Web Appendix

Targeting Revenue Leaders for a New Product

Michael Haenlein & Barak Libai

Web Appendix I: Verification and Validation of the Agent-based Model

In a recent review, Rand and Rust (2011) provide structured guidelines and motivations for the use of agent-based modeling (ABM) in a marketing context. ABM is especially suitable for situations that focus on interactions among many agents and where simple rules of behavior give rise to complex emergent patterns that are often difficult to track with straightforward mathematical modeling. This makes ABM an essential tool for the analysis of processes like the diffusion of innovations, which are largely driven by customer-to-customer interactions such as word of mouth and observational learning. Specifically, Rand and Rust (2011) emphasize two factors that should be considered when developing an agent-based model: *validation* and *verification*. In what follows we briefly explain how the approach presented in our study is consistent with their guidelines on these two matters.

Validation

At the micro level, our model, like the central example in Rand and Rust's (2011) paper, is a contagion model that follows the diffusion-of-innovation paradigm, which has been validated by numerous studies over the years. This supports our use of parameters of internal and external influence, as well as our assumption that agents possess a local social network that does not allow them to discover information about the entire population. In addition, the seeding process we model is consistent with practice both in its nature (getting the product to consumers randomly or to opinion leaders with launch) and in its extent, as a percentage of the population. At the macro level, the aggregated patterns of diffusion created by our model appear to suggest typical innovation adoption patterns of a sigmoid curve, in which at first a few adopt, and then more and more until the entire population has adopted.

At the empirical input level, the range of parameters for δ and q (external and internal influence, respectively) is consistent with previous cascade modeling research, which draws on the Bass model. While the value of δ is in the range of the corresponding parameter used in the aggregate Bass model, there is a need to adapt the q parameter to the size of the network. This adaptation has been done in the past (e.g., Goldenberg, Libai, and Muller 2002), where the parameters are set to achieve curves similar in magnitude to the empirical observations of the Bass model research. Recent research has indeed pointed to the close relationship between the individual-level cascade models and the aggregate, empirically supported, Bass model diffusion approaches (Fibich and Gibori 2010). See Web Appendix II for more details on the parameter range used in our simulation.

In addition, we constructed a variety of networks based on the Jackson Rogers (2007) algorithm. This algorithm was formulated to duplicate the processes that occur in the creation of real social network structures and has been validated by the authors. The Jackson Rogers social network structure is more realistic than models in which every consumer knows every other consumer. Because we examine a range of social network structures, we are able to cover a variety of social system scenarios.

In terms of empirical output, thus far there have been no published empirical studies that looked at the social value of revenue leaders. We hope that this work will generate more interest in this subject.

Verification

As outlined by Rand and Rust (2011), model verification deals with assessing the match between the implemented and conceptual model and consists of three steps: documentation, programmatic testing and the analysis of test cases.

In terms of documentation, the description of the agent-based model within the main body of our manuscript serves as the documentation of the conceptual model. Within Web Appendix VII we provide a description of the implemented model through the full R-code used to run our agent-based model simulation. This R-Code has been used within the R Computing environment (Version 2.15.1, http://cran.r-project.org/) in combination with the igraph package (Version 0.6-2) to generate all results presented in this paper. To ensure that the implemented and conceptual model are completely aligned, programming was carried out by one of the authors.

Regarding programmatic testing, we performed a series of sanity checks to ensure that our model is working as expected. We tested each routine of the code separately to ensure its functioning (unit testing). Since several of our methods involve social networks, which are complicated by nature, some of these tests were conducted on a simplified network of 10 members, with homogeneous ties and influence parameters. Once each function had been validated by itself, they were combined together in a step-wise fashion and verified again. This stepwise model build-up reduces the growth in model complexity which facilitates verification. By its nature this testing process required several walkthroughs to ensure that the code is generating correct results and its logic expressed the concept underlying the conceptual model. Note that since the code was programmed by one of the authors (vs. by an outside programmer) code walkthroughs and debugging walkthroughs essentially collapsed to the same actions.

Finally, we verified the proper function of the code using a series of test cases, especially when changes were applied to the code or new models were added. This included an analysis of corner cases (e.g., {p, q}={0, 1} and {p, q}={1, 0}) and sampled cases (i.e., randomly selected combinations of design parameters, i.e., network clustering coefficient, CLV assortativity, CLV SD and seeding percentage). Both helped to ensure that the code did not exhibit any aberrant behavior (e.g., negative adoption rates, lack of full adoption despite p>0). Finally we used relative value testing for parameters such as internal/ external influence, network size and number of periods to test whether changing one value while keeping the others constant impacted results in the expected direction.

Web Appendix II: The Diffusion Model

Like the Bass model, our model includes two states for consumers: adopters and non-adopters. Two factors impact the adoption decision of an individual: an external factor, representing the probability of being influenced by advertising, mass media, or other marketing efforts, and an internal factor, representing the probability of being influenced by a social interaction (e.g., word of mouth) with another individual who has already adopted the product. Formally, the probability that an individual *i* will adopt the product in period *t* (contingent on not having adopted it prior to *t*) can be determined as:

(W1) $$p_i(t) = 1 - (1-\delta_i)\,(1-q_i)^{Ni(t)},$$

where δ_i is the probability that actor *i* adopts the product due to external factors (external influence parameter); q_i is the probability that actor *i* adopts the product due to an interaction with one other individual who has already adopted the product (internal influence parameter); and *Ni(t)* is the number of individuals in *i's* personal network who have already adopted the product prior to *t*. In addition to being well accepted (Leskovec, Adamic, and Huberman 2007; Zubcsek and Sarvary 2011), an advantage of the cascade approach is that it follows an established tradition in marketing, which also allows us to build on past research when setting up model parameters. The diffusion process dynamics adhere to the following pattern:

- **Period 0**: This is the initial condition; generally, consumers have not yet adopted the product (that is, the adoption state of all network members is 0). Only the seeded group of customers receives the value of 1.

- **Period 1:** The probabilities for each consumer $p_i(t)$ are realized. An individual member can obtain the activation value of 1 through the combination of external influence and internal influence from the seeded people, if s/he is connected to any of them. To realize a given probability, a random number *U* is drawn from a uniform distribution in the range [0,1]. If $U< p_i(t)$ (based on the algorithm above), then the consumer moves from non-adopter to adopter (receiving the value of 1). Otherwise the consumer remains a non-adopter.

- **Period *n***: The process continues as gradually more non-adopters obtain the value of 1.

- **Period 30**: The process ends. By that time, on average, more than 90% have become adopters, and in addition, due to the discount rate, very little net present value (NPV) is left. The program calculates the customer equity, i.e., the NPV given the number of adopters and when each one adopted.

One should note that 30 periods may seem extensive given the discount rate, and possibly the retention rate of customers. However, there are at least three reasons to consider a longer period. First, while the monetary value in the later years is low, one may still want to capture it to make sure the results are unbiased. This perspective is equally consistent with the long term view of CLV, that sometimes uses infinite horizon, even when the early years are those who matter more. Second, because we consider a diffusion growth process, some customers enter later. Since we simulate different diffusion processes, there may be some processes that are characterized by a relatively longer "left tail" where the mass adoption starts later. In such cases, one would want to use a longer horizon to make sure the monetary effect is well captured (although the discount rate will indeed limit the monetary effect of such processes). Finally, a duration of 30 periods is largely consistent with other agent-based models that considered the monetary effect of diffusion processes (Goldenberg, Libai, and Muller 2010; Libai, Muller, and Peres 2012) and used this longer range for the same reasons. We have also re-run the program to check the case of 15 periods and found no substantial difference in the results.

