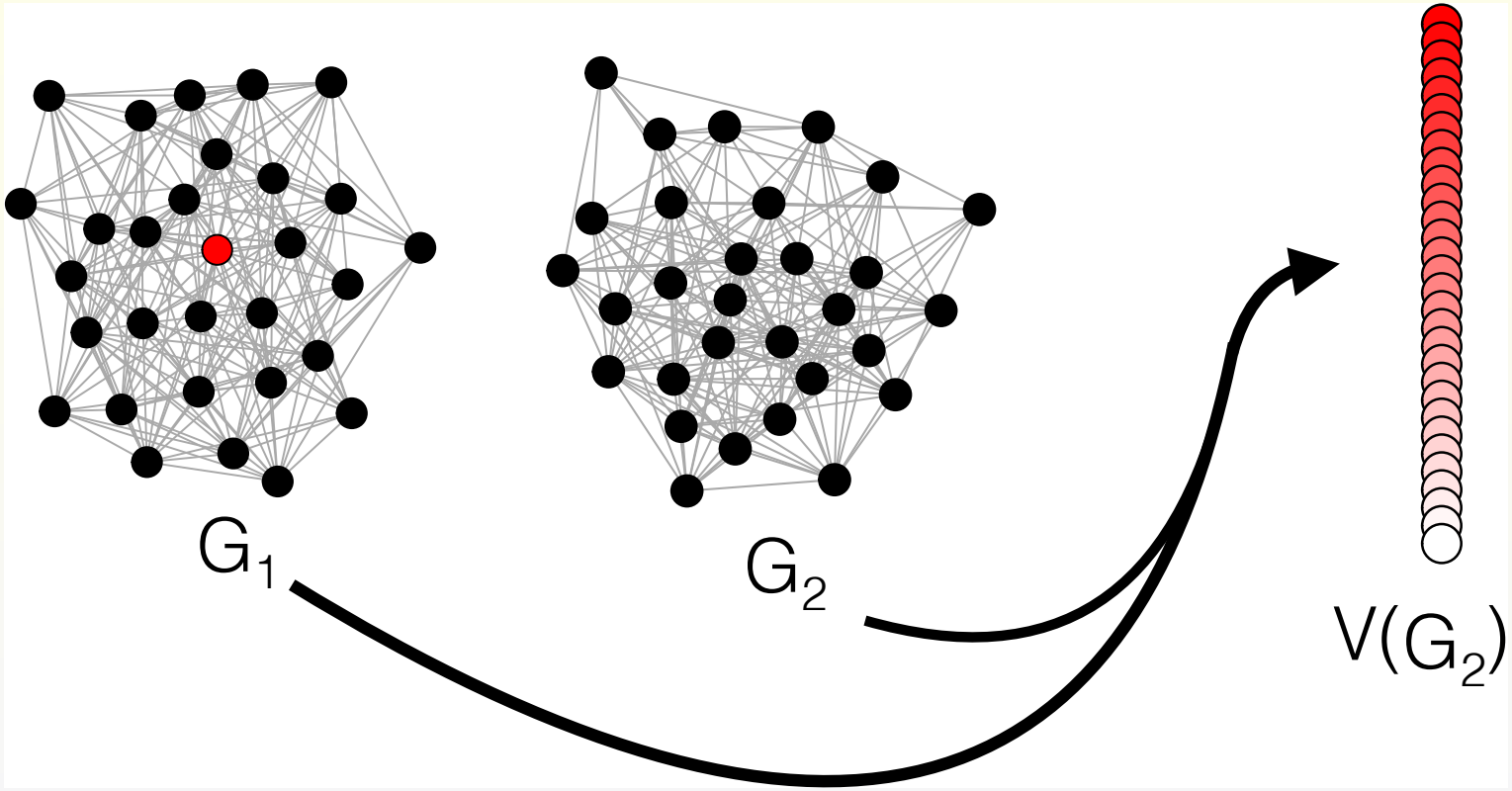
given a collection of vertices of interest,
find more with similar structural & functional role

Vertex Nomination: a general formulation

given a collection of vertices of interest,
find more with similar structural & functional role

one graph: within graph
multiple graphs: between graphs

Vertex Nomination



A visual representation of the VN framework:
Given vertex of interest v^* in graph $G_1 = (V_1, E_1)$,
find the corresponding $v^{*'}$ ($\exists$?) in graph $G_2 = (V_2, E_2)$,
ranking the vertices of G_2
so that $v^{*'}$ appears near the top of the nomination list.

Given a vertex of interest in a network G_1, the vertex nomination problem seeks to find the corresponding vertex of interest (if it exists) in a second network G_2.

Although the vertex nomination problem and related tasks have attracted much attention in the machine learning literature, with applications to social and biological networks, the framework has so far been confined to a comparatively small class of network models, and the concept of statistically consistent vertex nomination schemes has been only shallowly explored.

We extend the vertex nomination problem to a very general statistical model of graphs, and, drawing inspiration from the long-established classification framework in the pattern recognition literature, we provide a rigorous theoretical framework for defining the key notions of Bayes optimality and consistency in this expanded framework of vertex nomination, including a derivation of the Bayes optimal vertex nomination scheme.

We prove that no universally consistent vertex nomination scheme exists.

Given a vertex of interest in a network G_1, the vertex nomination problem seeks to find the corresponding vertex of interest (if it exists) in a second network G_2.

Although the vertex nomination problem and related tasks have attracted much attention in the machine learning literature, with applications to social and biological networks, the framework has so far been confined to a comparatively small class of network models, and the concept of statistically consistent vertex nomination schemes has been only shallowly explored.

We extend the vertex nomination problem to a very general statistical model of graphs, and, drawing inspiration from the long-established classification framework in the pattern recognition literature, we provide a rigorous theoretical framework for defining the key notions of Bayes optimality and consistency in this expanded framework of vertex nomination, including a derivation of the Bayes optimal vertex nomination scheme.

We prove that no universally consistent vertex nomination scheme exists.

Given a vertex of interest in a network G_1, the vertex nomination problem seeks to find the corresponding vertex of interest (if it exists) in a second network G_2.

Although the vertex nomination problem and related tasks have attracted much attention in the machine learning literature, with applications to social and biological networks, the framework has so far been confined to a comparatively small class of network models, and the concept of statistically consistent vertex nomination schemes has been only shallowly explored.

We extend the vertex nomination problem to a very general statistical model of graphs, and, drawing inspiration from the long-established classification framework in the pattern recognition literature, we provide a rigorous theoretical framework for defining the key notions of Bayes optimality and consistency in this expanded framework of vertex nomination, including a derivation of the Bayes optimal vertex nomination scheme.

We prove that no universally consistent vertex nomination scheme exists.

Given a vertex of interest in a network G_1, the vertex nomination problem seeks to find the corresponding vertex of interest (if it exists) in a second network G_2.

Although the vertex nomination problem and related tasks have attracted much attention in the machine learning literature, with applications to social and biological networks, the framework has so far been confined to a comparatively small class of network models, and the concept of statistically consistent vertex nomination schemes has been only shallowly explored.

We extend the vertex nomination problem to a very general statistical model of graphs, and, drawing inspiration from the long-established classification framework in the pattern recognition literature, we provide a rigorous theoretical framework for defining the key notions of Bayes optimality and consistency in this expanded framework of vertex nomination, including a derivation of the Bayes optimal vertex nomination scheme.

We prove that no universally consistent vertex nomination scheme exists.

Given a vertex of interest in a network G_1, the vertex nomination problem seeks to find the corresponding vertex of interest (if it exists) in a second network G_2.

