УДК 004.056

Моделирование информационного противоборства в социальных сетях на основе теории игр и динамических байесовских сетей

© С.В. Вельц

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

В статье рассматривается задача моделирования информационного влияния и противоборства в социальных сетях. Актуальность задачи обусловлена возрастающим влиянием социальных сетей на процессы в обществе и возрастающей конкуренцией в информационном пространстве. В данной работе предложен подход к решению указанной задачи на основе теории игр Штекельберга и динамических байесовских сетей. Также предлагается иерархический алгоритм оценки информационного влияния, что позволяет существенно ускорить вычисления в случае больших сетей. Практическая применимость подхода проверена в рамках вычислительного эксперимента на синтетических данных и данных сети Twitter. Предлагаемый подход обладает большой гибкостью и производительностью, что дает возможность решать широкий круг задач и делает перспективным его использование при построении информационно-аналитических систем.

Ключевые слова: *социальные сети, теория игр, динамические байесовские сети, максимизация информационного влияния.*

Введение. Сеть — это множество элементов и связей между ними. Стоит отметить важность изучения сетевых структур для решения задач информационной безопасности, включая, обнаружение вторжений в компьютерные системы, расследование инцидентов, борьба с ботнетами, изучение динамики распространения вирусов, слежение за телефонными и социальными сетями.

Начиная с конца 90-х годов изучается задача моделирования информационного влияния и управления в социальных сетях. Важность этой задачи обусловлена тем, что социальные сети слабо регулируются государством, в сети Интернет существует относительная анонимность, распространение информации часто носит вирусный характер («сарафанное радио») и ярко выражены явления самоорганизации и взаимного доверия.

В качестве практических примеров информационного противоборства в социальных сетях можно привести: события «арабской весны», в которой важную роль сыграли сервисы Twitter и Facebook [1]; использование социальных сетей в маркетинге и для конкурентной борьбы [2,3,5]; миротворческие операции и операции по подавлению восстаний [7].

В данной области можно выделить следующие современные направления исследований: построение моделей влияния (информационных каскадов (IC) [2,8], линейных порогов (LT) [2,8], вероятностные модели [3,8,11]); построение эффективных алгоритмов максимизации влияния (на основе аппарата субмодулярных функций (жадный алгоритм) и его улучшения, CELF [10], CELF++ [13]); с использованием локальных свойств графа (LDAG [14], SimPath [15]); прорежение графа [16]; имитация отжига [17]; алгоритмы оптимизации мониторинга сети [10]; вариации задачи максимизации влияния и алгоритмы решения (максимизация блокирования влияния[18], максимизация влияния с учетом времени [19], тематическое распространение влияния [20]); теоретико-игровые модели информационного влияния [23,24].

В данной работе предлагается новый поход к моделированию информационного противоборства в социальных сетях. Он заключается в использовании теории игр для нахождения оптимальных стратегий сторон информационного конфликта и вероятностных моделей [9] для описания информационного влияния и введения целевых показателей стратегий игроков.

Научная новизна работы заключается в предлагаемой вероятностной модели влияния, которая основана на динамических байесовских сетях и алгоритме иерархической выработки стратегии игроков.

Состоятельность результатов данной работы и их применимость на практике показана в вычислительном эксперименте. На основе реальных данных из социальных сетей (Twitter, arXiv) был получен оптимальный (с теоретико-игровой точки зрения) план их мониторинга.

Статья имеет следующую структуру. В разделе 1 описана структурная схема системы моделирования, описаны компоненты и принципы их работы. В разделе 2 приведены результаты вычислительного эксперимента. В заключении представлены выводы работы, включая возможные направления дальнейших исследований.

Моделирование влияния в социальных сетях. Постановка задачи. Рассмотрим игру двух игроков А и Б, которые борются за влияние в обществе (например, репутацию некоторой компании или политические настроения). Данное информационное противоборство отражается в некоторых сетях, например Twitter, Facebook, телефонной сети. Игрок А (атакующий) может выбрать начальное множество узлов сети, которые переводятся в активное состояние и распространяют выгодную для игрока А информацию. При этом игрок А может выбрать ограниченное количество начальных узлов.

