FluidRating: A Time-Evolving Rating Scheme in Trust-based Recommendation Systems Using Fluid Dynamics

Wenjun Jiang[†‡], Jie Wu[‡], Guojun Wang[†], and Huanyang Zheng[‡]
[†]School of Information Science and Engineering, Central South University, Changsha, Hunan Province, 410083, P. R. China
[‡]Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA

***Abstract*—The goal of a trust-based recommendation system is to predict unknown ratings based on the ratings expressed by trusted friends. However, most of the existing work only considers the ratings at the current time slot. In real life, a user receives the influence of different opinions sequentially; accordingly, his opinion evolves over time. We propose a novel rating prediction scheme, *FluidRating*, which uses fluid dynamics theory to reveal the *time-evolving* formulation process of human opinions. The recommendation is modeled as fluid with two dimensions: the temperature is taken as the "opinion/rating," and its volume is deemed as the "persistency," representing how much one insists on his opinion. When new opinions come, each user refines his opinion through a round of fluid exchange with his neighbors. Opinions from multiple rounds are aggregated to gain a final prediction; both uniform and non-uniform aggregation are tested. Moreover, Three sampling approaches are proposed and examined. The experimental evaluation of a real data set validates the feasibility of the proposed model, and also demonstrates its effectiveness.**

Keywords*—*Fluid dynamics theory, rating prediction, time-evolving, trust-based recommendation system.

I. Introduction

High-quality and personalized recommendations are a key feature in many online systems [1]–[3]. To recommend an item that a user may be interested in, explicit knowledge of social network structures, such as the trust relationships, can be incorporated. In recent years, trust-based recommendation systems have gained significant attention [1], [4]–[6]. Such systems use the knowledge of a trust network among users, to provide personalized recommendations by aggregating the opinions of their trusted friends. Several models have been proposed to aggregate trust information among trusted friends [7], such as TidalTrust [8], MoleTrust [9], FlowTrust [10], and RN-Trust [11]. These models work in one round, i.e., only the *current* trust information is considered, or, the information is taken as static. In real life, a user's opinion evolves with time, because he receives the influence of different opinions at different times, either directly from connected friends or indirectly from friends of friends. Therefore, going one step further, we propose a rating prediction scheme, *FluidRating*, to simulate the *time-evolving* opinion formulation process as fluid flows, using the fluid dynamics theory.

In this paper, we consider the setting where there is a single item of interest (e.g., a product). A subset of users (*raters*, denoted as R) have prior opinions about this item. A special non-rater, the *sink* (denoted as S), is one whose rating is being predicted. The remaining non-raters (denoted as N) have not formed their opinions, but can propagate the opinions from R to S. In addition, we convert the trust relations into the influence relations, which are based on the following intuition: the more user a' trusts user a, the higher the probability is that the opinion of a can influence a'. Then, the three sets of users, R, N, and S, and the influence relations among them, are used to construct a *rating network*. Fig. 1 shows an example of rating network, where $R = \{a_1, a_2\}$, $N = \{a_3, a_4, a_5\}$, and $S = \{a_6\}$. The number associated with a node in R corresponds to a *rating*. The higher the number is, the higher the rating is. The number associated with each edge represents the *influence strength* from one user to another, which is determined by the trust relationship between them.

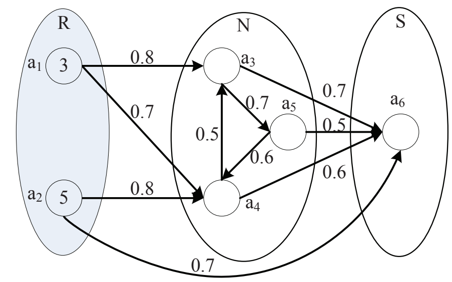


Fig. 1. An example of a rating network, where nodes represent users; numbers on nodes represent ratings on a given target item, and weighted edges represent influence relations.

We consider this question: *Upon receiving new opinions, how will a user refine his opinion*? For instance, for a given target item, a person first receives a positive opinion from a friend, and forms an initial opinion; some time later, he receives a negative opinion from another friend. Will the user change his initial opinion? We observe in real life that whether the user changes his opinion or not depends on how much he insists on his own opinion, as well as how much the others insist on theirs. We call this feature the *persistency*, and take it into consideration in the FluidRating scheme.

Main Ideas. The proposed FluidRating is a computational model using the fluid dynamics theory. A rating network is modeled as a fluid dynamics system: each node corresponds to a *container* with uniform size and unlimited volume. Each influence edge corresponds to a *pipe* connecting two containers; Pipes are installed at the bottom of the containers. The recommendation (or opinion influence) from friends is captured as *fluid*, which has two dimensions: the *temperature* is taken as the "opinion/rating," and its *volume* is deemed as the "persistency." Fig. 2 shows the resulting fluid dynamics system of the rating network in Fig. 1.

In FluidRating, fluids originate from raters and pass through non-raters. When there exists a fluid height difference

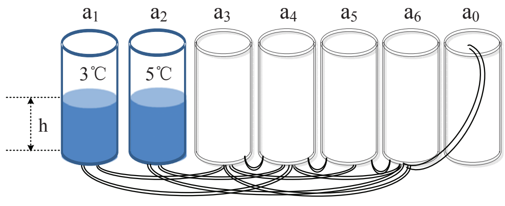


Fig. 2. The illustration of FluidRating. Each node corresponds to a container. The fluid temperature in the container is its "opinion/rating," and volume is the "persistency" of its opinion. The direction of fluid flow is consistent with that of influence, and is controlled by a one-way valve.

between two connecting containers, fluid will flow from one container to another, according to the fluid dynamics theory. Then, when fluid flows, the fluids flowing into a container will mix with the existing fluids, and the fluid temperature and volume in this container will change, which reflects the time-evolving properties of opinion formulation. Eventually, fluid will flow into the sink (i.e., a_6 in Fig. 1). In this paper, we adopt a discretized approach to computing the temperature change of each container over the *round*. The final rating of the sink is a collection of sampled temperatures. Among many desirable feasibilities, FluidRating can capture the effect of *first influence* through the shortest path from a rater.

Our contributions are threefold: (1) We present a clean-slate computational model, FluidRating, based on the fluid dynamics theory that can capture many subtle details in a time-evolving recommendation system. Instead of only considering a static influence at the current time slot, the model takes the opinion refinements over time into consideration, which clearly reveals the time-evolving formulation process of human opinions. (2) We take a discretized approach for efficient calculation. We also differentiate direct influence from directly connected friends and indirect influence from friends of friends. Both influence value and features, such as the persistency of a user over that of his neighbor, are captured through a simple system consisting of containers and pipes only. (3) We conduct extensive experiments in a real data set (Epinions.com), which validate the feasibility of the proposed model and demonstrate its effectiveness. The effects of several factors are tested, including the number of rounds, the duration of time slots, the weight sequence, and the sample approaches.