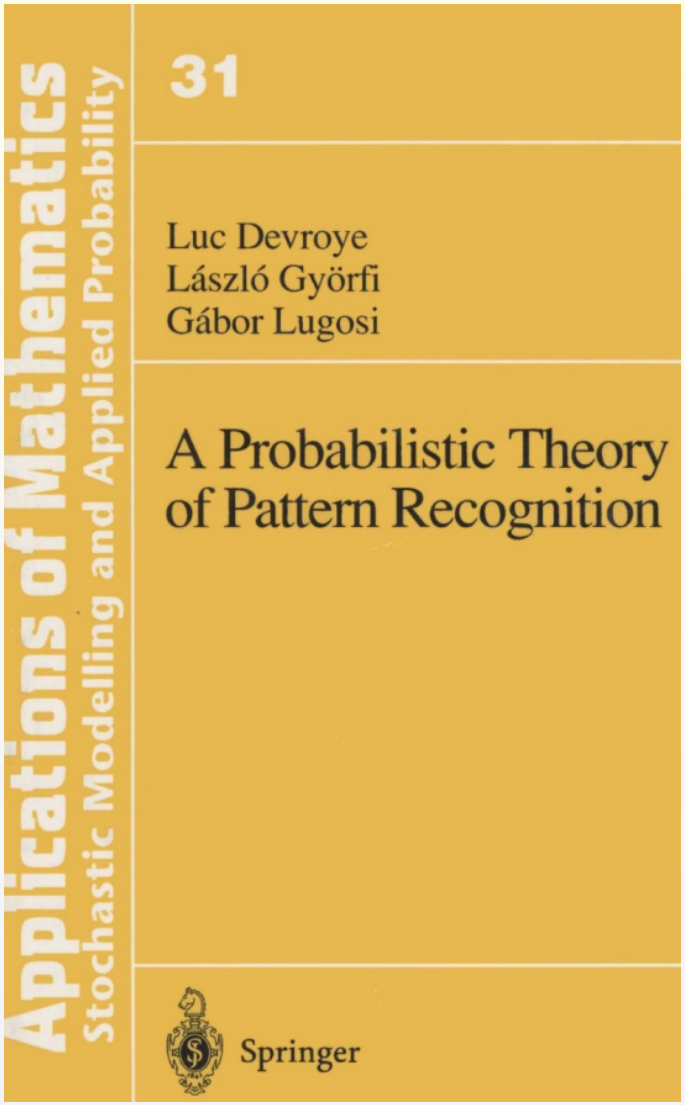
Although the vertex nomination problem and related tasks have attracted much attention in the machine learning literature, with applications to social and biological networks, the framework has so far been confined to a comparatively small class of network models, and the concept of statistically consistent vertex nomination schemes has been only shallowly explored.

We extend the vertex nomination problem to a very general statistical model of graphs, and, drawing inspiration from the long-established classification framework in the pattern recognition literature, we provide a rigorous theoretical framework for defining the key notions of Bayes optimality and consistency in this expanded framework of vertex nomination, including a derivation of the Bayes optimal vertex nomination scheme.

We prove that no universally consistent vertex nomination scheme exists.

On consistent vertex nomination schemes

no universally consistent VN scheme exists?



On consistent vertex nomination schemes

no universally consistent VN scheme exists?

(a) "Statistical inference on graphs is an important branch of modern statistics and machine learning."
(b) "the vertex nomination (VN) inference task" is a foundational inference task for graphs.
(c) L*, u.c., etc provide *the* "firm theoretical context in which to frame algorithmic progress".

$\Longrightarrow$

this is among the most interesting foundational contributions to "A Probabilistic Theory of Pattern Recognition" in many years, and among the most interesting foundational contributions to "A Probabilistic Theory of Pattern Recognition" for graphs ... ever.

no universally consistent VN scheme exists?

The classical classification setting:

- $(X, Y), (X_1, Y_1), \cdots, (X_n, Y_n) \sim^{iid} F$
- training data $\mathcal{D}_n = \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$
- to-be-classified observation X with unobserved class label Y
- $L_n(\hat{g}) = P[\hat{g}(X; \mathcal{D}_n) \neq Y]$
- $L^* = \inf_g P[g(X) \neq Y]$

no universally consistent VN scheme exists?

A general theorem by Stone (1977) allows us to deduce universal consistency of several classification rules.

In particular:

the k-nearest neighbors classifier
with $k(n) \to \infty$ and $k(n)/n \to 0$.

Charles J. Stone, Consistent Nonparametric Regression,
The Annals of Statistics, vol. 5, no. 4, pp. 595-620, 1977.

Graph Matching: Relax at Your Own Risk

Graph matching – aligning a pair of graphs to minimize their edge disagreements – has received wide-spread attention from both theoretical and applied communities over the past several decades, including combinatorics, computer vision, and connectomics. Its attention can be partially attributed to its computational difficulty. Although many heuristics have previously been proposed in the literature to approximately solve graph matching, very few have any theoretical support for their performance. *A common technique is to relax the discrete problem to a continuous problem, therefore enabling practitioners to bring gradient-descent-type algorithms to bear.* **We prove that an indefinite relaxation (when solved exactly) almost always discovers the optimal permutation, while a common convex relaxation almost always fails to discover the optimal permutation.**

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 60-73, 2016.

Graph Matching: Relax at Your Own Risk

Theorem 1: Suppose A and B are adjacency matrices for ρ-correlated Bernoulli(Λ) graphs, and there is an $\alpha \in (0, 1/2)$ such that $\Lambda_{i,j} \in [\alpha, 1-\alpha]$ for all $i \neq j$. Let $P^* \in \Pi$, and denote $A' := P^* A P^{*T}$.

a) If $(1-\alpha)(1-\rho) < 1/2$, then it almost always holds that

$$\arg\min_{D \in \mathcal{D}} -\langle A'D, DB \rangle = \arg\min_{P \in \Pi} \|A' - PBP^T\|_F = \{P^*\}.$$

b) If the between graph correlation $\rho < 1$, then it almost always holds that $P^* \notin \arg\min_{D \in \mathcal{D}} \|A'D - DB\|_F$.

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 60-73, 2016.

On consistent vertex nomination schemes

Definition: Bayes error of a VN scheme

Let $(G_1, G_2) \sim F_{c,n,m,\theta}$ with vertex of interest $v^* \in C$, and let $\mathfrak{o} : V_2 \to W$ be an obfuscating function. For a VN scheme $\Phi \in \mathcal{V}_{n,m}$, we define the *level-k error* of Φ at v^* to be

$$L_k(\Phi, v^*) = \mathbb{P}_{(G_1,G_2)\sim F_{c,n,m,\theta}} \left[\mathrm{rank}_{\Phi(G_1, \mathfrak{o}(G_2), v^*)}(\mathfrak{o}(v^*)) \geqslant k+1 \right].$$

We define the *level-k Bayes optimal* VN scheme to be any element $\Psi \in \arg\min_{\Phi \in \mathcal{V}_{n,m}} L_k(\Phi, v^*)$, and define the *level-k Bayes error* to be $L_k^*(v^*) = L_k(\Psi, v^*)$ for Bayes optimal Ψ.

The level-1 Bayes error would correspond to the Bayes error if we viewed vertex nomination as a simple classification problem rather than a more complicated ranking problem. That is, $L_1(\Phi, v^*)$ is simply the probability that Φ fails to "classify" $\mathfrak{o}(v^*)$ as the vertex corresponding to v^* in $\mathfrak{o}(G_2)$.

On consistent vertex nomination schemes

Remark:

We note that the error defined above depends on n and m in a manner distinct from that defined in the classical classification framework. In the classical setting, $L(h_n)$ denotes the error rate of a classifier that classifies a single observation X based on n training instances $\{(X_i, Y_i)\}_{i=1}^n$. In the case of VN, the notion of labeled training instances is, at best, more hazy. Indeed, in the present setting, the training data and test data are inseparable—the graphs (more specifically, their edges) *are* the training data.

