

On Organizing Online Soirees with Live Multi-Streaming

Chih-Ya Shen,¹ C. P. Kankeu Fotsing,^{4,5} De-Nian Yang,² Yi-Shin Chen^{1,4} Wang-Chien Lee³

¹ Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

² Institute of Information Science, Academia Sinica, Taipei, Taiwan

³ Department of Computer Science and Engineering, The Pennsylvania State University, PA, USA

⁴ Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan

⁵ Social Networks and Human-Centered Computing Program, Taiwan International Graduate Program,
Institute of Information Science, Academia Sinica, Taipei, Taiwan

chihya@cs.nthu.edu.tw, cedricfotsing@yahoo.fr, dnyang@iis.sinica.edu.tw, yishin@cs.nthu.edu.tw, wlee@cse.psu.edu

Abstract

The popularity of live streaming has led to the explosive growth in new video contents and social communities on emerging platforms such as Facebook Live and Twitch. Viewers on these platforms are able to follow multiple streams of live events simultaneously, while engaging discussions with friends. However, existing approaches for selecting live streaming channels still focus on satisfying individual preferences of users, without considering the need to accommodate *real-time* social interactions among viewers and to diversify the content of streams. In this paper, therefore, we formulate a new *Social-aware Diverse and Preferred Live Streaming Channel Query (SDSQ)* that jointly selects a set of diverse and preferred live streaming channels and a group of socially tight viewers. We prove that SDSQ is NP-hard and inapproximable within any factor, and design SDSSel, a 2-approximation algorithm with a guaranteed error bound. We perform a user study on Twitch with 432 participants to validate the need of SDSQ and the usefulness of SDSSel. We also conduct large-scale experiments on real datasets to demonstrate the superiority of the proposed algorithm over several baselines in terms of solution quality and efficiency.

Introduction

With the birth of Twitch, Facebook Live, Twitter Periscope, and Ustream, *live streaming* has recently become very popular. In addition to allowing users to create streaming channels on various topics in real time (e.g., news, sports, games), these new platforms support two unique features:

1) *Live Interactions*. Live streaming platforms have been regarded as *virtual third places* for social interactions (Hamilton, Garretson, and Kerne 2014). They enable viewers to interact (e.g., chat and send virtual gifts) with each other and the broadcaster in real time as events unfold. As Facebook reported (Facebook 2016), people frequently interact with friends and comment 10 times more on Facebook Live videos than on a regular video. Moreover, the feature of allowing users to invite friends to watch live streams together and have fruitful interactions (e.g., for sports games) has become a cornerstone in Facebook Live and Twitch

(Mashable 2017), and has been used for more than 300 million times (Twitch 2016b). Indeed, social interaction is an important feature that brings users and their friends together to enjoy real-time live streaming (Hamilton, Garretson, and Kerne 2014). From our analysis on datasets collected from bilibili¹ and Twitch, nearly 30% of friends watch 51% overlapping channels. 21% of friends in bilibili simultaneously watch the same videos and chat with more than 26.5 lines on a 3-minute long video. Moreover, the users in Twitch join *Twitch Teams* with friends, and 62% of the users in a team watch the same channels.

2) *Multi-Streaming*. This feature, supported by Youtube Live, Facebook Live, Teevox, Dualstreamnow, and Twitchster, allows viewers to simultaneously watch multiple channels with correlated but different perspectives. NBC sports and Fox sports live streaming channels, both offering multiple (at least four) viewing angles in sports games (e.g., NFL, MLB, UFC), have attracted more than 25M and 48M viewers, respectively. From our analysis of a real dataset on 2017 Taipei Summer Universiade (FISU 2017) broadcasted on Youtube Live, at least 30% of viewers watch multiple streams of the same or different sports games at the same time. The user study (detailed in the experimental results section) on 432 Twitch users also manifests that more than 90% of them have enjoyed multi-streaming. Moreover, research also shows that user satisfaction has been boosted by the immersive experience of watching different camera angles of an event at the same time (Haimson and Tang 2017; Hamilton et al. 2016; Mostafa et al. 2016; Hamilton, Garretson, and Kerne 2014; Jain, Sarda, and Haritsa 2004).