Игрок Б (защищающийся) наблюдает за сетью, используя ограниченное количество ресурсов (вычислительные мощности и персонал).

Поскольку атакующий потенциально может наблюдать за действиями по мониторингу сетей, используется теоретико-игровая модель «лидер — последователь», также известная как модель Штекельберга (Stackelberg). Данная модель часто применяется к задачам, связанным с безопасностью [22].

Требуется определить смешанные стратегии игроков — набор узлов для атаки (для игрока А) и набор узлов для мониторинга (для игрока Б). Игра продолжается до тех пор, пока не будет достигнут горизонт моделирования или игрок Б не обнаружит факт атаки со стороны игрока А.

Введем следующие обозначения:

$G = (V, E)$ — учетные записи в сети и связи между ними;

T — горизонт моделирования;

X_t^i — состояние, в котором находится объект $v_i \in V$ в момент времени t. Допустим, что возможно только 2 состояния: $X_t^i = 1$, если узел скомпрометирован, иначе, $X_t^i = 0$;

$O(t): V \to X_t$ — данные наблюдений для сети в момент времени t.

$S \subset V$ — множество начальных вершин для атаки игрока А.

$M \subset V$ — множество вершин, наблюдаемых игроком Б.

$P(S)$ и $P(M)$ — распределения вероятностей.

Требуется найти оптимальные стратегии игроков: множество атакуемых вершин и множество M узлов для мониторинга:

$$A: \max_{P(S), s.t. |S| \le R_A, P(M)} Q_A(S, M), \tag{1}$$

$$B: \max_{P(M), s.t. |M| \le R_B, P(S)} Q_B(S, M). \tag{2}$$

где R_A, R_B — доступное количество ресурсов у игроков, $Q_A(S, M)$, $Q_B(S, M)$ — целевые функции игроков (будут описаны далее).

Для решения этих задач предлагается система, схема которой показана на рис. 1.

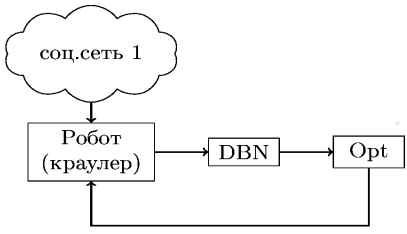

Рис. 1. Структурная схема системы

Кратко опишем принцип функционирования каждой из указанных компонент.

Робот (краулер). Робот (краулер) выполняет следующие функции: во-первых, реализует протоколы общения с социальными сетями (например, RESTful API); во-вторых, многие сайты имеют ограничения на количество запросов в единицу времени, поэтому робот использует несколько серверов с разными IP адресами для обхода этих ограничений; в-третьих, при получении информации происходит извлечение признаков и приведение сообщения к стандартному виду, пригодному для сохранения в БД (преобразуют JSON или XML в вектора признаков).

Вероятностная модель на основе динамической байесовской сети. Байесовская сеть — это графическая вероятностная модель, представляющая собой ациклический ориентированный граф (DAG), где вершины соответствуют случайным величинам, а ребра показывают условную зависимость случайных величин между собой. Динамическая байесовская сеть (DBN) — байесовская сеть для меняющегося во времени процесса, полученная связыванием байесовских сетей для каждого отдельного момента времени между собой за счет учета условной зависимости случайных величин в разные моменты времени.

Структура модели показана на рис. 2.