On consistent vertex nomination schemes

Theorem:

Let $\Phi^* = \ldots.$
Then $L_k(\Phi^*, v^*) = L_k^*(v^*).$

On consistent vertex nomination schemes

Definition: Consistency of a VN scheme

Let $\mathbf{F} = \left(F_{c(n),n,m(n),\theta(n)}\right)_{n=n_0}^{\infty}$ be a sequence of nominatable distributions in $\mathcal{N}$ with nested cores satisfying $\lim_{n\to\infty} m(n) = \infty$. For a given non-decreasing sequence $\{k_n\}$, we say that a VN rule $\boldsymbol{\Phi} = (\Phi_{n,m(n)})_{n=n_0}^{\infty}$ is *level-*$\{k_n\}$ *consistent* at v^* with respect to $\mathbf{F}$ if

$$\lim_{n\to\infty} L_{k_n}(\Phi_{n,m(n)}, v^*) - L_{k_n}^*(v^*) = 0,$$

for any sequence of obfuscating functions of V_2 with $|V_2| = m(n)$. If a scheme Φ is level-$\{k_n\}$ consistent at v^* for a constant sequence $k_n = k$, $n = 1, 2, \ldots$, then we say simply that Φ is *level-k consistent*.

On consistent vertex nomination schemes

In the VN problem, the complexity of the model generating the data can also grow in n, which effectively thwarts the ability of a VN rule to asymptotically overcome a sequence of adversarial graph models.

On consistent vertex nomination schemes

Theorem:

Let $\epsilon \in (0,1)$ be arbitrary, and consider a VN rule $\boldsymbol{\Phi} = (\Phi_{n,m})$.
For any nondecreasing sequence (k_n) satisfying $k_n = o(m)$,
there exists a sequence of distributions $(F_{c,n,m,\theta})$ in $\mathcal{N}$
with nested cores such that

$$\text{limsup}_{n\to\infty} L^*_{k_n}(v^*) = \epsilon < 1 = \lim_{n\to\infty} L_{k_n}(\Phi_{n,m}, v^*).$$

Unlike in the classification setting,

no universally consistent VN scheme exists!

But it gets worse

Definition

Let $\mathfrak{N}$ be the collection of all nested-core nominatable sequences. For a nondecreasing sequence (k_n), we say that $\mathfrak{C} \in \mathfrak{N}$ is a maximal (k_n)-consistency class if the following two conditions hold.

i. There exists a VN rule Φ that is jointly (k_n)-consistent for each $\mathbf{F} \in \mathfrak{C}$;

ii. If $\mathbf{F}' \notin \mathfrak{C}$, then there does not exist a VN rule Φ that is jointly (k_n)-consistent for each $\mathbf{F} \in \mathfrak{C} \cup \{\mathbf{F}'\}$.

A natural question to ask is whether it is possible to partition $\mathfrak{N}$ into a finite number of maximal (k_n)-consistency classes for a particular sequence $(k_n)_{n=1}^{\infty}$? If so, then the lack of universally consistent VN-schemes can be operationally mitigated via an ensemble nomination approach.

However . . .

But it gets worse: an infinite number of consistency classes ...

For any sequences (k_n), any partition of $\mathfrak{N}$ into maximal (k_n)-consistency classes must include countably infinite parts.

Theorem

Let (k_n) be a sequence of nondecreasing numbers satisfying $k_n = o(n)$. If $\mathfrak{N} = \cup_{\alpha \in \mathcal{A}} \mathfrak{C}_\alpha$ is a partition of $\mathfrak{N}$ into maximal (k_n)-consistency classes, then $|\mathcal{A}|$ is at least $\aleph_0$.

But it gets worse: an infinite number of consistency classes ...

For any sequences (k_n), any partition of $\mathfrak{N}$ into maximal (k_n)-consistency classes must include countably infinite parts.

Theorem

Let (k_n) be a sequence of nondecreasing numbers satisfying $k_n = o(n)$. If $\mathfrak{N} = \cup_{\alpha \in \mathcal{A}} \mathfrak{C}_\alpha$ is a partition of $\mathfrak{N}$ into maximal (k_n)-consistency classes, then $|\mathcal{A}|$ is at least $\aleph_0$.

And still worse . . .

One of the consequences of a lack of universal consistency is that, for any given VN rule and nondecreasing sequence (k_n), nominatable distributions exist outside of the (k_n)-consistency class

$$\mathfrak{C}_{\Phi}^{(k_n)} = \{\mathbf{F} \in \mathcal{N} \text{ s.t. } \Phi \text{ is } (k_n)\text{-consistent for } \mathbf{F}\}.$$

Given Φ and $\mathbf{F}$, is it possible to verify $\mathbf{F} \in \mathfrak{C}_{\Phi}^{(k_n)}$?

This is a central (theoretical) question in practice, as the first step in correcting for an inconsistent VN scheme is being able to detect if a VN scheme is, in fact, inconsistent.

Unfortunately, such a verification is impossible sans metadata/supervision.

Adversary / Contamination – an outlier model

Consider the one-sample symmetric *location* model $\mathcal{P}$ defined by

$$X_i = \mu + \varepsilon_i, \qquad i = 1, \ldots, n, \tag{3.5.1}$$

where the errors are independent, identically distributed, and symmetric about 0 with common density f and d.f. F. If the error distribution is normal, $\bar{X}$ is the best estimate in a variety of senses.

In our new formulation it is the X_i^* that obey (3.5.1). A reasonable formulation of a model in which the possibility of gross errors is acknowledged is to make the ε_i still i.i.d. but with common distribution function F and density f of the form

$$f(x) = (1-\lambda)\frac{1}{\sigma}\varphi\left(\frac{x}{\sigma}\right) + \lambda h(x). \tag{3.5.2}$$

Here h is the density of the gross errors and λ is the probability of making a gross error. This corresponds to,

$$\begin{aligned} X_i &= X_i^* \text{ with probability } 1-\lambda \\ &= Y_i \text{ with probability } \lambda \end{aligned}$$

where Y_i has density $h(y-\mu)$ and (X_i^*, Y_i) are i.i.d. Note that this implies the possibly unreasonable assumption that committing a gross error is independent of the value of X^*.

Bickel & Doksum

Adversary / Contamination

To model adversarial attacks in the VN-framework, we introduce the concept of an *adversary*.

We say $\mathcal{A} = \{f_{\mathcal{A}}, V_{\mathcal{A}}\}$ is an *adversary* if

1. $f_{\mathcal{A}} : \mathcal{G}_m \mapsto \mathcal{G}_m$ such that $V(f_{\mathcal{A}}(G)) = V(G)$,
2. $V_{\mathcal{A}} \subset V(G)$,
3. If

$$K = \left\{ v, w \in V(G) \text{ s.t. } (v, w) \in E(f_{\mathcal{A}}(G)) \setminus E(G) \right\} \cup \left\{ v, w \in V(G) \text{ s.t. } (v, w) \in E(G) \setminus E(f_{\mathcal{A}}(G)) \right\},$$

then $K \subset V_{\mathcal{A}}$.

We say that $\mathcal{A}$ is an *edge-augmenting adversary* if $E(G) \subset E(f_{\mathcal{A}}(G))$, and similarly for an *edge-removing adversary*.

In other words, $f_{\mathcal{A}}$ is just a function that operates only on the edges incident to the vertices of $V_{\mathcal{A}}$.

We refer to $V_{\mathcal{A}}$ as the vertices *controlled* by $\mathcal{A}$.

Example Adversarial Model

Consider a stochastic blockmodel on $\mathcal{G}_n$, with two blocks, B_1 and B_2 with $n/2$ vertices in each block. The edge-probability matrix B is given by

$$B = \begin{pmatrix} p & r \\ r & q \end{pmatrix},$$

with $p \geqslant q \geqslant r > 0$.

Consider the following model-contamination procedure:

1. First, c_+ vertices are selected at random from V (call them W_+), and c_- vertices are selected at random from $V \setminus W_+$.
2. For each vertex $v \in W_+$, and each vertex $u \in V \setminus W_-$, if there is not an edge, an edge is then created with probability s_+.
3. For each vertex $v \in W_-$, and each vertex $u \in V \setminus W_+$, if there is an edge, it is deleted with probability s_-.

