Influence Maximization with Viral Product Design

Nicola Barbieri * Francesco Bonchi †

Abstract

Product design and *viral marketing* are two popular concepts in the marketing literature that, although following different paths, aim at the same goal: maximizing the adoption of a new product. While the effect of the social network is nowadays kept in great consideration in any marketing-related activity, the interplay between product design and social influence is surprisingly still largely unexplored.

In this paper we move a first step in this direction and study the problem of designing the features of a novel product such that its adoption, fueled by peer influence and "word-of-mouth" effect, is maximized. We model the viral process of product adoption on the basis of social influence and the features of the product, and devise an improved iterative scaling procedure to learn the parameters that maximize the likelihood of our novel feature-aware propagation model. In order to design an effective algorithm for our problem, we study the property of the underlying propagation model. In particular we show that the expected spread, i.e., the objective function to maximize, is monotone and submodular when we fix the features of the product and seek for the set of users to target in the viral marketing campaign. Instead, when we fix the set of users and try to find the optimal features for the product, then the expected spread is neither submodular nor monotone (as it is the case, in general, for product design). Therefore, we develop an algorithm based on an alternating optimization between selecting the features of the product, and the set of users to target in the campaign. Our experimental evaluation on real-world data from the domain of social music consumption (LastFM) and social movie consumption (Flixster) confirms the effectiveness of the proposed framework in integrating product design in viral marketing.

1 Introduction

Product design and *viral marketing* are two widely studied concepts in the marketing literature. The former concept refers to all those activities (such as market share forecasting or estimation of customer utilities for a given set of product features), that are aimed at designing a new product, with the objective of maximizing the number of customers adopting the product. The latter concept refers to marketing techniques which are able to take advantage of peer influence among customers, through modern social media and communication platforms. The idea is to exploit a pre-existing social network in order to increase brand awareness or to achieve other marketing objectives (such as product sales) through self-replicating viral processes.

The *share-of-choice* (SOC) problem is the combinatorial optimization problem at the basis of product design. In the SOC problem we are given a set of product features $\mathcal{F}$ with their levels (e.g., the size of the screen of a laptop), and a set of customers $V = \{v_1, v_2, \ldots, v_n\}$. Customers have different utilities for the feature levels of a product (called *part-worth utilities* in the marketing literature [9]): we let $u^i_{f,l}$ denote the part-worth utility for customer v_i if level l is chosen for feature $f \in \mathcal{F}$. Moreover we are given a "hurdle" h_i for each user v_i, representing the utility value of equilibrium between making or not making the purchase.

By denoting with $x_{f,l} \in \{0, 1\}$ an indicator variable, which is 1 if the level l has been selected for the feature f and zero otherwise, the assumed adoption model is that a customer v_i decides to adopt the product when the sum of utilities exceeds her hurdle:

$$\sum_{f \in \mathcal{F}} \sum_{l} x_{f,l} \cdot u^i_{f,l} \geq h_i. \tag{1.1}$$

The objective is to select the feature levels for a new product so to maximize the number of customers that will adopt it.

The basic computational problem behind viral marketing instead, is that of selecting the set of initial users that are more likely to influence the largest number of users in a social network, known as *influence maximization* (MAXINF) [11]. Given a directed social network $G = (V, E)$, where a link $(v_j, v_i) \in E$ means that v_j can potentially influence v_i, and given a budget k, the problem requires to find k "seed" nodes in the network, such that by activating them we can maximize the expected number of nodes that eventually get activated, according to a probabilistic propagation model that governs how influence propagates through the network.

One of the most studied propagation models is the *linear threshold* (LT) model. In this model at a given timestamp, each node is either active (a customer which already purchased the product) or inactive, and each node's tendency to become active increases monotonically as more of its neighbors become active. An active node never

*Yahoo Labs, Barcelona, Spain. E-mail: barbieri@yahoo-inc.com
†Yahoo Labs, Barcelona, Spain. E-mail: bonchi@yahoo-inc.com

becomes inactive again. In particular, each node v_i is influenced by each neighbor v_j according to a weight $b_{j,i}$, such that the sum of incoming weights to v_i is no more than 1. At the beginning, each node v_i picks a threshold θ_i uniformly at random from $[0, 1]$. If at time t, the total weight from the active neighbors of an inactive node v_i is at least θ_i,

$$\sum_{\substack{(v_j,v_i)\in E \\ v_j \text{ is active}}} b_{j,i} \geq \theta_i, \tag{1.2}$$

then v_i becomes active at timestamp $t + 1$.

Observe that the two problems, SOC and MAXINF, share various similarities: both have a threshold-based activation model, they have the same objective to maximize (the number of adoptions), and they are both **NP**-hard [12, 11].

They have also several differences. In particular, in SOC only a set of user V is considered, overlooking the possible relations among the users, i.e., the social network. On the other hand MAXINF considers all the products the same, thus overlooking an important aspect: different product features interest different users to different extents, so one should not forget the characteristics of the product, when modeling influence propagation.

In this paper *we incorporate product design in viral marketing*. This leads to the new problem of influence maximization with viral product design (MAXINF_VPD), which builds upon the similarities of SOC and MAXINF, and overcomes their limitations by considering both the social influence and the features of the product to be promoted by a viral marketing campaign.

In a recent paper [10], the economists Gunnec and Raghavan provide the "dual" of our effort: they incorporate peer influence in SOC. In particular peer influence is injected into the objective function of SOC as a decrease in one person's hurdle. The more social contacts adopt the product, the smaller becomes the hurdle. By means of an analytical example, Gunnec and Raghavan show the difference in market share among SOC with and without peer influence: in the worst case, the loss of market share for ignoring the social network effects can be as large as the whole market.