The remainder of this paper is organized as follows: Section II surveys related work. Section III states the problem we address. Sections IV and V present the overview of the model and the algorithm details, respectively. Section VI analyzes the properties of FluidRating. Section VII describes the experimental evaluation. Finally, Section VIII concludes this paper and suggests future work.

II. Related Work

We review the related literature of trust models, user opinions, and social influence used in recommendation systems.

Trust model. Trust models usually conduct trust inference based on trusted graphs [12], which can be shortest and strongest paths [8], or paths with a restricted depth [9]. Existing models calculate trust values in one round at the current time slot [8]–[10]. Personalized PageRank [1] is conducted with multiple rounds, based on the Markov chain model.

TABLE I. Notations.

Symbol	Description
$G=(V,E)$	rating network with node set V and edge set E
R	rater set $\{a_1, a_2, ..., a_m\}$
N	non-rater set $\{a_{m+1}, ..., a_{n-1}\}$
S	set $\{a_n\}$ with sink a_n
$V=R\bigcup N\bigcup S$	node set with a total of n nodes
$a/a'/a''$	node/outgoing neighbor/incoming neighbor of a
$N_a/N_a^{out}/N_a^{in}$	whole/outgoing/incoming neighbor set of a
$e_{aa'}/w_{aa'}/v_{aa'}$	edge from a to a'/weight $w_{aa'}$/flow velocity $v_{aa'}$
i/Δ	sample index/sample interval
$h_a(i)/s_a(i)/t_a(i)$	height/volume/temperature in a at the i'th sample
$s_{aa'}(i)/t_{aa'}(i)$	volume/temperature from a to a' at the i'th sample

FluidRating uses the fluid dynamics theory to relate time-evolving temperature changes to opinions/rating refinements. It is more powerful in modeling for its ability to describe persistency or other features.

User Opinion/Rating. User opinion is usually represented as a numeric value in online web sites. Anderson et al. [1] uses a finite integer set with {+, -, 0} representing positive, negative, and no (neutral) ratings. In FluidRating, opinion is measured by fluid temperature, which can be easily updated based on the volume and temperature of new fluid.

Social influence. [13] finds that a person's opinion is significantly swayed by others' opinions. [14] validates that stronger ties are individually more influential, while weak ties are responsible for the propagation of novel information. Our FluidRating takes the finding in [13] as a foundation: when new opinions come, each person refines his opinion through a round of fluid exchanges with his neighbors. In addition, the concept of direct and indirect influences are modeled through fluid exchanges among neighbors and neighbors' neighbors; meanwhile, the strength of influence, strong or weak, is modeled by the cross-sectional area of the corresponding pipe.

III. Problem Formulation

We first describe the settings of a trust-based recommendation system. Then, we formulate the problem we address. Notations used in this paper are described in Table I. We first define the rating network as follows:

Definition 1: A rating network is a directed graph $G=(V,E)$ where V is a set of nodes, and $E \subseteq V^2$ is a set of directed edges. Each edge $e_{aa'}$ has the direction from node a to node a', associated with a weight $w_{aa'}$ indicating the influence value from a to a'.

The node set $V = \{a_1, a_2, ..., a_m, a_{m+1}, ..., a_{n-1}, a_n\}$ consists of three types of nodes, *rater*s who have formed their opinions $R=\{a_1, a_2, ..., a_m\}$, *non-rater*s who haven't formed their opinions $N=\{a_{m+1}, ..., a_{n-1}\}$, and the *sink* $S=\{a_n\}$, a special non-rater whose opinion is being predicted. Note that, generally, the sink is also a non-rater. However, in order to isolate the roles of non-raters and the sink as in [1], we put the sink into a single set. Currently, we only collect the temperature of sink for k rounds. A natural extension is to collect the temperatures for all non-raters, where the ratings of multiple non-raters can be predicted at the same time.

Three types of users take different roles: raters (R) have formed their opinions, and thus, serve as the source of opinions. Non-raters (N) connect raters and the sink, and take the role of propagating influence, i.e., they will be influenced by some raters, and then, will propagate the influence to the sink. Sink (S) is the target of the influence; he can refer to multiple opinions from friends, and form his own opinion. There are some key features of the influence among users: (1) The opinions of raters will not change with time, while the opinions of non-raters (including the sink) change. (2) The opinions of raters and non-raters can influence the opinion of a non-rater (including the sink). (3) The influences are independent. The opinions of raters and non-raters influence several non-raters (including the sink), independently. Then, we define the problem as follows:

Time-evolving Rating Prediction Problem. Given a rating network $G = (V, E)$ with $V = R \bigcup N \bigcup S$, and a sink a_n whose rating on a target item needs to be predicted, the task is to design a scheme to predict the rating of a_n efficiently, and to capture the time-evolving opinion formulation process; another task is to refine opinions using the features of users, such as the persistency.

IV. Overview of The Model

We first describe two social principles that our model should obey. Then, we describe the overview of FluidRating, where we use a novel approach of applying the fluid dynamics theory in trust-based recommendation systems.

A. Basic Social Principles

We extract the following two ground truths in real life, which serve as the general rules for the model design:

Principle 1: First Influence Dominates. In real life, the first influence makes more of an impact on people's opinions (the *first impressions* [15] phenomenon in psychology). As mentioned in a proverb, "first impressions are lasting impressions."

Principle 2: Stronger Influence Dominates. When a new opinion influence comes to a user, only if it is stronger than his current opinion, will the user refine his own opinion accordingly.

We will use Principle 1 to guide the selection of aggregation sequence, and Principle 2 to model the refinement of opinion. In a rating network, direct friends' influences come earlier than indirect ones; the influences from 1-hop neighbors are taken as the first influences.

B. Model the Recommendation

We view the time-evolving formulation process of human opinions as follows: each user first receives the influence from directly connected friends, and updates his own opinion accordingly; then, he propagates his opinion to other friends. The process can be done iteratively. In this way, for each user, both the direct influences he receives from connected friends and indirect influences from friends of friends are captured.

Based on this, a rating network is modeled as a fluid dynamics system (Algorithm 1): each user corresponds to a container with uniform size, and with unlimited volume, so that fluid will never overflow. Containers are connected through pipes, which correspond to the influence edges in the rating network. Recommendations are modeled in terms of fluids, which originate from raters, pass through non-raters, and finally reach the sink. The ratings of users are modeled as the fluid temperature, and the persistency of the corresponding user is measured as the fluid volume. Both the direct and indirect influences are modeled through fluid exchanges among connecting containers. Using the basic fluid dynamics theory, we can obtain the speed of efflux, and cope with the updating of fluid temperature and volume in containers.