Parameter ranges

When deciding on the ranges of parameters δ and q for ABM cascade models, a fundamental criterion is to choose parameters that will create growth processes of the type observed for real products. It has been suggested that while the external influence parameter δ is in the range of the aggregate-level parameter of external influence for the Bass model, the individual level parameter will vary and depends on the specific social network (Goldenberg et al. 2007). Here, the ranges we use for δ (0.001-0.02) and q (0.04-0.16) match those used by Libai, Muller and Peres (2010), who looked at seeding in a network of similar size. To allow heterogeneity in δ and q we draw different parameters for each actor from a uniform distribution in the corresponding range.

One way to see the connection between the individual level and the aggregate level approach is as follows (Goldenberg et al. 2007): If one takes the average internal influence parameter (0.1) and the average number of ties of an individual in our simulations (5.7) and assumes that about half of the social network has adopted (middle of the diffusion process), then the average probability that a given person will be affected by internal influence is $(1-(1-0.1)^{5.7})=0.45$, a number that is comparable with findings on the average value of the aggregate parameter for internal influence in market level diffusion processes (Sultan, Farley, and Lehmann 1990).

To further explore the effect of parameter range, we repeated our analysis using the δ and *q* parameters used by Goldenberg, Libai and Muller (2001). Thus, in the new analysis, *q* was sampled in the range of 0.01-0.07, and δ was sampled in the range 0.0005-0.01. The fundamental results of this analysis are presented in Table WA2A below. These results are similar to those presented in Table 2 in terms of their direction and the relative sizes of the direct and social value created by the different seeding targets.

Table WA2A:

Values using diffusion parameters from Goldenberg, Libai, and Muller (2001)

	Opinion leader			Revenue leader			Random		
Seed size	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**
0.50%	69.5	5.8	**75.3**	33.8	31.6	**65.3**	24.2	8.1	**32.4**
1.00%	43.8	5.7	**49.6**	26.0	30.7	**56.7**	20.6	8.2	**28.7**
1.50%	33.6	5.6	**39.1**	22.9	29.9	**52.8**	17.9	8.1	**26.0**
2.00%	27.3	5.4	**32.8**	20.2	29.7	**49.8**	16.7	8.0	**24.7**
2.50%	23.1	5.4	**28.6**	17.8	29.8	**47.6**	15.1	7.9	**23.0**
3.00%	20.3	5.3	**25.7**	16.4	29.6	**46.0**	14.4	7.8	**22.2**
3.50%	17.5	5.4	**22.9**	14.6	29.8	**44.4**	13.0	8.0	**21.0**
4.00%	15.8	5.4	**21.3**	13.4	29.6	**43.0**	12.4	8.0	**20.4**
Average	**31.4**	**5.5**	**36.9**	**20.6**	**30.1**	**50.7**	**16.8**	**8.0**	**24.8**

Customer Profitability in the diffusion process

The profit that a firm derives from customers who adopt a new product varies according to the type of product and may be distributed heterogeneously across customers. For durable products the profit of a given customer equals the profit obtained from a single purchase. For services and repeat-purchase products, the profitability is the expected lifetime value of each customer at the time of adoption, which is further discounted to the time of the product introduction (Gupta, Lehmann, and Stuart 2004).

Web Appendix III: CLV in the European Cellular Case

Our CLV calculation in the European Cellular dataset follows the conventional calculation of customer lifetime value for cases of *lost for good* retention, that sums the discounted cash flows from the surviving customers at each period (Gupta et al. 2006; Gupta, Lehmann, and Stuart 2004). Here we combined data we had on previous purchases, with expected purchases, using the average retention rate among customers to determine the calculation horizon. Specifically, for each actor in our dataset we estimated customer lifetime value as the sum of discounted known revenue and discounted expected revenue:

(W2) *CLV = Discounted known revenue + Discounted expected revenue*

Discounted known revenue**:** To calculate discounted pasted revenue, we obtained historical monthly revenue per user for a 26-month period (from May 2004 to June 2006). For customers who had been acquired less than 26 months previously (i.e., after May 2004), we used monthly revenue since the date of acquisition. We then applied an annual discount rate of 10%

(corresponding to a monthly discount rate of approximately 0.80%) to discount this past revenue up to the end of the 26-month period (i.e., June 2006). Discounted known revenue was subsequently defined as the sum of all discounted past revenue. This leads to the following expression:

(W3) $$Discounted\ past\ revenue = \sum_{i=1}^{l_j} z_i (1+dm)^{i-1}, \text{ with}$$

(W4) $$dm = (1+d)^{\frac{1}{12}} - 1,$$

where l_j is the past lifetime of customer j in months (i.e., either 26 or the number of months elapsed between acquisition and June 2006), z_i is the revenue generated by customer j in month i (counting backwards from June 2006, so i=1 represents June 2006, i=2 May 2006 and i=26 May 2004) and d is the annual discount rate (10% in our case).

Discounted expected future revenue: Consistent with the overall churn rate of the company during the time of data collection (approximately 13.6%), we assumed a total lifetime of eight years, or 96 months, for each actor. We determined the expected future lifetime for each user at the end of the 26 month period (i.e., June 2006) in the following way: For each customer acquired less than 26 months previously (i.e., after May 2004), we calculated the months elapsed since acquisition and subtracted this number from the total expected lifetime of 96 months. For all customers acquired more than 26 months previously we assumed an expected future lifetime of 70 months. Using this expected future lifetime and the average past revenue per user, we calculated the present value of expected future revenue over the remaining lifetime. Again, a discount rate of 10% annually was applied in this case. This results in the following expression:

(W5) $$Discounted\ expected\ future\ revenue = \bar{z}\sum_{i=1}^{96-l_j} q^i = \bar{z}\frac{q-q^{(96-l_j)+1}}{1-q}, \text{ with}$$

(W6) $q = \frac{1}{1+dm}$ and

(W7) $\bar{z} = \frac{1}{l_j}\sum_{i=1}^{l_j} z_i$

Combining equations W2, W3 and W5 leads to the following overall expression for CLV:

(W8) $$CLV = \sum_{i=1}^{l_j} z_i(1+dm)^{i-1} + \bar{z}\frac{q-q^{(96-l_j)+1}}{1-q}$$

In practice, there are various approaches for CLV forecasting (e.g., Rust, Kumar, and Venkatesan 2011). Note that our focus in this example is not on forecasting, but on the comparison of social value creation via the different seeding approaches. Given that, we our focus is on an empirical distribution of CLV rather than its prediction. The CLV approach we use is the indeed the fundamental one, yet it is well used in assessments of customer lifetime value, in particular when limited information is available (Gupta et al. 2006; Gupta, Lehmann, and Stuart 2004). However, for predictive purposes they are more sophisticated approaches that take advantage of rich customer data. See Rust, Kumar, and Venkatesan (2011) for a review and an example of an advanced predictive approach.