Notice that this gives rise to a new stochastic block model with the edge-probability matrix $\tilde{B}$ given by

Example Adversarial Model

$$\tilde{B} = \begin{array}{c} \\ \tilde{B}_1 \\ \tilde{B}_1^+ \\ \tilde{B}_1^- \\ \tilde{B}_2 \\ \tilde{B}_2^+ \\ \tilde{B}_2^- \end{array} \begin{array}{c} \begin{array}{cccc} \tilde{B}_1 & \tilde{B}_1^+ & \tilde{B}_1^- & \tilde{B}_2 \end{array} \\ \left(\begin{array}{cccc} p & p+s_+(1-p) & p-s_-p & r \\ p+s_+(1-p) & \begin{array}{c} p+s_+(1-p) \\ +(1-p)(1-s_+)s_+ \end{array} & p & r+s_+(1-r) \\ p-s_-p & p & \begin{array}{c} p-s_-p \\ -p(1-s_-)(s_-) \end{array} & r-s_-(1-r) \\ r & r+s_+(1-r) & r-s_-r & q \\ r+s_+(1-r) & \begin{array}{c} r+s_+(1-r) \\ +(1-r)(1-s_+)s_+ \end{array} & r & q+s_+(1-q) \\ r-s_-r & r & \begin{array}{c} r-s_-r \\ -r(1-s_-)s_- \end{array} & q-s_-q \end{array} \right) \end{array}$$

where $\tilde{B}_1^+$ are the vertices in $W_+ \cap B_1$, $\tilde{B}_1^-$ are the vertices in $B_1 \cap W_-$, and $\tilde{B}_1$ are the vertices in $B_1 \setminus (\tilde{B}_1^+ \cup \tilde{B}_1^-)$, with $\tilde{B}_2$ defined analogously.

We will assume that our vertex of interest was unchanged in $\tilde{B}_1$.

Example Adversarial Model

The Annals of Statistics
2015, Vol. 43, No. 3, 1027–1059
DOI: 10.1214/14-AOS1290

ROBUST AND COMPUTATIONALLY FEASIBLE COMMUNITY DETECTION IN THE PRESENCE OF ARBITRARY OUTLIER NODES[1]

BY T. TONY CAI AND XIAODONG LI

University of Pennsylvania

Community detection, which aims to cluster N nodes in a given graph into r distinct groups based on the observed undirected edges, is an important problem in network data analysis. In this paper, the popular stochastic block model (SBM) is extended to the generalized stochastic block model (GSBM) that allows for adversarial outlier nodes, which are connected with the other nodes in the graph in an arbitrary way. Under this model, we introduce a procedure using convex optimization followed by k-means algorithm with $k = r$.

Adversarial Effect

Theorem

Suppose that Φ is a VN scheme that runs spectral clustering on the contaminated graph by first selecting the number of communities in a consistent manner and nominating the vertices in the group with the highest probability of within-group connection. Suppose that we are interested in rank $k_n = n/2$. If either

1. $\frac{p-q}{p} < s_-(2-s_-)$ *for all n, or*
2. $\frac{p-q}{1-q} < s_+(2-s_+)$ *for all n,*

then Φ is no longer consistent with respect to the contaminated model (provided $p \neq q$ for all n).

We have a VN scheme.
We have a distribution.
And that distribution is in our scheme's consistency class.
Now, after contamination,
the new distribution is not in our scheme's consistency class.

Adversarial Effect

Theorem

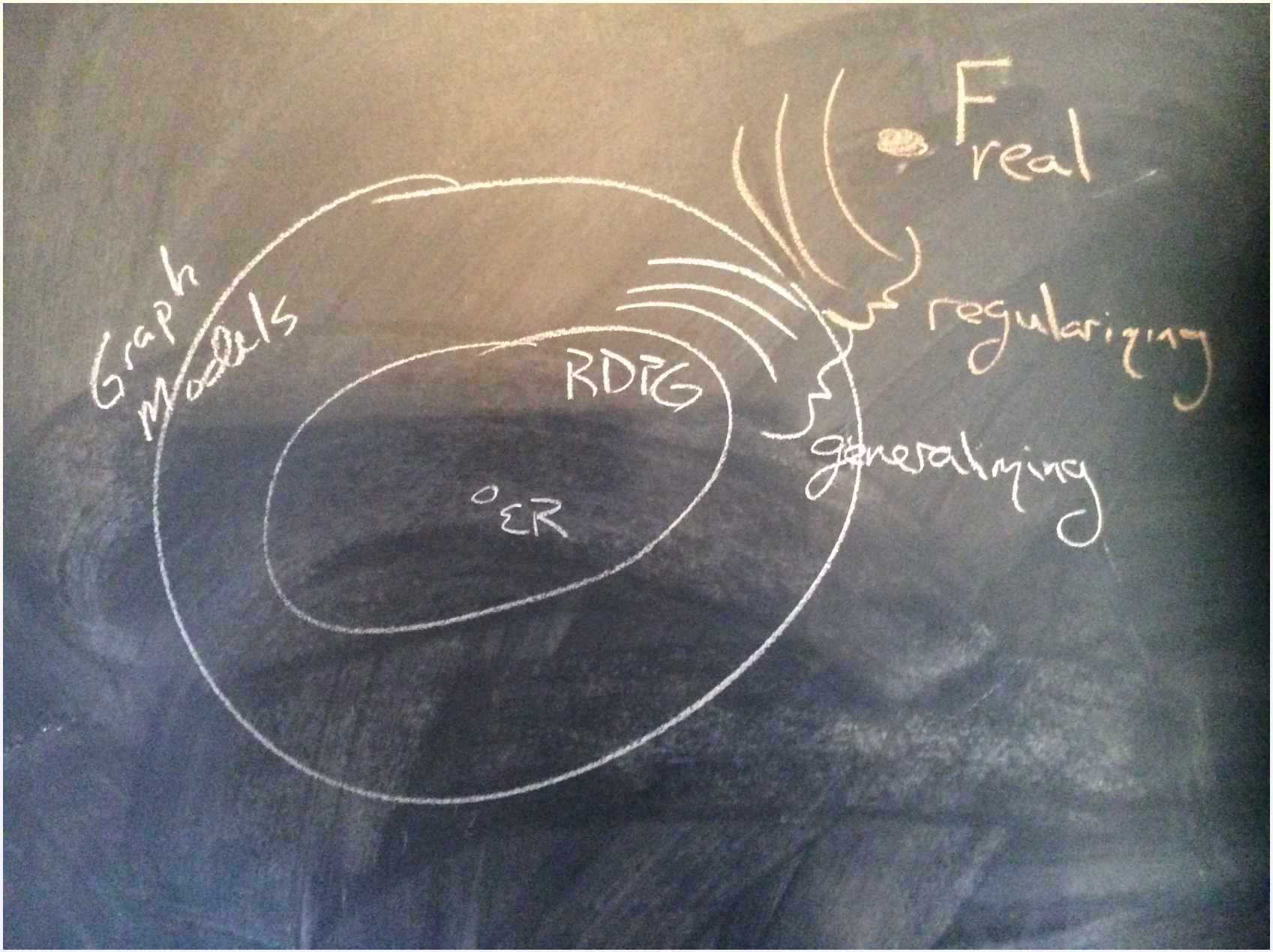
Suppose that Φ is a VN scheme that runs spectral clustering on the contaminated graph by first selecting the number of communities in a consistent manner and nominating the vertices in the group with the highest probability of within-group connection. Suppose that we are interested in rank $k_n = n/2$. If either

1. $\frac{p-q}{p} < s_-(2-s_-)$ *for all n, or*
2. $\frac{p-q}{1-q} < s_+(2-s_+)$ *for all n,*