Algorithm 1 Initialization(G, a_n)

Input: G, a rating network for predicting a_n's rating.
Output: G', a FluidRating system for a_n.

for each rater/non-rater in G **do**
  Set up a container with unlimited volume in G'. Set the fluid temperature in the rater's container to be equal to its rating, and height to h. Let the non-rater's container be empty.
for each influence edge from a to a' in G **do**
  Set up a single-direction pipe from a to a' in G'.

In addition, we adopt a discretized approach to computing the temperature change over the round (or time slot), with each slot having a duration of Δ. A total of k samples of the fluid temperatures in a_n are collected, and are aggregated to get the final temperature (i.e., the ultimate opinion). Then, the process of time-evolving rating prediction is converted into the fluid temperature, and volume updates through multiple rounds.

Assumptions: (1) The whole system is a closed one, in which there is enough fluid to supply each rater's container. (2) The fluid temperature in each container will not change until it is mixed with the incoming fluid. That is, the container, the pipes, and the one-way valve are associated with temperature-insulating material. (3) Raters' ratings and persistencies will not change (similar to [1]). The insight is that, when a user has enough first-hand experience, he will not listen to the others. (4) A rating network is available to be used, based on which, the FluidRating system can be set up. It can be constructed as follows: Add the sink, the neighbors who are raters, and those who can reach raters within a given maximum number of hops, into the rating network; then, add in all edges among those nodes (including the sink).

C. FluidRating System Setup

The FluidRating system consists of three parts: the containers, the pipes between containers, and the fluid flowing among the containers and pipes. Algorithm 1 shows the initialization process. Fig. 2 illustrates an example of the FluidRating system corresponding to the rating network in Fig. 1.

1) The Containers: We relate each node to a container with unlimited volume (i.e., large enough so that fluid can never overflow). All the containers have the same size. The cross-sectional area is denoted as b, which essentially reflects the persuasiveness of the corresponding node, i.e., how effectively he can persuade neighbors. Due to page limitations, in the future we will consider the work of accurately modeling the feature of persuasiveness. Then, we take b as a constant of 1. As a result, the fluid hight is proportional to the fluid volume. Moreover, all containers are put in the same horizontal level, so that they have the same surface air pressure.

(a) (b)

Fig. 3. The one-way valve: (a) a symbol; (b) an example in real life.

2) The Pipes: Each edge $e_{aa'}$ is related to a directional pipe from the container of a to a', the cross-sectional area of which is equal to $w_{aa'}$. The directional pipe is implemented through installing a one-way valve in it. Fig. 3 shows the symbol and a real-world example (from Google Images) of the one-way valve. All the pipes are installed at the bottom of the connecting containers. The only exception is for an additional container a_0, which is used to store the fluid in a_n. A special pipe is installed to absorb all the fluid from a_n to a_0. The setting of directional pipes is based on Principle 2, i.e., stronger influence dominates. Only when a container has a larger fluid height (indicating larger persistency), will the fluid flow to its neighbors and mix with their fluids.

3) The Fluid: We assume that there is a single type of fluid in FluidRating. We maintain the fluids in the container of each rater to be a height of h by injecting fluid continuously (line 2 in Algorithm 1), indicating the stable persistency. Their ratings are initialized as the fluid temperatures. Meanwhile, we assume that initially all non-raters have no opinions on the target item. Then, their containers are set to be empty. We do not consider the external influence [16] that may impact users' opinions.

In Algorithm 1, each rater, non-rater, and the sink is considered once (lines 1-2), the time complexity of which is $O(|V|)$. Each edge is transformed into a pipe (lines 3-4), the time complexity of which is $O(|E|)$. Therefore, the total time complexity of Algorithm 1 is $O(|V| + |E|)$.

D. Basic Physical Principles

Given a FluidRating system as has been set up above, fluid will flow from raters to the sink a_n, directly or via non-raters in N. Before going further into the detailed process, we describe the following two basic physical principles that FluidRating should obey as a closed system:

Principle 3: Mass Conservation [17]. The mass of any closed system must remain constant over time.

Principle 4: Energy Conservation [18]. The total energy of an isolated system cannot change over time.

Since FluidRating takes a single type of fluid which has a constant density and $mass = density \cdot volume$, fluid volume is proportional to the mass. Principle 3 will be used for maintaining the fluid volume conservation when we conduct fluid updates. Principle 4 will be applied to calculate the updated temperature when fluids are mixed.

V. FluidRating: Algorithm Details

The FluidRating scheme consists of three steps: fluid updating preparation, fluid updating execution, and sample aggregation. First, we present how the fluid will flow and how the flowed fluids will mix up, according to fluid dynamics theory. Then, sampled temperatures of multiple rounds are aggregated to gain a final opinion, using a non-increasing weight sequence. The earlier opinions are given larger weights than are the later opinions, based on Principle 1.

Algorithm 2 FluidRating(G', a_n)

Input: G', a FluidRating system for predicting a_n's rating.
Output: t_{a_n}, a_n's temperature/rating.

```
Let k be the total number of samples (time slots).
for i = 0 to k do
    for each pipe from a to a' do
        if h_a(i) > h_a'(i) then
            Record the volume and temperature of the flowed
            fluid (Eqs. 1 and 2).
    for each rater's container do
        Fill fluid into it to maintain its height/temperature.
    for each non-rater's container do
        Update fluid height and temperature (Eqs. 4 and 5).
    Record fluid temperature in a_n.
    Draw all the fluid in a_n to a_0.
Aggregate the k fluid temperatures in a_n (Eq. 6).
```

We show how fluid flows among containers and pipes, from the view of a *discrete time system*. Without loss of generality, we consider that the fluid updating is done synchronously at the end of each time slot (k time slots in total). A high-level fluid flowing process is shown in Algorithm 2. Meanwhile, Fig. 4 illustrates the updating process, which corresponds to the example of rating network in Fig. 1. At the beginning of the i^{th} time slot, we prepare fluid updating and check if fluid will flow in each pipe, by comparing the fluid heights of two connected containers. If there is a directional pipe from a to a', and the fluid height in a is larger than that of a', then the fluid will flow from a to a'; if either of the two conditions is not met, no fluid will flow. Then we record the volume and temperature of the flowed fluid, i.e., varied amounts of fluid in containers, in this time slot. At the end of each time slot, we mix up the flowed fluid and the remaining fluid in each container, and conduct fluid updating.