Web Appendix IV: The Agent-based Model Code

Generate random graph using Jackson-Rogers algorithm

```
jackson.rogers.game <- function(nodes) {

    ## Step 1: Set parameter values

        m               <- runif(1, min=1.00, max=5.00)
        r               <- runif(1, min=0.50, max=5.00)
        prob            <- runif(1, min=0.10, max=0.50)
        pR              <- prob
        pS              <- prob
        mR              <- ceiling(r*m/(r+1))
        mS              <- m - mR
        m0              <- round(runif(1, min=m+1, max=0.20*nodes),0)
        T               <- nodes
        mInit           <- ceiling(runif(1, min=0, max=0.5*(m0-1)))
        ForceInDegree   <- 0
        defaultDeg      <- 0


    ## Step 2: Define function randSample, which generates a random sample of size k from integers 1,...,n w/o replacement

        randSample <- function (n,k) {
            if (k <= n) {sample <- sample(c(1:n),k,replace=FALSE)}
            else {sample <- 0}
            sample}


    ## Step 3: Generate networks with given number of nodes

        total_nodes <- 1
        T           <- T-(nodes-1)
        while (abs(nodes/total_nodes - 1) > 0.05) {
            T <- T + (nodes-total_nodes)


    ## Step 3.1: Define initial connections

            nEdges              <- 0
            DirectedOutLinks    <- vector("list",m0+T)
            DirectedInLinks     <- vector("list",m0+T)
            inDegrees           <- rep(0,m0 + T)
            for (i in 1:m0) {
                dorepeat <- 1
                while (dorepeat==1) {
                    cons <- randSample(m0,mInit)
                    if(i%in%cons) {dorepeat <- 1}
                    else {dorepeat <- 0}}
                DirectedOutLinks[[i]] <- cons
                for (j in 1:length(cons)) {
                    DirectedInLinks[[cons[j]]]      <- append(DirectedInLinks[[cons[j]]],i)
                    inDegrees[cons[j]]              <- inDegrees[cons[j]]+1
                    nEdges                          <- nEdges+1}}
            for (i in (m0+1):(m0+T)) {inDegrees[i] <- defaultDeg}
```

Step 3.2: Make connections

```
for (t in (m0+1):(m0+T)) {
    parents <- randSample(t-1,mR)
    nodesToSearch <- NULL
    for (p in 1:mR) {
        curPHood <- DirectedOutLinks[[parents[p]]]
        nodesToSearch <- sort(append(nodesToSearch,curPHood))
        if (runif(1,0,1)<= pR) {
            DirectedOutLinks[[t]] <- append(DirectedOutLinks[[t]],parents[p])
            DirectedInLinks[[parents[p]]] <- append(DirectedInLinks[[parents[p]]],t)
            inDegrees[parents[p]] <- inDegrees[parents[p]]+1
            nEdges <- nEdges+1}
        if (p <= ForceInDegree) {
            DirectedOutLinks[[parents[p]]] <- append(DirectedOutLinks[[parents[p]]], t)
            DirectedInLinks[[t]] <- append(DirectedInLinks[[t]], parents[p])
            inDegrees[t] <- inDegrees[t]+1
            nEdges <- nEdges+1}}
    searchNodePositions <- randSample(length(nodesToSearch), mS)
    for (j in 1:mS) {
        if(runif(1,0,1) <= pS & length(searchNodePositions) == mS) {
            DirectedOutLinks[[t]] <- append(DirectedOutLinks[[t]],nodesToSearch[searchNodePositions[j]])
            DirectedInLinks[[nodesToSearch[searchNodePositions[j]]]] <- append(DirectedInLinks[[nodesToSearch[searchNodePositions[j]]]],t)
            inDegrees[nodesToSearch[searchNodePositions[j]]] <- inDegrees[nodesToSearch[searchNodePositions[j]]]+1
            nEdges <- nEdges+1}}}
```

Step 3.3: Convert DirectedOutLinks and DirectedInLinks into an adjacency matrix and define graph object

```
    mat_out <- matrix(0, length(DirectedOutLinks), length(DirectedOutLinks))
    for (i in seq(along = DirectedOutLinks)) mat_out[i, as.numeric(DirectedOutLinks[[i]])] <- 1
    graph_directed <- graph.adjacency(mat_out)
    graph_1 <- as.undirected(graph_directed)
    graph_2 <- simplify(graph_1)
    matrix <- get.adjacency(graph_2)
    graph_3 <- graph.adjacency(matrix, mode="undirected")
    graph_4 <- delete.vertices(graph_3, V(graph_3)[degree(graph_3)==0])
    total_nodes <- length(V(graph_4))}
graph_4}
```

Agent-based model

```
library(igraph)
```

Step 1: Define parameter values

```
target_nodes        <- 1000
log_normal_mean     <- 0
target_rho_error    <- 0.01
periods             <- 30
discount_rate       <- 0.10
repetitions         <- 20
runs                <- 3
max_counter         <- 100000
ext_delta_min       <- 0.001
ext_delta_max       <- 0.020
int_q_min           <- 0.040
int_q_max           <- 0.160
leader_percentage   <- 0.10
```

Step 2: Define functions: adoption_process, fastcor and do_adoption

```
adoption_process    <- function(seed) {
        time_period         <- 0
        adoption            <- rep(999, nodes)
        while (time_period < periods) {
                adoption[adoption==999 & seed == 1] <- time_period
                if(sum(seed)==nodes) break
                time_period         <- time_period+1
                connected_seeds     <- (seed %*% m)[1,]
                adoption_prob       <- 1-(1-delta_vec)*(1-q_vec)^connected_seeds
                Rnd_vec             <- runif(nodes, min=0, max=1)
                seed                <- sign(seed + (adoption_prob > Rnd_vec))}
        adoption}

fastcor <- function (x, y) .Internal(cor(x, y, 4L, FALSE))

do_adoption <- function(seed_to_use) {
        adoption_base       <- adoption_process(numeric(nodes))
        adoption_time       <- adoption_process(seed_to_use)
        CLV_diff            <- z/(1+discount_rate)^adoption_time - z/(1+discount_rate)^adoption_base
        list(CLV = sum(CLV_diff*seed_to_use), Social = sum(CLV_diff*(1-seed_to_use)))}
```

Step 3: Set-up simulation space

```
sdev_space          <- c (0.10, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75)
rho_space           <- c (0.00, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70)
seed_space          <- c (0.005, 0.010, 0.015, 0.020, 0.025, 0.030, 0.035, 0.040)
cluster_space       <- c (0.07, 0.10, 0.13, 0.17, 0.23, 0.31, 0.40, 0.52)

for (log_normal_sdev in sdev_space){
for (target_rho in rho_space){
for (seed_percent in seed_space){
for (target_clustering in cluster_space) {
```

Step 4: Set-up different runs

```
Mean_OL_CLV <- Mean_RLea_CLV <- Mean_RLoo_CLV <- Mean_Rnd_CLV <- Mean_OL_Social <- Mean_RLea_Social <- Mean_RLoo_Social <- Mean_Rnd_Social <- Mean_delta <- Mean_q <-
numeric(runs)

column_names        <- c("CLV SD", "CLV Correlation", "Avg. Clustering", "Degree Assort", "Degree Mean", "Degree SD", "CLV-Degree Correlation")
row_names           <- c("Run 1", "Run 2", "Run 3")
Summary_stat        <- matrix(rep(0,7*runs), nrow=runs, dimnames=list(row_names, column_names))
for (run_count in 1:runs) {
```