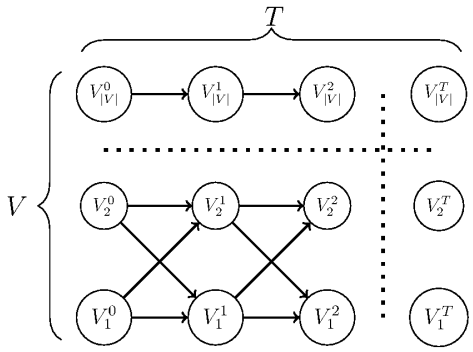

Рис. 2. Структура динамической байесовской сети

Множество вершин V_{DBN} разбивается на *слои* (непересекающиеся подмножества), соответствующие моментам времени $t = 1,...,T$:

$$V_{DBN} = V^1 \cup V^2 \cup \ldots \cup V^T \tag{3}$$

Внутри одного слоя ребер нет. Ребра идут от слоя V^{t-1} к слою V^t. Ребро существует в одном из двух случаев: между верши-

нами в сети G было ребро; вершины слоев V^{t-1} и V^t соответствуют одной и той же вершине в G.

Для задания DBN также необходимо описать условные распределения вероятностей между вершинами. На практике используют следующие решения:

• табличные распределения, заданные экспертом. Данный вариант хорошо подходит для небольших сетей, но в масштабах рассматриваемой задачи неприменим;

• обучение по данным [26]. Требует наличия качественных данных большого объема;

• моделирование Монте-Карло. Предполагается независимость вершин и принадлежность вероятности $p(u,v)$ равномерному распределению $U(0,1)$ [2];

• эвристические методы. Вероятности активации выбираются специальным образом, для получения некоторых полезных свойств (например, $p(u,v)=1/d_v$ [2]).

В данной работе используется метод моделирования Монте-Карло.

На основе описанной модели вводится следующий показатель качества стратегии: математическое ожидание суммы количества активных вершин в каждый момент времени до детектирования атаки.

Вероятности состояний вершин вычисляются по следующей формуле:

$$P\left(S_t^u=1\right)=1-P\left(S_{t-1}^u=0\right)\prod_{v\in V_{t-1},\exists p(v,u)\in E}P\left(S_{t-1}^v=0\right)+ \\ +P\left(S_{t-1}^v=1\right)\left(1-p\left(v,u\right)\right), \tag{4}$$

где $P(S_0)$ — априорные вероятности состояния вершин.

Вероятность детектирования атаки на шаге t (детектирование происходит, если хотя бы одна из вершин, находящихся под наблюдением, становится активна):

$$P_{alarm}(t)=\prod_{v\in V}P_M(v)P\left(S_t^v=1\right) \tag{5}$$

Вероятность того, что в момент времени t атака еще не обнаружена:

$$P_{cont}(t) = P_{cont}(t-1)\left(1 - P_{alarm}(t)\right), P_{cont}(0) = 1 \quad (6)$$

Функционал качества:

$$Q(S,D) = \sum P_{\text{cont}}(t)\left(\sum P\left(S_t^v = 1\right)\right), \quad (7)$$

где

$$Q_A(S,D) = Q(S,D), \quad (8)$$

$$Q_B(S,D) = -Q(S,D). \quad (9)$$

Оптимизация мониторинга. Модуль оптимизации решает задачу нахождения оптимальной стратегии мониторинга социальной сети (для игрока Б) и оптимальной стратегии атаки (для игрока А). Эта задача разбивается на две: нахождение равновесия в смешанных стратегиях и эффективное нахождение чистых стратегий (максимизация функционалов 8 и 9).

Алгоритм поиска равновесия в смешанных стратегиях. При применении модели Штекельберга к поставленной задаче возникает проблема с заданием игры. Рассмотрим упрощенный случай, когда защитник выбирает k узлов из N для мониторинга в начальный момент времени и на протяжении игры они остаются фиксированными. Аналогично атакующий выбирает m узлов, из которых будет начата атака. В этом случае матрица выигрышей будет иметь размерность C_N^k на C_N^m. Для значений параметров, которые на $N > 1000$, $k \sim 100$, $m \sim 100$ практике можно считать весьма скромными, данная задача вычислительно неразрешима. Поэтому требуются специальные методы решения.

Избежать представления игры в нормальной форме можно за счет подхода на основе двух оракулов [23]. Отличие данной работы от [24] заключается в использовании нового иерархического алгоритма в качестве оракулов.