A. Fluid Updating Preparation

First, let us consider a single pipe, say the pipe connecting a and a', with cross-sectional area $w_{aa'}$. At the beginning of the i^{th} time slot, if a has more fluid than a' (i.e., $h_a(i) > h_{a'}(i)$), then the fluid will flow from a to a' during this time slot, with a duration of Δ. The basic theory behind flowed fluid is Torricelli's law [19]. It states that the speed of efflux, v, of a fluid through a sharp-edged hole at the bottom of a tank filled to a depth h is the same as the speed that a body (in this case a drop of water) would acquire in falling freely from a height h, i.e., $v = \sqrt{2gh}$, where g is the acceleration due to gravity. As an application of this law, the speed of flowed fluid in our case will be $v_{aa'} = \sqrt{2g(h_a - h_{a'})}$. Considering the cross-sectional area $w_{aa'}$ and the duration of the time slot Δ, the volume of flowed fluid in this time slot can be calculated as follows:

$$s_{aa'} = v_{aa'} \cdot w_{aa'} \cdot \Delta = \sqrt{2g[h_a(i) - h_{a'}(i)]} \cdot w_{aa'} \cdot \Delta \quad (1)$$

The insight behind Eq. 1 is that, the influence received by a person from a friend is proportional to the square root of their

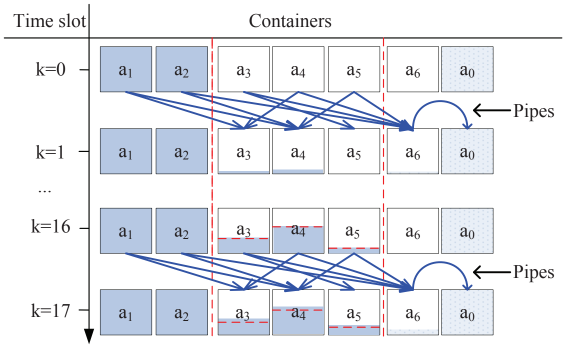


Fig. 4. The illustration of fluid updating from the i^{th} to $(i+1)^{th}$ time slot.

persistency difference, the influence strength from this friend to him, and the duration of the time slot. In another word, the influence strength and the duration of time are proportional to the actual amount of influences. As for the temperature of flowed fluid from a to a', we consider it to be the same as that of a, as in the following:

$$t_{aa'}(i) = t_a(i). \tag{2}$$

B. Fluid Updating Execution

In this subsection, we describe how the flowed fluid mixes up with the existing fluid in the containers. According to the law of mass conservation, the fluid in the amount of $s_{aa'}$, will flow out from a, and flow into a'. For a given container a, at the end of the i^{th} time slot, the volume of fluid in a will be:

$$s_a(i+1) = s_a(i) - \sum_{a' \in N_a^{out}} s_{aa'} + \sum_{a'' \in N_a^{in}} s_{a''a}, \tag{3}$$

where N_a^{out} and N_a^{in} represents the outgoing and incoming containers of a, respectively. As we have mentioned before, fluid height will impact whether fluid will flow, which we calculate as follows:

$$h_a(i+1) = s_a(i+1)/b, \tag{4}$$

where b is the cross-sectional area.

Since we use a single type of fluid, the specific heat is a constant. According to Principle 4, the law of energy conservation, the fluid temperature after mixing up is calculated as follows:

$$t_a(i+1) = \frac{t_a(i) \cdot \left[s_a(i) - \sum_{a' \in N_a^{out}} s_{aa'}\right] + \sum_{a'' \in N_a^{in}} \left[t_{a''a} \cdot s_{a''a}\right]}{s_a(i+1)} \tag{5}$$

Eq. 5 is essentially $\sum(\text{volume} \cdot \text{temperature})/\sum \text{volume}$. The first part of the numerator is the existing fluid in container a, while the second part is the flowed fluid from other containers.

For the fluid system in Fig. 2, its fluid updating process is shown in Fig. 4. The known conditions are: $h_{a_3} = h_{a_4} = h$, $t_{a_3} = 5$ and $t_{a_4} = 3$. During the 1^{st} time slot Δ, due to the height difference of the fluid in two connected containers, the pipes from a_1 to a_3, a_1 to a_4, a_2 to a_4, and a_2 to a_6 will have some flows. We first calculate the volumes and temperatures of fluid that will flow, using Eqs. 1 and 2. Without loss of generality, we let the raters' fluid height be $h = 10$. At the end of the 1^{st} Δ, the heights and the temperatures of fluids in each container can be calculated using Eqs. 4 and 5; the process and results are shown in Table II.

TABLE II. FLUID UPDATING IN THE 1^{st} Δ.

Parameter	Calculation	Value
$s_{a_1a_3}(1)$	$\sqrt{2gh} \cdot 0.8 \cdot \Delta$	0.4482
$s_{a_1a_4}(1)$	$\sqrt{2gh} \cdot 0.7 \cdot \Delta$	0.3922
$s_{a_2a_4}(1)$	$\sqrt{2gh} \cdot 0.8 \cdot \Delta$	0.4482
$s_{a_2a_6}(1)$	$\sqrt{2gh} \cdot 0.7 \cdot \Delta$	0.3922
$t_{a_1a_3}(1)$	$t^{a_1}(0)$	3
$t_{a_1a_4}(1)$	$t^{a_1}(0)$	3
$t_{a_2a_4}(1)$	$t^{a_2}(0)$	5
$t_{a_2a_6}(1)$	$t^{a_2}(0)$	5
$s_{a_3}(1)$	$s_{a_1a_3}(1)$	0.4482
$s_{a_4}(1)$	$s_{a_1a_4}(1) + s_{a_2a_4}(1)$	0.8404
$s_{a_5}(1)$	0	0
$s_{a_6}(1)$	$s_{a_2a_6}(1)$	0.3922
$t_{a_3}(1)$	$s_{a_1a_3}(1) \cdot t_{a_1a_3}(1)/s_{a_1a_3}(1)$	3
$t_{a_4}(1)$	$\frac{s_{a_1a_4}(1) \cdot t_{a_1a_4}(1) + s_{a_2a_4}(1) \cdot t_{a_2a_4}(1)}{s_{a_1a_4}(1) + s_{a_1a_4}(1)}$	4.067
$t_{a_5}(1)$	0	0
$t_{a_6}(1)$	$s_{a_2a_6}(1) \cdot t_{a_2a_6}(1)/s_{a_2a_6}(1)$	5

C. Sample Aggregation

We aggregate the temperature of sink in different time slots to gain a final temperature (i.e., a final opinion or rating). Based on Principle 1 "first influence dominates," a basic aggregation rule is that: the earlier samples take no less weights than do the samples that come later. The insight behind this is that the first impression is not less important than the latter impressions. An example of the sample aggregation is:

$$t_{a_n} = \sum_{i=1}^{k} q^{1+c(i-1)} \cdot t_{a_n}(i), \tag{6}$$

where $q^{1+c(i-1)}$ is the weight for the i^{th} sample, and $\sum_{i=1}^{k} q^{1+c(i-1)} = 1$. Meanwhile, $c \in [0, \infty], q \in [0, 1]$. For example, when $c = 1$, the sequence is nonuniform, $\{q, q^2, q^3, ...\}$. When $c = 0$, the sequence is uniform, $\{q, q, q, ...\}$. When $c = \infty$, the sequence is $\{q, 0, 0, ...\}$. Therefore, we can use different c to control the method of sample aggregation.

