SUPPLEMENT TO "IDENTIFYING PREFERENCES IN NETWORKS WITH BOUNDED DEGREE"

(*Econometrica*, Vol. 86, No. 1, January 2018, 263–288)

ÁUREO DE PAULA
University College London, São Paulo School of Economics, CeMMAP, IFS, and CEPR

SETH RICHARDS-SHUBIK
Lehigh University and NBER

ELIE TAMER
Harvard University

APPENDIX B: SUFFICIENCY AND SHARP IDENTIFICATION

CONDITIONS 1 AND 2 ARE SUFFICIENT (as well as necessary) for the existence of a pairwise stable network under further restrictions on preferences, which may be reasonable in some applications. The main restriction is that the marginal utility from adding a link to a distant individual ($> 2D$) must be greater than that from adding a link to a nearby individual ($\leq 2D$) of the same type. Also, a form of separability is required in the utility function, as defined below. Then, because Conditions 1 and 2 are necessary *and* sufficient for equilibrium, we can use them to recover exactly the set of preference parameters that are compatible with the observed type shares (i.e., the sharp identified set).

To state the main restriction, we need to denote the type(s) that an individual of type t could become if he or she were to add a link to an individual of type s who is within distance $2D$. So, abstractly, we define the correspondence $\Psi(t, s)$ to collect all such types.[31] For example, in Figure 4, $\hat{t} \in \Psi(t, s)$ and $\hat{s} \in \Psi(s, t)$ (note the arguments are reversed). In general, these sets could be constructed via enumeration: first check each alter node in type t to see whether an individual of type s could be located there, then determine what type the individual of type t would become if a link were added to an individual of type s at that location. To define the separability requirement, we refer to separate components in the network type *when the ego is removed*. These components consist of subsets of alters that are connected with each other (i.e., paths exist among them) but not with other alters in the network type, if the ego is removed. We require additive separability of the utility function across such subsets of alters. Bringing these together, the assumption is as follows.

ASSUMPTION 3: *For an individual of type t, let $\bar{t}$ be the network type that would be obtained by linking to an individual of type s who is at a distance greater than $2D$, and let $\Psi(t, s)$ collect the type(s) that could be obtained by linking to an individual of type s who is at a distance of $2D$ or less. Assume the following*:

Áureo de Paula: apaula@ucl.ac.uk
Seth Richards-Shubik: sethrs@lehigh.edu
Elie Tamer: elietamer@fas.harvard.edu

[31] There can be multiple types in $\Psi(t, s)$ if it is feasible to have alters of type s located at multiple nodes in the subnetwork defined by type t.

 https://doi.org/10.3982/ECTA13564

(a) $u(\bar{t};\varepsilon) \geq u(\hat{t};\varepsilon), \forall \hat{t} \in \Psi(t,s)$.

(b) *For any type t, the utility function is additively separable across sets of edges and vertices that belong to different components of the graph obtained when the ego is removed from the network type*.

The first part of this assumption is plainly at odds with the clustering observed in many networks, because it does not encourage links among individuals who already share connections. Nevertheless, it is valid for certain preference structures that have received attention in the literature. It is trivially satisfied in models where only direct connections matter (e.g., Currarini, Jackson, and Pin (2009)), because the resulting types yield the same utility (i.e., $u(\bar{t};\varepsilon) = u(\hat{t};\varepsilon)$). It is also satisfied in some models with indirect connections, such as the connections model in Jackson and Wolinsky (1996). The key point is that Assumption 3(a) is more a statement about marginal utility than about total utility.[32] It is possible for network types with more nearby alters to provide greater total utility, while adding links to more distant nodes yields greater marginal utility. This holds in the connections model because the marginal utility of adding a link to some node is reduced by the current value of the indirect connection to that node, which decreases in distance.[33] More generally, it seems that this would be the case when one's closeness (or distance) to other nodes in the network matters more than the clustering among one's alters. In our example utility specification (1), which values both closeness and clustering in some sense, Assumption 3(a) is satisfied in a subset of the parameter space where the value of mutual friendships (ω) is bounded relative to the value of friends of friends (ν). (This is shown in Corollary 1 after the theorem.)

The second part of Assumption 3 makes the utility function additive across links going to separate subsets of alters (i.e., in separate components of the graph without the ego, as defined above). For example, the utility of type $\bar{t}$ in Figure 3 must be the sum of the payoffs obtained from having each of the two links. (Removing the ego from this type leaves two components, each consisting of a W linked to a B, so the assumption is binding.) On the other hand, the alters in type $\hat{t}$ in Figure 4 remain connected without the ego, so the utility of this type does not need to be separable across the ego's two links. Utility specification (1) indeed satisfies Assumption 3(b) for any values of the parameters, while allowing for important complementarities across links such as with mutual friendships (see Corollary 1). However, the assumption is restrictive in general, for example, ruling out convex costs in the number of links. (Having convex costs might be natural in many applications, but in most specifications in the econometric literature on networks, they are not included.)

With this assumption, we then state the result as follows.[34]

THEOREM 3: *For models satisfying Assumption 3, given a probability distribution of preference classes $\{P_{H|v_1}\}$, there exists a network that is pairwise stable (except possibly for a set*

[32]Note that Assumption 3(a) is equivalent to $u(\bar{t};\varepsilon) - u(t;\varepsilon) \geq u(\hat{t};\varepsilon) - u(t;\varepsilon)$.

[33]In the connections model, adding a link to a node currently at distance d yields $w_{ij}(\delta^1 - \delta^d) - c_{ij}$ (where $w_{ij} \geq 0$ is the benefit of being connected to node j, $0 < \delta < 1$ discounts that benefit by the distance between i and j, and c_{ij} is the cost of a direct link with node j). The marginal utility of adding link ij also depends on changes in the distances to other nodes reached via node j, and those marginal benefits are similarly reduced by the current distance to j.

[34]Because Conditions 1 and 2 assess the measures of sets of agents, they cannot guarantee pairwise stability for sets with zero measure. This issue does not arise in Theorem 1 because it does not affect the necessity of the conditions.

of agents with zero measure) *where the proportion of agents of type* t *is equal to* π_t *for each* $t \in \mathcal{T}$ *if and only if there exists a vector of allocation parameters* α *satisfying Conditions* 1 *and* 2 *such that* $\pi_t = \frac{1}{\mu} \sum_H \mu_{v_1(t)} P_{H|v_1(t)} \alpha_H(t)$ *for every* $t \in \mathcal{T}$.

The necessity of Conditions 1 and 2 was established by Theorem 1. The proof of their sufficiency below uses a contrapositive argument. It shows that if there is no pairwise stable network with the given type shares, then allocation parameters cannot be found satisfying the two conditions. The key point is that if positive measures of nearby individuals mutually desire to add links, then, under Assumption 3(a), these individuals must also desire to add links with distant individuals of the same types, which violates Condition 2.

PROOF OF THEOREM 3: To start, fix a vector of preference parameters and, hence, a distribution of preference classes, as well as the observed proportions of network types. Suppose that under these preferences, any network with these type shares is unstable. Therefore, for any such network, there must be a positive measure of pairs of individuals for whom the presence or absence of a link between them is unstable. To translate this into our conditions, first note that for any network among a set of players N, there is a unique vector of allocation parameters that expresses the allocation of the individuals from each preference class to each network type. This is because each individual is associated with one and only one preference class, and one and only one network type.

First, we consider existing links ($G(i, j) = 1$). If there is a positive measure of pairs of individuals who are linked but one or both of them would prefer to drop the link, then there must be some preference class H where a positive measure of individuals with preferences in this class is some network type that is not in H. Therefore, $\alpha_H(t) > 0$ for some $t \notin H$ and so Condition 1 would be violated.

If all existing links are stable, there must be nonexisting links ($G(i, j) = 0$) that are unstable (i.e., both i and j would prefer to add the link). We first consider nonexisting links between individuals who are distant from each other in the network (i.e., $d(i, j; G) > 2D$). If there is a positive measure of such pairs of individuals who would prefer to be linked with each other, then there is at least one pair of network types (t, s) such that positive measures of individuals of these two types would prefer to add links with each other. A link between two individuals of types t and s who are distant from each other would transform them to types $\bar{t}$ and $\bar{s}$, respectively, and this tuple of types, (t, s) and $(\bar{t}, \bar{s})$, pertains to one of the equations in Condition 2.

We now use Assumption 3(b) to show that $\bar{t}$ and $\bar{s}$ are in the respective preference classes of these individuals of types t and s, which leads to a violation of Condition 2. Consider $\bar{t}$. This type contains the same subnetwork of alters as type t, plus an additional subnetwork from the link to an individual of type s. If the ego is removed from type $\bar{t}$, the additional subnetwork is a separate component in the graph (there are no other paths from the ego in $\bar{t}$ to the alter formerly of type s, because that alter was previously beyond $2D$). The utility of type $\bar{t}$ is therefore additively separable between the subnetwork from type t and the additional subnetwork. Denote the links in type t as $l = 1, \ldots, l_t$, the additional link in type $\bar{t}$ as $l_t + 1$, and the representations of these types as (A_t, v_t) and $(A_{\bar{t}}, v_{\bar{t}})$, respectively. (For simplicity, suppose that the alters from type t appear in the same rows of $A_{\bar{t}}$ as they do in A_t.) Given the additive separability, if type t is in an individual's preference class (i.e., $u(A_t, v_t; \varepsilon) \geq u(A_{t,-l}, v_t; \varepsilon)$, $l = 1, \ldots, l_t$), then the corresponding links in type $\bar{t}$ are preferred as well (i.e., $u(A_{\bar{t}}, v_{\bar{t}}; \varepsilon) \geq u(A_{\bar{t},-l}, v_{\bar{t}}; \varepsilon)$, $l = 1, \ldots, l_t$). This indeed holds for the relevant individuals of type t, because here we are supposing that all existing links are stable (otherwise Condition 1 is violated). These

individuals also prefer type $\bar{t}$ over type t, so the additional link in $\bar{t}$ is preferred as well (i.e., $u(A_{\bar{i}}, v_{\bar{i}}; \varepsilon) \geq u(A_{\bar{i},-(l_t+1)}, v_{\bar{i}}; \varepsilon)$). Therefore, all the links in type $\bar{t}$ satisfy the preference inequalities needed for this type to be in the preference classes of these individuals. Hence, the expression $\mu_{v_1(t)} \sum_{\tilde{H} \in \mathcal{H}} P_{\tilde{H}|v_1(t)} \alpha_{\tilde{H}}(t) 1_{\bar{i} \in \tilde{H}}$ is strictly positive. The same argument applies for type $\bar{s}$ and the relevant individuals of type s. Hence, the expression $\mu_{v_1(s)} \sum_{\check{H} \in \mathcal{H}} P_{\check{H}|v_1(s)} \alpha_{\check{H}}(s) 1_{\bar{s} \in \check{H}}$ is strictly positive as well. Therefore, the product of the measures given by these expressions is strictly positive, which violates Condition 2.