Step 5: Define network and adjust CLV distribution to reach specified correlation value

```
        rho                 <- Inf
        while(abs(rho-target_rho) > target_rho_error) {
                counter             <- 0
                actual_clustering   <- 0
                while(abs(actual_clustering - target_clustering)>0.02) {
                        sample_net_1        <- try(jackson.rogers.game(target_nodes))
                        if (as.numeric(is.igraph(sample_net_1))==1) {
                                sample_net_2        <- simplify(sample_net_1)
                                actual_clustering   <- transitivity(sample_net_2)}
                        if (as.numeric(is.igraph(sample_net_1))==0) {actual_clustering <- 0}}
        g_1     <- sample_net_1
        nodes   <- length(V(g_1))
        g       <- simplify(g_1)
        m       <- get.adjacency(g, sparse=TRUE)
        m_std   <- m/rowSums(m)
        z       <- rlnorm(nodes, meanlog = log_normal_mean, sdlog = log_normal_sdev)
        z_net   <- (m_std %*% z)[,1]
        rho     <- fastcor(z,z_net)
        while (abs(rho-target_rho) > target_rho_error && counter < max_counter) {
                for (i in 1: nodes) {
                        z_net   <- (m_std %*% z)[,1]
                        rho     <- fastcor(z,z_net)
                        rho_1   <- -2
                        while (abs(rho-target_rho) < abs(rho_1-target_rho) && abs(rho_1-target_rho) > target_rho_error && counter < max_counter) {
                                z[i]    <- rlnorm(1, meanlog = log_normal_mean, sdlog = log_normal_sdev)
                                z_net   <- (m_std %*% z)[,1]
                                rho_1   <- fastcor(z,z_net)
                                if (counter != max_counter) counter <- counter+1}}}}
```

Step 6: Calculate network and CLV statistics

```
z_nor                   <- z/mean(z)
z_net_nor               <- (m_std %*% z_nor)[,1]
net_degree              <- degree(g)
net_degree_avg          <- (m_std %*% net_degree)[,1]
z_sdev                  <- sd(z_nor)
z_table                 <- data.frame(aggregate(z_nor, by=list(degree(g)), mean))
correlation             <- fastcor(z_nor, z_net_nor)[1]
clustering              <- transitivity(g)
degree_assort           <- fastcor(net_degree, net_degree_avg)[1]
degree_mean             <- mean(degree(g))
degree_sdev             <- sd(degree(g))
CLV_Degree_cor          <- cor(z_table[,1],z_table[,2])
Summary_stat[run_count,] <- cbind(z_sdev, correlation, clustering, degree_assort, degree_mean, degree_sdev, CLV_Degree_cor)
```

Step 7: Define vectors (seed, delta, q)

```
z                       <- z_nor
degree                  <- degree(g)
random                  <- runif(nodes, min=0, max=1)
top_z                   <- sort(z)[((1-leader_percentage)*nodes):nodes]
bottom_z                <- sort(z)[1:(leader_percentage*nodes)]
top_degree              <- sort(degree)[((1-leader_percentage)*nodes):nodes]
top_random              <- sort(random)[((1-leader_percentage)*nodes):nodes]
top_z_seed_space        <- 1*(z>min(top_z))
bottom_z_seed_space     <- 1*(z<=max(bottom_z))
top_degree_seed_space   <- 1*(degree>min(top_degree))
top_random_seed_space   <- 1*(random>min(top_random))
z_seed                  <- 1*(rank(runif(nodes, min=0, max=1)*top_z_seed_space)>(nodes*(1-seed_percent)))
z_seed_looser           <- 1*(rank(runif(nodes, min=0, max=1)*bottom_z_seed_space)>(nodes*(1-seed_percent)))
degree_seed             <- 1*(rank(runif(nodes, min=0, max=1)*top_degree_seed_space)>(nodes*(1-seed_percent)))
rnd_seed                <- 1*(rank(runif(nodes, min=0, max=1)*top_random_seed_space)>(nodes*(1-seed_percent)))
delta_vec               <- runif(nodes, min=ext_delta_min, max=ext_delta_max)
q_vec                   <- runif(nodes, min=int_q_min, max=int_q_max)
```

Step 8: Calculate customer equity

```
RLea_CLV <- RLea_Social <- OL_CLV <- OL_Social <- RLoo_CLV <- RLoo_Social <- Rnd_CLV <- Rnd_Social <- numeric(repetitions)
for (rep in 1: repetitions) {
        foo             <- do_adoption(z_seed)
        RLea_CLV[rep]   <- foo$CLV
        RLea_Social[rep] <- foo$Social

        foo             <- do_adoption(degree_seed)
        OL_CLV[rep]     <- foo$CLV
        OL_Social[rep]  <- foo$Social

        foo             <- do_adoption(z_seed_looser)
        RLoo_CLV[rep]   <- foo$CLV
        RLoo_Social[rep] <- foo$Social

        foo             <- do_adoption(rnd_seed)
        Rnd_CLV[rep]    <- foo$CLV
        Rnd_Social[rep] <- foo$Social}
```

Step 9: Calculate adoption process and uplift statistics and generate output file

```
Mean_OL_CLV[run_count]       <- mean(OL_CLV)
Mean_OL_Social[run_count]    <- mean(OL_Social)
Mean_RLea_CLV[run_count]     <- mean(RLea_CLV)
Mean_RLea_Social[run_count]  <- mean(RLea_Social)
Mean_RLoo_CLV[run_count]     <- mean(RLoo_CLV)
Mean_RLoo_Social[run_count]  <- mean(RLoo_Social)
Mean_Rnd_CLV[run_count]      <- mean(Rnd_CLV)
Mean_Rnd_Social[run_count]   <- mean(Rnd_Social)
Mean_delta[run_count]        <- mean(delta_vec)
Mean_q[run_count]            <- mean(q_vec)}

output                       <- cbind(target_clustering, log_normal_sdev,target_rho, seed_percent, Summary_stat, Mean_OL_CLV, Mean_OL_Social, Mean_RLea_CLV, Mean_RLea_Social,
Mean_RLoo_CLV, Mean_RLoo_Social, Mean_Rnd_CLV, Mean_Rnd_Social, Mean_delta, Mean_q)
write.table(output, "c:/Output.csv", sep=";", append=TRUE)}}}}
```

Web Appendix V: The Jackson Rogers Model

The Jackson-Rogers (2007) algorithm creates networks that have characteristics similar to the ones found in empirical networks, specifically, a relatively small average distance between pairs of nodes, a clustering coefficient larger than the ones found in random networks, an approximately scale-free degree distribution, assortativity in degree and a negative relationship between degree and clustering. The idea underlying the Jackson-Rogers (2007) algorithm is that networks grow over time. Each period, a new node is added to the existing network structure, and this node connects to the nodes already present in the network by linking either to "strangers" or to "friends of friends". First, the node identifies m_r nodes uniformly at random and links with each with probability p_r (the nodes it links to are referred to as "parent" nodes), and then it identifies m_n nodes uniformly at random from the union of the connections of the parent nodes and links with them with probability p_n.