The above new features have boosted the variety and volume of available contents and channels in live streaming platforms, e.g., Twitch has reported 2.1 million broadcasters and more than 100 million users (Twitch 2016a). Nevertheless, the wide variety and large volume of channels and contents also bring prominent challenges to users who would like to exploit these platforms to organize some social events or gather some friends to watch live streaming together. To meet the need for an *online soiree organization service*, in this paper, we propose a new framework that helps organize an online soiree of live streaming by finding a group of

¹<https://www.bilibili.com/>.

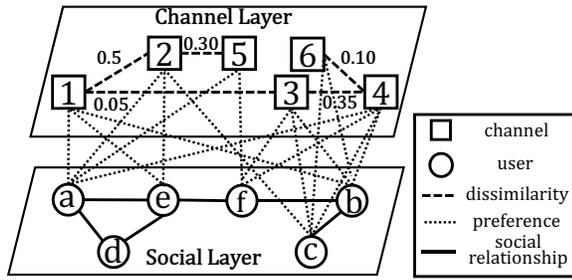


Figure 1: Illustrative example.

close friends (e.g., for *Twitch Team*) and a number of diverse and preferred live streaming channels². For live streaming soiree groups such as *Twitch Teams*, an organizer may fail to extract a group of socially tight members or fail to find a highly diverse set of preferred channels to enjoy due to the tediousness of the manual selection. Another way is randomly forming live streaming groups in an ad hoc manner, but this approach does not carefully consider the above factors to address the potential user needs.

To support the online soiree organization service framework, we formulate a new query, namely *Social-aware Diverse and Preferred Live Streaming Channel Query (SDSQ)*, aiming to select a set of diverse and preferred channels along with a group of socially tight viewers, such that 1) every selected viewer is interested in at least one chosen channel (coverage constraint), 2) each selected channel interests at least p selected viewers (minimum interest constraint), and 3) the selected viewers form a socially tight group, i.e., each pair of selected viewers are at most h hops away in the social network (social tightness constraint)³. SDSQ aims to maximize the diversity of the selected preferred channels (i.e., the average dissimilarity) for multi-streaming while selecting a group of socially tight friends in order to organize an online soiree.

Figure 1 illustrates the strength of using SDSQ for organizing an online soiree. Assuming that the dissimilarity (dashed lines) of channels (denoted by squares) and social relationship (solid lines) among users (denoted by circles) are given. e.g., the content of channels 1 and 2 are 50% dissimilar, and users a and d are friends. We also assume that the preferences (dotted lines) of users over channels are known, e.g., user a prefers channel 1. A preference-based approach may return channels $\{3, 5\}$ along with viewers $\{a, c, f\}$ as each selected channel is preferred by at least two selected viewers. For an online soiree, this is not a good configuration, as the social connections between selected viewers are sparse, while the channels are very similar, e.g., no

²While social closeness is essential for engaging social conversations, the diverse and preferred channels bring rich and different viewing/camera angles of the event. These are two key ingredients to the success of an online soiree.

³Parameters h and p can be acquired from real data. From the data collected in our experiments detailed later, we observe that users prefer to watch channels with at least 4.71 friends who share the same interests, while the friends are selected within 1.57 hops.

dissimilarity edge between channels 3 and 5. In contrast, SDSQ chooses channels $\{3, 4\}$ and viewers $\{b, c, f\}$ because the viewers are friends (or friends-of-friends) and they all prefer the chosen channels, while the channels $\{3, 4\}$ are dissimilar from each other.

Solving SDSQ is very challenging because it needs to examine the channel diversity, social tightness, and user preference jointly, for choosing not only channels but also viewers. We prove that SDSQ is NP-hard and inapproximable within any factor unless $P = NP$. Nevertheless, we observe that by first introducing a small bounded error, an efficient approximation algorithm can be designed. We thus propose a 2-approximation algorithm, namely SDSSel, with a guaranteed error bound. The contributions of this paper are summarized as follows.