Суть метода двух оракулов заключается в постепенном построении смешанных стратегий для каждого из игроков, в предположении, что у нас есть некоторый алгоритм (оракул), который вычислит оптимальную чистую стратегию в ответ на смешанную стратегию противника.

Алгоритм 1 . Алгоритм поиска оптимального плана мониторинга с двумя оракулами.

Вход: G

Выход: смешанные стратегии защитника и атакующего

1. Инициализируем M произвольной стратегией (распределением ресурсов) защитника.

2. Инициализируем S произвольной статегией атакующего.

repeat

$(m,s) \leftarrow CoreLP(M,S)$

$\alpha \leftarrow DO(S)$

$M \leftarrow M \cup \{\alpha\}$

$\gamma \leftarrow AO(M)$

$S \leftarrow S \cup \{\gamma\}$

until условие сходимости

Условие сходимости выполняется, когда ни один из оракулов DO и AO не может вычислить чистую стратегию, которая была бы лучше, чем лучшая смешанная стратегия соответствующего игрока при фиксированной текущей лучше стратегии противника.

В качестве подпрограммы CoreLP может использоваться любой метод решения задачи линейного программирования, например, симплекс-метод. В [27] доказана корректность алгоритма двух оракулов.

Алгоритм поиска чистых стратегий. Чистые стратегии представляют из себя множества вершин либо для атаки, либо для мониторинга. Задача поиска этих множеств может быть сведена к задаче о вершинном покрытии (SCP или VCP), которая является NP-полной, следовательно необходим эффективный приближенный алгоритм ее решения. На практике применяются варианты жадного алгоритма (CELF, LDAG, SimPath). В данной работе предлагается новый подход, основанный на иерархическом представлении графа. Пример такого представления показан на рис. 3. Подход позволяет, во-первых, существенно ускорить жадный алгоритм, во-вторых, гибко выбирать компромисс между точностью и скоростью вычислений. Вычислительные эксперименты показывают, что при потере точности около 1%, алгоритм быстрее CELF в десятки раз.

Алгоритм построения иерархического представления показан в листинге 2. Основная идея заключается в решении задачи SCP для графа и создании из полученного решения нового графа, который служит новым слоем в представлении. Данный процесс продолжает-

ся до тех пор, пока в слое не останется одна вершина, либо мы уже не сможем уменьшить количество вершин (эта ситуация может возникнуть, если граф состоит из нескольких связных компонент, не связанных между собой).

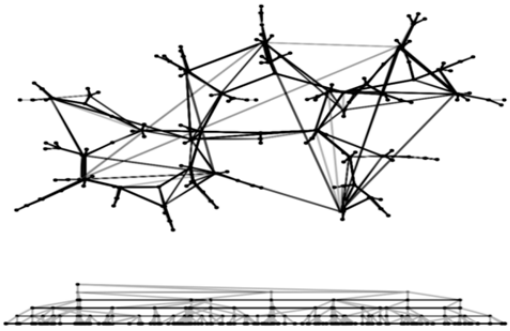

Рис. **3.** Пример иерархического представления графа

Каждый слой в представлении состоит из: графа $G_k = (V_k, E_k)$, который задает структуру данного слоя; отображения вершин данного слоя в родительские вершины в следующем слое $parents_k : V_k \to V_{k+1}$; отображения вершин данного слоя в множество вершин предыдущего слоя, которые покрываются данной вершиной $children_k : V_k \to V_{k-1}$; весов вершин, которые равны сумме весов покрываемых вершин: $weight_k : V_k \to R_+$ (для нижнего слоя все веса равны единице).

Алгоритм 2. buildHierarchicalGraph – жадный алгоритм построения иерархического представления графа.

Вход: $G = (V, E), w: E \to [0,1]$.