Last, we consider nonexisting links between individuals who are $2D$ or less from each other in the network. If there is a positive measure of such pairs of individuals who would prefer to be directly linked, then there is some pair of network types (t, s) and some distance $d \leq 2D$ such that positive measures of individuals of these two types who are at distance d from each other would prefer to add links with each other. A link between two such individuals would transform them to some types $\hat{t} \in \Psi(t, s)$ and $\hat{s} \in \Psi(s, t)$, respectively. These individuals prefer $\hat{t}$ over t and $\hat{s}$ over s, respectively. By Assumption 3(a), they weakly prefer $\bar{t}$ over $\hat{t}$ and $\bar{s}$ over $\hat{s}$, respectively (where, as before, the types $\bar{t}$ and $\bar{s}$ would result from a link being added between distant individuals of types t and s). Hence, they prefer $\bar{t}$ over t and $\bar{s}$ over s, respectively. Then the same argument applies as for nonexisting links between distant individuals. For these individuals, all the links in type t (respectively, s) are preferred, as is the additional link in type $\bar{t}$ ($\bar{s}$); hence, $\bar{t}$ ($\bar{s}$) is in their preference classes. Therefore, as above, the expressions $\mu_{v_1(t)} \sum_{\tilde{H} \in \mathcal{H}} P_{\tilde{H}|v_1(t)} \alpha_{\tilde{H}}(t) 1_{\bar{i} \in \tilde{H}}$ and $\mu_{v_1(s)} \sum_{\check{H} \in \mathcal{H}} P_{\check{H}|v_1(s)} \alpha_{\check{H}}(s) 1_{\bar{s} \in \check{H}}$ are strictly positive, and so Condition 2 would be violated. *Q.E.D.*

Thus, in cases where Assumption 3 is appropriate, Conditions 1 and 2 can be used to characterize the identified set precisely. We next show how this applies to specification (1).

COROLLARY 1: *In utility specification* (1), *if* $\nu \geq 0$ *and* $\omega \leq \frac{L}{L-1}\nu$, *then there exists a network that is pairwise stable* (*except possibly for a set of agents with zero measure*) *where the proportion of agents of type* t *is equal to* π_t *for each* $t \in \mathcal{T}$ *if and only if there exists a vector of allocation parameters* α *satisfying Conditions* 1 *and* 2 *such that* $\pi_t = \frac{1}{\mu} \sum_H \mu_{v_1(t)} P_{H|v_1(t)} \alpha_H(t)$ *for every* $t \in \mathcal{T}$.

PROOF: It suffices to show that specification (1) satisfies Assumption 3 when ν and ω are in the stated region. To establish Assumption 3(a), we compare the utility from adding a link to a distant individual of some type against the utilities from adding a link to individuals of the same type at each distance $\leq 2D$ (supposing this is feasible).

Without loss of generality, consider the possible changes in utility for an individual i of type t if a link is added to some individual k of type s. If $d(i, k; G) > 2D$, then type $\bar{t}$ is obtained. The change in utility is $f(X_i, X_k) + \varepsilon_{il}(X_k) + |N(k)|\nu$, where $N(k)$ are k's existing friends (i.e., in type s). There is no intersection between $N(k)$ and i's existing friends of friends in type t; hence, there are $|N(k)|$ additional friends of friends in type $\bar{t}$. If $d(i, k; G) = 2D = 4$, then some type in $\Psi(t, s)$ is obtained. This type yields the same utility as $\bar{t}$, because again there is no intersection between $N(k)$ and the existing friends of friends in type t (otherwise i and k would be at distance 3). If $d(i, k; G) = 3$, then k's friends and i's friends of friends must intersect (this is how i and k are at distance 3). An example of this was shown in Figure 4. Hence, there are fewer than $|N(k)|$ additional friends of friends in the resulting type, and so the change in utility is less than for type $\bar{t}$ (given $\nu \geq 0$).

Finally, suppose that $d(i,k;G)=2$. In this case, the change in utility from adding a link to k would include the value of mutual friendship (ω). Specifically, the marginal utility would include a term $\sum_{j\in N(i)} G(j,k)\omega$ for the value of the new mutual friendships that are created by adding this link (where $N(i)$ are i's existing friends in type t).[35] However, for each new mutual friendship, there would also be one fewer new friend of a friend, compared with the result of adding a link to a distant individual of type s. (Note that k's neighbors would be either a mutual friend or a friend of a friend to i in the resulting type, but not both.) In addition, individual k would no longer be a friend of a friend to i. So there would be $\sum_{j\in N(i)} G(j,k)+1$ fewer friends of friends in the resulting type, call it $\hat{t}$, compared with type $\bar{t}$. Hence, the difference between the utilities of these types is

$$u(\hat{t};\varepsilon)-u(\bar{t};\varepsilon)=\sum_{j\in N(i)} G(j,k)(\omega-\nu)-\nu.$$

The maximum possible number of new mutual friendships is $L-1$ (because type t must have at least one link to spare), so having $\omega\leq\frac{L}{L-1}\nu$ ensures that this difference is weakly negative: $(L-1)(\omega-\nu)-\nu\leq 0$.

We have thus established that for each distance $d\leq 2D$, the utility of adding a link to someone of type s at that distance (if feasible) is weakly less than the utility of adding a link to someone of type s beyond $2D$. Therefore, Assumption 3(a) is satisfied.

Now we show that Assumption 3(b) is satisfied in specification (1) in any region of the parameter space. The first line of (1) is always additively separable, whether or not the direct friends are in separate components of the graph without the ego. The second line is separable because there is no intersection among the friends of friends in separate components of the graph without the ego. To see how this delivers the required additive separability, let J and K contain the direct friends in two different components of the graph without the ego. Then

$$\left(\bigcup_{j\in J} N(j)-N(i)-\{i\}\right)\cap\left(\bigcup_{k\in K} N(k)-N(i)-\{i\}\right)=\emptyset,$$

and so

$$\left|\bigcup_{\hat{j}\in J\cup K} N(\hat{j})-N(i)-\{i\}\right|\nu=\left|\bigcup_{j\in J} N(j)-N(i)-\{i\}\right|\nu+\left|\bigcup_{k\in K} N(k)-N(i)-\{i\}\right|\nu.$$

The third line of (1) is additively separable because $G(j,k)=0$ for any pair of direct alters $j\in J$ and $k\in K$ (with J and K defined as above). Hence,

$$\sum_{\hat{j}\in J\cup K}\sum_{\substack{\hat{j}'>\hat{j}\\ \hat{j}'\in J\cup K}} G\big(\hat{j},\hat{j}'\big)\omega=\sum_{j\in J}\sum_{\substack{j'>j\\ j'\in J}} G\big(j,j'\big)\omega+\sum_{k\in K}\sum_{\substack{k'>k\\ k'\in K}} G\big(k,k'\big)\omega.$$

Therefore, Assumption 3(b) is satisfied as well. *Q.E.D.*

This result applies in both simulation exercises in Section 7, given the parameterizations that are used. In the first we have $\omega=\nu=0$ and in the second we have $\omega=\nu=0.2$. Hence, in both cases we recover the sharp identified set.

[35]There could be multiple paths of length 2 from i to k, so there could be multiple new mutual friendships.

APPENDIX C: STATISTICAL INFERENCE

We take a sampling approach to inference whereby the target population is a single, large network (e.g., a school) and the statistical randomness (to be summarized through standard errors) comes from the i.i.d. sampling assumption (think here of choosing a set of students at random from the school). Naturally, a different sampling procedure from the large network would lead to a different characterization of the statistical uncertainty, and here we illustrate our ideas with simple random sampling. See Kolaczyk (2009) for various sampling procedures in networks.

PROOF OF PROPOSITION 1: Let π be a scalar constant which represents the proportion of interest. (Here, we show the pf for the scalar case for simplicity.) The sampling approach to inference[36] maintains that we have a realization of the random process that generates a single, large network. Conditional on this population network, a simple random sample without replacement is drawn from the network.[37] Observations in this sample are by definition i.i.d. conditional on the network, and the only randomness in this second stage is coming from the act of sampling conditional on the realization of the large network from the random process. We take the population of interest to be a network of size N_p (unobserved) where we maintain that this population is large (but countable). Here, we restrict ourselves to a finite population for simplicity, but similar results would hold for a more complex population. We subscript N with a p to indicate the population. Let there be a *sampled* network of size N_s from the population network where an observation in this sample is chosen using simple random sampling. This means that we choose a sample of size N_s from the population of size N_p conditional on the target (population) network. We enumerate the population by $\mathbf{T}_p' = (T_1, \ldots, T_{N_p})$ and the observed sample as $\mathbf{T}_s' = (T_1^s, \ldots, T_{N_s}^s)$, where the sample is a subset from $\mathbf{T}_p$. We can take the T's to be, for example, whether a given individual has any black friends. The sampling process is conditional on the outcomes (the T's) and so the probability of choosing a sample of size N_s from N_p is $1/\binom{N_p}{N_s}$. Now define a sequence of binary indicator variables $(W_i)_{i=1}^{N_p}$. These are random variables such that $W_i = 1$ if observation i in the population is observed in the sample. We have $E[W_i|\mathbf{T}_p] = P(W_i = 1|\mathbf{T}_p) = \frac{\binom{N_p-1}{N_s-1}}{\binom{N_p}{N_s}} = \frac{N_s}{N_p}$. Also, the W's are correlated with correlation $E[W_iW_j|\mathbf{T}_p] = \frac{N_s}{N_p}\frac{N_s-1}{N_p-1}$ for $i \neq j$ (the event that i is in the sample is not independent from whether $j \neq i$ is in the sample since the sample must always be of size N_s). Here, note that the expectation and the probability are conditional on the T's and so this randomness is purely from the simple random sampling process where $\mathbf{T}_p$ is taken as a vector of constants (i.e., the T's can be arbitrarily correlated). So, our estimator for π is

$$\hat{\pi} = \frac{1}{N_s}\sum_{i=1}^{N_s} T_i^s = \frac{1}{N_p}\sum_{i=1}^{N_p} \frac{W_i}{\frac{N_s}{N_p}} T_i.$$

[36]Here, we follow the general sampling approach of Cochran (1977) and more recently of Imbens and Rubin (2015) (in the context of inference on causal effects).