In Figure 1 we report empirical example based on real data, to show that in viral marketing the features of the product being promoted cannot be overlooked. In particular we show the different expected spread achieved by our algorithm when considering items with different features set.[1] The first plot, from a LastFM dataset, shows the expected spread (for different sizes of the seed set) of a viral marketing campaign of two different new songs that we might want to promote. One song has the features {Electronic, Metal}, indicating a song combining Heavy Metal and Electronic music, while the second song has the features {Lana del Rey, Kate Perry}, indicating an

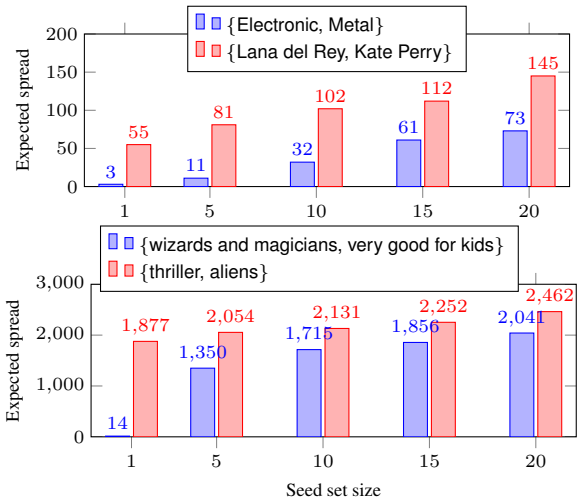

Figure 1: *The different spread of influence achieved by our algorithm when considering items with different feature sets. The plot above refers to an experiment on the LastFM dataset, while the bottom plot is from the Flixster dataset.*

hypothetical collaboration among these two pop singers. The expected size of market share in our experiments clearly speaks in favor of the second song. Therefore, if we were about commercializing a new hit, we would better consider setting up a collaboration between Lana del Rey and Kate Perry, than a metal-electronic extravaganza.

2 Background and related work

Influence maximization. Given a probabilistic propagation model m (for instance, LT) and a seed set $S \subseteq V$, the expected number of active nodes at the end of the process is denoted by $\sigma_m(S)$. The MAXINF problem requires to find the set $S \subseteq V$, $|S| = k$, such that $\sigma_m(S)$ is maximum. Kempe *et al.* [11] show that the problem is **NP**-hard, however, the function $\sigma_m(S)$ is *monotone*[2] and *submodular*[3]. When equipped with such properties, the simple algorithm that at each iteration greedily extends the set of seeds with the node w providing the largest marginal gain $\sigma_m(S \cup \{w\}) - \sigma_m(S)$, produces a solution with $(1 - 1/e)$ approximation guarantee [16].

Following [11], considerable effort has been devoted to develop methods for improving the efficiency of MAXINF [13, 5, 8], or for learning the strength of influence along each link [18, 7] (which is a needed input for MAXINF). Regardless the fact that users' authoritativeness, expertise, trust and influence are evidently topic-dependent, the research on social influence has surprisingly largely overlooked this aspect: only recently, researchers have started looking at social influence while keeping in consideration the characteristics of the item that is propagating [2].

[1] In this paper for simplicity's sake and due to the nature of our datasets, the features we consider have binary values: either the product has a feature or not.

[2] $\sigma_m(S) \leq \sigma_m(T)$ whenever $S \subseteq T$.

[3] $\sigma_m(S \cup \{w\}) - \sigma_m(S) \geq \sigma_m(T \cup \{w\}) - \sigma_m(T)$ whenever $S \subseteq T$.

Product design can be roughly divided into two main phases. In the first phase, data about customers preferences is collected and analyzed to produce part-worth utilities. The most popular tool for this task is a statistical technique used in market research known as *conjoint analysis* [9].

The second phase takes part-worth utilities as input and aims at selecting the levels of the product features that maximizes the product adoption: this is the *share-of-choice* (SOC) problem which has been shown to be **NP**-hard [12] and for which several heuristics have been proposed [1, 21].

Aral and Walker coin the term *viral product design* [19]. They distinguish between "viral characteristics" and "viral features" of a product. Viral characteristics refer to the content of the product, whereas viral features correspond directly to viral mechanism associated to the product.In this paper we are interested in the former, while Aral and Walker conduce randomized field experiments with the latter.

Gunnec and Raghavan [10] incorporate peer influence in the SOC problem and propose a genetic algorithm for their new version of the problem. There are many differences with our contribution. First, they consider all social contacts of a person having the same influence weight, thus overlooking any distinction between more or less influential users. Second, they assume part-worth utilities as given in input, as it is always the case for SOC. Instead in this paper we study the problem of *learning* the part-worth utilities, the hurdle for each user and the influence strength for each link *jointly*, by analyzing past data. Last, we incorporate product design in MAXINF, so the overall objective is different.

Miah *et al.* [14] study the problem of selecting which attributes of a product the seller should highlight in order to maximize the visibility of the product in a database representing an e-marketplace. Das *et al.* [6] introduce a new problem for web item design: given a training dataset of existing items with their user-submitted tags, and a query set of desirable tags, design the k best new items expected to attract the maximum number of desirable tags.

3 Paper contributions and roadmap

Our contributions are summarized as follows:

- A feature-aware propagation model (F-TM), which models the process of product adoption by considering social influence and part-worth utilities for the features of the items being propagated (Section 4).
- An *improved iterative scaling* procedure to learn part-worth utilities, hurdles, and social influence *jointly* (Section 4.1): those are the parameters that maximize the likelihood for the F-TM model, on a log of observed product propagations.
- The definition of the influence maximization with viral product design (MAXINF_VPD) problem and the study of its properties under the F-TM propagation model (Section 5).
- An iterative alternating optimization algorithm for the MAXINF_VPD problem (Section 6).
- A thorough experimental assessment using two real-world, semantically rich datasets from the domain of social music consumption (LastFM) and social movie consumption (Flixster). The datasets contain (i) the social network among users, (ii) a database of past item propagations in the social network, and (iii) a set of features for each item (Section 7).

4 Feature-aware propagation model

We next introduce our feature-aware propagation model. For simplicity's sake, we focus on binary features: either the product has a feature or not.