Выход: 2

$HG = (G_0 = (V_0, E_0), ..., G_L = (V_L, E_L))$, L — количество слоев,

$parents_k : V_{k-1} \to V_k, k = 1...L$

$children_k : V_k \to V_{k-1}, k = 1...L$

$weight_k : V_k \to R_+, k = 1...L$

1. Инициализация

$HG \leftarrow G$

$k \leftarrow 0$

$G_k \leftarrow G$

$weight_k(i) = 1, \quad i = 1..(V_k)$

2. Построение слоев.

while $|V_k| > 1$ **do**

$X \leftarrow solveVertexCoverProblem(G_k)$

if $|V_k| = |X|$ **then** уменьшить слой не удалось

break

end if

$covered \leftarrow \varnothing$ Задаем родительские связи между вершинами слоев

$children_{k+1}(i) \leftarrow \varnothing, i = 1..|X|$

$weight_{k+1}(i) \leftarrow 1, i = 1..|X|$

for $v \in X, u \in neighbours(v)$ **do**

if $u \notin covered$ **then**

$parents_k(u) \leftarrow v$

$children_{k+1}(v) \leftarrow children_{k+1}(v) \cup \{u\}$

$weight_{k+1}(v) \leftarrow weight_{k+1}(v) + weight_k(u)$

$covered \leftarrow covered \cup \{u\}$

end if

end for

$E_{k+1} \leftarrow \varnothing$ Создаем ребра между вершинами нового слоя

for $v \in X$ **do**

for $u \in neighbours(v)$ **do**

if $v = parents_k(u)$ **then** не создаем петли

continue

end if

запоминаем ребро (v,u) в списке ребер ведущих из v в $parents_k(u)$

end for

for $u \in X$ **and** список ребер из v в u не пуст

$E_{k+1} \leftarrow E_{k+1} \cup \{(v,u)\}$

$p(u,v) = 1 - \prod(1 - p(e))$ вычисляем вероятность активности ребра

end for

end for

$V_{k+1} \leftarrow X$

$HG \leftarrow HG \cup (V_{k+1}, E_{k+1})$ запоминаем созданный слой

$k \leftarrow k+1$

end while

После того, как иерархическое представление построено, для оптимизации используется жадный алгоритм. Алгоритм движется от меньших слоев (верхних) к большим (нижним), пока не достигнет начального графа. Для каждого слоя используется жадная стратегия выбора вершин с учетом их весов. При этом используется следующая эвристика: в выборе участвуют только вершины, чьи родители были выбраны в предыдущем (вышележащем) слое. Данный алгоритм показан в листинге 3.

Алгоритм 3. Иерархический алгоритм поиска оптимальной стратегии.

Вход: $G = (V, E)$, $P(M)$ (для А) или $P(S)$ (для Б), R

Выход: A – множество узлов

$HG \leftarrow buildHierarchicalGraph(G)$ Строим иерархическое представление графа

$k \leftarrow 0$ Находим начальный уровень в иерархии

while $k < HG.numberOfLayers \wedge H.G[k].numberOfNodes > 2R$ **do**

$k \leftarrow k+1$

end while

$disabledNodes \leftarrow \varnothing$ Для каждого из слоев решаем задачу максимизации

$A \leftarrow \varnothing$

for $i \leftarrow k, 0$ **do**

$A \leftarrow maximize(HG[k], P(M) \vee P(S), R, disabledNodes, weight_k)$

if k=0 **then**

break

end if

$disabledNodes \leftarrow \{v \in V_{k-1} : parent(v) \notin A\}$

end for

return A

Результаты вычислительного эксперимента. Вычислительный эксперимент состоял из двух частей. В первой части было проведено

сравнение точности и времени выполнения алгоритмов CELF и предложенного алгоритма HA. Использовались синтетические данные в виде случайных графов (модель Эрдеша-Реньи). Результаты сравнения показаны на рис. 4. Видно, что при незначительных потерях точности алгоритм HA значительно быстрее. Стоит отметить, что при малом количестве ребер в графе точность алгоритма HA снижается. В этой ситуации помогает увеличение коэффициента размера начального слоя.