[37]This is simpler but not essential as the difference between sampling with replacement is small especially in cases where the sample is much smaller than the population.

First, it is simple to see that $\hat{\pi}$ is unbiased for π:

$$E[\hat{\pi}] = \frac{1}{N_p}\sum_{i=1}^{N_p} \frac{E[W_i|\mathbf{T}_p]}{\frac{N_s}{N_p}} T_i = \frac{1}{N_p}\sum_{i=1}^{N_p} T_i = \pi.$$

Now, for the variance, let σ_p^2 be the population variance: $\frac{1}{N_p}\sum_{i=1}^{N_p}(T_i - \pi)^2$:

$$\begin{aligned}\operatorname{Var}(\hat{\pi}) &= \frac{\sigma_p^2}{N_s} - \frac{\sigma_p^2}{N_p} - \frac{1}{N_p^2}\sum_{i\neq j, i,j=1}^{N_p} (T_i - \pi)(T_j - \pi) \\ &\sim \frac{\sigma_p^2}{N_s},\end{aligned}$$

where the second line holds if indeed N_p is much larger than N_s and so the last two terms are negligible. Note here that the estimator is unbiased and its variance is proportional to $\frac{1}{N_s}$, the inverse of the sample size. So, as N_s increases, the variance approaches zero. In this finite population, N_s becomes closer to N_p, and the variance will be small. Note that the notion of "consistency" still holds in the finite population setup in that, as N_s approaches N_p, we learn the parameter of interest exactly since now (in a finite population) $\frac{1}{N_p}\sum_{i=1}^{N_p} T_i$ is the parameter of interest (rather than π) and here $P(W_i = 1)$ approaches 1 as N_s approaches N_p (i.e., as everyone gets sampled). This is exactly Cochran's definition of consistency in sampling whereby "the estimate becomes exactly equal to the population value." See Cochran (1977, p. 21).

An approximate $(1-\alpha)$ two sided confidence interval for π can then be constructed as $[\hat{\pi} - 1.96\sqrt{\widehat{\operatorname{Var}(\hat{\pi})}}, \hat{\pi} + 1.96\sqrt{\widehat{\operatorname{Var}(\hat{\pi})}}]$.[38] *Q.E.D.*

Asymptotic normality of sample means under simple random sampling from finite populations was studied under the approximate setup whereby both N_s and N_p approach infinity and results can be used to show that $N_s^{-\frac{1}{2}}\sum_{i=1}^{N_s}(T_i^s - \pi) \to^d \mathcal{N}(0, \pi(1-\pi))$. For example, Hájek (1960) provided necessary and sufficient conditions for the normal approximation to hold in sampling from finite populations when both N_s and N_p approach infinity. Essentially, he showed that a Lindberg-type condition must hold for the normal approximation to be valid. Here, this condition holds trivially since the T's are binary. See Cochran (1977) and Hájek (1960).

We now discuss how we can use the confidence intervals to map the uncertainty in sampling to θ which is the vector that characterizes the payoff structure. The data are informative only on the measure of network types, $\pi \equiv (\pi_t)_{t\in\mathcal{T}}$—see Proposition 1 above.

Let there be a given vector π of observed type probabilities. Then, the identified set $\Theta \subset \mathrm{R}^k$ in a given (large) network can be defined as follows (without conditioning on $\mathbf{X}$):

$$\Theta \equiv \Theta(\pi) = \left\{\theta \in R^k : F(\theta, \pi) = 0\right\},$$

[38] Again, the coverage for this interval holds only approximately for large N_p and N_s. In principle, one can use better approximations to the sampling distribution of the sample mean, but this is given here for simplicity.

where

$$F(\theta;\pi) = \min_{\{\alpha_{H(\theta)}(t):t\in H(\theta)\}} \alpha^\top Q\alpha \quad \text{subject to:}$$
$$\sum_{t\in H} \alpha_{H(\theta)}(t) = 1, \quad \forall H(\theta),$$
$$\alpha_{H(\theta)}(t) \geq 0, \quad \forall t, H(\theta),$$
$$\sum_{H} P_{H(\theta)}\alpha_{H(\theta)}(t) = \pi_t, \quad \forall t$$

(see Section 6 for more on the quadratic matrix Q). Again, the key here is that if we know π, then constructing Θ becomes a family of quadratic programming problems, that is, Θ collects all θ's where $F(\theta,\pi) = 0$. To obtain a confidence region for Θ, we can first construct a confidence region for π and then, for every element π_s in this confidence region, solve for the corresponding Θ_s and take the union $\bigcup_s \Theta_s$. This heuristic relies fundamentally on being able to construct a valid confidence region for π. A sampling approach to inference delivers such a confidence region by sampling nodes independently within one large network conditional on outcomes. In this approach, the target parameter of interest and the objective of the analysis would be to learn about the population type shares (and then using those to back out the structural parameters).

Take then as given that we have an approximation for the distribution of the vector of type shares. The previous discussion provides a sampling approach that can be used to obtain such an approximation.[39] Given the (approximate) distribution of types, we use standard methods to provide a confidence region for the identified set of the structural parameters.

A sample analog of the measure of each type is

$$\hat{\pi}(t_k) = \frac{1}{n}\sum_i 1[i \in t_k]$$

for $k = 1,\ldots,|\mathcal{T}|$ and where these types are mutually exclusive. Moreover, let $\hat{\pi}^\top = (\hat{\pi}(t_1),\ldots,\hat{\pi}(t_{|\mathcal{T}|}))$, which is the vector of estimated type probabilities. We make the following assumption on the population choice probabilities and also maintain the approximate asymptotic distribution for the type vector. This result can easily be derived under standard assumptions (e.g., Theorem 17.2 in van der Vaart (1998)).

ASSUMPTION 4: *Let the network type proportions be such that*

$$\pi(t_k) > 0, \quad \forall k = 1,\ldots,|\mathcal{T}|; \qquad \sum_{k=1}^{|\mathcal{T}|} \pi(t_k) = 1.$$

Also, assume that, as $n \to \infty$,

$$G(\hat{\pi},\pi) = n\sum_{k=1}^{|\mathcal{T}|} \frac{\left(\hat{\pi}(t_k) - \pi(t_k)\right)^2}{\hat{\pi}(t_k)} \to_d \chi^2_{|\mathcal{T}|-1}, \tag{4}$$

where $\pi^\top \equiv (\pi(t_1),\ldots,\pi(t_{|\mathcal{T}|}))$.

[39] Other approaches may be possible. For example, the Bayesian bootstrap can be used to approximate via simple simulations the posterior for the vector of types using draws from gamma distributions.

Again, the exact asymptotic distribution as stated in (4) is one way to characterize sampling uncertainty and is not strictly required.

Given the above assumption, to build a (frequentist) confidence region for Θ, we "invert" the above statistic for multinomial probabilities. In particular, define

$$CI_{1-\alpha}(\pi) = \left\{\pi \in S^{|\mathcal{T}|} : G(\hat{\pi}, \pi) \leq c_{1-\alpha}\left(\chi^2_{|\mathcal{T}|-1}\right)\right\}, \tag{5}$$

where $S^{|\mathcal{T}|}$ is the unit simplex of size $|\mathcal{T}|$, $\hat{\pi}(t)$ are the sample analogues of the type probabilities, $c_{1-\alpha}(\chi^2_{|\mathcal{T}|})$ is the $(1-\alpha)$ critical value of the $\chi^2_{|\mathcal{T}|}$ distribution. The confidence region in (5) is standard and collects the set of network type probabilities that covers the truth with probability $(1-\alpha)$ (in repeated samples). It is also possible to consider a Bayesian approach to inference here where obtaining a posterior for $\pi(t)$ given standard priors can be easily done also (using a Bayesian bootstrap, for example).

Now, for every $\pi \in CI_{1-\alpha}(\pi)$, we can solve our model in terms of the *set* of θ's using the quadratic programming function $F(\theta, \pi) = 0$. The collection of these sets would be a confidence region for the identified set:

$$CI_{1-\alpha}(\theta) = \left\{\Theta(\pi) : F\left(\Theta(\pi), \pi\right) = 0 \text{ for } \pi \in CI_{1-\alpha}(\pi)\right\}. \tag{6}$$

Here, the notation for $\Theta(\pi)$ in $F(\Theta(\pi), \pi) = 0$ implicitly means that $\Theta(\pi)$ is the set of θ's such that $F(\theta, \pi) = 0$.

For other promising approaches to inference, see, for example, the results in Leung (2015).

APPENDIX D: DETAILS OF SIMULATION PROCEDURES

D.1. *Objective Matrix in First Exercise*

The matrix Q for the first simulation exercise is shown here in Figure D.1. The rows and columns correspond to the allocation parameters listed under the heading "parameter."