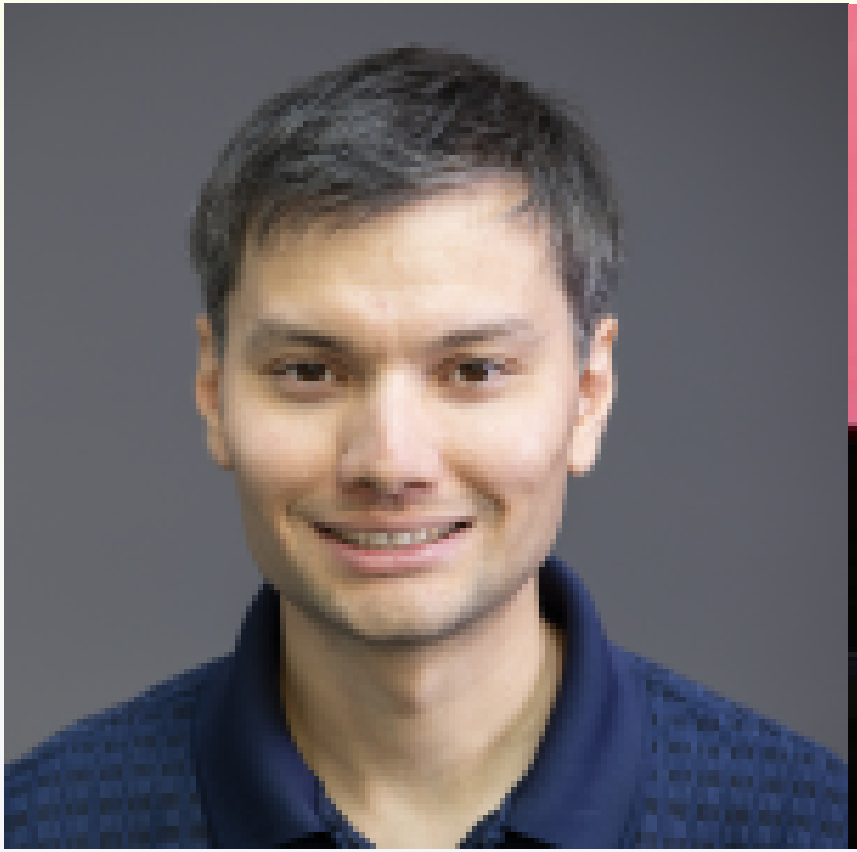
then Φ is no longer consistent with respect to the contaminated model (provided $p \neq q$ for all n).

We have a VN scheme.
We have a distribution.
And that distribution is in our scheme's consistency class.
Now, after contamination,
the new distribution is not in our scheme's consistency class.

“reality is for people who can’t handle drugs”



Larson's Rule:
regularization of contaminated graphs
for VN inference task



Jonathan Larson, MSR

Chris White, MSR

regularization

Trimmed estimator. From **Wikipedia**, the free encyclopedia. In statistics, a **trimmed** estimator is an estimator derived from another estimator by excluding some of the extreme values, a process called truncation. This is generally done to obtain a more robust statistic, and the extreme values are considered outliers.

Trimmed estimator - Wikipedia

https://en.wikipedia.org/wiki/Trimmed_estimator

regularization

Trimmed estimator. From **Wikipedia**, the free encyclopedia. In statistics, a **trimmed** estimator is an estimator derived from another estimator by excluding some of the extreme values, a process called truncation. This is generally done to obtain a more robust statistic, and the extreme values are considered outliers.

Trimmed estimator - Wikipedia
https://en.wikipedia.org/wiki/Trimmed_estimator

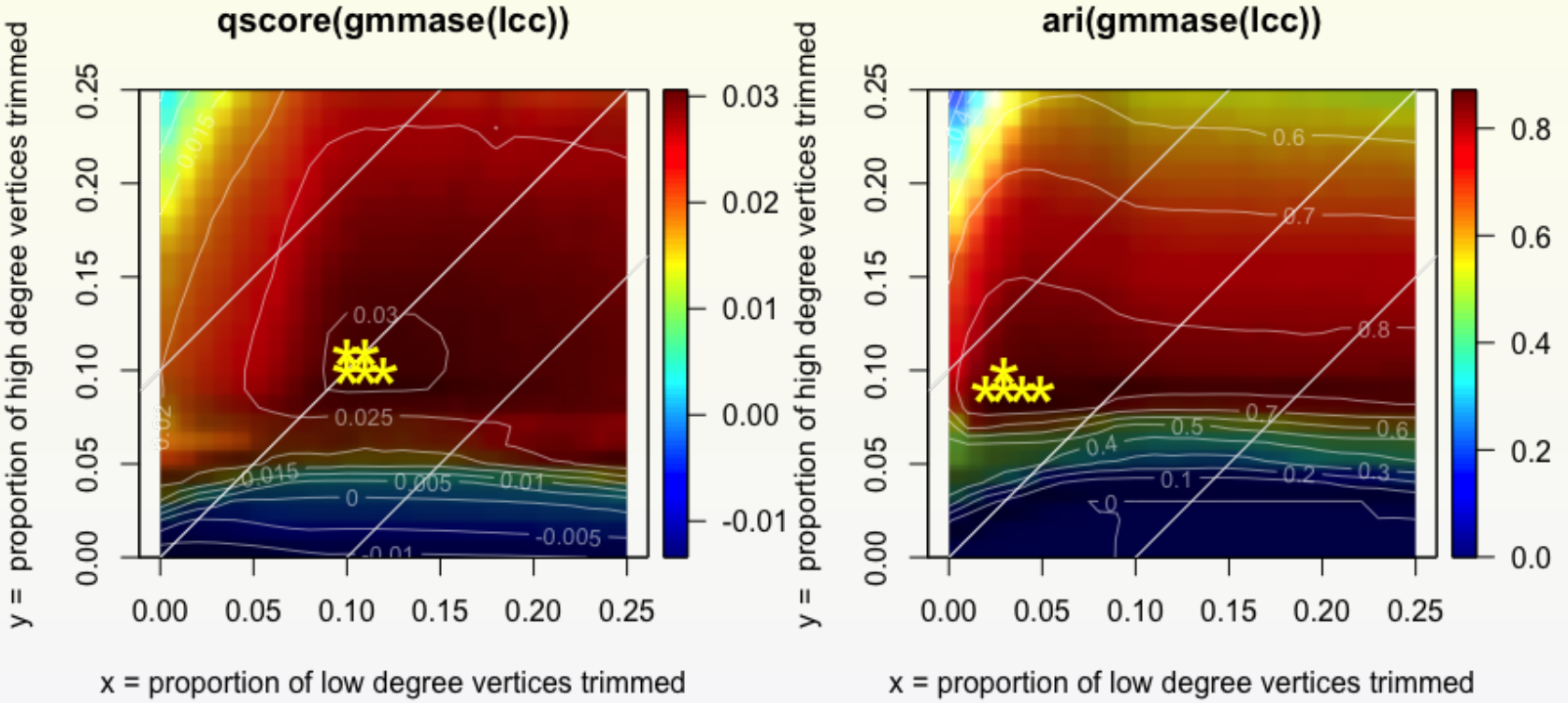
$$\bar{X}_\alpha = \frac{X_{([n\alpha]+1)} + \cdots + X_{(n-[n\alpha])}}{n - 2[n\alpha]}$$