Besides the above sample approach (denoted as *FluidRating 1*), we design two additional methods, *FluidRating 2* and *FluidRating 3*, to collect the sampling temperatures of the sink. FluidRating 2 has the same setting as FluidRating 1, that is, at the end of each time slot, a_0 absorbs the fluid in a_n. The difference is that we collect the sample of fluid temperature in a_0, instead of in a_n. This method will mix all the incoming fluids together. As the fluid volume in a_0 gets larger, the later incoming fluids will make less of an impact. FluidRating 3 has a different setting than do both FluidRating 1 and 2. We do not use a_0 to absorb fluid in a_n, we just let a_n keep all the incoming fluid, and we collect the fluid temperature in a_n for k time slots. This method will mix the incoming fluid whose amount is larger than the current amount in a_n. When a_n has enough fluid, no more fluid can flow in and mix with its fluid.

The time complexity of Algorithm 2 can be calculated as follows: in a single time slot, each container and pipe is considered once, the time complexity of which is $O(|V|+|E|)$; there are a total of k time slots; so the final time complexity is $O(k|V| + k|E|)$. Over a time period, the updating of fluid temperature and volume in containers can be deemed as a state transition from one to the next. At a specific time slot, say $t(i)$, only one array is used to record the current fluid state in pipes; two arrays are needed to store the current

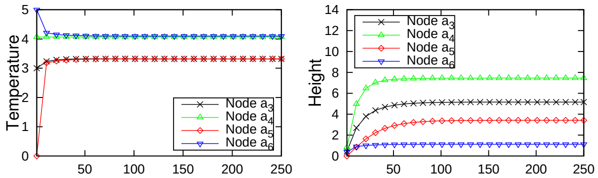


Fig. 5. The fluid temperature and height for the example scenario in Fig. 1, $k = 250, \Delta = 0.04, h = 10, b = 1$.

state (temperature/height) and the next state of containers, respectively. The space cost is $O(|E|)$ for pipes, and $O(|V|)$ for containers. An additional array is used for recording the k samples of fluid temperatures in a_n, with space cost $O(|k|)$. Therefore, the total space complexity is $O(|V| + |E| + |k|)$.

D. Convergence of the Overall Method

As shown in Fig. 5, the height and temperature become stable after some time. The insight behind this phenomenon is the opinion formulation process. At the very beginning, a user (sink) has no opinion on the given target item. Upon receiving opinions from others, he formulates and refines his own opinion. In addition, a person's opinion increases in maturity, indicating the increased persistency. Moreover, the opinion matures quickly at the beginning, and slows down later, i.e., the amount of increased persistency decays with time. This simulation result is consistent with our real-world experiences.

VI. The Analysis

In this section, we analyze the properties of FluidRating on two aspects: its conformity with social and physical principles, and its explainability of recommendation.

A. Conformity with Basic Principles

FluidRating is consistent with both social and physics principles. (1) Conformity with Principle 1: In FluidRating, when aggregating the samples of fluid temperatures, a non-decreasing sequence is used as weight, so that the earliest fluid is being given the highest weight. Since the earlier incoming fluid represents the earlier influence, it emphasizes the importance of the first influence. (2) Conformity with Principle 2: In FluidRating, the larger height indicates stronger persistency. For updating fluid, we look at each pipe. For the pipe from a to a', according to Torricelli's law, only when $h_a > h_{a'}$, will fluid flow, and mix with others. This is consistent with Principle 2, in that only when other's persistency is larger, will the current user take the advice and refine his own opinion. (3) Conformity with Principle 3: Since we consider a single type of fluid in FluidRating, the conservation of volume is equivalent to that of mass. The fluid volume in FluidRating remains unchanged, because for each step of updating, the fluid flows in a pipe are always equal to the fluid flows out of that pipe. Therefore, the total amount of fluid remains unchanged. (4) Conformity with Principle 4: According to the physics, the fluid energy is equivalent to the product of fluid mass, temperature, and specific heat. In FluidRating, with several parts of the fluid mixed, the final temperature is calculated according to the energy before mixture (Eq. 5). Therefore, Principle 4 holds.

B. Explainability of Recommendations

Explainability means that the recommendation system is able to explain how it predicted the rating, to help users understand why particular items have been suggested [5]. In FluidRating, we model the opinions as the temperatures of fluid, as to predict t_{a_n}. In total, k samples of the temperature of fluid flowing into a_n in a given time period are collected. During this process, we can get both the final rating, and the rating variations over time. Positive and negative opinions from trusted friends are modeled as fluids with high or low temperatures. These fluids mix with each other and finally get to steady states, similar to how friends' opinions may influence each other and finally lead to an agreement.

Now we can use these k samples to explain why our prediction is reasonable. We can explain to users that this prediction is based on ratings from their trusted users, according to the time they reach and the persistency they have. A more interesting phenomenon is that, the temperature's first sample equals the temperature of the nearest rater, which stands for the first impression. Essentially, the fluid from the nearest rater arrives at the sink the earliest. In addition, all the three aggregation methods make sense: (1) In *FluidRating 1*, a_n does not keep the fluid, and all later fluids can reach a_n (since $h_a = 0$). We collect the temperature of fluid that reaches a_n (then the fluid flows into a_0). In other words, this approach collects the variation of opinions. (2) In *FluidRating 2*, we collect the fluid temperature in a_0, where all the incoming fluids of a_n are kept. In this way, the sample that we collect each time is actually the temperature of all those mixed fluids. (3) In *FluidRating 3*, sink a_n keeps all the incoming fluids in its own container. We collect the fluid temperature in a_n for k time slots. In this way, when there is enough fluid in a_n's container, no more fluid can flow into it. It makes sense in that, if one has enough persistency, he will not listen to others.