D.2. *Simplification of the QP Problem in First Exercise*

As noted in the text, the QP problem for this model can be simplified to the point that it is trivial to verify whether the optimal value is zero. We use this result to confirm the identified set obtained using the MCMC search procedure with the original QP problem, and to construct identified sets based on different observations (i.e., different type shares).

The simplification is obtained as follows. There are 16 potentially nonzero allocation parameters in the QP problem for this model (listed in Figure D.1), but 12 can be eliminated with simple manipulations. (Specifically, the four allocation parameters with a positive diagonal element in their row of the matrix Q are set equal to zero, as this is necessary for an optimal value of zero to be attainable, and eight other parameters are eliminated using the constraint $\sum_{t \in H} \alpha_H(t) = 1$.) Expressions for the equilibrium type shares as a function of the remaining four allocation parameters and the structural parameters are listed in Table D.I. Unique values of these remaining allocation parameters can then be recovered, given a vector of structural parameters and the vector of type shares. It is then trivial to compute the objective function value and to assess whether these allocation parameters satisfy the constraint $0 \leq \alpha_H(t) \leq 1$.

(Row)	(Parameter)																
1	$\alpha_1(B,0)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	$\alpha_2(B,0)$	0	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0
3	$\alpha_2(B,B)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	$\alpha_3(B,0)$	0	0	0	0	0	0	0	0	0	0	0	**1**	0	**1**	0	0
5	$\alpha_3(B,W)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	$\alpha_4(B,0)$	0	**1**	0	0	0	**1**	0	0	0	0	0	**1**	0	**1**	0	0
7	$\alpha_4(B,B)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	$\alpha_4(B,W)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	$\alpha_5(W,0)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	$\alpha_6(W,0)$	0	0	0	0	0	0	0	0	0	**1**	0	0	0	**1**	0	0
11	$\alpha_6(W,W)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	$\alpha_7(W,0)$	0	0	0	**1**	0	**1**	0	0	0	0	0	0	0	0	0	0
13	$\alpha_7(W,B)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	$\alpha_8(W,0)$	0	0	0	**1**	0	**1**	0	0	0	**1**	0	0	0	**1**	0	0
15	$\alpha_8(W,W)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	$\alpha_8(W,B)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE D.1.—Matrix Q for model with $D = 1$, $L = 1$, and $\mathcal{X} = \{B, W\}$.

It is also possible to use the expressions in Table D.I to find all the equilibrium type shares for a given vector of structural parameters. Rather than evaluate different parameter vectors given a fixed vector of type shares (to find the identified set), one can instead evaluate different type shares given a fixed vector of structural parameters. Either way, for any pair of vectors of type shares and structural parameters, one recovers the four allocation parameters using the expressions in the table and verifies whether the objective function value is zero and the allocation parameters each fall within the unit interval. Then, because Conditions 1 and 2 are necessary *and* sufficient for pairwise stability in this model (because $D = 1$; see Appendix B), this guarantees that the type shares are obtainable in equilibrium under the given values of the structural parameters. (There is one additional restriction on the admissible vectors of type shares, which is that the measure of blacks linked to whites must equal the measure of whites linked to blacks.) Thus, to find the set of equilibria shown in Figure 5, we fix the structural parameters at the stated values and evaluate a grid of points in the space of admissible vectors of type shares.

TABLE D.I

EQUILIBRIUM TYPE SHARES IN THE FIRST SIMULATION

Type $v = (x, y)$	Proportion (conditional on race of the ego)
$(B,0)$	$\pi_{(B,0)} = (1 - f_{BB})(1 - f_{BW}) + (1 - \alpha_3(B,W))(1 - f_{BB})f_{BW}$
(B,B)	$\pi_{(B,B)} = f_{BB}(1 - f_{BW}) + (1 - \alpha_4(B,W))f_{BB}f_{BW}$
(B,W)	$\pi_{(B,W)} = \alpha_3(B,W)(1 - f_{BB})f_{BW} + \alpha_4(B,W)f_{BB}f_{BW}$
$(W,0)$	$\pi_{(W,0)} = (1 - f_{WW})(1 - f_{WB}) + (1 - \alpha_7(W,B))(1 - f_{WW})f_{WB}$
(W,W)	$\pi_{(W,W)} = f_{WW}(1 - f_{WB}) + (1 - \alpha_8(W,B))f_{WW}f_{WB}$
(W,B)	$\pi_{(W,B)} = \alpha_7(W,B)(1 - f_{WW})f_{WB} + \alpha_8(W,B)f_{WW}f_{WB}$

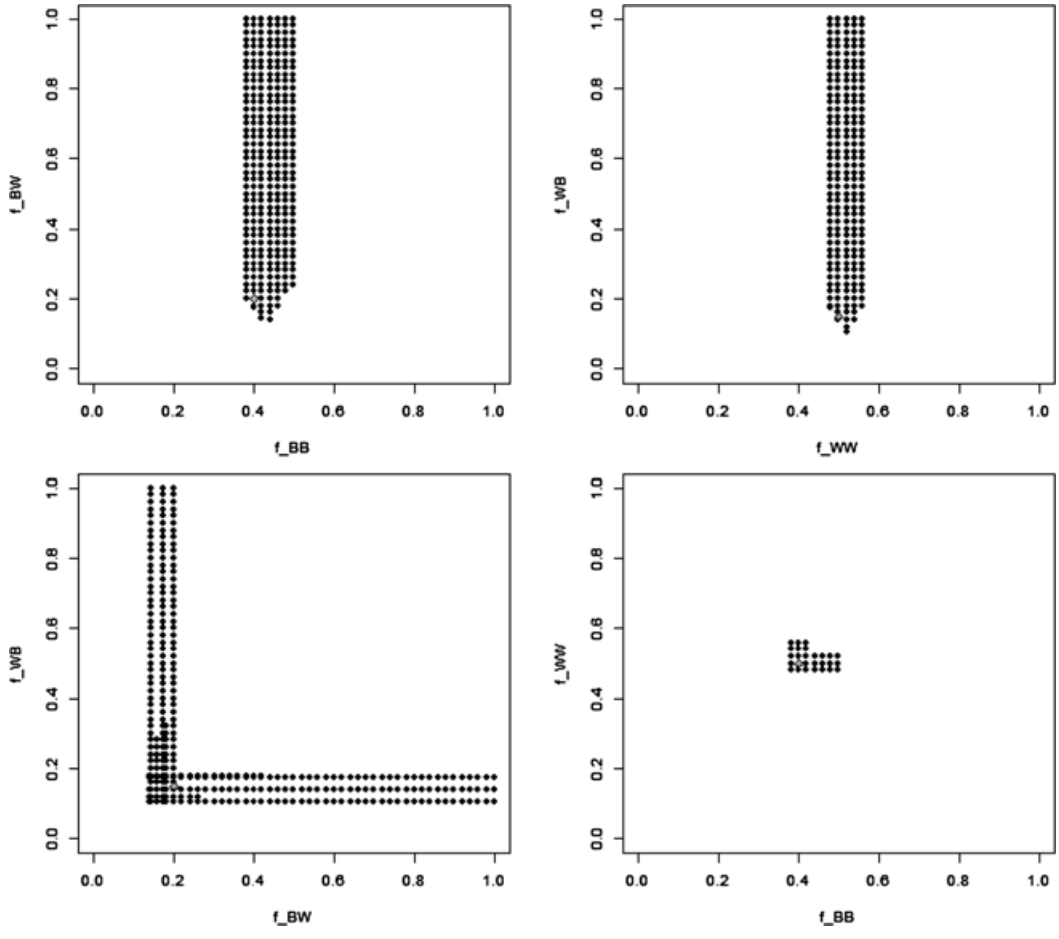

FIGURE D.2.—Identified set from one finite network, using simplified QP problem. *Notes*: Diamonds indicate true parameter values: $f_{BB} = 0.40$, $f_{BW} = 0.20$, $f_{WB} = 0.15$, and $f_{WW} = 0.50$.

D.3. *Utility Specification in the Second Exercise*

This exercise uses $D = 2$, $L = 3$, and $\mathcal{X} = \{B, W\}$. Written in terms of the matrix-vector pairs (A, v) that represent network types, the utility specification (1) is as follows:

$$
\begin{aligned}
u(A, v; \varepsilon) \equiv & \sum_{l=2}^{L+1} a_{1l}\left(f_{v_1, v_l} + \varepsilon_{l-1}(v_l)\right) && \text{(direct connections)} \\
& + \nu \sum_{k>L+1} 1\left\{\sum_{l=2}^{L+1} a_{1l} a_{lk} > 0\right\} && \text{(friends of friends)} \\
& + \omega \sum_{l=2}^{L+1} \sum_{k>l}^{L+1} a_{1l} a_{lk} && \text{(mutual friends)}
\end{aligned} \tag{8}
$$

(recall that row 1 of A corresponds to the ego, rows 2 to $L + 1$ correspond to direct connections, and rows $k > L + 1$ correspond to friends of friends).