- We identify an urgent need of online soiree organization service for emerging live streaming platforms and formulate a new query, namely SDSQ, to support the service.
- We prove that SDSQ is NP-hard and inapproximable within any ratio and propose an error-bounded 2-approximation algorithm named SDSSel.
- We prototype a new online soiree organization service on Twitch and conduct a user study to validate the need of SDSQ. The results indicate that users are more satisfied with the selection results from SDSQ as compared to other baselines and users' manual selections.
- We compare SDSSel with various baseline approaches on three large real datasets. The results show that SDSQ outperforms the baselines in solution quality and efficiency on the large real datasets.

Problem Formulation

Given a heterogeneous graph $G = (V, C, E, D, P)$, where V is the set of viewers and C is the set of channels. E represents the set of *social edges*, D is the set of *diversity edges*, and P is the set of *preference edges*. A social edge $(u, v) \in E$ with $u, v \in V$ indicates that viewers u and v are friends. A diversity edge $[q, r] \in D$ with weight $W[q, r] \in [0, 1]$ quantifies the dissimilarity between two channels $q, r \in C$ (the larger $W[q, r]$ is, the more dissimilar channels q and r are). A preference edge $\langle u, q \rangle \in P$ indicates that the viewer $u \in V$ has a preference for channel $q \in C$. We denote $\Delta_q(C) = \sum_{r \in C} W[q, r]$ the *total incidence diversity* of channel q in C , i.e., the sum of weights of the diversity edges incident to q . The *channel graph* $G(C) = (C, D)$ contains the set of channels and their corresponding diversity edges. The *social graph* $G(V) = (V, E)$ represents the set of viewers and their corresponding friendships. For a channel $q \in C$, we denote the set of viewers that have preferences for q as $V(q)$. Table 1 summarizes the important notations. In the following, we first define the h -dense group.

Definition 1. A subgraph $F \subseteq G(V)$ is an h -dense group if for any pair of viewers $u, v \in F$, $d_G^E(u, v) \leq h$ holds, where $d_G^E(u, v)$ is the shortest hop distance between u and v on $G(V)$.

Table 1: Table of notations.

Notation	Description
$\omega(K)$	Average diversity of channel set K
$C(v)$	Channels preferred by viewer v
$V(q)$	Viewers preferring channel q
$V(K)$	Viewers preferring at least one channel in K
$\Delta_q(C)$	Total incident diversity of q in C
$\Delta(K)$	Sum of diversity edge weights in K
$H_v^2(V)$	$2h$ -dense viewer set of v
$H_v^2(C)$	$2h$ -dense channel set of v
$d_G^E(u, v)$	Shortest hop distance between u and v on G

With the concept of h -dense groups in hand, we formally formulate SDSQ as follows.

Problem: Social-aware Diverse and Preferred Live Streaming Channel Query (SDSQ).

Given: A heterogeneous graph $G = (V, C, E, D, P)$, social constraint h , and minimum interest constraint p .

Find: A set of viewers $F \subseteq V$ and a set of channels $K \subseteq C$ such that i) every selected channel $q \in K$ is preferred by at least p viewers in F (minimum interest constraint), i.e., $\forall q \in K, |V(q) \cap F| \geq p$, ii) every viewer $u \in F$ prefers at least one chosen channel $q \in K$ (coverage constraint), i.e., $\forall u \in F, \exists q \in K, \langle u, q \rangle \in P$, and iii) the selected viewers collectively form an h -dense group (social constraint).

Objective: To maximize the diversity of the selected channels, i.e., $\omega(K) = \frac{\sum_{[q,r] \in D(K)} W[q,r]}{|K|}$, where $D(K)$ is the set of diversity edges induced by K on $G(C)$ ⁴.