Let $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$ be the social graph, where each node v_i has an associated hurdle $h_i \in \mathbb{R}^+$, that represents an internal resistance of the user to be influenced. For each arc $(v_j, v_i) \in E$, let $b_{j,i} \in \mathbb{R}^+$ represent the strength of the influence exerted by v_j on v_i. Let $\mathcal{F}$ denote the universe of possible product features, and let $\mathcal{F}(p)$ represent the set of features of a given product p. The part-worth utility for user v_i and feature f is denoted $u_f^i \in \mathbb{R}$: it is natural that each feature may contribute positively for some users, while its contribute may be negative for others.

The *feature-aware threshold model* (F-TM) is a variant of the LT model, where peer influence and feature utilities cooperate, while the hurdle is used as a reduction of the utilities. In accordance with the literature on MAXINF, we keep the propagation model probabilistic: at the beginning each node v_i picks a threshold θ_i uniformly at random from $[0, 1]$. Time unfolds in discrete steps. For a given product p, the set of newly activated nodes at time t is denoted by $D_p(t)$, while $C_p(t) = \cup_{t' \leq t} D_p(t')$ represents the set of all nodes that adopted p so far. Product adoption is modeled using a logistic function. At time t the probability that user v_i adopts product p is:

$$Pr(p|i,t) = \frac{\exp\{\psi_t^i(p)\}}{1 + \exp\{\psi_t^i(p)\}} \quad \text{where:} \tag{4.3}$$

$$\psi_t^i(p) = \sum_{\substack{(v_j,v_i)\in E \\ v_j \in C_p(t)}} b_{j,i} + \sum_{f \in \mathcal{F}(p)} u_f^i - h_i. \tag{4.4}$$

The activation probability is zero if the node v_i has no active neighbors at the considered time, or if $\mathcal{F}(p) = \emptyset$. At time t if $Pr(p|i,t) \geq \theta_i$ then v_i adopts p at time $t+1$. The diffusion ends when no new nodes can be activated.

With the choice of the logistic as activation function, we model log-odds as a linear function of the user overall utility (social influence and subjective preference) in adopting the product. Moreover, this choice allows us employing maximum likelihood for estimating the parameters of the model.

$$B \geq C + \sum_{p\in\mathcal{P}}\sum_{i:v_i\in V}\delta(i,p)\left\{\sum_{v_j\in N^p_{in}(v_i)}\beta_{j,i} + \sum_{f\in\mathcal{F}(p)}\upsilon^i_f - \eta_i - \left(1-\psi^i_{t_p(i)}(p)\right)\right\} - \sum_{p\in\mathcal{P}}\sum_{i:v_i\in V}(1-\delta(i,p))\left(1-\psi^i_{t_p(i)}(p)\right)$$
$$- \sum_{p\in\mathcal{P}}\sum_{i:v_i\in V}\psi^i_{t_p(i)}(p)\cdot\frac{1}{\lambda^i_p}\left\{\sum_{v_j\in N^p_{in}(v_i)}\exp\left\{\beta_{j,i}\lambda^i_p\right\} + \sum_{f\in\mathcal{F}(p)}\exp\left\{\upsilon^i_f\lambda^i_p\right\} - \exp\left\{\eta_i\lambda^i_p\right\}\right\}$$

Figure 2: *Lower bound on the likelihood improvement.*

4.1 Learning the parameters of the model. We next focus on the problem of learning social influence strength, part-worth utilities and hurdles from past observations. The input of the learning problem is composed by: (i) the directed social graph $G = (V, E)$, (ii) a log of past product propagations $\mathbb{D}$, and (iii) for each product p the list of its features $\mathcal{F}(p)$. The log $\mathbb{D}$ is a relation $(User, Product, Time)$, where each tuple $\langle i, p, t\rangle$ indicates that the user v_i adopted the product p at the time t. We assume that the projection of $\mathbb{D}$ over the first column is contained in V, the projection of $\mathbb{D}$ over the second column corresponds to the universe of products $\mathcal{P}$, and the third column is contained in $[t_0, T]$. We also denote the time at which user v_i adopted the product p by $t_p(i)$. If v_i does not adopt p, then we set $t_p(i) = T + 1$.

The output of the learning problem is the set of parameters of the F-TM model: (i) the influence strength $b_{i,j}$ for each arc $(v_i, v_j) \in E$, (ii) the part-worth utility u^i_f for each user $v_i \in V$ and each feature $f \in \mathcal{F}$, and (iii) the hurdle h_i for each user $v_i \in V$. Let Θ denote the overall set of parameters of the model. Assuming that each propagation trace is independent from the others, the likelihood of the data given the model parameters Θ can be expressed as

$$\mathcal{L}(\mathbb{D};\Theta) = \sum_{p\in\mathcal{P}}\log\mathcal{L}(\mathbb{D}_p;\Theta) \quad \text{where:}$$
$$\mathcal{L}(\mathbb{D}_p;\Theta) = \prod_{i\,:\,v_i\in V} Pr(p|i,t_p(i))^{\delta(i,p)}\left(1-Pr(p|i,t_p(i))\right)^{1-\delta(i,p)}$$

and $\mathbb{D}_p$ represents the propagation of p, i.e., the selection of the relation $\mathbb{D}$ where the product is p. Moreover $\delta(i, p) = 1$ if v_i adopted p and 0 otherwise.