For our empirical application we create networks with a predefined level of clustering. In total we define eight levels of clustering, ranging between 0.07 and 0.54 with an average of 0.24 over all conditions. These values are consistent with the degree of clustering observed in our empirical analysis (0.34) and are in line with the range of clustering coefficients previously reported for social networks (Jackson and Rogers 2007; Libai, Muller, and Peres 2010; Watts and Strogatz 1998). For each level of clustering we generate a total of 1,536 networks by first drawing random values for the key parameters (i.e., m_r, p_r, m_n, and p_n, as well as m_0, which denotes the number of nodes in the initial network, i.e., before additional nodes are added over time) from ranges that are consistent with the ones observed for real-life social networks.[1] We

[1] Specifically, we make the following four assumptions: first, the total number of links formed by each incoming node ($m=m_r\ p_r + m_n\ p_n$) ranges between 1 and 5; second, the ratio between links formed at random and links formed with the connections of parent nodes ($m_r\ p_r/\ m_n\ p_n$) ranges between 0.5 and 5; third, the probability of linking with a

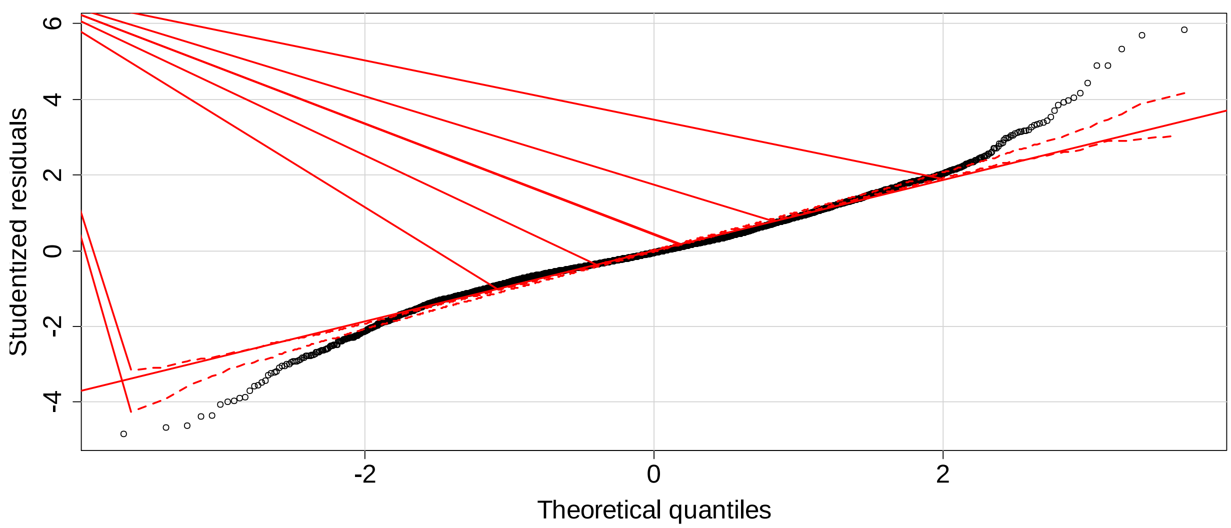
then let the network grow over time until it reaches approximately 1,000 nodes[2]. If the clustering coefficient of the resulting network falls within the target range (± 0.02), we keep the network. If not, we restart the process from the beginning with a different set of values for m_r, p_r, m_n, p_n and m_0.

node ($p_r = p_n = p$) ranges between 0.1 and 0.5; and fourth, the size of the initial network m_0 ranges between m+1 and 200; see Table 1 in Jackson and Rogers (2007) for a justification of these ranges.

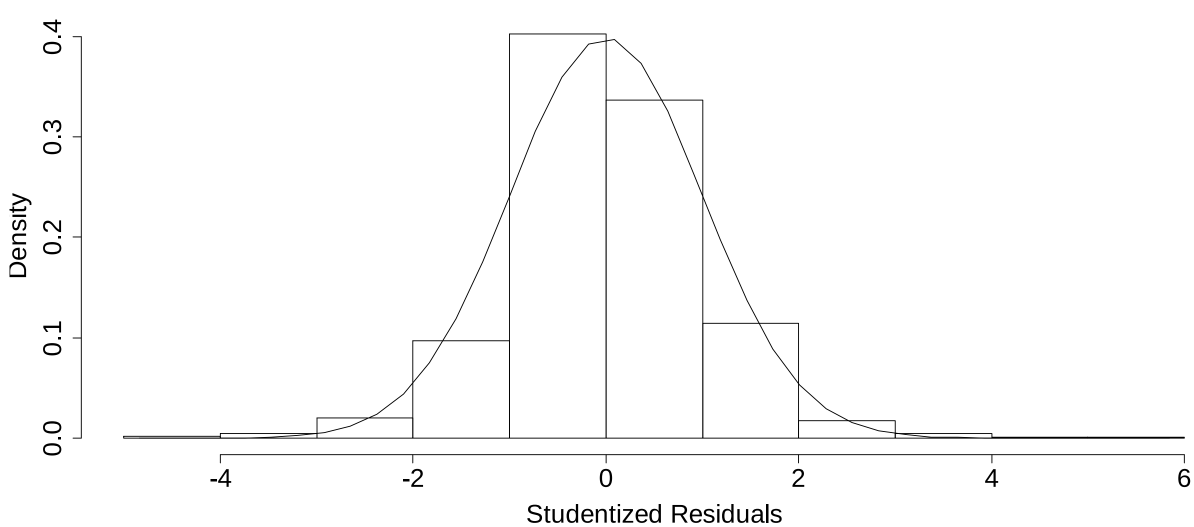
[2] Due to the random nature of the link creation process, the number of nodes is slightly different in each network. Among the 12,288 networks used for our analysis, the number of nodes ranges from a minimum of 953 to a maximum of 1,052 with an average of 988.

Web Appendix VI: Diagnostics for OLS Regression (Table 3)

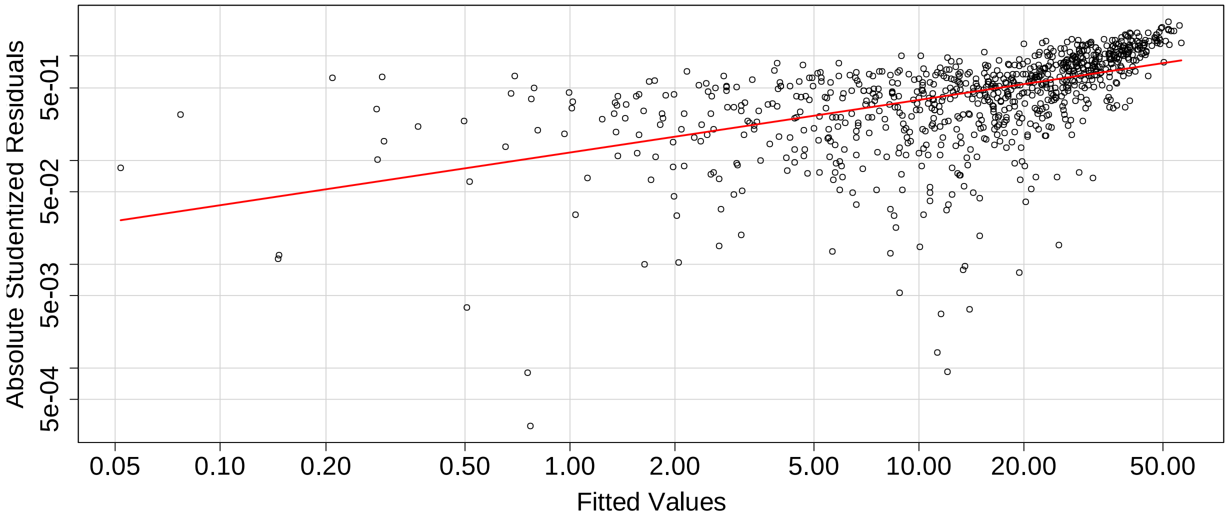
QQ-Plot



Distribution of Studentized Residuals



Spread-Level Plot



Web Appendix VII: Additional Analyses and Robustness Checks

Part 1: Higher influence for heavy users

Note that in the basic diffusion model, for each node *i* in period *t*, we determine the probability of adoption as a function of an external influence parameter δ_i and an internal influence parameter q_i such that:

(1) $$p_i(t) = 1 - (1-\delta_i)\,(1-q_i)^{Ni(t)},$$

where $N_i(t)$ is defined as the number of nodes in *i*'s personal network who already adopted prior to *t*.