SDSQ includes two parameters h and p , which are to be set by soiree organizers (or guided by an analysis on the historical query data). In our user study in the experimental results section, we also analyze the users' choices of the two parameters. Note that the *average dissimilarity*, instead of the sum of dissimilarity, is adopted in the objective function to avoid favoring a large number of channels in the solution (Hurley and Zhang 2011). Through our initial study, we find that SDSQ is NP-hard and inapproximable within any factor as stated below, i.e., no approximation algorithm exists.

Theorem 1. *SDSQ is NP-hard and inapproximable within any factor unless $P = NP$.*

Proof. We prove that SDSQ is NP-hard with the reduction from the maximum DBS (Diameter Bounded Subgraph) problem (Asahiro, Miyano, and Samizo 2010), which is NP-hard. The decision problem of DBS, given a graph $G_m = (V_m, E_m)$ and two integers k, d , decides whether G_m contains a connected subgraph $G_d = (V_d, E_d)$ such that for any vertices u and v in V_d , $d_{G_m}^E(u, v) \leq d$ and

⁴An alternative objective is to employ the weighted sum of the four factors. However, the weights for each individual are different and difficult to evaluate. In contrast, a large objective value here is inclined to include more preferred channels especially when the diversity becomes larger.

$|V_d| \geq k$ hold. For each instance of DBS, we build an instance of SDSQ as follows. The input heterogeneous graph $G = (V, C, E, D, P)$ is constructed by letting $V = V_m$, $E = E_m$, $C = \{q_1, q_2\}$, $D = \{\{q_1, q_2\}\}$, $P = \{\langle u, c \rangle | \forall u \in V, c \in C\}$, (i.e., each viewer $u \in V$ prefers all the channels), $W[q_1, q_2] = 1$, $p = k$, and $h = d$.

We first prove the sufficient condition. If the maximum DBS returns TRUE with a diameter bounded subgraph G_d , then $F = V_d$ and $K = C$ is a feasible solution to SDSQ. F forms an h -dense group since for any pair of vertices $u, v \in V_d$, $d_{G_m}^E(u, v) \leq d = h$ holds. Moreover, every channel in C is preferred by all the viewers in F , and F is an h -dense group of size at least $p = k$. For the necessary condition, If SDSQ returns a solution with \hat{F} and \hat{K} , then for any vertices $u, v \in \hat{F}$, $d_{G_m}^E(u, v) \leq h$ because \hat{F} is also an h -dense group on $G_m = (V_m, E_m)$, i.e., $V = V_m$ and $E = E_m$. Furthermore, since $|\hat{F}| \geq p = k$, \hat{F} is a diameter bounded subgraph on $G_m = (V_m, E_m)$. Therefore, SDSQ is NP-hard. Moreover, if there exists a σ -approximation algorithm of SDSQ with $\sigma < \infty$, then given any instance of DBS, the same algorithm can solve the instance in polynomial time because any returned feasible solution of SDSQ implies TRUE for DBS. Therefore, SDSQ is inapproximable within any ratio unless $P=NP$. \square

Integer Linear Programming Formulation. Here, we formulate the Integer Linear Programming (ILP) formulation of the proposed SDSQ problem. We first define a binary decision variable x_i with $1 \leq i \leq |V|$ to indicate whether a viewer $v_i \in V$ is selected. That is, $x_i = 1$ if and only if v_i is selected in the group F . Similarly, let binary decision variable y_j , $1 \leq j \leq |C|$, indicate whether a channel $q_j \in C$ is included in K . Moreover, we define another binary decision variable p_{ij} where $p_{ij} = 1$ if and only if channel q_i is preferred by viewer v_j . A new binary variable z_{ij} is also introduced to capture whether the two endpoints of a diversity edge $[q_i, q_j]$ are selected in K . That is, $z_{ij} = 1$ if and only if q_i and q_j both are included in K .