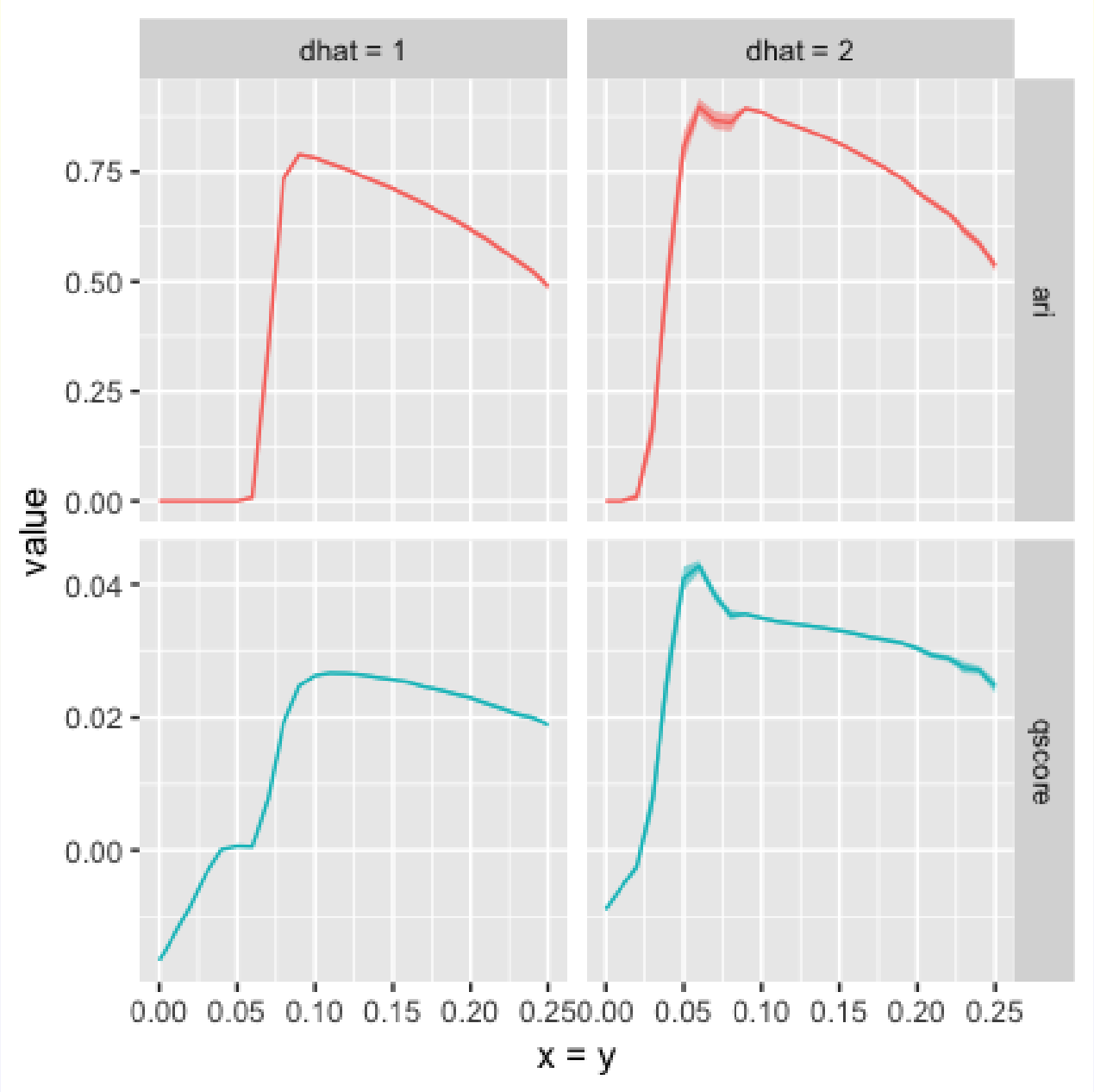
The **modularity** of a graph with respect to some vertex partition measures how good the partition is, or how separated the different partitions are from each other. It is defined as

$$Q = \frac{1}{2m} * \sum_{i,j} \left(\frac{A_{ij} - k_i * k_j}{2m} \right) \delta(c_i, c_j),$$

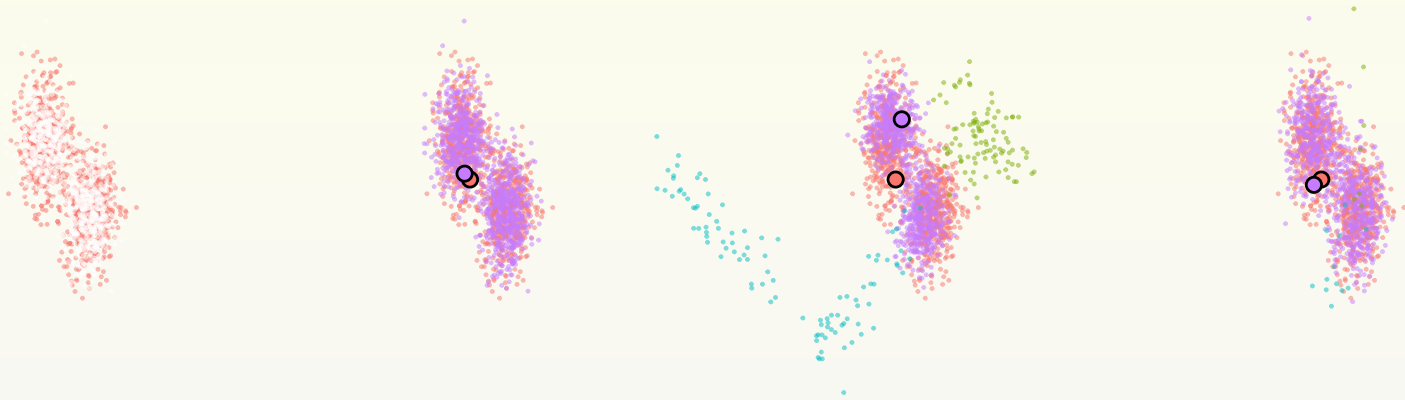
where

- m is the number of edges,
- A_{ij} is the element of the adjacency matrix,
- k_v is the degree of v,
- c_v is the type (or component) of v,
- $\delta(x, y)$ equals 1 if $x = y$ and 0 otherwise.


qscore(gmmase(lcc))
y = proportion of high degree vertices trimmed
x = proportion of low degree vertices trimmed
0.00
0.05
0.10
0.15
0.20
0.25
0.03
0.02
0.01
0.00
-0.01
0.03
0.025
0.015
0.005
0
-0.005
-0.01
ari(gmmase(lcc))
y = proportion of high degree vertices trimmed
x = proportion of low degree vertices trimmed
0.8
0.6
0.4
0.2
0.0
0.6
0.7
0.8
0.5
0.4
0.1
0.2
0.3
0