VII. Experimental Evaluation

In this section, we evaluate the performance of FluidRating with experiments in a real social network data set.

A. Experimental Design

We use the leave-one-out method to evaluate the performance [4], [5]. If there is a user providing a rating to an item, the rating is masked and predicted through algorithms based on the rating network. Then, we compare the calculated value with the masked value. The metric of root mean squared error (RMSE) [5] is used to measure the error in rating prediction: $RMSE = \sqrt{\sum (r_{u,i} - \hat{r_{u,i}})^2 / D}$, where D is the total number of of user/item pairs that can be predicted, and $r_{u,i}$ and $\hat{r_{u,i}}$ denote the real and predicted ratings respectively. A smaller RMSE indicates a higher prediction accuracy.

Epinions is a good testbed [4] that is widely used in the research of trust-based recommendation. This is because it includes both the information of user trust relationships and user/item ratings. Users can review items and assign them numeric ratings in the range of [1,5]. They can also build their own trust network by adding the people whose reviews they think are valuable. We use the data set published by Massa [4]. It consists of 49,290 users who rated a total of 139,738 different items at least once. The total number of

TABLE III. THE COVERAGE WITH DIFFERENT LENGTH.

Length	1	2	3	4	5	6
Coverage(%)	62.49	78.84	85.74	89.76	96.4	100

TABLE IV. PARAMETER SETTING.

Parameter	Description	Default value
h	fluid height in rater's container	10
b	cross-sectional area of containers	1
k	number of interations	250
Δ	time slot	0.04
c	(non-)uniform aggregation	(1)0
q	uniform aggregation	$1/k$
	nonuniform aggregation	[0.1,0.9]

reviews is 664,824. The total number of issued trust statements is 487,181. Since we focus on the time-evolving opinion formulation, we do not run experiments on the whole data set. Alternatively, we extract a subset. First of all, we restrict the maximum length to be 6, i.e., if a user has some friends within 6 hops who have rated the same item, we say the user/item pair can be predicted. We randomly choose 1,000 users. For each user, we randomly choose at most 6 items to which he has given a rating which can be predicted. A total of 5,548 pairs of user/items that can be predicted are selected. Table III shows the coverage of the selected sub data set, measuring the percentage of test pairs that can be predicted.

For a given test pair, we first construct a rating network, by identifying raters and non-raters: the former are the users' friends within a given hop who have rated the item, while the latter are those have not. Based on this, we relate users with containers and ratings as fluid temperatures, and then conduct multiple rounds of fluid updating using the FluidRating scheme. The temperatures of the sink are collected and aggregated as the final predicted rating. Next, we select the following methods for comparison: (1) TidalTrust [8]. It finds all trusted raters with the shortest distance from the sink, and aggregates their ratings, weighted by the trust between the sink and these raters. (2) MoleTrust [9]. It considers all raters up to a maximum-depth, which is given as an input, and is independent of any specific user and item. (3) Random Walk. Similar with [5], we set different thresholds on the number of steps in a random walk. (4) Personalized PageRank [1]. We take the result when it converges (e.g., the variation is small enough). Table IV shows the parameter settings. We use FluidRating 1 and uniform aggregation as the defaults.

B. Experimental Results and Analysis

In this subsection, we present the results of our experiments. First we describe the findings of "first influence," then we analyze how each factor can impact the performance.

1) The Existence of the "First Influence": We test the RMSE of 1-hop, 2-hop, 3-hop neighbors, and we get the results of 1.186, 1.436, and 1.639, respectively. It validates our claim that closer friends help for more accurate prediction, which is also found in [4], and indicates the existence of "first influence". In fact, we observe the first influence phenomenon through all experiments. Figs. 6(a) and 6(b) show 4 different patterns of 4 user/item pairs, where the real ratings of TestPairs 1, 2, and 4 are higher than the predicted rating, and that of TestPair 3 is lower. For TestPair 1, the real rating is 5, and the predicted rating first decreases when the number of samples, that is k, increases from 1 to 6, then keeps stable during the

TABLE V. THE RESULTS FOR A TEST PAIR WITH DIFFERENT Δ.

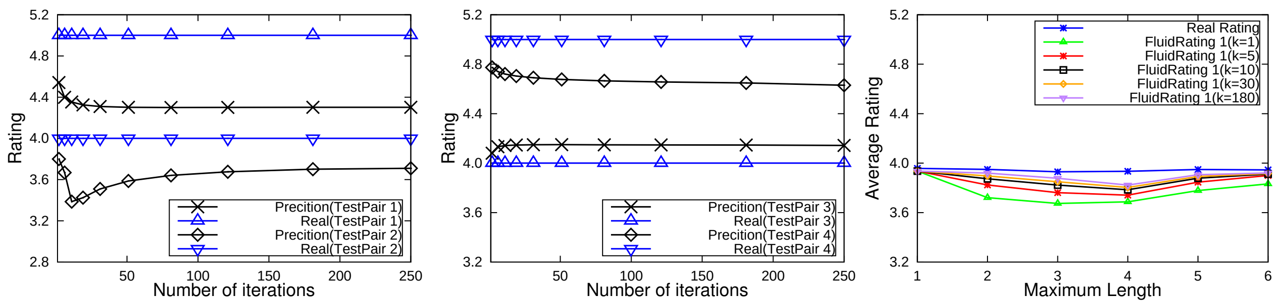
k \ Δ	0.001	0.01	0.1	0.4
2	4	4	4	4
6	4	4	4.0027	3.9478
19	4	4	4.0084	3.961
51	4	4	4.0146	3.929
81	4	4	4.0168	3.9148
121	4	4	4.018	3.9036
181	4	4	4.0189	3.894
250	4	4	4.0193	3.8878

period afterward. For TestPair 2, there is a fluctuation when k varies from 2 to 6, then to 11 and 19. It first decreases, then increases, and finally stabilizes. TestPairs 3 and 4 show other patterns. In all the four patterns, the first samples give predictions close to the real truth. This finding is a general phenomenon in the data set; as shown in Fig. 6(c), for the sub-data set we use, the average rating when k is small is very close to the the real average rating, and when k becomes larger, the gap between real and predicted ratings decreases gradually; this also indicates the refinements of users' opinions.

Fig. 7(a) displays the results of 100 TestPairs, which shows that the predicted rating is very close to the real rating in some cases, while some other cases are not. We analyze the meta-results, and find the reason is: for the latter test pairs, it usually happens that those users have few raters in their subgraph, or the raters' opinions are largely different from one another. In fact, this is what usually happens in real life. FluidRating is proposed exactly for modeling this process by combining the influence from different opinions over time.