The constraints are listed as follows.

$$\sum_{j \in [1, |V|]} p_{ij} x_j - p \cdot y_i \geq 0, \forall i \in [1, |C|], \quad (1)$$

$$\sum_{j \in [1, |V|]} p_{ji} y_j - x_i \geq 0, \forall i \in [1, |V|], \quad (2)$$

$$\begin{aligned} x_i + x_j &\leq 1, \forall d_G^E(v_i, v_j) > h, \\ z_{ij} &\geq y_i + y_j - 1, \forall 1 \leq i < j \leq |C|, W[i, j] > 0, \end{aligned} \quad (3)$$

$$z_{ij} \leq \frac{y_i + y_j}{2}, \forall 1 \leq i < j \leq |C|, W[i, j] > 0. \quad (5)$$

For the objective function, please note that the denominator includes the term $|K|$. If we simply use $\sum_{i=1}^{|C|} y_i$ to represent $|K|$, the formulation becomes *fractional*. To avoid this case, we perform a transformation as follows. We first introduce a new binary decision variable z_n , $1 \leq n \leq |C|$, to represent the size of the obtained channel set K . In this

case, $s_n = 1$ if and only if $|K| = n$. Then, s_n is derived with the following constraints.

$$\sum_{n \in [1, |V|]} s_n = 1, \quad (6)$$

$$\sum_{q \in C} y_i = \sum_{n \in [1, |C|]} n \cdot s_n. \quad (7)$$

Then, let the objective value be ω , SDSQ aims to maximize ω , where ω is derived as follows.

$$\omega \leq \frac{\sum_{[q_i, q_j] \in D} W[q_i, q_j] z_{ij}}{n} + (1 - s_n) \sum_{\forall [q_i, q_j] \in D} W[q_i, q_j], \forall 1 \leq n \leq |V|. \quad (8)$$

Related Work

The *group formation* problem aims to find a group of users for various social activities. Lappas et al. propose algorithms to form an expert team with all the required skills for business applications (Lappas, Liu, and Terzi 2009). Shen et al. propose methods to form an impromptu social activity group and suggest a suitable activity location for the group (Shen et al. 2016). Social connectivity and user interests have been considered for activity group formation to maximize the willingness of group members (Shuai et al. 2013). Preference has also been considered to group users according to a set of items (Roy, Lakshmanan, and Liu 2015) but without a guarantee of tight social relations among group members. Even though these existing approaches aim at forming activity groups, they are not designed for live streaming that needs to consider the channel diversity (Haimson and Tang 2017; Jain, Sarda, and Haritsa 2004; Mostafa et al. 2016; Hamilton et al. 2016), social tightness (Hamilton, Garretson, and Kerne 2014) and user preference for channels simultaneously in order to ensure a successful soiree event.

Video channel selection algorithms search and recommend channels or programs, based on personal viewing behaviors of users in Youtube (Covington, Adams, and Sargin 2016; Koren and Sill 2013; Lu et al. 2015; Park et al. 2016; Zhao, McAuley, and King 2014). Recently, diversity has drawn increasing attention in content search. DisC diversity combines diversity and coverage to summarize search results (Drosou and Pitoura 2012). Diversification to geo-spatial keyword search that considers both the relevance and spatial diversity has also been studied (Zhang et al. 2014). Nevertheless, the social connectivity among friends is not considered for the selection of viewers to foster warm interactions.

In addition, various item and product recommendation algorithms have been proposed (Koren and Sill 2013; Lu et al. 2015; Park et al. 2016; Zhao, McAuley, and King 2014). However, the SDSQ problem studied in this paper is for soiree configuration and planning that considers the channel diversity and social tightness to jointly select a group of viewers and channels. SDSQ is totally different from the traditional recommendation algorithms, and thus recommendation algorithms cannot be applied to solve SDSQ.

Algorithm Design for SDSQ

To maximize the average diversity, a simple approach is to iteratively extract the top channels with the highest *total incident diversity*, which is the total weight of the diversity edges incident to a channel, and then find viewers who like these channels. However, this *channel-oriented* approach has two pitfalls: i) the selected viewers are not guaranteed to be socially tight, and ii) each selected channel may not be preferred by at least p selected viewers. Alternatively, a *social-oriented* approach is to select the viewers to form an h -dense group and then find a set of channels with maximal average diversity. Nevertheless, finding an h -dense group is computationally intensive. Especially when $h = 1$, the viewers need to form a clique, which is NP-complete.