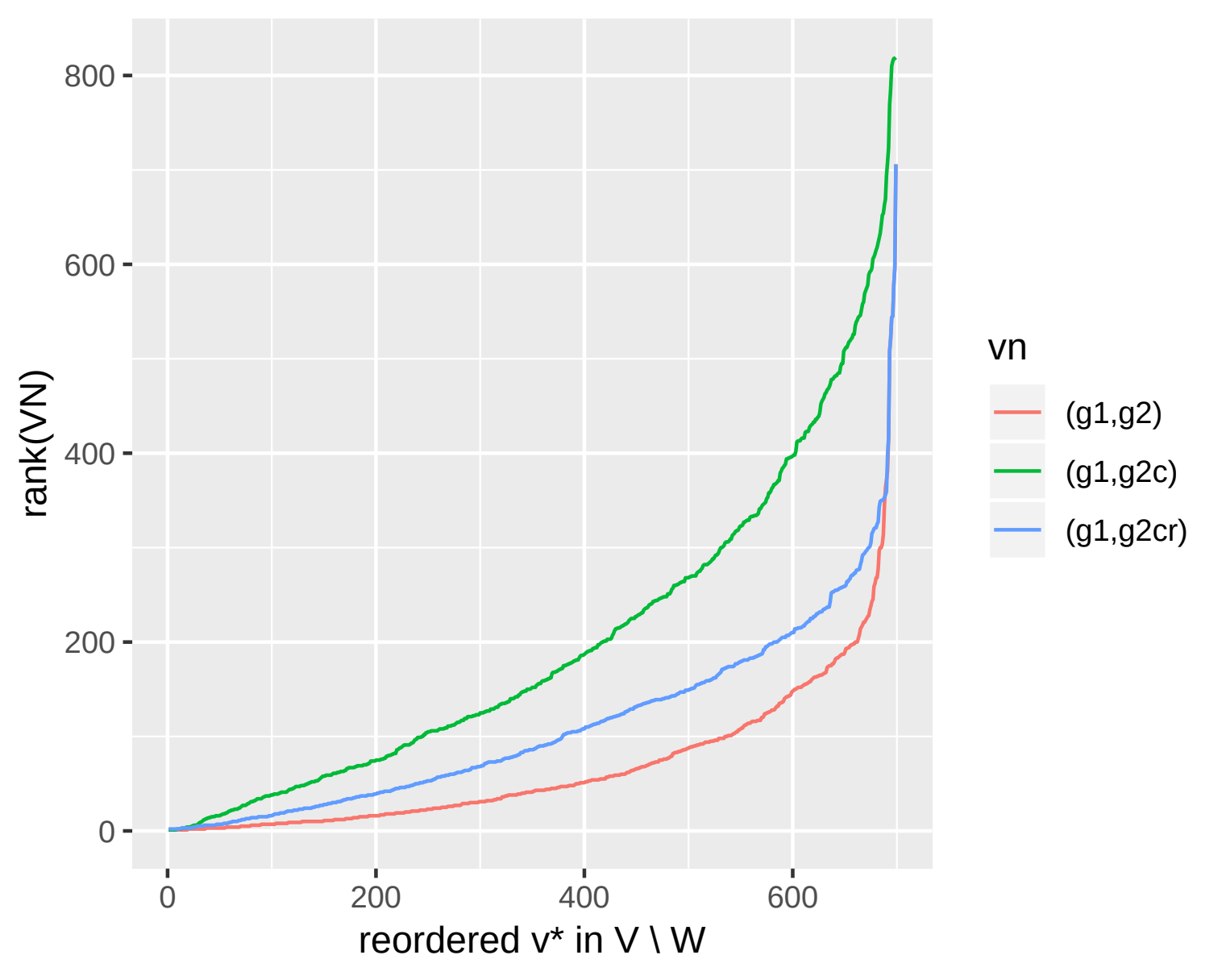

dhat = 1
dhat = 2
ari
qscore
value
0.75
0.50
0.25
0.00
0.04
0.02
0.00
0.00 0.05 0.10 0.15 0.20 0.25
0.00 0.05 0.10 0.15 0.20 0.25
x = y

Illustrative Simulation Result: VN◦GMM◦ASE



(a) G_1
(b) G_1 & G_2: VN succeeds
(c) G_1 & G_2^c: VN fails under contamination
(d) G_1 & G_2^{cr}: VN succeeds after regularization

Simulation Results



Yogi Berra (purportedly):

"In theory there is no difference between theory and practice. In practice, there is."

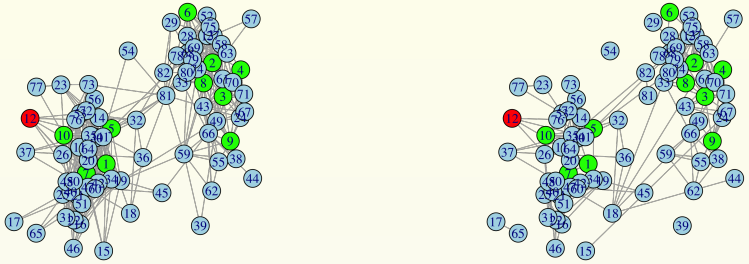
(cf. *"That's all well and good in practice, but how does it work in theory?"*)

High School Data: Facebook & Friendship

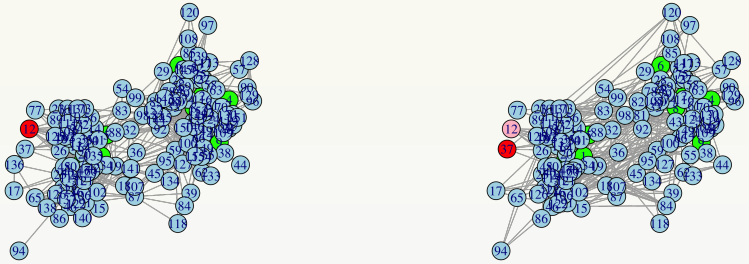
About the data:

	FB full	Survey full	FB core	Survey core
$\|V\|$	156	134	82	82
$\|E\|$	1437	406	513	214
average degree	18.42	6.06	12.51	5.22

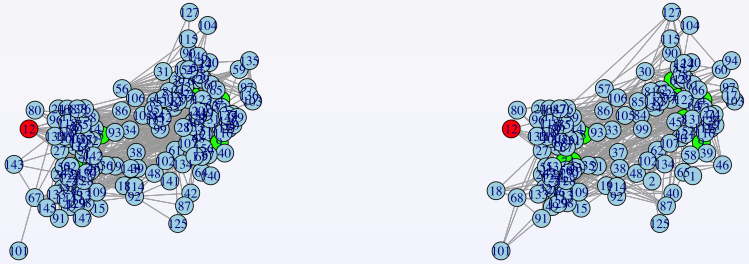
High School Data: Facebook & Friendship



Core: VN succeeds.

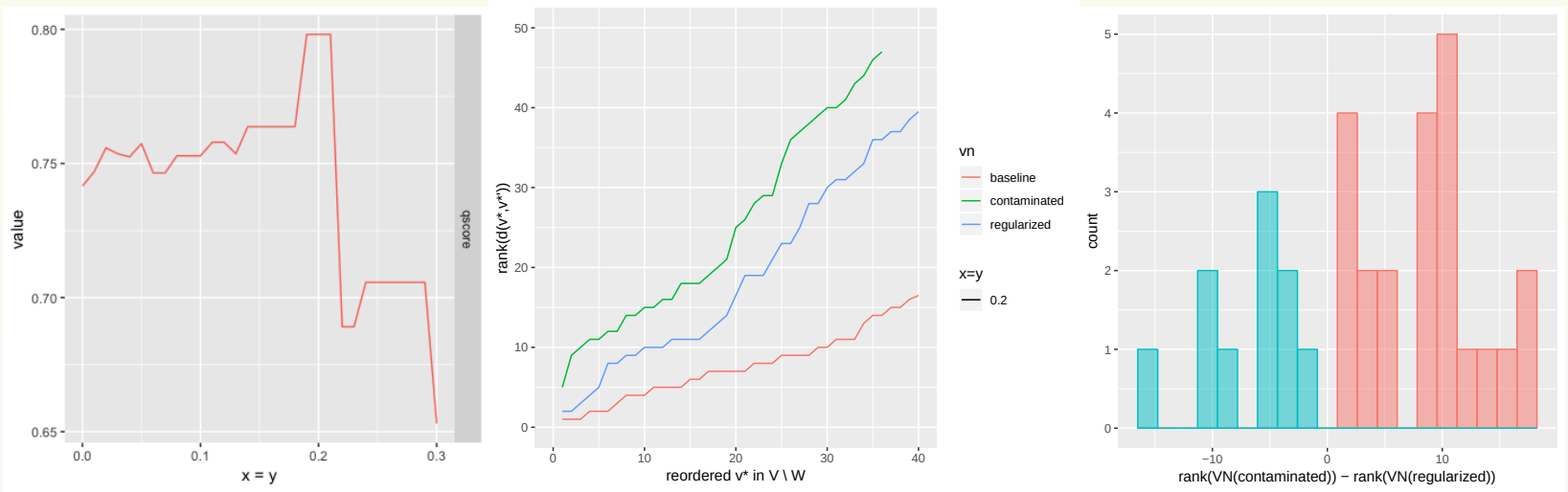


Full: VN fails.



Regularized: VN succeeds.