Во второй части был протестирован алгоритм выработки смешанных стратегий на реальных данных сети микроблогов Twitter (11 тыс. учетных записей, 25 тыс. связей) (подмножество данных из [28]). В качестве подпрограммы CoreLP использовалась библиотека COIN-OR. Вычисление 100 оптимальных узлов для мониторинга заняло 1721 мин (28.7 ч).

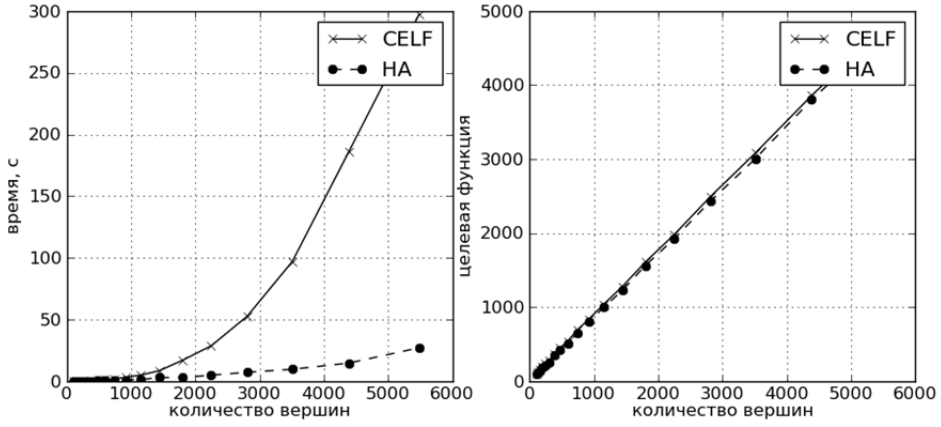

Рис. 4. Сравнение времени выполнения и точности алгоритмов CELF и HA

Заключение. Преимуществом предложенного подхода является возможность вероятностного вывода в задачах информационного противоборства. Это позволяет не только решить задачу оптимального мониторинга социальных сетей, но и ставить новые задачи. Например, обратную задачу — исследование инцидентов, когда по данным наблюдений мы пытаемся восстановить начальное воздействие на социальную сеть. Кроме того, перспективно развитие данной работы в направлении исследования кооперативных игр. Улучшение предложенного иерархического алгоритма нахождения стратегий возможно за счет использования более эффективных алгоритмов, вместо жадного алгоритма (например, использование идей из алгоритма CELF), и эвристик при построении иерархического представления зависящих от структуры входных данных.

ЛИТЕРАТУРА

[1] Khondker H. H. Role of the New Media in the Arab Spring. *Globalizations*, 2011, vol.8 (5), pp. 675–679.

[2] Kempe D., Kleinberg J., Tardos É. Maximizing the spread of influence through a social network. *Proceeding KDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining,* 2003, ACM New York, NY, USA, pp. 137–146.

[3] Domingos P., Richardson M. Mining the Network Value of Customers. *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, 2002.

[4] Goldenberg J., Libai B., Muller E. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Journal Marketing Letters,* 2001, vol. 12, Issue 3.

[5] Kempe D., Kleinberg J., Tardos E. Influential nodes in a diffusion model for social networks. *32nd International Colloquium on Automata, Languages and Programming (ICALP)*, 2005, pp. 1127–1138.

[6] Hill S., Provost F., Volinsky C. Network-based marketing: Identifying likely adopters via consumer networks. *Journal of Computational and Graphical Statistics*, 2006, 21(2), pp. 256―276.

[7] U.S. Dept. of the Army and U.S. Marine Corps. The U.S. Army. Marine Corps Counterinsurgency Field Manual. University of Chicago Press, 2007, pp. 3–24.

[8] Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. *Социальные сети: модели информационного влияния, управления и противоборства*. Москва, Физматлит, 2010, 228 с.