Our idea is to first explore a larger but slightly sparse social group (i.e., a $2h$ -dense group) and then tailor it to the final solution. The above *relax-and-tailor* strategy has been widely adopted by the research community of approximation algorithms (Williamson and Shmoys 2011) for difficult NP-hard problems. Finding a $2h$ -dense group with more candidate viewers for SDSQ is simple and thereby suitable for large-scale online social networks (OSN). Moreover, for each viewer v , a $2h$ -dense group of v contains all possible h -dense groups of v that are feasible to SDSQ. If a channel q has at least p viewers in an h -dense group of v , it also has at least p viewers in the corresponding $2h$ -dense group of v . Later we show that the above nice properties foster promising solutions effectively and efficiently.

The above observations allow us to divide a large-scale OSN into smaller search subspaces with $2h$ -dense groups. Accordingly, we design a 2-approximation algorithm, namely *Social-aware Diverse and Preferred Live Streaming Selection (SDSSel)* with a guaranteed error bound h in social tightness. SDSSel iteratively performs the following two steps for each viewer v to extract the best candidate solution: 1) *2h-Dense Sets Generation*: it identifies the viewers in the $2h$ -dense group and a candidate channel set that satisfies the minimum interest constraint (*interest constraint* for short). 2) *Channel-Viewer Selection*: it first selects some channels with high diversity from the candidate channel set and then extracts a set of socially cohesive viewers to ensure that each selected channel is preferred by at least p selected viewers. A candidate solution is generated after the above two steps for each viewer, and SDSSel records the best candidate solution among all viewers. Finally, a post-processing step fixes the error incurred by the relaxation strategy in the best candidate solution and tailors it into the final solution with the same ratio.

Specifically, for each channel q , let $\Delta_q(C) = \sum_{r \in C} W[q, r]$ denote the total incident diversity of q . For each viewer $v \in V$, let $H_v^2(V)$ denote the $2h$ -dense group (also called *2h-dense viewer set*) of v extracted from V . It can be efficiently extracted by including every viewer $u \in V$ with $d_G^E(u, v) \leq h$. That is, $H_v^2(V) = \{u | d_G^E(u, v) \leq h\}$. Further, we define the *2h-dense channel set* of v , i.e., $H_v^2(C)$, as the largest subset of C where each channel in the subset has at least p viewers from $H_v^2(V)$. By introducing $H_v^2(V)$ and $H_v^2(C)$, later we show that if the optimal

solution includes a viewer v , the channel set in the optimal solution must be included in $H_v^2(C)$. The above observation plays a critical role in the derivation of the approximation ratio. Intuitively, SDSSel iteratively performs *2h-Dense Sets Generation* and *Channel-Viewer Selection* on each $v \in V$. During the process, SDSSel records the best candidate solution (K^{APX}, F^{APX}) , where K^{APX} is the best channel set obtained so far, and F^{APX} is the corresponding viewer set. Finally, SDSSel employs a post-processing strategy on K^{APX} and F^{APX} to tailor the solution and fix the error. The pseudo code is presented in Algorithm 1.

Algorithm 1 Social-aware Diverse and Preferred Live Streaming Selection (SDSSel)