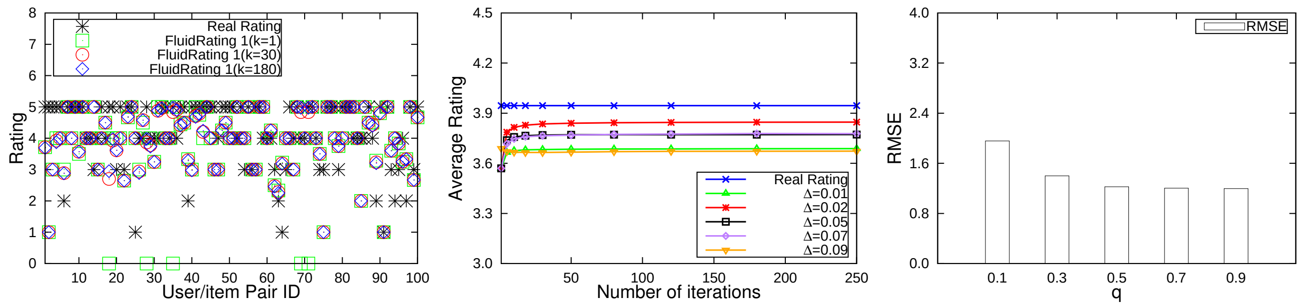
2) The Effects of Time Slot: We test the effects of the duration of time slot, Δ, as shown in Table V and Fig. 7(b). Generally speaking, a smaller Δ leads to slower convergence, while a larger Δ may lead to an inaccurate prediction because of the discrete approach. Fig. 7(b) shows the average rating of real and prediction, with different $\Delta \in [0.01, 0.1]$. It indicates that, when Δ is small enough, the rating prediction performance is stable. Table V shows the detailed results for an example test pair $7/213$, where 7 is the user ID, and 213 is the item ID. We let Δ switch among $\{0.001, 0.01, 0.1, 0.4\}$, and we find that the predicted ratings change insignificantly from $\Delta = 0.001$ to $\Delta = 0.1$, and it changes sharply when Δ changes from 0.1 to 0.4. Thus, Δ should not be too large.

3) The Effects of Aggregation Methods: As mentioned before, we design a set of sequences, $q^{1+c(i-1)}$, to measure the weights of k samples. The default setting of FluidRating 1 is $c = 0$ and $q = 1/k$. We also try a non-uniform sequence of $c = 1$, and let q change in the range of $[0.1, 0.9]$, with an increment of 0.2. The results are shown in Figs. 7(c), 8(a) and 8(b). Fig. 7(c) shows that when q is smaller, the RMSE is larger, and vise versa. This finding indicates that when we give earlier samples a larger weight, the accuracy is higher. It again validates the existence of "first influence." In addition, when $q \geq 0.5$, the change of RMSE becomes insignificant. Based on this, we can say that, approximately, first influence takes at least an importance of 50% in the whole process of opinion formulation. Fig. 8(a) shows that FluidRating 1* has a better and more stable performance than FluidRating 1 with respect to the maximum length. It indicates that, when we put more weight on the earlier influence, the accuracy is improved. This finding validates the feasibility of



(a) Test pairs 1 and 2 (b) Test pairs 3 and 4 (c) All test pairs

Fig. 6. The phenomenon of the first influence. (a) and (b) show four different patterns of 4 user/item test pairs. (c) provides a comparison of the average ratings with respect to k and the maximum length.



(a) 100 test pairs (b) FluidRating 1 with changing Δ (c) FluidRating 1 with changing q

Fig. 7. The effects of impact factors. (a) describes the detailed results of 100 test pairs. (b) depicts the average predicted rating with FluidRating 1 and different settings of Δ. (c) shows the RMSE of FluidRating 1 with $c = 1$ and q changing from 0.1 to 0.9.

our sample aggregation approach, which prefers a decreasing weight sequence. Fig. 8(b) shows that FluidRating 2 and 3 have performances stable and good as, or even better than, those of FluidRating 1*, which again indicates flexibility in that multiple choices are available to collect the samples.

4) The Effects of Sample Approaches: Figs. 9(a)-9(c) show the RMSE of using FluidRating 1, FluidRaing 2 and FluidRaing 3, respectively. The trends of performance changes are similar when using FluidRating 1 and 3, and the performance varies quite significantly when the maximum length changes. FluidRating 2 shows two notable differences: with the increase of k, its performance becomes stable gradually; the gap of RMSE is insignificant with respect to different maximum lengths. In a trust-based recommendation system, a larger maximum length usually indicates a larger coverage, however, this also indicates a decreased accuracy. The above findings indicate that FluidRating 2 is more resistant to maximum length changes. However, it takes a longer time to converge.

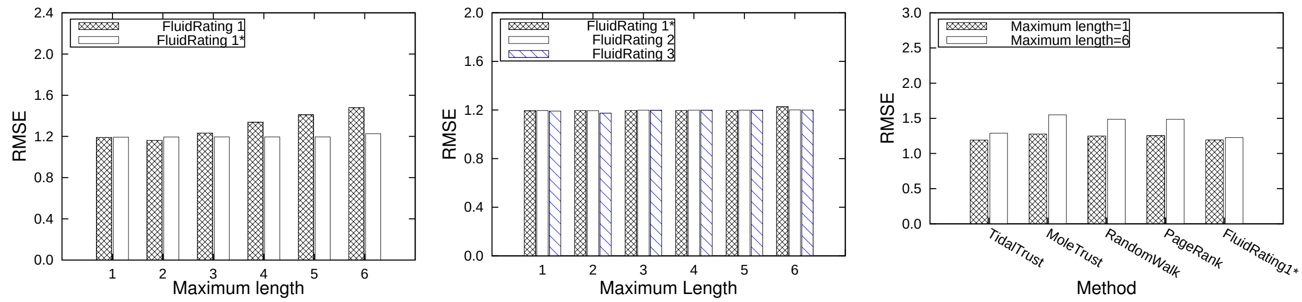
5) Comparison of Multiple Methods: We compare the RMSE of using several trust-based recommendation methods. As shown in Fig. 8(c), FluidRating beats MoleTrust, RandomWalk, and Personalized PageRank; meanwhile, TidalTrust performs almost as well as FluidRating. We analyze the reason to be that, TidalTrust takes the shortest and strongest recommendation path for rating prediction, which is exactly taking the first influence. In addition, the RMSE of using FluidRating 1* is 4.812% less than that of using TidalTrust when maximum length=6, which indicates the existence of varieties. That is, in most cases, "first influence dominates" works, while in a few other cases, it does not. This finding also indicates the necessity of considering influence over time in multiple rounds, rather than only one round. It is worth noting that, the RMSE of the randomly selected subset is larger than that of the whole data set, as in [5] and [4]. However, the sub-data set is enough for us to reveal the opinion formulation process. In the future we will consider extending the experiments in the whole data set.