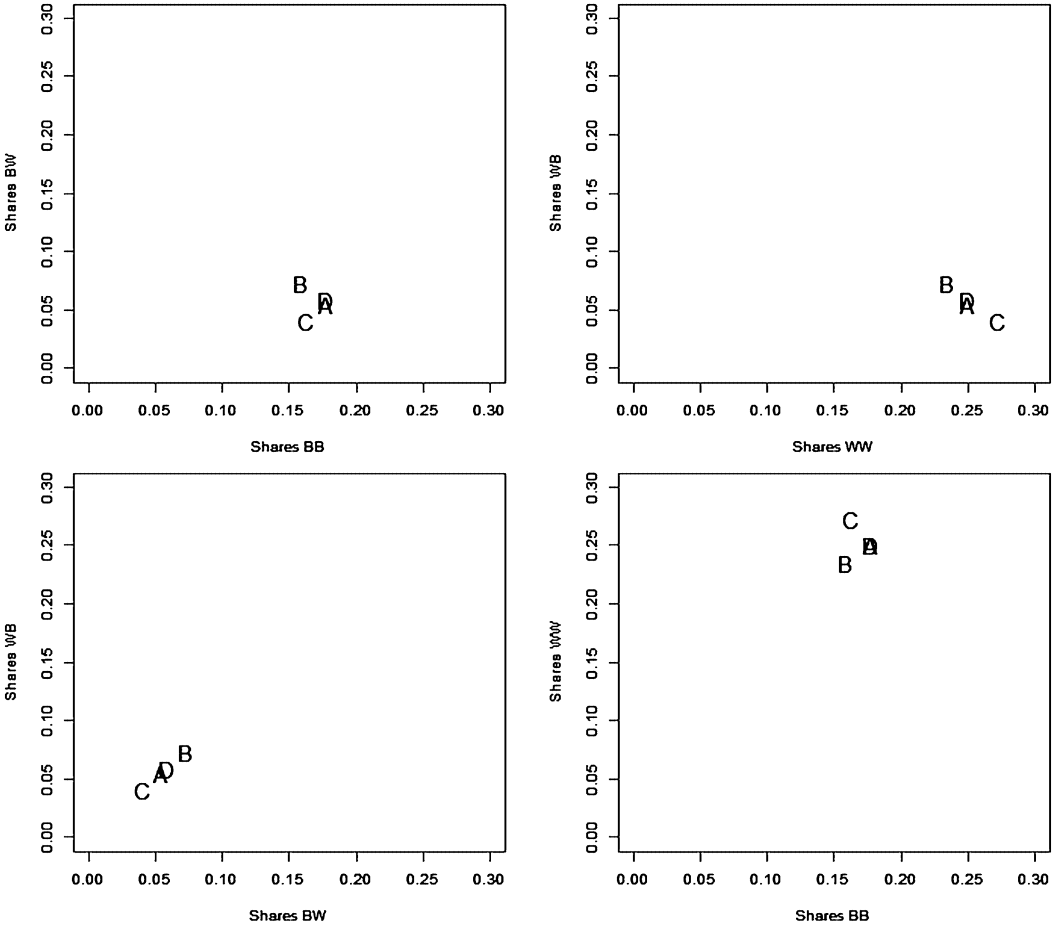

FIGURE D.3.—Type shares in four randomly selected networks (A, B, C, D). *Notes*: Letters A, B, C, D correspond to four vectors of type shares that were randomly selected from the full set of equilibrium type shares in Figure 6. Positions of the letters indicate the values of the type shares.

The order of the direct alters in the first line above is somewhat arbitrary because it is set by the convention we adopt to select a canonical representation for each network type (see Appendix D.5.1). This raises a question of how to assign the shocks for the direct connections. Rather than associate each shock with the same row of A and element of v in every type, we instead use the highest valued shocks (within each race) to compute the utility of each type. Then, for example, the utility of a type with one black friend and one white friend does not depend on whether the black alter corresponds to row 2 of A and the white alter corresponds to row 3, or vice versa. More generally, by assigning the shocks in this way, the utility of each network type does depend on the particular convention used to select the canonical representations.

D.4. *Microsimulation Procedure*

To simulate vectors of type shares to use as data, we generate equilibrium networks from which the shares can be extracted. The description below of this microsimulation

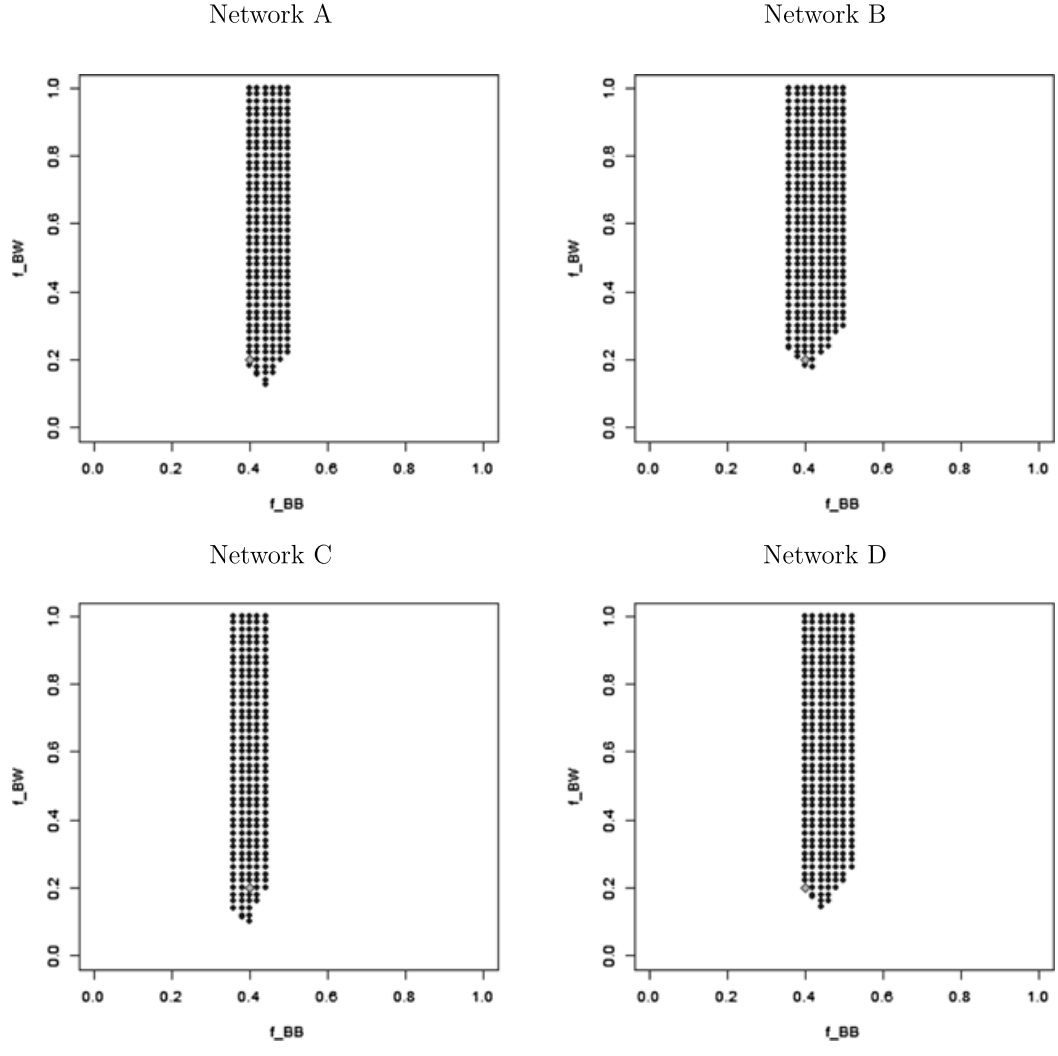

FIGURE D.4.—Identified sets from four separate networks: black preferences. *Notes*: Diamonds indicate true parameter values: $f_{BB} = 0.40$, $f_{BW} = 0.20$.

procedure is focused on the second exercise, but the overall procedure is the same for both.

Although only one network is needed for the type shares, we generate a number of pairwise stable networks to illustrate the variation that can arise in these models. In the second exercise, each network has $n = 500$ individuals, with $n_B = 100$ blacks and $n_W = 400$ whites. For each network, we first draw vectors of preference shocks for all the individuals. Then, the procedure to find a pairwise stable network starts with a random initial graph. These initial graphs are generated by independently establishing links with probability $1/(2n)$ and then removing links at random from individuals with more than L links. The success rate of $1/(2n)$ is chosen to limit the number of individuals with greater than L links in the initial draw while yielding a degree distribution that is somewhat similar to the equilibrium distribution.