[9] Рассел С., Норвиг П. *Искусственный интеллект: современный подход*. Москва, Вильямс, 2006, 1406 с.

[10] Leskovec J. Cost-effective outbreak detection in networks. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2007, pp. 420–429.

[11] Zhang D., Gatica-perez D., Bengio S., Roy D. Learning influence among interacting Markov chains. *Advances in Neural Information Processing Systems,* 2005, 18.

[12] Pelkowitz L. A continuous relaxation labeling algorithm for Markov random fields. *IEEE Transactions on Systems, Man and Cybernetics*, 1990, vol. 20, pp. 709–715.

[13] Goyal A., Lu W., Lakshmanan L.V.S. *CELF++: Optimizing the greedy algorithm for influence maximization in social networks*, 2011.

[14] Chen W., Yuan Y., Zhang L. *Scalable influence maximization in social networks under the linear threshold model*. ICDM, 2010.

[15] Goyal A. SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. *Proceeding ICDM '11 Proceedings of the 2011 IEEE 11th International Conference on Data Mining,* 2011.

[16] Mathioudakis M., Bonchi F., Castillo C., Gionis A., Ukkonen A. *Sparsification of influence networks*. KDD, 2011, pp. 529–537.

[17] Jiang Q. Song G., Cong G., Wang Y., Si W., Xie K. *Simulated Annealing Based Influence Maximization in Social Networks*, AAAI, 2011.

[18] He X., Song G., Chen W., Jiang Q. Influence blocking maximization in social networks under the competitive linear threshold model. *Proceedings of the 12th SIAM International Conference on Data Mining (SDM'2012)*, 2012.

[19] Chen W., Lu W., Zhang N. Time-critical influence maximization in social networks with time-delayed diffusion process. *Proceedings of the 26th Conference on Artificial Intelligence (AAAI'2012)*, 2012.

[20] Tang J. *Social influence analysis in large-scale networks*, KDD, 2009.

[21] Goyal A. *Learning Influence Probabilities In Social Networks Proceedings of the Third ACM international conference on Web search and data mining,* 2010.

[22] M. Jain, J. Pita, M. Tambe, F. Ordóñez, P. Paruchuri, S. Kraus. Bayesian Stackelberg games and their application for security at Los Angeles international airport. Newsletter ACM SIGecom Exchanges, 2008.

[23] Jain M., Korzhyk D., Vaněk O., Conitzer V., Pěchouček M., Tambe M. A double oracle algorithm for zero-sum security games on graphs. *Proceeding AAMAS '11 The 10th International Conference on Autonomous Agents and Multiagent Systems*, 2011, vol. 1.

[24] Tsai J., Nguyen T. H., Tambe M. *Security Games for Controlling Contagion*, AAAI, 2012.

[25] Conte D., Foggia P., Sansone C., Vento M. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004, vol. 18, no. 3, pp. 265–298.

[26] Goyal A., Bonchi F., Lakshmanan L. Learning influence probabilities in social networks. *WSDM '10 Proceedings of the Third ACM international conference on Web search and data mining*, 2010, pp. 241–250.

[27] McMahan, Gordon, Blum. *Planning in the presence of cost functions controlled by an adversary*. ICML, 2003, pp. 536–543.

[28] Zafarani R., Liu H. Social Computing Data Repository at ASU. URL: http://socialcomputing.asu.edu

Статья поступила в редакцию 28.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Вельц С.В. Моделирование информационного противоборства в социальных сетях на основе теории игр и динамических байесовских сетей. *Инженерный журнал: наука и инновации*, 2013, вып. 11. URL: http://engjournal.ru/catalog/it/security/991.html

Вельц Сергей Владимирович родился в 1987 г., окончил МГТУ им. Н.Э. Баумана в 2011 г. Аспирант кафедры «Информационная безопасность» МГТУ им. Н.Э. Баумана. Область научных интересов: применение методов машинного обучения и теории игр в задачах информационной безопасности, создание распределенных отказоустойчивых систем. e-mail: svelts@gmail.com