To examine a situation in which heavier users have a higher influence on others, we modify this specification by defining $N_i(t)$ as the *sum of the revenue* of all nodes in *i*'s personal network who already adopted prior to t. This definition implies that nodes with higher revenue are likely to exert a stronger influence on other users. Our measure of $N_i(t)$ therefore corresponds to a weighted count in which the weight received by each node is proportional to the node's revenue. Note that revenue has been normalized to an overall average of one, so the overall influence in the system does not change. The results of this analysis are presented in Table WA5A:

Table WA5A:

Higher influence for heavier users

	Opinion leader			Revenue leader			Random		
Seed size	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**
0.50%	65.9	5.0	**71.0**	58.7	28.3	**87.0**	24.0	7.8	**31.9**
1.00%	42.1	5.1	**47.2**	43.5	27.8	**71.3**	20.1	7.6	**27.8**
1.50%	32.2	5.0	**37.2**	36.4	27.4	**63.8**	17.7	7.4	**25.2**
2.00%	26.4	5.1	**31.5**	31.4	27.2	**58.6**	16.6	7.3	**23.9**
2.50%	22.0	4.9	**26.9**	27.6	27.2	**54.8**	15.0	7.4	**22.4**
3.00%	19.3	4.9	**24.2**	24.8	27.0	**51.8**	14.2	7.3	**21.5**
3.50%	16.8	4.9	**21.8**	22.1	27.4	**49.6**	13.2	7.4	**20.5**
4.00%	15.0	4.9	**19.9**	20.0	27.1	**47.1**	12.5	7.3	**19.8**
Average	**30.0**	**5.0**	**35.0**	**33.1**	**27.4**	**60.5**	**16.7**	**7.4**	**24.1**

Part 2: Decay of word of mouth over time

This analysis relates to possible changes in the value of word of mouth over time. Recent results suggest that in various cases the effect of word of mouth after an event such as adoption or defection will gradually decrease in an exponential way (Nitzan and Libai 2011; Trusov, Bucklin, and Pauwels 2009). To test for the impact of such an effect within our simulation, we assumed that the value of internal influence parameter q decays over time following adoption, according to an exponential decay function with a decay rate of 0.1 per period (Trusov, Bucklin, and Pauwels 2009).

The results are presented in Table WA5B. We see that the results are consistent with those presented in Table 2 in terms of their direction as well as the relative magnitudes of the social and direct values of the different seeding strategies. Interestingly, we see that the direct value of all groups is higher with a decaying q than with a constant q. The reason is that with a decaying q, the overall adoption pattern is slower than with a constant q, and adoptions, on average, occur later on, generating less value. Hence, accelerating the purchases of the seeded customers creates relatively more value.

Table WA5B:

Decay of word of mouth over time

	Opinion leader			**Revenue leader**			**Random**		
Seed size	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**
0.50%	69.5	5.8	**75.3**	33.8	31.6	**65.3**	24.2	8.1	**32.4**
1.00%	43.8	5.7	**49.6**	26.0	30.7	**56.7**	20.6	8.2	**28.7**
1.50%	33.6	5.6	**39.1**	22.9	29.9	**52.8**	17.9	8.1	**26.0**
2.00%	27.3	5.4	**32.8**	20.2	29.7	**49.8**	16.7	8.0	**24.7**
2.50%	23.1	5.4	**28.6**	17.8	29.8	**47.6**	15.1	7.9	**23.0**
3.00%	20.3	5.3	**25.7**	16.4	29.6	**46.0**	14.4	7.8	**22.2**
3.50%	17.5	5.4	**22.9**	14.6	29.8	**44.4**	13.0	8.0	**21.0**
4.00%	15.8	5.4	**21.3**	13.4	29.6	**43.0**	12.4	8.0	**20.4**
Average	**31.4**	**5.5**	**36.9**	**20.6**	**30.1**	**50.7**	**16.8**	**8.0**	**24.8**

Part 3: Periodic variation of the influence of word of mouth

This analysis examines the case in which social effects are not stable. Social networks are inherently dynamic, i.e., they may be in a state of flux in which ties and the influence they exert on one another change. To see how such change might affect our results, we performed an analysis in which the internal influence parameter q was not constant over time but rather sampled out of the range used in our initial simulation, for each node in each period. While this analysis still assumes a constant basic network structure (i.e., the group of nodes and links between them), it allows for random changes in the level of influence between different nodes over time. The results of this robustness check can be found in Table WA5C. As can be seen from a comparison to Table 2, our main findings are generally similar to those obtained when q is constant over time.

Table WA5C:

Periodic variation of word of mouth

	Opinion leader			Revenue leader			Random		
Seed size	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**
0.50%	69.5	5.0	**74.5**	34.5	29.0	**63.5**	25.7	7.8	**33.5**
1.00%	44.0	4.9	**48.9**	27.4	27.6	**55.0**	21.5	7.2	**28.7**
1.50%	33.1	4.9	**38.0**	23.7	27.3	**51.1**	19.1	7.3	**26.4**
2.00%	27.3	4.9	**32.2**	21.1	27.3	**48.4**	17.4	7.2	**24.6**
2.50%	23.1	4.8	**27.9**	18.7	26.9	**45.6**	16.0	7.3	**23.3**
3.00%	20.1	4.8	**24.9**	17.1	26.8	**43.9**	15.0	7.2	**22.2**
3.50%	17.5	4.8	**22.4**	15.4	27.1	**42.5**	13.8	7.4	**21.2**
4.00%	15.8	4.9	**20.7**	14.2	27.2	**41.4**	13.1	7.3	**20.4**
Average	**31.3**	**4.9**	**36.2**	**21.5**	**27.4**	**48.9**	**17.7**	**7.3**	**25.0**

Part 4: Opinion leader identification based on influence

In this analysis we ask what happens when opinion leaders are chosen according to their influence (assuming that the firm can assess individual-level influence) instead of their connectivity. This is independent of the case in which heavy use is correlated with high influence, which we discuss above. To do so, we repeated the analysis, but instead of targeting the 10% of people with the highest degree, we targeted the 10% with the highest value of internal influence parameter q. The results are presented in Table WA5D. While the general direction of the results is similar, we do see that, overall, the total value of opinion leaders is lower than in the case of hub targeting and is now considerably lower than that of revenue leaders. On one hand, the direct value of opinion leaders is higher than in the case of hub targeting, since hubs are expected to adopt relatively early in any case, such that their acceleration brings less added value. On the other hand, the social value of opinion leaders is substantially lower in the current analysis, a result that is consistent with recent empirical findings on the relative advantage of targeting hubs (Hinz et al. 2011). Notice, however, that in our case, the distribution of degree is approximately scale free, while the individual *q* is taken from a uniform distribution, which means that influence-based opinion leaders are not as different from others to begin with. Given the very limited information on the empirical distribution of individual-level *q* in different markets, more research is needed in order to carry out an in-depth comparison of seeding strategies targeting different types of opinion leaders.