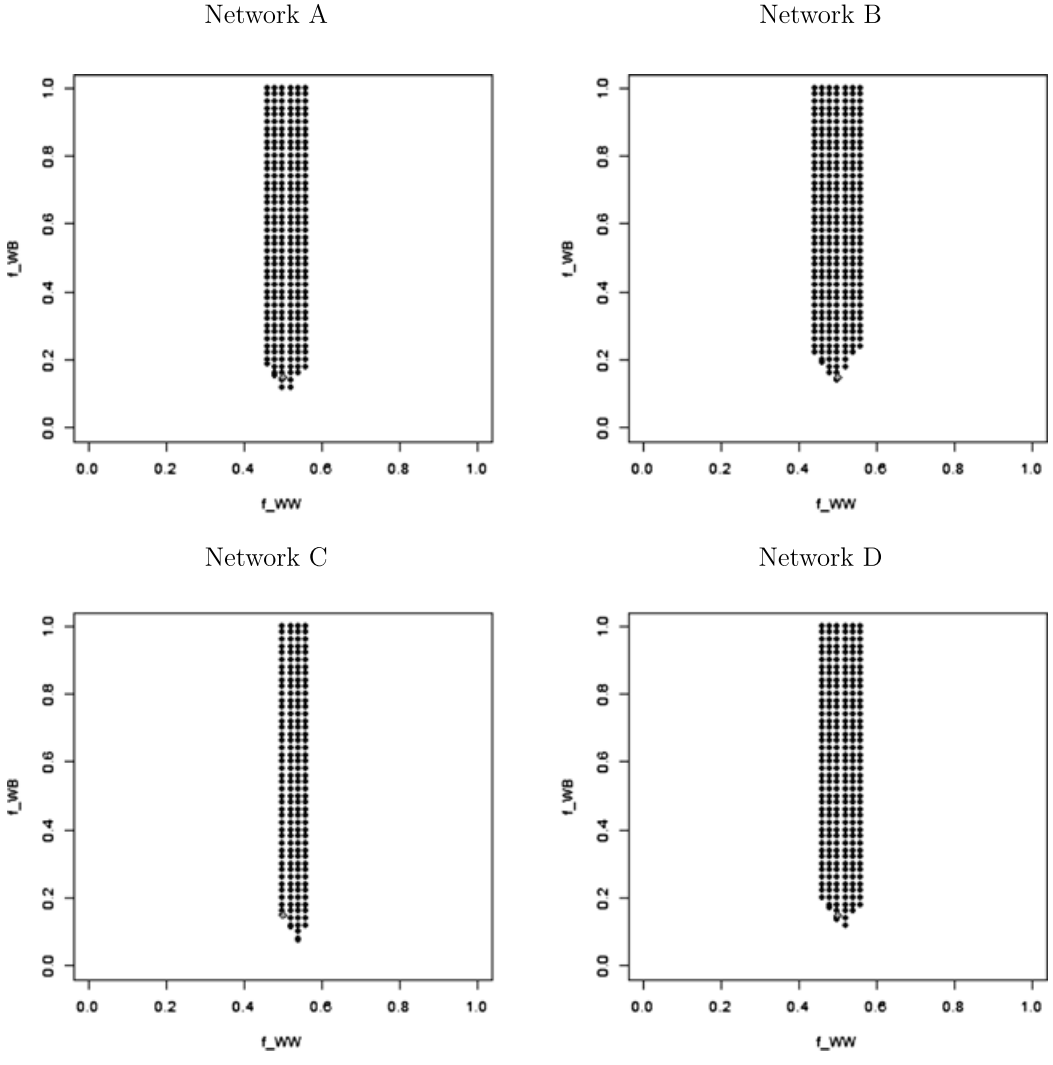

FIGURE D.5.—Identified sets from four separate networks: white preferences. *Notes*: Diamonds indicate true parameter values: $f_{WW} = 0.50$, $f_{WB} = 0.15$.

Given a random initial graph, the following sequential process is then used to find a stable network:

(i) Draw a random sequence over all unordered pairs of players (i.e., a permutation of the numbers 1 to $n(n-1)/2$, which index the pairs).

(ii) For each pair (i, j) in the sequence, myopically update $g_{ij} = g_{ji}$ based on the conditions for pairwise stability, using the network as it has evolved up to that point.

(iii) If no links or non-links were updated in an entire sequence over all the pairs, stop: the network is pairwise stable.

(iv) Otherwise, go through another random sequence of all pairs: repeat steps (i) to (iii), up to `#seqs` times (`#seqs` was set to 100).

If the network does not converge after `#seqs` of such random sequences over all pairs, a new random initial graph is used and steps (i) to (iv) are repeated. If the network still

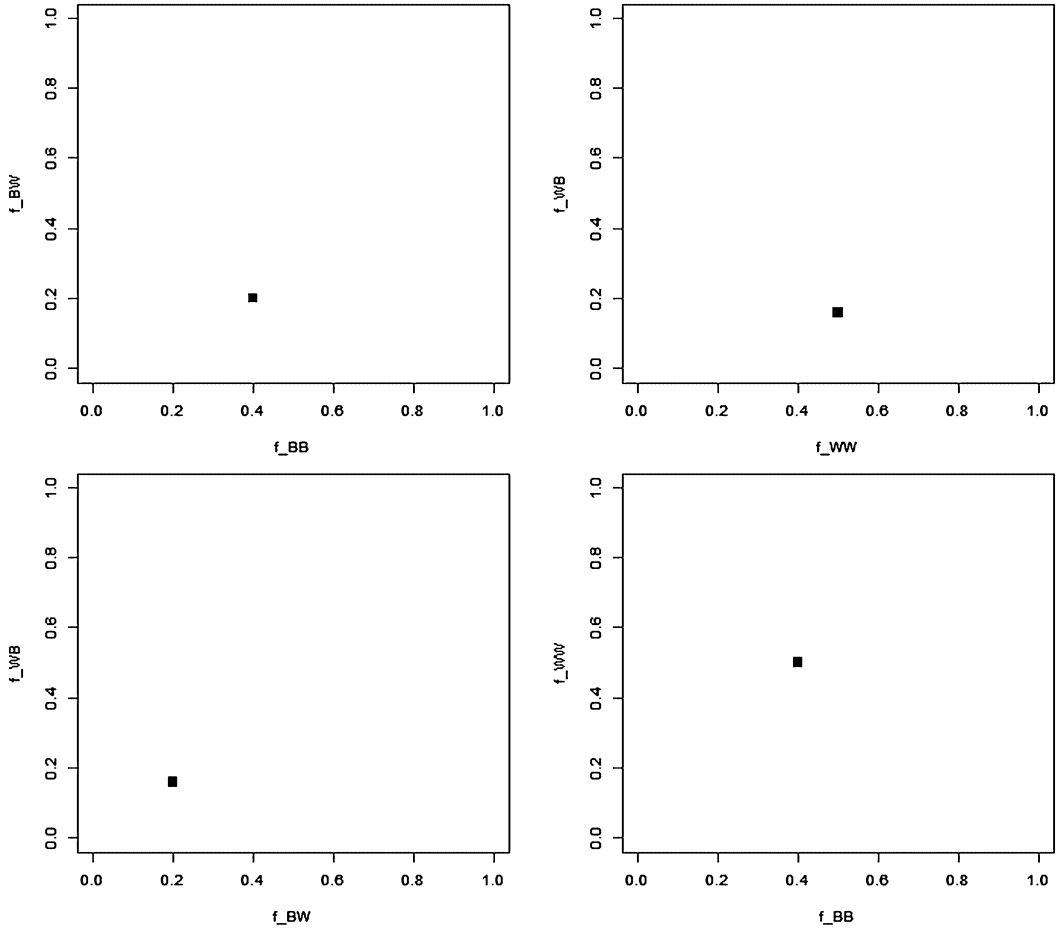

FIGURE D.6.—Identified set from four networks used jointly. *Notes*: Blocks illustrate the identified set obtained using four vectors of type shares (shown in Figure D.3) that were randomly selected from the full set of equilibrium type shares in Figure 5. A grid of parameter vectors with intervals of size 0.02 in each dimension was evaluated, and the identified set consists of one vector in this grid: $\hat{f}_{BB} = 0.40$, $\hat{f}_{BW} = 0.20$, $\hat{f}_{WB} = 0.16$, and $\hat{f}_{WW} = 0.50$. The true parameter values are: $f_{BB} = 0.40$, $f_{BW} = 0.20$, $f_{WB} = 0.15$, and $f_{WW} = 0.50$.

does not converge after this process is repeated with multiple initial graphs (up to seven), we say we have failed to find an equilibrium for this set of preference shocks and move on to draw a new set of shocks. Networks that do not converge are discarded.

We generated a total of 47 pairwise stable networks in this way (out of 50 attempts). The degree distribution from these networks appears in Figure D.7, along with the degree distribution of same-sex friendships from all schools in the Add Health data. Our simulated networks have fewer isolates and more individuals with one link, but otherwise the two distributions are broadly similar, and the average degree is the same at 1.05. Also, it turns out that the shares of network types with any mutual friends are zero in most simulated networks (see Figure 7 in the article). This is a consequence of having $\omega = \nu$ along with the values of the other parameters that were chosen to generate a degree distribution like that in Add Health. Under these parameter values, there is very low probability that three randomly selected individuals would all desire to be connected with each other in a

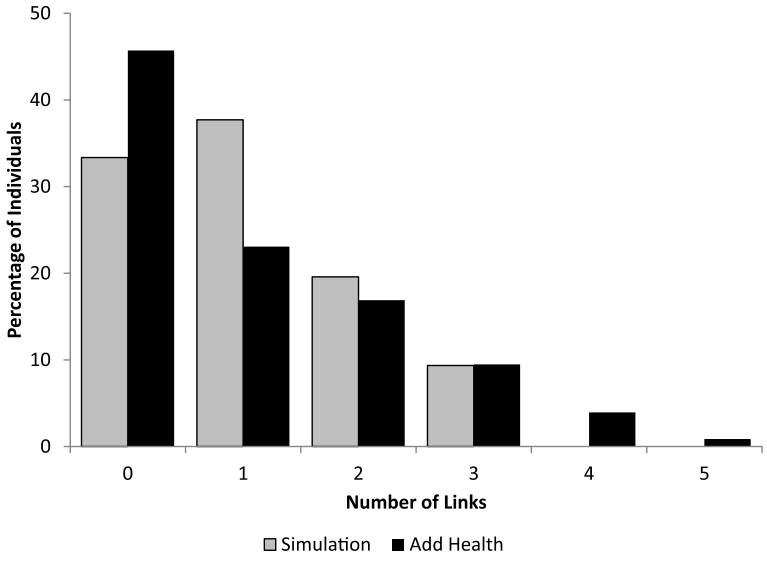

FIGURE D.7.—Degree distribution in second exercise compared with Add Health study.

triad, rather than at least one of them preferring to drop one link (thereby gaining ν while losing ω and $f_{xy} + \varepsilon_{l-1}(y)$).[40]

D.5. *Network Types and Transformations of Types*

Our approach requires an enumeration of all possible network types under a given preference structure. In addition, we define two sets of transformations on the types: what they become if one of their links is deleted, and what they become if a link is added to some other type. These transformations are needed to generate preference classes and to assess Conditions 1 and 2. None of these depend on a particular parameterization of the model, so they can be constructed prior to the recovery of the identified set. Hence, any computational burden here does not directly impact the time it takes to search through the parameter space.

D.5.1. *Enumeration of Network Types*

Each network type is an equivalence class of isomorphic subnetworks (with a root node, which is the ego). There can be multiple matrix-vector pairs (A, v) representing the same type, which are related to each other by permutation of the rows and columns for the alters.[41] So for computational convenience, we adopt a convention to single out one (A, v) pair from each class, which we refer to as the canonical representation of that network type. The enumeration of network types is then a list of these canonical representations.

[40] Given $\omega = \nu$, the marginal payoff from dropping one link in a triad of players (with no other connections) is $-(f_{xy} + \varepsilon_l(y))$. The highest value of any f_{xy} is $f_{WW} = -0.7$, and with this value the probability that a white individual would *not* want to drop such a link to the alter at position l is $\Pr(f_{WW} + \varepsilon_l(W) > 0) = 0.24$. Among three arbitrary whites, the probability that *none* of them would want to drop either one of their two links in the triad is $\Pr(f_{WW} + \varepsilon_l(W) > 0)^6 = 0.0002$. This, therefore, is the probability that a triad (with no other connections) would be pairwise stable among three randomly selected whites.