The optimization problem can be defined as follows:

$$\begin{array}{lll} \underset{\Theta}{\text{Maximize}} & \log L(\mathbb{D};\Theta) & \\ \text{subject to} & b_{i,j} \geq 0, & \forall\ (v_i, v_j) \in E \\ \text{and} & h_i \geq 0 & \forall\ v_i \in V \end{array}$$

For solving this learning problem we adopt the *improved iterative scaling* (IIS) approach[17, 4]. In practice we seek for an improvement Δ of the parameters, such that we obtain a higher (log) likelihood:

$$L(\mathbb{D};\Theta+\Delta) \geq L(\mathbb{D};\Theta)$$

The idea is to express the change in log likelihood $B = L(\mathbb{D};\Theta+\Delta) - L(\mathbb{D};\Theta)$ and optimize it with respect to Δ. The Δ is the set of improvements for all parameters, i.e., a $\beta_{i,j}$ for each $(v_i, v_j) \in E$, a η_i for each $v_i \in V$, and a υ^i_f for each $v_i \in V$ and $f \in \mathcal{F}$, such that $\Theta + \Delta$ actually means $b_{i,j} := b_{i,j} + \beta_{i,j}$, $h_i := h_i + \eta_i$, and $u^i_f := u^i_f + \upsilon^i_f$.

Let $N_{in}(v_i)$ denote the set of nodes which may potentially influence the node v_i, i.e: $N_{in}(v_i) = \{v_j \in V | (v_j, v_i) \in E\}$, while $N_{out}(v_i)$, symmetrically defined, is the set of nodes which can be influenced by v_i. For a given product p and a timestamp t, let $N^p_{in}(v_i) = C_p(t_p(i)) \cap N_{in}(v_i)$ denote the set of nodes which are exerting influence on v_i for product p at time t. Moreover, let $\lambda^i_p = |N^p_{in}(v_i)| + |\mathcal{F}(p)| + 1$.

By exploiting the inequality $-\log x \geq 1 - x$ and by applying Jensen's inequality, we can derive a lower bound on the improvement of the likelihood as reported in Figure 2, where C is the sum of all terms in the derivation which do not depend on the IIS parameters. Due to lack of space we omit the complete derivation which will be provided in an extended version of this paper.

Following the IIS approach we compute the set of updates of parameters Δ that maximizes our lower bound on the improvement of the loglikelihood. This is summarized in Algorithm 1, where we compute $\frac{\partial B}{\partial(\cdot)} = 0$ for each IIS parameter by applying Newton Raphson, in order to obtain an incremental update. Note that $b_{i,j} + \beta_{i,j}$ or $h_i + \eta_i$ may result negative. In this case, due to the monotonicity of the activation function with respect to $b_{i,j}$ and h_i, the lowest legal value that meets our non-negative constraints is zero.

5 Influence maximization with viral product design

We next focus on the MAXINF_VPD problem defined over the F-TM propagation model. The input to the problem is the social graph and the parameters of F-TM for which we defined a learning procedure in the previous section.

PROBLEM 1. (MAXINF_VPD) *Given a directed graph $G = (V, E)$, and the parameters of the F-TM propagation model: the influence strength $b_{i,j}$ for each arc $(v_i, v_j) \in E$, the part-worth utility u^i_f for each user $v_i \in V$ and feature $f \in \mathcal{F}$, and the hurdle h_i for each user $v_i \in V$.*

Let $\sigma(S, H)$ denote the expected spread of a set of nodes $S \subseteq V$ and a set of features $H \subseteq \mathcal{F}$, according to F-TM. Given a budget k of nodes, the problem requires to find S, with $|S| \leq k$, and H that maximize $\sigma(S, H)$.

Algorithm 1: Learning the parameters of F-TM

Input : Social graph $G = (V, E)$, propagation log data $\mathbb{D}$
Output: All parameters of F-TM: social influence $b_{i,j}$, part-worth utilities u^i_f, and hurdle h_i.

```
init(Θ); //Random initialization of parameters
repeat
    forall the v_i ∈ V do
        forall the v_j ∈ N_out(v_i) do
            Solve ∂B/∂β_{i,j} = 0 for β_{i,j}
            Set b_{i,j} ← max(0, b_{i,j} + β_{i,j})
        end
        forall the f ∈ F do
            Solve ∂B/∂v^i_f = 0 for v^i_f
            Set u^i_f ← u^i_f + v^i_f
        end
        Solve ∂B/∂η_i = 0 for η_i
        Set h_i ← max(0, h_i + η_i)
    end
until log likelihood converge;
```

THEOREM 5.1. MAXINF_VPD *is* **NP***-hard for* F-TM*.*

Proof. We show that the problem MAXINF under the *general threshold model* (GTM)[11], is a special instance of our problem MAXINF_VPD under F-TM.

Consider a fixed set of features H. Then according to Equations (3) and (4) the node's decision to become active is a monotone function of the set of its neighbors that are already active. Moreover, the activation probability is zero if the set of active neighbors is empty. These two properties correspond to the definition of GTM [11]. Moreover MAXINF_VPD, when the set of features H is fixed, corresponds to MAXINF, which has been shown to be **NP**-hard for GTM [11]. □

THEOREM 5.2. *Given the* F-TM *model, for any fixed choice of the feature-set H, the spread function $\sigma(S, H)$ is monotone and submodular in S.*

Proof. We can apply the submodularity conjecture for the General Threshold model of Kempe et al. [11], which was later proven by Mossel and Roch in [15]. This conjecture states that if the likelihood that a node become active increases as more of its neighbors become active and if the marginal effect of each neighbor satisfies the property of "diminishing returns", then the expected spread for a given set of initial nodes is submodular. Roughly, this means that "local" submodularity is preserved "globally". Therefore, if the local activation function is monotone and submodular, so it will be the expected spread function.