Table WA5D:

Opinion leader identification based on influence

	Opinion leader			Revenue leader			Random		
Seed size	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**
0.50%	24.1	7.1	**31.2**	34.6	28.7	**63.3**	26.1	7.5	**33.6**
1.00%	20.5	6.9	**27.4**	26.7	28.1	**54.8**	21.1	7.5	**28.5**
1.50%	18.3	7.0	**25.3**	23.0	27.6	**50.6**	18.8	7.3	**26.1**
2.00%	16.8	7.1	**23.9**	20.5	27.5	**48.0**	16.9	7.3	**24.2**
2.50%	15.4	6.9	**22.3**	18.4	27.3	**45.7**	15.6	7.3	**22.9**
3.00%	14.4	6.9	**21.3**	16.6	27.1	**43.7**	14.7	7.3	**22.1**
3.50%	13.2	7.0	**20.1**	15.0	27.4	**42.4**	13.3	7.2	**20.6**
4.00%	12.4	6.9	**19.4**	13.8	26.9	**40.7**	12.6	7.3	**19.9**
Average	**16.9**	**7.0**	**23.9**	**21.1**	**27.6**	**48.7**	**17.4**	**7.3**	**24.7**

Table WA5E:

Social and direct value for different seeding percentages (per 1% seeding), including revenue laggards

	Opinion leader			Revenue leader			Random			Revenue laggards		
Seed size	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**
0.50%	67.4	4.9	72.3	34.2	28.8	63.0	24.6	7.6	32.2	24.1	2.0	26.1
1.00%	42.6	5.0	47.6	27.3	28.2	55.5	20.9	7.4	28.4	20.7	2.0	22.6
1.50%	32.3	4.9	37.2	23.0	27.5	50.5	18.4	7.5	25.8	17.8	1.9	19.8
2.00%	26.5	4.9	31.4	20.6	27.6	48.2	17.1	7.3	24.5	16.1	1.9	18.0
2.50%	22.6	5.0	27.6	18.4	27.3	45.7	15.7	7.3	23	15.0	1.9	16.9
3.00%	19.9	4.9	24.8	16.7	27.0	43.7	14.1	7.2	21.9	14.1	1.9	16.0
3.50%	17.0	4.9	21.9	15.0	27.3	42.3	13.5	7.2	20.7	12.8	1.9	14.7
4.00%	15.4	4.9	20.3	13.8	27.2	41.0	12.9	7.3	20.2	12.1	1.9	14.0
Average	**30.5**	**4.9**	**35.4**	**21.1**	**25.1**	**46.2**	**17.2**	**6.0**	**23.1**	**16.6**	**1.9**	**18.5**

Table WA5E reports the direct and social value for revenue laggards, which complement the results already reported in Table 2. We see that the direction of the results is consistent with that of the results on revenue leaders: Compared with seeding random customers, seeding revenue laggards creates lower social value and lower direct value. However, for both social and direct value, the differences between revenue laggards and random customers are smaller than the corresponding differences between revenue leaders and random customers.

Table WA5F.

Seeding characteristics for high and low total value

		Revenue leader seeding (High total value)	Revenue leader seeding (Low total value)
Opinion leader seeding (High total value)	Percent of cases	29%	8%
	CLV SD	2.2	0.5
	CLV assortativity	0.36	0.32
	Degree SD	10.6	8.8
	Degree assortativity	0.64	0.63
	Clustering coefficient	0.24	0.20
	CLV-Degree correlation	0.065	0.033
		I	II
Opinion leader seeding (Low total value)	Percent of cases	25%	38%
	CLV SD	2.2	0.5
	CLV assortativity	0.36	0.33
	Degree SD	9.5	9.6
	Degree assortativity	0.63	0.62
	Clustering coefficient	0.25	0.25
	CLV-Degree correlation	-0.047	0.005
		III	IV

Web Appendix VIII: Cost Analysis

As highlighted in our manuscript, the total value created by seeding is equal to the direct value plus the social value minus the cost of seeding. For this analysis we assume that the cost of seeding (program cost) can be expressed as a certain fraction p of the CLV of the customers targeted for the seed. We assume that $0 \leq p \leq 1$, since the firm would, at maximum, invest the full CLV of the targeted customer as the cost of seeding. Seeding revenue leaders is recommendable when it generates a net value that is at least as high as that of seeding opinion leaders. This implies that:

(W9) $$\text{Gross value}_{RL} - p_{RL} \times CLV_{RL} \geq \text{Gross value}_{OL} - p_{OL} \times CLV_{OL}$$

where *Gross value* is the gross value of seeding (i.e., direct value plus social value), *CLV* is the customer lifetime value of customers selected as seeds, p is the cost of seeding as a percentage of CLV, and the subscripts *RL* and *OL* correspond to revenue leader seeding and opinion leader seeding, respectively. Equation W9 is equivalent to:

(W10) $$p_{RL} \leq (\text{Gross value}_{RL} - \text{Gross value}_{OL} + p_{OL} \times CLV_{OL})/CLV_{RL} = \pi_{RL}$$

Equation W10 implies that revenue leader seeding creates a higher total value than opinion leader seeding as long as p_{RL} is smaller than a certain threshold value π_{RL}.

Table WA6A provides average values of π_{RL} estimated according to our simulations for different values of p_{OL} (ranging from 0% to 100%) and different seeding percentages. It shows that for low seeding percentages (0.5%), revenue leader seeding does not generate a higher net benefit than opinion leader seeding, regardless of the cost associated with targeting opinion leaders. This is consistent with our observation in Table 2 that for low seeding percentages, the total value generated by revenue leader seeding is lower than that generated by opinion leader seeding.

However, for higher seeding percentages, it is worthwhile to invest in revenue leader seeding, as long as it is relatively cheaper than opinion leader seeding. Take the case of 1.0% seeding as an example. If seeding opinion leaders is associated with a cost of 30% CLV, revenue leader seeding generates a higher total value if the cost associated with it is not greater than 9.8% of revenue leader CLV.

One should, however, note that due to the difference in CLV between revenue leaders and opinion leaders, translating these relative differences into absolute ones is not straightforward. In Table 1 we see that, on average, the CLV of a revenue leader is 3.83 times that of the average customer. Hence, on a comparative absolute scale, it may be worthwhile to invest in revenue leader seeding even for relatively low cost percentage thresholds. In the example above, for a seed constituting 1% of the population, if the cost of seeding an opinion leader is 30% of the opinion leader's CLV, it is worthwhile to invest in revenue leaders if the cost of targeting one is 9.8%×3.83=37.5% of the CLV of an opinion leader. For high seeding percentages, revenue leader seeding is clearly more efficient even if it is more costly than opinion leader seeding.

Table WA6A: Maximum cost of revenue leader seeding as a percentage of revenue leader CLV (π_{RL})

Seeding percentage	Cost of opinion leader seeding as percentage of opinion leader CLV: p(OL)										
	0%	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
0.5%	-71.7%	-67.6%	-63.5%	-59.4%	-55.3%	-51.2%	-47.1%	-43.0%	-38.9%	-34.8%	-30.7%
1.0%	-2.6%	1.6%	5.7%	9.8%	14.0%	18.1%	22.3%	26.4%	30.5%	34.7%	38.8%
1.5%	18.5%	22.6%	26.7%	30.7%	34.8%	38.9%	42.9%	47.0%	51.1%	55.1%	59.2%
2.0%	29.8%	33.9%	38.0%	42.1%	46.2%	50.2%	54.3%	58.4%	62.5%	66.6%	70.6%
2.5%	36.6%	40.7%	44.8%	48.9%	53.1%	57.2%	61.3%	65.4%	69.6%	73.7%	77.8%
3.0%	40.6%	44.7%	48.8%	52.9%	57.0%	61.1%	65.2%	69.3%	73.4%	77.5%	81.6%
3.5%	45.3%	49.4%	53.4%	57.5%	61.5%	65.6%	69.6%	73.7%	77.7%	81.8%	85.8%
4.0%	47.2%	51.2%	55.3%	59.3%	63.4%	67.4%	71.5%	75.5%	79.6%	83.6%	87.7%

Web Appendix IX: Issues for Future Research in Revenue Leader Targeting

Part 1. The effect of a hybrid strategy.