C. Summary of Experiments

The experimental results show that users' opinions do evolve with time, and verify the existence of the first influence phenomenon. The proposed FluidRating model can flexibly handle those key points. Multiple factors, such as the number (k), the duration (Δ) of time slots, and the weighted sequence of k samples, can affect the prediction performance. The main findings are summarized as follows: (1) First influence does exist in the experiments. It holds in many cases. (2) Exceptions also occur. For some test pairs, influence other than first influence dominates. This is exactly why we propose "stronger influence dominates," consider the persistency of users, and form opinions over time.

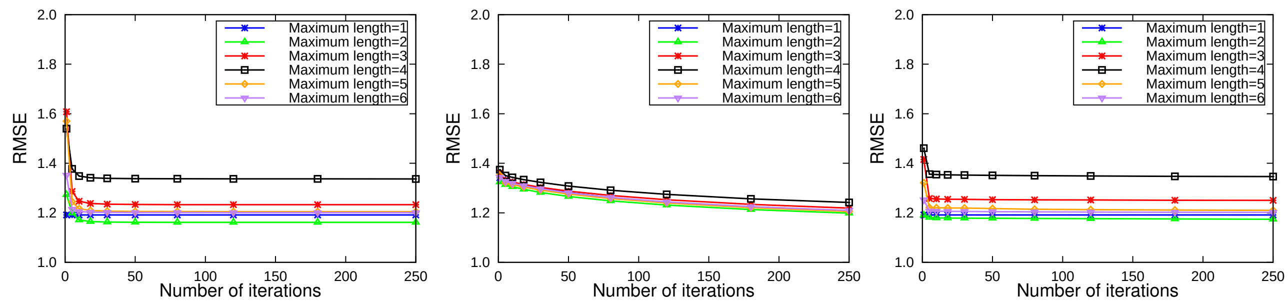
VIII. Conclusion and Future Work

Recommendation systems aim to predict the opinions of users on a target item, in order to decide whether to recommend the item to them. However, existing work focuses on the static rating prediction at the current time. We identify two challenges of capturing ratings and refining them, based on direct and indirect influences. We propose a novel fluid dynamics theory-based rating prediction model, where we consider two dimensions of a dynamic fluid. That is, fluid temperature is taken as "opinion/rating," and volume is then deemed as the "persistency" of its opinion. In this way, the two challenges are solved naturally and gracefully. The experiments in a real social network data set, Epinions, validate the



(a) FluidRating 1 V.S. FluidRating 1* (b) FluidRating 1*, 2, and 3 (c) Multiple methods

Fig. 8. The effects of weight sequence, sample approaches, and the comparison with other methods. (a) compares FluidRating 1 with $c = 0$, $c = 1$, and $q = 0.5$, the latter is denoted as FluidRating 1*. (b) compares FluidRating 1*, FluidRating 2, and FluidRating 3. (c) compares multiple methods.



(a) FluidRating 1 (b) FluidRating 2 (c) FluidRating 3

Fig. 9. The effects of sample approaches.

effectiveness of the proposed model, as well as the existence of phenomenon we observe in real life, that is, first influence. Currently, we mainly consider the feature of persistency for opinion influence, and we assume that the cross-sections of all containers are uniform. For future works, we are interested in extending FluidRating in dealing with more features in real life. For instance, the persuasiveness of a user can be reflected by the cross-sectional area of a container. In addition, once a user has formed his opinion, he may actually buy the product, and will form his own opinion based upon not only his expected opinion, but also his experience. This will then change his rating, which may in turn change the rating of his trusted friends. This evolution will also be studied in the future.

ACKNOWLEDGMENTS

This work is supported by NSFC grants 61272151 and 61073037, ISTCP grant 2013DFB10070, the China Hunan Provincial Science & Technology Program under Grant Number 2012GK4106, the Ministry of Education Fund for Doctoral Disciplines in Higher Education under Grant Number 20110162110043. This work is also supported in part by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

REFERENCES

[1] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: An axiomatic approach. *Proc. ACM WWW*, pages 199–208, 2008.

[2] X. Amatriain. Mining large streams of user data for personalized recommendations. *SIGKDD Explorations*, 14(2):37–48, 2012.

[3] N. Koenigstein, G. Dror, and Y. Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proc. ACM RecSys*, pages 165–172, 2011.

[4] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proc. ACM RecSys*, pages 17–24, 2007.

[5] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proc. ACM KDD*, pages 397–406, 2009.

[6] X. Yang, H. Steck, and Y. Liu. Circle-based recommendation in online social networks. In *Proc. ACM KDD*, pages 1267–1275, 2012.

[7] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei. Reputation and trust-based systems for ad hoc and sensor networks. *Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks, A. Boukerche (ed.), Wiley, 2008, ISBN: 978-0-470-38358-2*.

[8] J. Golbeck. Computing and applying trust in web-based social networks. *PhD thesis, University of Maryland*, 2005.

[9] P. Massa and P. Avesani. Trust metrics on controversial users: Balancing between tyranny of the majority and echo chambers. *International Journal on Semantic Web and Information Systems*, 3:39–64, 2007.

[10] G. Wang and J. Wu. FlowTrust: Trust inference with network flows. *Frontiers of Computer Science in China*, 5(2):181–194, 2011.

[11] M. Taherian, M. Amini, and R. Jalili. Trust inference in web-based social networks using resistive networks. In *Proc. ICIW*, pages 233–238, 2008.

[12] W. Jiang, G. Wang, and J. Wu. Generating trusted graphs for trust evaluation in online social networks. *Future Generation Computer Systems*, 31:48–58, 2014.

[13] H. Zhu, B. Huberman, and Y. Luon. To switch or not to switch: understanding social influence in online choices. In *Proc. ACM CHI*, pages 2257–2266, 2012.

[14] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In *Proc. ACM WWW*, pages 519–528, 2012.

[15] E. R. Smith and D. M. Mackie. *Social Psychology*. Psychology Press, 2007.

[16] S. A. Myers, Ch. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *Proc. ACM KDD*, pages 33–41, 2012.

[17] http://en.wikipedia.org/wiki/Conservation_of_mass.

[18] http://en.wikipedia.org/wiki/Conservation_of_energy.

[19] http://en.wikipedia.org/wiki/Torricelli's_law.