As the feature-set H is fixed, the sum of part-worth utilities and the hurdle are constant terms which can be incorporated into the user-specific threshold. For simplicity sake, let us us introduce as simplified activation function for each user v_i:

$$f^i(S) = \frac{\exp\{\psi^i(S)\}}{1 + \exp\{\psi^i(S)\}}, \text{ where } \psi^i(S) = \sum_{v_j \in S} b_{j,i}.$$

We can easily verify that, assuming positive influence weights, the monotonicity inequality $f^i(T) \geq f^i(S)$ holds for each $S \subseteq T \subseteq V$. The submodularity condition is formalized by the following inequality:

$$f^i(T \cup \{w\}) - f^i(T) \leq f^i(S \cup \{w\}) - f^i(S),$$

where again $S \subseteq T \subseteq V$ and $w \in V$. By standard algebraic manipulations, the above condition is equivalent to:

$$\frac{(e^{\psi^i(\{w\})} - 1)(e^{\psi^i(T)} - e^{\psi^i(S)})(e^{\psi^i(S \cup T \cup \{w\})} - 1)}{(e^{\psi^i(S)} + 1)(e^{\psi^i(T)} + 1)(e^{\psi^i(S \cup \{w\})} + 1)(e^{\psi^i(T \cup \{w\})} + 1)} \geq 0$$

which is satisfied for each $S \subseteq T \subseteq V$ and for positive influence weights. Summarizing we have shown that, by fixing the feature set H, the activation function adopted in this paper is locally monotone and submodular in S. This allow us to apply the submodularity result presented in [15] and conclude that, fixing the feature set H, the expected spread $\sigma(S, H)$ is monotone and submodular in S. □

The result above is not surprising. In practical terms it means that if we fix the product, then MAXINF_VPD boils down to standard MAXINF, and thus we can apply the standard algorithms developed for MAXINF to find the seed set of nodes. In particular, when the product is given, and thus the set of features H is fixed, the optimal seed set can be approximated by means of the simple greedy algorithm, which at each iteration adds to the seed set, the node that exhibits the highest marginal gain in the expected spread. To estimate the spread for a given seed set we employ Monte Carlo simulations. The overall procedure for seed selection provides an $(1 - 1/e - \epsilon)$ approximation [11]. The computational burden of the Monte Carlo simulations can be reduced by employing the lazy evaluation procedure CELF [13] which exploits the submodularity of the objective function.

Unfortunately, when doing influence maximization with product design, there is no clear property that we can use. In fact adding a feature to the current feature set, might well decrease the viral potentialities of the new product, as many influential users might have a negative part-worth utility for such feature. Therefore, when we aim at finding both the seed set of influential users S and the feature set H such as to maximize $\sigma(S, H)$ under the F-TM propagation model, we have to rely on heuristic methods. In the next section we propose our algorithm for MAXINF_VPD.

6 Algorithm

In the MAXINF_VPD problem we have to jointly select a set of users $S \subseteq V$ and a set of features $H \subseteq \mathcal{F}$, to maximize the expected viral spread of a viral marketing campaign started with nodes S, for a new product described by H. Given the enormous size of the search space of our problem, i.e., $2^{|\mathcal{F}|}\binom{|V|}{k}$, we need some property, or at least a good heuristics. A promising direction has recently been investigated by Singh *et al.* in [20], where the authors extends the concept of submodularity to set functions with two arguments (dubbed bisubmodularity). In particular, they show that for a restricted class of those functions, where (i) the two considered sets are not disjoint and (ii) fixing one of the coordinates of the function yields a submodular function in the free coordinate, that is, if the following property holds

$$f(A,B)+f(A',B') \geq f(A \cup A', B \cup B')+f(A \cap A', B \cap B')$$

then the coordinate-wise maximization, in which each set is optimized by considering the other fixed, is approximately optimal. Unfortunately, the same procedure does not provide, in general, any approximation guarantee if the two sets are disjoint (although authors prove a weaker result that requires some additional assumptions, i.e, the existence of a submodular extension of f).

A further obstacle in our case is that, due to the possibility of obtaining negative part-worth utilities, the spread function under F-TM is not even monotone. This is a common challenge in product design and SOC, for which heuristic approaches, or search techniques based on simulated annealing or genetic algorithms have been adopted [1, 3, 21].

Although we cannot rely on the bisubmodularity results, we can nevertheless borrow, as an heuristic, the coordinate-wise maximization approach. Following this idea we propose a procedure which alternates local greedy choices for the update of the seed set and the feature set. The procedure is presented in Algorithm 2.

Our algorithm starts by detecting the pair of singletons $\langle v, f \rangle$ in the cartesian product $V \times \mathcal{F}$, which maximizes the spread (Line 2). Then, iteration by iteration, we alternate the procedure of greedy seed selection and feature update in such a way that, at the generic iteration (i), we generate an improvement of the objective function, by forcing one of the two following inequalities to hold:

$$\sigma(S^{(i)}, H^{(i)}) \leq \sigma(S^{(i+1)}, H^{(i)}) \tag{6.5}$$

$$\sigma(S^{(i)}, H^{(i)}) \leq \sigma(S^{(i)}, H^{(i+1)}). \tag{6.6}$$

When we keep the feature set H fixed, if we increase the size of the seed set S then inequality in Equation (6.5) trivially holds thanks to the monotonicity in S. Actually, the step in which we make the greedy selection of the next node to add to the seed set (Line 9), has the usual $(1-1/e)$ quality guarantees, thanks to submodularity.

Algorithm 2: MAXINF_VPD

Input : Social graph $G = (V, E)$, budget $K \in \mathbb{N}$, and the set of all parameters of F-TM: social influence $b_{i,j}$, part-worth utilities u_f^i, and hurdle h_i.
Output: $S \subset V$, $|S| = k$, and a set of features $H \subseteq \mathcal{F}$.