In order to see the effect of a hybrid strategy on the value created we re-ran the basic analysis, this time adding a situation of splitting the targeting so half would target revenue leaders and half would target opinion leaders. In Table WA9A we see the results, compared to equivalent results (from Table 2) with pure strategies.

Table WA9A: Value created in a hybrid strategy (equal parts revenue and opinion leaders) compare to pure strategies

	Opinion leader			Revenue leader			Random			Hybrid: Revenue Leaders and Opinion leaders		
Seed size	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**
1.00%	42.6	5.0	47.6	27.3	28.2	55.5	20.9	7.4	28.4	39.4	16.7	56.1
1.50%	32.3	4.9	37.2	23.0	27.5	50.5	18.4	7.5	25.8	31.4	17.2	48.6
2.00%	26.5	4.9	31.4	20.6	27.6	48.2	17.1	7.3	24.5	26.0	16.3	42.3
2.50%	22.6	5.0	27.6	18.4	27.3	45.7	15.7	7.3	23	22.6	16.5	39.1
3.00%	19.9	4.9	24.8	16.7	27.0	43.7	14.1	7.2	21.9	19.8	15.9	35.7
3.50%	17.0	4.9	21.9	15.0	27.3	42.3	13.5	7.2	20.7	17.7	16.1	33.8
4.00%	15.4	4.9	20.3	13.8	27.2	41.0	12.9	7.3	20.2	16.1	16.1	32.2
Average	**25.2**	**4.9**	**30.1**	**19.3**	**27.4**	**46.7**	**16.1**	**7.3**	**23.5**	**24.7**	**16.4**	**41.1**

We see that overall, the pattern of results is similar among the pure and hybrid strategy. The total value of the hybrid strategy is on average lower than that of the best pure strategy, with the exception of 1% where the total value is slightly higher. Of course, other combinations of pure strategies should be examined to determine the optimal strategy by the firm.

Part 2. The effect of prediction error.

In order to examine the effect of prediction error, we considered a case in which 50% of the revenue leader targeting has not reached revenue leaders, but rather random customers, and in another situation, 50% of the opinion leader targeting have not reached opinion leaders, but rather random customers. Table WA9B presents the results compared to equivalent results (from Table 2) with no error targeting.

Table WA9B: Value created with and without error in targeting.

	Opinion leader			Revenue leader			50% error opinion leader targeting			50% error revenue leader targeting		
Seed size	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**	**Social**	**Direct**	**Total**
1.00%	42.6	5.0	47.6	27.3	28.2	55.5	37.8	6.3	44.1	24.5	18.2	42.7
1.50%	32.3	4.9	37.2	23.0	27.5	50.5	29.8	6.5	36.3	21.6	18.5	40.1
2.00%	26.5	4.9	31.4	20.6	27.6	48.2	24.8	6.3	31.1	19.4	17.5	36.8
2.50%	22.6	5.0	27.6	18.4	27.3	45.7	21.8	6.2	28.0	17.7	17.6	35.3
3.00%	19.9	4.9	24.8	16.7	27.0	43.7	19.4	6.1	25.5	16.2	17.1	33.3
3.50%	17.0	4.9	21.9	15.0	27.3	42.3	17.0	6.2	23.2	14.7	17.3	32.0
4.00%	15.4	4.9	20.3	13.8	27.2	41.0	15.7	6.1	21.8	13.9	17.3	31.2
Average	**25.2**	**4.9**	**30.1**	**19.3**	**27.4**	**46.7**	**23.8**	**6.2**	**30.0**	**18.3**	**17.6**	**35.9**

We see that as expected, the value created in the case of targeting error goes down compared to no error, yet the pattern of value creation does not change. This will be different, of course, if the percent of error will differ among revenue leader and opinion leader targeting.

Additional References for the Web Appendix

Fibich, Gadi and Ro'i Gibori (2010), "Aggregate Diffusion Dynamics in Agent-Based Models with a Spatial Structure," *Operations Research*, 58 (5), 1450 - 68.

Goldenberg, Jacob, Barak Libai, Sarit Moldovan, and Eitan Muller (2007), "The Npv of Bad News," *International journal of research in marketing*, 24 (3), 186 - 200.

Goldenberg, Jacob, Barak Libai, and Eitan Muller (2010), "The Chilling Effects of Network Externalities," *International journal of research in marketing*, 27 (1), 4 - 15.

---- (2002), "Riding the Saddle: How Cross-Market Communication Can Create a Major Slump in Sales," *Journal of marketing*, 66 (2), 1 - 16.

---- (2001), "Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth," *Marketing letters*, 12 (3), 211 - 23.

Gupta, Sunil, Dominique Hanssens, Bruce Hardie, Wiliam Kahn, V. Kumar, Nathaniel Lin, Nalini Ravishanker, and S. Sriram (2006), "Modeling Customer Lifetime Value," *Journal of service research*, 9 (2), 139 - 55.

Gupta, Sunil, Donald R. Lehmann, and Jennifer Ames Stuart (2004), "Valuing Customers," *Journal of marketing research*, 41 (1), 7 - 18.

Hinz, Oliver, Bernd Skiera, Christian Barrot, and Jan U. Becker (2011), "Seeding Strategies for Viral Marketing: An Empirical Comparison," *Journal of marketing*, 75 (6), 55 - 71.

Jackson, Matthew O. and Brian W. Rogers (2007), "Meeting Strangers and Friends of Friends: How Random Are Social Networks?," *The American economic review*, 97 (3), 890 - 915.

Leskovec, Jure, Lada A. Adamic, and Bernardo A. Huberman (2007), "The Dynamics of Viral Marketing," *ACM Transactions on the web*, 1 (1), 1 - 39.

Libai, Barak, Eitan Muller, and Renana Peres (2012), "Decomposing the Value of Word-of-Mouth Seeding Programs: Acceleration Vs. Expansion," *Journal of marketing research*, Forthcoming.

---- (2010), "Sources of Social Value in Word-of-Mouth Programs," *MSI Working paper series*, Report No. 10-103.

Nitzan, Irit and Barak Libai (2011), "Social Effects on Customer Retention," *Journal of marketing*, 75 (6), 24 - 38.

Rand, William and Roland T. Rust (2011), "Agent-Based Modeling in Marketing: Guidelines for Rigor," *International journal of research in marketing*, 28 (3), 181 - 93.

Rust, Roland T., V. Kumar, and Rajkumar Venkatesan (2011), "Will the Frog Change into a Prince? Predicting Future Customer Profitability," *International journal of research in marketing*, 28 (4), 281 - 94.

Sultan, Fareena, John U. Farley, and Donald R. Lehmann (1990), "A Meta-Analysis of Applications of Diffusion Models," *Journal of marketing research*, 27 (1), 70 - 77.

Trusov, Michael, Randolph E. Bucklin, and Koen Pauwels (2009), "Effects of Word-of-Mouth Versus Traditional Marketing: Findings from an Internet Social Networking Site," *Journal of marketing*, 73 (5), 90 - 102.

Watts, Duncan J. and Steven H. Strogatz (1998), "Collective Dynamics of Small-World Networks," *Nature*, 393 (6684), 440 - 42.

Zubcsek, Peter Pal and Miklos Sarvary (2011), "Advertising to a Social Network," *Quantitative Marketing and Economics*, 9 (1), 71 - 107.