[41] In general, finding whether or not two graphs are isomorphic has an unknown computational complexity. It is known to be in *NP*, but not whether it is in *NP*-complete or *P*.

Our convention is as follows. After the first line (row/column of A or element of v), which corresponds to the individual of interest (the ego), the next L lines correspond to her direct connections (or direct alters). Then the subsequent $L-1$ lines correspond to the $L-1$ additional possible direct connections of the first direct alter, and so on. Should the ego have fewer than L links, we leave vacant lines at the end of her block. For example, if the ego only has $L-1$ links, the $L+1$ row and column of A and element of v are zero. This also applies to the blocks for any alter who does not have his full set of links. Second, if an indirect alter is reached through multiple direct alters, she appears in the block corresponding to the direct alter with the most links. Finally, an ordering over the set of characteristics $\mathcal{X}$ in the vector v fixes the permutation and selects the canonical element from the equivalence class of (A, v) pairs for this type.

Then, given such a convention, it is useful to have an automated procedure to generate the list of canonical representations. First, we generate all non-isomorphic adjacency matrices A. This is similar to generating all unlabeled graphs with up to $1+L\sum_{d=1}^{D}(L-1)^{d-1}$ nodes (10 nodes in the model with $D=2$ and $L=3$). Various algorithms for graph generation are available.[42] However, for this model, given the limitation on the number of links per node ($L=3$), it was easiest to write our own simple procedure to generate the non-isomorphic adjacency matrices. First we make all the tree structures (i.e., graphs with no cycles), then all graphs with one mutual friendship, then all with two mutual friendships, and finally the one graph with three mutual friendships. There are a total of 36 non-isomorphic adjacency matrices that are relevant under this preference structure.[43]

Finally, to construct the network types, we consider all possible combinations of characteristics of the ego and the direct alters. (The characteristics of the alters at distance 2 are not relevant under this preference structure, so they can be omitted from the vectors v.) We then compare permutations of the alter characteristics and retain only those (A, v) pairs that are unique, following our convention.[44] This yields the list of canonical representations of network types. In the model for the second exercise, there are 356 distinct network types.

D.5.2. *Link Deletion*

The construction of preference classes involves comparing the utility of each type against what would be obtained if a link were deleted. To facilitate these comparisons, we make a list containing the results of link deletion from each type. Links are easily deleted from a network type by setting the relevant elements of A to zero. We do not then need to check which canonical representation is isomorphic to the result; only the utility of the resulting type is needed. Utility in specification (1) is computed as a function of the characteristics of the direct alters, the number of friends of friends, and the number of mutual friends. These are easily extracted from any (A, v) pair regardless of the ordering of the rows and columns, and nodes that are not connected to the ego following the deletion of a link can be ignored automatically. Accordingly, the list regarding link

[42]See, for example, http://www3.cs.stonybrook.edu/~algorith/files/generating-graphs.shtml for a list with recommendations.

[43]This is considerably less than the number of unlabeled graphs among 10 nodes for three reasons. First, here the nodes have at most three links. Second, we restrict to graphs with one connected component (which contains the ego). Third, we do not consider links among nodes at distance 2 from the ego, as they are not relevant for the ego's utility.

[44]General algorithms to test for isomorphisms between graphs with node characteristics (i.e., "colors") are also available, for example, the nauty and Traces programs (http://pallini.di.uniroma1.it/).

deletion contains only the characteristics of the direct alters and the numbers of indirect and mutual friends in the resulting types. The number of elements in the list is the number of types multiplied by the number of links. In the second exercise, this is $356 \times 3 = 1{,}068$.

D.5.3. *Link Addition*

In order to construct the objective matrix Q in our QP problem, we need a mapping that gives the types which would result if individuals of two types, say t and s, were linked (given an initial distance $> 2D$). This mapping can be stored as a matrix where each row and each column corresponds to a type, and the entry at position $[t, s]$ gives the type $\bar{t}$ that would result for an individual of type t if a link were added to an individual of type s. (The matrix is $|\mathcal{T}| \times |\mathcal{T}|$.) For any cases where either t or s already has L links, this entry is blank, which makes the matrix sparse. Otherwise, the resulting type $\bar{t}$ is found by: (1) adding a link in the first unoccupied row and column for a direct alter in the adjacency matrix for type t, A_t; (2) inserting the characteristic of the ego from type s into the corresponding element of the vector of characteristics for type t, v_t; and (3) adding links to indicate any direct alters in type s into the appropriate unoccupied rows for *indirect* alters in A_t. This yields an adjacency matrix and vector of characteristics representing the new type $\bar{t}$. The resulting (A, v) pair may not be the canonical representation of that type, however, so we apply an algorithm to test for graph isomorphisms to find the matching element within the list of canonical representations. We wrote our own simple algorithm, which considers certain permutations of A and v, but more general algorithms could be used.[45]

D.6. *Specification of the QP Problem*

In order to accommodate error in the type shares estimated from a finite sample, we modify QP problem (3) to allow the predicted type shares to be within fixed bands around the observed shares. To do this, we define two slack variables for each type share, one for a positive difference, $\beta^+(t)$, and one for a negative difference, $\beta^-(t)$. The constraints for matching predicted shares to observed shares then become

$$\frac{1}{\mu}\sum_H \mu_{v_1(t)} P_{H|v_1(t)}(\theta)\alpha_H(t) + \beta^+(t) - \beta^-(t) = \pi_t, \quad \forall t.$$

The slacks above are additional variables in the modified QP problem (although the objective function is unchanged). Their magnitudes are limited based on functions of the sample size n, denoted $\delta^+(n)$ and $\delta^-(n)$, by the following additional constraints:

$$0 \le \beta^+(t) \le \delta^+(n) \quad \text{and} \quad 0 \le \beta^-(t) \le \delta^-(n), \quad \forall t.$$

Thus, the slacks define fixed bands around the observed type shares, allowing errors from $-\delta^-(n)$ to $+\delta^+(n)$ for each type.

The slacks are not minimized in the modified QP problem (e.g., by including their sum of squares in the objective function) because we found that doing so would add greatly

[45]See footnote 44 for references. We only need to consider a limited number of permutations of A and v because the canonical representations always place the ego in the first row, the direct alters in the next L rows, and the indirect alters in specific rows based on the direct alter through which they are reached. Only permutations among blocks for direct alters and for indirect alters need to be checked.

to the solution time. However, the sums of the slacks ($\sum_t \beta^+(t)$ and $\sum_t \beta^-(t)$) are further constrained with an upper bound. Without this, the total absolute error between the predicted and observed type shares could be equal to $\delta^+(n)$ or $\delta^-(n)$ multiplied by the number of types. Given the number of types in the second exercise, even small amounts for these errors, like 0.01, could then result in a large total absolute error—greater than 1, for example, which would be the total absolute error if we just predicted each type share to be equal to zero. Hence, we include the constraints

$$\sum_t \beta^+(t) \leq \Delta^+ \quad \text{and} \quad \sum_t \beta^-(t) \leq \Delta^-.$$

The Δ^+ and Δ^- above, along with the $\delta^+(n)$ and $\delta^-(n)$, can be thought of as tuning parameters. The specific values we choose are described in Appendix D.7.3.

The exact formulation of the QP problem used in our simulations is then as follows:

$$\min_{\{\alpha_H(t):t\in H, H\in\mathcal{H}(\theta)\},\beta^+,\beta^-} \alpha^\top Q(\theta)\alpha \quad \text{subject to:}$$

$$\frac{1}{\mu}\sum_{H\in\mathcal{H}(\theta)} \mu_{v_1(t)} P_{H|v_1(t)}(\theta)\alpha_H(t) + \beta^+(t) - \beta^-(t) = \pi_t, \quad \forall t, \tag{9}$$

$$\sum_{t\in H} \alpha_H(t) = 1, \quad \forall H \in \mathcal{H}(\theta); \qquad 0 \leq \alpha_H(t) \leq 1, \tag{10}$$

$$0 \leq \beta^+(t) \leq \delta^+(n), \qquad 0 \leq \beta^-(t) \leq \delta^-(n), \quad \forall t, \tag{11}$$

$$\sum_t \beta^+(t) \leq \Delta^+, \qquad \sum_t \beta^-(t) \leq \Delta^-. \tag{12}$$

The dependence of the objective matrix and the set of preference classes on the structural parameter vector (i.e., $Q(\theta)$ and $\mathcal{H}(\theta)$) is a further aspect of our implementation, discussed in the next section.

D.7. *Evaluation of a Parameter Vector*

Given a candidate vector of preference parameters θ, we wish to solve the QP problem above to determine whether the optimal value is zero. There are three main steps in this process: (1) finding the distribution of preference classes, (2) constructing the objective matrix for the QP problem, and (3) solving the QP problem.

D.7.1. *Distribution of Preference Classes*

The probability distribution of preference classes is approximated by Monte Carlo integration with independent draws of the preference shock vectors (we used 10,000 draws). For each draw ε_i, we find the preference class of a black individual and a white individual with those particular shocks. These are the two sets of types such that $u(A, v; \varepsilon_i) \geq u(A_{-l}, v; \varepsilon_i)$, $1 \leq l \leq L$, given $v_1 = B$ (a black ego) and given $v_1 = W$ (a white ego). The number of times a particular preference class appears with different draws then approximates its true probability. Thus, we have the preference class probabilities, $P_{H|x}(\theta)$, and the set of preference classes that have appeared in this procedure, $\mathcal{H}(\theta)$. As the notation indicates, the contents of $\mathcal{H}(\theta)$ can change with θ, because many preference classes

have low probabilities of occurrence and so may not be realized even with 10,000 draws, depending on the values of θ.