1 $S, H, H_{old} \leftarrow \emptyset$
2 $(v_i', f') \leftarrow \arg\max_{(v_i,f) \in V \times \mathcal{F}} \sigma(\{v_i\}, \{f\})$
3 $S \leftarrow S \cup \{v_i'\}$, $H \leftarrow H \cup \{f'\}$
4 $\sigma_{curr} \leftarrow \sigma(S, H)$
5 $iterate \leftarrow true$
6 $it \leftarrow 0$
7 **while** $(iterate)$ **do**
8 **if** $(it \mod 2 = 0 \wedge |S| < k)$ **then**
9 $[s, \sigma_{curr}] \leftarrow greedySelection(S, H)$
10 $S \leftarrow S \cup \{s\}$
11 **else**
12 $H_{old} \leftarrow H$
13 $[H, \sigma_{curr}] \leftarrow updateFeatureSet(S, H)$
14 **if** $(|S| = k \wedge H_{old} = H)$ **then**
15 $iterate \leftarrow false$
16 **end**
17 $S' \leftarrow \arg\max_{\substack{S' \subseteq V \\ |S'| \leq |S|}} \sigma(S', H)$
18 **if** $(\sigma(S', H) > \sigma_{curr})$ **then**
19 $S \leftarrow S'$
20 $\sigma_{curr} \leftarrow \sigma(S', H)$
21 **end**
22 **end**
23 $it \leftarrow it + 1$
24 **end**

For the features-update step we do not have any property, but we can adopt several heuristic techniques for generating "good" solutions, such that Equation (6.6) is always satisfied. In this paper, we experiment with two alternatives for the the $updateFeatureSet(S, H)$ procedure (Line 13):

Local Update: in this case the update procedure provides the best feature set (w.r.t. the objective function) as output, which can be obtained by performing *only one* addition/removal of a feature to/from the current feature set. Although the procedure is expected to converge naturally to a limited number of features, as the algorithm will reach a point in which no local change can improve the spread, we put a constraint (≤ 10) of the size of the selected feature set to make the interpretability of the feature set provided.

Genetic Algorithm Update: the update of the feature set is performed by employing a genetic algorithm. Our implementation is built upon the popular JGAP (`http://jgap.sourceforge.net/`) framework, by using elitist selection on a initial population of 30 feature sets and by allowing 20 evolutions. Better results may be, in principle, obtained by using a larger number of evolutions, but we found this combination of parameters as a good compromise between

	Last.fm	Flixster
nodes ($\|V\|$)	1,372	29,275
links ($\|E\|$)	7,354	212,106
number of product adoptions ($\|\mathbb{D}\|$)	1,208,640	5,423,310
propagations episodes	322,932	2,768,654
avg number of products per user	880	185
avg number of users per product	23	953
products ($\|P\|$)	51,495	5,685
distinct features ($\|\mathcal{F}\|$)	13,510	5,116
avg number of features per product	5	8
avg number of products per feature	18	9

Table 1: *Summary of the datasets.*

computational time and effectiveness. In this case the fitness function can be naturally instantiated by the expected spread. However, to penalize large feature sets, we impose a penalty $(|H| - t)^2$ if the number of feature exceeds the "desired" length t of the feature set (10 in our experiments).

Once updated the feature set, we apply the standard MAXINF greedy algorithm with F-TM model, with budget $|S^{(i)}|$ and fixed feature set, to check if a better selection of seeds exists (Line 17). The procedure ends when we have reached the seed budget and no further update of the feature set produces an improvement of the spread (Line 15).

7 Experimental analysis

Datasets. For our purposes we need information-rich datasets containing (i) the social network among users, (ii) a database of past propagations of items in the social network, and (iii) a set of features for each item. Gathering such dataset is a non-trivial task. Indeed, the majority of MAXINF literature uses synthetically-generated data: usually the social graph is a real one, but the influence information is synthetic. We argue that for the task of studying and analysing the impact of part-worth utilities and peer influence on the overall diffusion process it is fundamental to rely on real-world data, mainly because real-world data exhibits patterns which may not be found on synthetic data. Therefore, we collected two real-world datasets: one from the domain of social music consumption (`lastfm.com`) and one from social movie consumption (`flixster.com`).

Our LastFM dataset was created starting from the *HetRec 2011 Workshop* dataset[4], and enriching it by crawling. Here the action log $\mathbb{D}$ contains information about the first time that a user listens to a song. If a user v_i listens to a song and shortly after one of her peers v_j does the same, we assume that the song propagated from v_i to v_j. More formally, we assume that a generic product p propagates from v_i to v_j iff $(v_i, v_j) \in E$ and $t_p(i) < t_p(j)$. Hence, we say that a tuple $\langle i, p, t\rangle \in \mathbb{D}$ is a *propagation episode* if exists $v_j \in C_p(t)$ such that $v_i \in N_{out}(v_j)$.

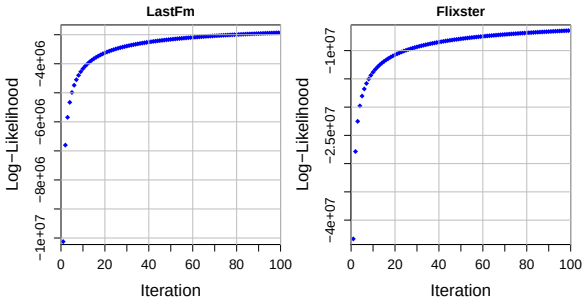

Figure 3: *Model fitting (Algorithm 1): convergence rate.*

Also our Flixster dataset was obtained by conjoining crawled information[5] with movies' tags coming from the *HetRec 2011 Workshop* dataset[5]. Here a movie propagates from a user to one of her peers, when both post a rating for it. In both datasets the social graph is bidirectional and the features are the tags associated to each product (song in LastFM and movie in Flixster).

Analysis of social influence and part-worth utility. Here we report the analysis of the parameters (peer influence, part-worth utility, and hurdle) learnt from real data, using Algorithm 1. To fit the F-TM model we run Algorithm 1 procedure for 100 iterations on both datasets: the convergence rate is reported in Figure 3.

Figure 4 shows that influence weights are distributed according to an exponential distribution, with a high density of zero-valued influence relationships, while part worth utilities are distributed according to a normal distribution, centered in the interval $(0, 20)$ and $(0, 50)$ on LastFm and Flixster, respectively. Moreover, we can observe that the Flixster dataset exhibits a higher level of influence, while in LastFm part-worth utilities play a more important role. Hurdles are distributed similarly in the two datasets.

Viral vs popular features. With the goal of showing that different features trigger viral cascades differently, we next compare the spread achievable under our propagation model by "viral" and "popular" features.