High School Data: Facebook & Friendship Results



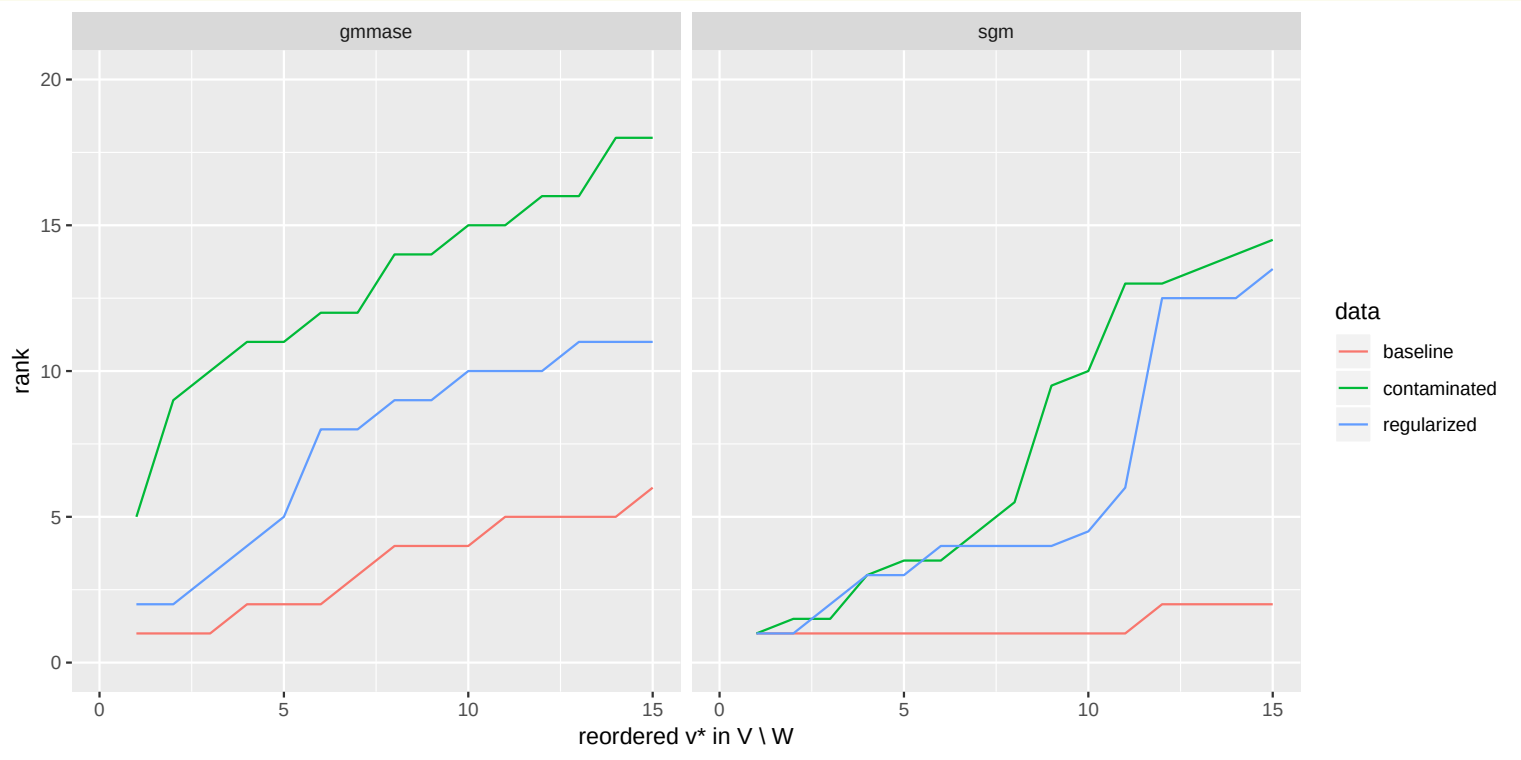
all: G_1 core ; G_2 core $\rightarrow$ full $\rightarrow$ regularized

left: Louvain modularity against regularization trim %

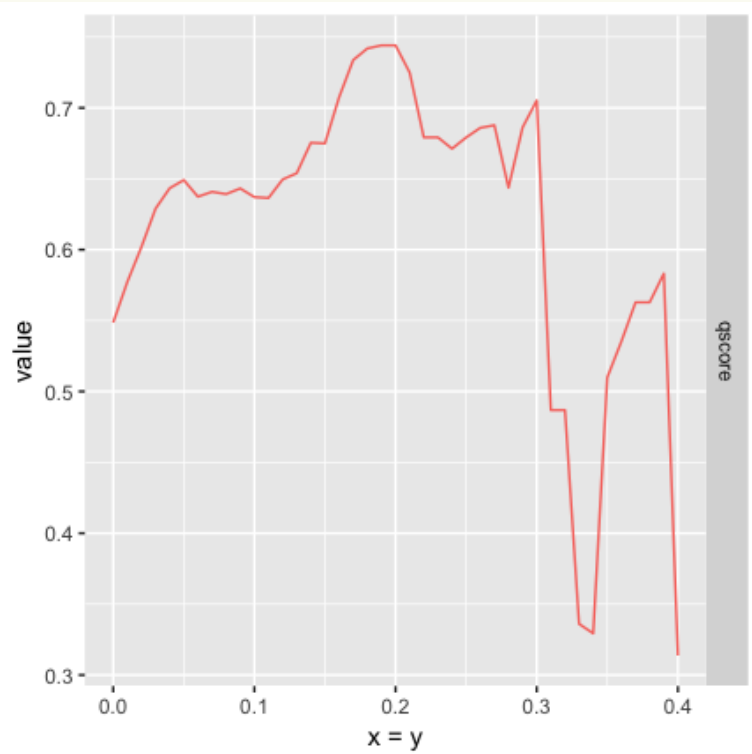
middle: vn o gmm o ase

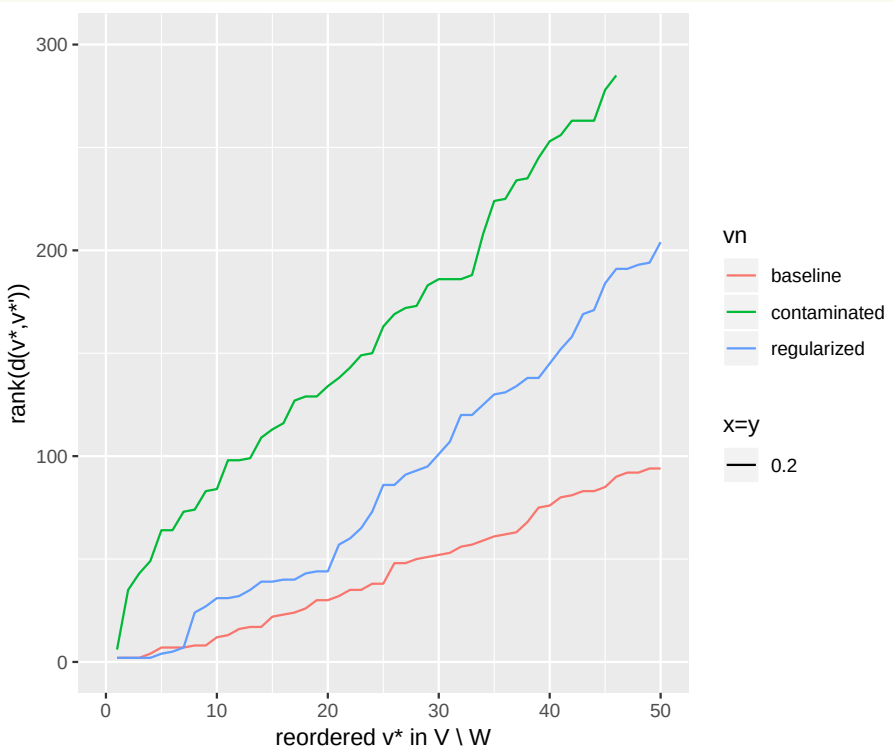
right: $p \approx 0.025$

High School Data: Facebook & Friendship Results



Zebrafish Results





People

Y. Park

V. Lyzinski

K. Levin

J. Agterberg

H. Patsolic

Z. Mousavi

M. Tang

A. Athreya

J. Larson

C. White

Leopold Kronecker to Hermann von Helmholtz (1888):

"The wealth of your practical experience
with sane and interesting problems
will give to mathematics
a new direction and a new impetus."

Kronecker

Helmholtz