To give a sense of the magnitudes of these sets, at the true parameter values we generate 249 preference classes (i.e., $|\mathcal{H}(\theta_\circ)| = 249$). The number of potentially nonzero allocation parameters (which are variables in the QP problem) is equal to the sum of the cardinalities of these preference classes: $\sum_{H\in\mathcal{H}(\theta_\circ)} |H| = 5{,}013$. These are the results when we restrict to network types that are either observed in the data or adjacent via addition or deletion of a link (all other types can be ignored). If we do not remove the unobserved and non-adjacent types (216 out of the 356 types in this model), we would generate 278 preference classes with a total of 12,812 potentially nonzero allocation parameters.

D.7.2. *Construction of the QP Objective Matrix*

Section 6 gives an overview of the construction of the objective matrix Q. Here, we provide some additional detail on the construction of the matrix S (the precursor to Q). Each row in S corresponds to an allocation parameter, as does each column. The row for parameter $\alpha_H(t)$ indicates which allocation parameters (in the columns) correspond to individuals of types that someone of type t with preferences in class H would like to add a link to. More specifically, the entries of S are defined as $S_{[\alpha_H(t),\alpha_G(s)]} = 1_{\bar{t}(s)\in H}$, where type $\bar{t}(s)$ is the type that an individual of type t (from $\alpha_H(t)$, for the row) would become if they added a link to someone of type s (from $\alpha_G(s)$, for the column) at a distance greater than $2D$.

To construct S, we first extract and store the sequence of types associated with all the allocation parameters in the columns (e.g., the type s in $\alpha_G(s)$). In practical terms, this is the concatenation of the contents of all the preference classes in $\mathcal{H}(\theta)$. We then proceed by row, as follows. Given the allocation parameter for the row, $\alpha_H(t)$, we first use the matrix defined in Section D.5.3 to find the type(s) s that someone of type t with preferences in class H would like to add a link to. As described in Section D.5.3, entry $[t, s]$ in that matrix indicates the type $\bar{t}$ that an individual of type t would become after adding a link to an individual of type s (or the entry is blank if either types t or s already have L links). Accordingly, we take row t of the matrix from Section D.5.3 and identify any entry whose value $\bar{t}$ is contained in H. The column positions of these entries then indicate the types s that someone of type t with preferences in class H would like to add a link to. Thus, we have a list of desired alter types for these individuals of type t. This list is then compared with the sequence of types from the allocation parameters in the columns (e.g., the s in $\alpha_G(s)$). In columns where there is a match, the entries of this row for allocation parameter $\alpha_H(t)$ are set to 1. Those columns correspond to allocation parameters $\alpha_G(s)$ such that $\bar{t}(s) \in H$. The other entries in this row are set to 0. This yields the desired result: $S_{[\alpha_H(t),\alpha_G(s)]} = 1_{\bar{t}(s)\in H}$. The main advantage of this approach is that the match procedure can be applied to the entire row at once, and it runs quickly even though the sequence of types from the columns of S is large.

To save memory, S is stored as a sparse binary matrix. Also, because the contents of $\mathcal{H}(\theta)$ can change with θ (see Section D.7.1), the matrices S and Q are reconstructed for each candidate parameter vector θ. While this adds a small amount of computational time (relative to the time to solve the QP), it turns out to be much better for memory usage compared with trying to maintain a fixed list of preference classes and a constant version of the matrices. As noted earlier, many preference classes have very low probabilities and do not appear in the list $\mathcal{H}(\theta)$ that is generated from a particular vector θ. A fixed matrix Q that could accommodate all preference classes found with any vector in the parameter space would be vastly larger than the matrices $Q(\theta)$ that are constructed for particular values of θ.

D.7.3. *Solution of the QP Problem*

To speed the solution of the QP problem, we first use a linear programming (LP) problem to obtain starting values for the allocation parameters, which are those that minimize the sum of absolute deviations between the observed and predicted type shares. The LP problem for this is similar to the QP problem presented in Section D.6, except for the objective function and an absence of upper bounds on the slack variables. It is specified as follows:

$$\min_{\{\alpha_H(t):t\in H,H\in\mathcal{H}(\theta)\},\beta^+,\beta^-} \sum_t \big(\beta^+(t)+\beta^-(t)\big) \quad \text{subject to:}$$

$$\frac{1}{\mu}\sum_{H\in\mathcal{H}(\theta)} \mu_{v_1(t)}P_{H|v_1(t)}(\theta)\alpha_H(t)+\beta^+(t)-\beta^-(t)=\pi_t, \quad \forall t,$$

$$\sum_{t\in H}\alpha_H(t)=1, \quad \forall H\in\mathcal{H}(\theta); \qquad 0\le\alpha_H(t)\le 1,$$

$$0\le\beta^+(t),\beta^-(t), \quad \forall t.$$

The solution to this problem provides a vector of allocation parameters and slack variables that are used as starting values in the QP problem. Also, the sums of the slacks in the solution here are used to define the limits Δ^+ and Δ^- in constraint (12) of the QP problem. Specifically, we set the values of Δ^+ and Δ^- equal to $\max\{\frac{1}{2}\sum_t[b^+(t)+b^-(t)], 6/n\}$, where $b^+(t)$ and $b^-(t)$ are the slacks in the solution to the LP problem. This limits the sum of absolute errors in the QP problem to the (optimal) sum of absolute errors from the LP problem, but with a floor of $6/n$. The floor is required in order to maintain some minimal size for the bands around the observed type shares. Last, for constraint (11) of the QP problem, we set $\delta^+(n)=2/n$ and $\delta^-(n)=1/(2n)$. These are roughly the amounts required in order for the solver to converge easily when we use the true parameter values and the observed type shares from the one randomly selected network. These values and the floor of $6/n$ in the formula for Δ^+ and Δ^- function as tuning parameters, which can be adjusted based on the performance of the solver.

To solve the QP problem, we use the active set algorithm in the program KNITRO, which is a variant of a sequential linear and quadratic programming optimization method (Byrd, Gould, Nocedal, and Waltz (2003)). As detailed below, this routine performs well on our problem. Over a range of values for the preference parameters, which yield on the order of 2,000 to 10,000 allocation parameters, the solution time averages less than 25 seconds.

D.8. *Construction of the Identified Set*

In concept, the identified set is a level set in the space of structural parameters, where the optimal value of the QP is zero and the predicted type shares match the observed type shares. Our approach to find this level set involves the use of Markov Chain Monte Carlo (MCMC) procedures. The results from the solution of the QP problem for a given parameter vector are converted into a pseudo-density, which an MCMC algorithm can then use to draw parameter vectors and move through the parameter space.[46]

[46]For the results shown here, we used both the Metropolis–Hastings and slice sampler algorithms in Matlab.

We use a log pseudo-density that is proportional to $-[\alpha^*(\theta)^\top Q(\theta)\alpha^*(\theta) + \beta^*(\theta)^\top\beta^*(\theta)]$, where $(\alpha^*(\theta), \beta^*(\theta))$ denotes a solution to the QP problem for θ. This allows for small positive values of the objective function as well as the errors between observed and predicted type shares. We found this to be helpful to address issues in computational precision, along with the sampling error in the observed type shares. Structural parameter vectors where the value of this pseudo-density is at least 95% of its maximum are then considered to be in the identified set and hence are shown in the figures.[47] For the results plotted in Figure 8 in the article, we generated a total of 7,090 such vectors.

D.9. *Computational Performance*

The above procedures to construct the identified set were run on machines with Intel® Xeon® 5160 processors (3 GHz base frequency) and 16 GB of physical memory. Computations were not parallelized, except in the "embarrassingly" simple sense that multiple Markov chains were run on different machines. The time required to evaluate a single parameter vector θ consists mainly of three steps: generating the preference class distribution, constructing the objective matrix, and solving the QP problem (relative to these, the time to solve the preliminary LP problem is trivial). On average, the first two steps each account for only 10% of the total compute time, so the majority of the computational burden comes from the solution of the QP (i.e., 80% of the compute time).

Based on evaluations of 15,000 structural parameter vectors in total, the average time to evaluate a single parameter vector (i.e., to generate the pseudo-density for a given θ) was 29.8 seconds. The number of allocation parameters in the QP problems for these parameter vectors ranged roughly from 2,000 to 10,000, with an average of 5,955.3.

REFERENCES

BYRD, R. H., N. I. M. GOULD, J. NOCEDAL, AND R. A. WALTZ (2003): "An Algorithm for Nonlinear Optimization Using Linear Programming and Equality Constrained Subproblems," *Mathematical Programming*, 100 (1), 27–48. [21]

COCHRAN, W. G. (1977): *Sampling Techniques*. New York: Wiley. [6,7]

CURRARINI, S., M. JACKSON, AND P. PIN (2009): "An Economic Model of Friendship: Homophily, Minorities and Segregation," *Econometrica*, 77 (4), 1003–1045. [2]

HÁJEK, J. (1960): "Limiting Distributions in Simple Random Sampling From a Finite Population," *Publications of the Mathematics Institute of the Hungarian Academy of Science*, 5, 361–374. [7]

IMBENS, G. W., AND D. B. RUBIN (2015): *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge: Cambridge University Press. [6]

JACKSON, M. O., AND A. WOLINSKY (1996): "A Strategic Model of Social and Economic Networks," *Journal of Economic Theory*, 71 (1), 44–74. [2]

KOLACZYK, E. (2009): *Statistical Analysis of Network Data*. New York: Springer. [6]

LEUNG, M. (2015): "A Random-Field Approach to Inference in Large Models of Network Formation," Working Paper, Stanford University. [9]

VAN DER VAART, A. W. (1998): *Asymptotic Statistics*. Cambridge: Cambridge University Press. [8]

Co-editor Liran Einav handled this manuscript.

Manuscript received 23 June, 2015; final version accepted 3 August, 2017; available online 21 August, 2017.

[47]The 95% threshold is somewhat arbitrary, but the results are robust to the value that is used. Relatively few parameter vectors produce pseudo-densities that are between 90% and 99% of the maximum value, for example.