We define the level of "virality" of each feature by two different measures. Let $prop(p, i, j)$ be a predicate that is true if in the data we observe a propagation of product p from v_i to v_j, false otherwise. We define the *Propagation Coefficient*(PC) of the feature f as the ratio between the number of propagations and the total number of product adoptions which involve the considered feature:

$$PC(f) = \frac{|\{\langle i, p, t\rangle \in \mathbb{D} | f \in \mathcal{F}(p) \wedge \exists\, v_j \in V : prop(p, j, i)\}|}{|\{\langle i, p, t\rangle \in \mathbb{D} | f \in \mathcal{F}(p)\}|}$$

As second measure we consider the likelihood that an adoption of a product which exhibits the feature f will trigger adoptions. The *Propagation Likelihood*(PL) of the

[4] `http://www.grouplens.org/node/462`

[5] `http://www.cs.sfu.ca/~sja25/personal/datasets/`

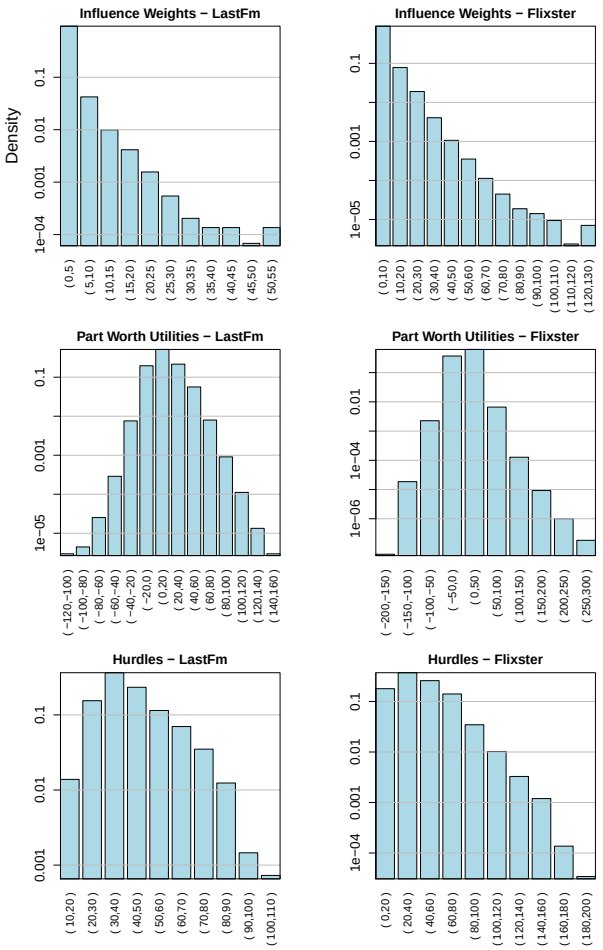

Figure 4: *Distribution of influence weights, part-worth utilities, and hurdles on the two datasets.*

feature f can be defined as:

$$PL(f) = \frac{\sum_{\substack{\langle i,p,t\rangle \in \mathbb{D} \\ f \in \mathcal{F}(p)}} \frac{|\{v_j \in V | prop(p,i,j)\}|}{|N_{out}(v_i)|}}{|\{\langle i,p,t\rangle \in \mathbb{D} | f \in \mathcal{F}(p)\}|}$$

Both scores have values $\in [0,1]$ and the higher the value, the higher the ability of the feature to trigger viral cascades.

To compare the expected spread achieved by product which exhibit different kind of features, we compare on both datasets the spread achieved by selecting the top-5 and the top-10 features selected according to three different criteria: (i) popularity (i.e, frequency, the number of products in which the feature is present), (ii) propagation coefficient and (iii) propagation likelihood. Before reporting the MAXINF results, it is worth mentioning that propagation coefficient and propagation likelihood are positively correlated on both datasets (ρ equals to 0.68 on LastFm and 0.91 on Flixster), while the correlation between the two virality indices and popularity of the feature is significantly lower ($\approx$ 0.03 on LastFm, and $\approx$ 0.14 on Flixster). Therefore, being popular does not make a feature viral.

The MAXINF results, given in Figure 5, highlight significant differences in terms of the expected spread achieved by

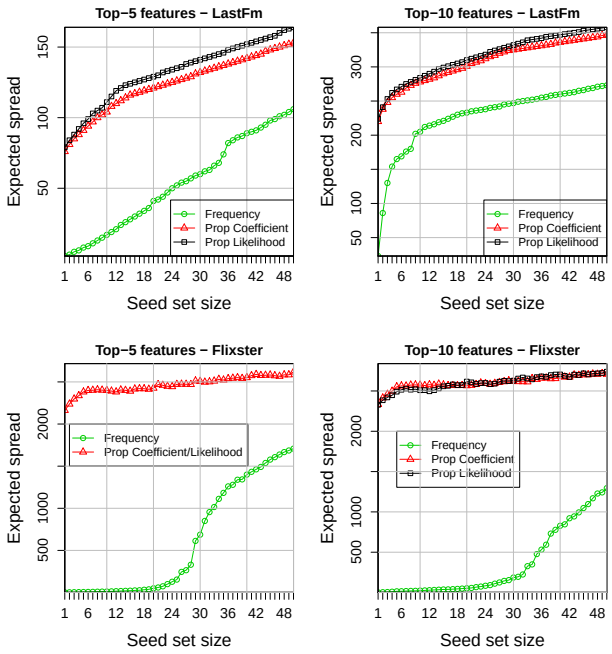

Figure 5: MAXINF *with "viral" and "popular" features.*

considering popular features versus "viral" ones. The highest spread in these cases can be achieved by selecting the features according to their propagation likelihood value. On the Flixster dataset, where we have already highlighted an albeit linear correlation between the values of propagation coefficient and likelihood, the top-5 feature selected according to the two different virality measures corresponds.

Influence maximization with viral product design. We next evaluate the performance of our framework for MAXINF_VPD. To initialize the algorithm, we select the 30 features with highest part-worth utility for each user v_i, and we pick the pair $\langle v_i, f\rangle$ which achieves the largest expected spread. Moreover, to reduce the feature search space, we consider only the top-$2,000$ features according to their propagation likelihood (PL), since this value, as shown in Figure 5, seems to be a good heuristic.

We report in Figure 6 the results of the maximization algorithm with a budget of 20 seed nodes on LastFm and 10 on Flixster. The version of the algorithm which employs a genetic algorithm optimization exhibits better performances in terms of spread (330 vs 305 on LastFm and $3,104$ vs $2,811$ on Flixster), at the cost of a higher computational overhead (around 3 times slower than the simpler version on LastFm and 5 times on Flixster).

The incremental selection of seed nodes is generally robust: on both datasets, the greedy algorithm is rarely able to provide a better combination of seeds after updating the feature set (considering the version based on genetic optimization, about 3 times on LastFm and 2 on Flixster).

While the local update version provides 10 features as output on both the datasets (which corresponds to the maximum budget), the solution provided by the genetic

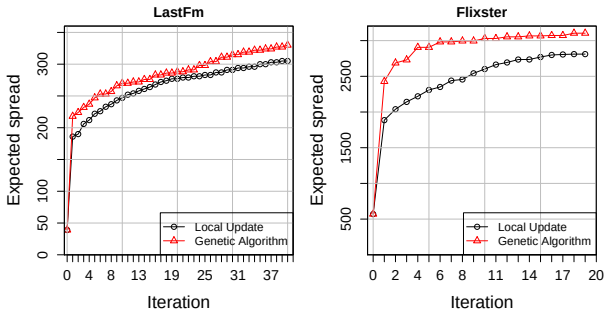

Figure 6: *Influence maximization with viral product design.*

version includes 18 features on LastFm, and 23 on Flixster. A very interesting observation is that some features belong to both feature sets provided by the different versions of the algorithm ($6/18$ on LastFm and $4/23$ on Flixster, starting from a set of 2,000 features).

Finally, the most important question of our experimental analysis is whether incorporating product design in viral marketing produces any benefit, w.r.t. keeping the two tasks separated. By comparing Figure 6 with Figure 5 we can observe clearly that the spread achieved by integrating the product design process into the influence maximization frame is significantly greater than the one achieved by the heuristic selection of features presented. On LastFm, employing 20 seed nodes and 10 features, the MAXINF_VPD procedure achieves a spread of 305, while the best result reported for a heuristic selection of the same number of features is 170. Similar gains are visible in Flixster.

8 Conclusions

The wide diffusion of social media and e-commerce platforms provides a great source of information for understanding customers' needs, interests and opinions, which can be exploited to design better marketing strategies and better products. In this paper we explore a novel direction in the field of *computational marketing*, by proposing a framework which allows the integration of the product design process into viral marketing.

We incorporate part-worth utilities into a social-influence-driven propagation model and define a procedure to fit the model to real data, so that we can learn peer influence strength and part-worth utilities jointly. Based on these parameters learnt from real data, we introduce the problem of influence maximization with product design and devise a heuristic optimization algorithm, which is empirically shown to provide good solutions on two real-world datasets. In particular, the expected market share gained by our procedure is larger than that achieved by standard influence maximization without product design, where the product is simply created putting together the most "viral" features.

Acknowledgments. This work was supported by MULTISENSOR project, funded by the European Commission, under the contract number FP7-610411.

References

[1] P. V. S. Balakrishnan and V. S. Jacob. Genetic algorithms for product design. *Manage. Sci.*, 42(8):1105–1117, 1996.

[2] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *ICDM*, 2012.

[3] A. Belloni, R. Freund, M. Selove, and D. Simester. Optimizing product line designs: Efficient methods and comparisons. *Manage. Sci.*, 54(9):1544–1552, 2008.

[4] A. Berger. The improved iterative scaling algorithm: A gentle introduction. 1997.

[5] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, 2010.

[6] M. Das, G. Das, and V. Hristidis. Leveraging collaborative tagging for web item design. In *KDD*, 2011.

[7] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.

[8] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. A data-based approach to social influence maximization. *PVLDB*, 5(1):73–84, 2011.

[9] P. Green and V. Rao. Conjoint measurament for quantifying judgmental data. *Journ. of Marketing Research*, 8(3):355–363, 1971.

[10] D. Gunnec and S. Raghavan. Integrating social network effects in the share-of-choice problem. Unpublished working paper. `http://terpconnect.umd.edu/~raghavan/preprints/snesoc.pdf`.

[11] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[12] R. Kohli and R. Krishnamurti. Optimal product design using conjoint analysis: Computational complexity and algorithms. *Europ. Journ. of Operational Research*, 40(2):186–195, 1989.

[13] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.

[14] M. Miah, G. Das, V. Hristidis, and H. Mannila. Determining attributes to maximize visibility of objects. *IEEE Trans. on Knowl. and Data Eng.*, 21(7):959–973, 2009.

[15] E. Mossel and S. Roch. Submodularity of influence in social networks: From local to global. *SIAM J. Comput.*, 39(6):2176–2188, 2010.

[16] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - i. *Mathematical Programming*, 14(1):265–294, 1978.

[17] K. Nigam. Using maximum entropy for text classification. In *IJCAI*, 1999.

[18] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES*, 2008.

[19] A. Sinan and W. Dylan. Creating social contagion through viral product design: A randomized trial of peer influence in networks. *Manage. Sci.*. Forthcoming.

[20] A. Singh, A. Guillory, and J. Bilmes. On bisubmodular maximization. *Journal of Machine Learning Research*, 2012.

[21] S. Tsafarakis and N. Matsatsinis. Designing optimal products: Algorithms and systems. In *Marketing Intelligent Systems Using Soft Computing*, 2010.