Require: Graph $G = (V, C, E, D, P)$, p, h

- 1: Delete from C every channel q such that $|V(q)| < p$
- 2: Delete from V every viewer v incident to no preference edge
- 3: $K^{APX} \leftarrow \emptyset, F^{APX} \leftarrow \emptyset$
- 4: $I \leftarrow V$ $\triangleright I$ is the set of viewers not yet processed
- 5: **while** $I \neq \emptyset$ **do**
- 6: Select a viewer $v \in I, I = I - \{v\}$
- 7: Let $H_v^2(V)$ and $H_v^2(C)$ be respectively the $2h$ -dense viewer and channel set of v
- 8: **if** $|H_v^2(V)| < p$ **then**
- 9: Continue;
- 10: $K_v^1 \leftarrow H_v^2(C)$
- 11: **for** $i \leftarrow 1$ to $|H_v^2(C)|$ **do**
- 12: $q_i \leftarrow \operatorname{argmin}_{q \in K_v^i} (\Delta_q(K_v^i))$
- 13: $K_v^{i+1} \leftarrow K_v^i - q_i$
- 14: **if** $\omega(K_v^i) > \omega(K^{APX})$ **then**
- 15: $K^{APX} \leftarrow K_v^i, F^{APX} \leftarrow V(K_v^i) \cap H_v^2(V)$
- 16: **end for**
- 17: Perform the post-processing step
- 18: **output** (K^{APX}, F^{APX})

$2h$ -Dense Sets Generation. As mentioned, SDSSel first generates $H_v^2(V)$. It iteratively adds a viewer within h hops from v , i.e., $H_v^2(V) = \{u \in V \mid d_G^E(u, v) \leq h\}$. A node v is discarded if $|H_v^2(V)| < p$ because in this case, $H_v^2(V)$ is not a feasible solution. Then, $H_v^2(C)$ is constructed by extracting from C every channel preferred by at least p viewers in $H_v^2(V)$.

Channel-Viewer Selection. Next, SDSSel iteratively extracts from $H_v^2(C)$ the channel with the smallest total incident diversity. Specifically, at iteration i , SDSSel trims off the channel q_i with the minimum total incident diversity in K_v^i to generate K_v^{i+1} , where K_v^i denotes the set of channels in the beginning of iteration i . That is, $q_i = \operatorname{argmin}_{q \in K_v^i} \Delta_q(K_v^i)$. Let $V(K_v^i)$ denote the set of viewers who prefer at least one channel in K_v^i . Note that $(K_v^i, V(K_v^i) \cap H_v^2(V))$ is a feasible solution to SDSQ with a bounded error h because (1) every channel in $K_v^i \subseteq H_v^2(C)$ has at least p viewers in $H_v^2(V)$ (recall that $H_v^2(C)$ is the set of all channels preferred by at least p viewers from $H_v^2(V)$), (2) each viewer in $V(K_v^i) \cap H_v^2(V)$ is guaranteed to have at least one preferred channel in K_v^i , and (3) $d_G^E(x, y) \leq 2h, \forall x, y \in H_v^2(V)$.

At each iteration i , SDSSel examines whether the channel set K_v^i leads to a better solution, i.e., $\omega(K_v^i) > \omega(K^{APX})$.

If it does, K^{APX} is replaced by K_v^i and the corresponding viewer set F^{APX} is updated as $V(K_v^i) \cap H_v^2(V)$. SDSSel then proceeds to the next iteration with K_v^{i+1} . After all channels from the $H_v^2(C)$ are extracted (i.e., at iteration $j = |H_v^2(C)|, K_v^{j+1} = \emptyset$), SDSSel completes the Channel-Viewer Selection step for the current v and continues to explore the $2h$ -dense sets of another viewer until all viewers in V are carefully examined.

Consider Figure 1 as an example with $p = 2$ and $h = 4$. At the beginning, SDSSel deletes viewer d since $C(d) = \emptyset$. Let a be the first viewer examined by SDSSel. SDSSel first finds $H_a^2(V) = \{a, b, c, e, f\}$ and then finds $H_a^2(C) = \{1, 2, 3, 4, 5, 6\}$. Then, the Channel-Viewer Selection step starts. SDSSel sets K_a^1 to $\{1, 2, 3, 4, 5, 6\}$ and starts extracting the channels with the minimum incident diversity. Channel 6 is the first extracted since $\Delta_6(H_a^2(C)) = 0.1$ is the minimum. Therefore, SDSSel removes channel 6 and K_a^2 becomes $\{1, 2, 3, 4, 5\}$ and SDSSel sets K^{APX} to K_a^1 with $F^{APX} = V(K_a^1) \cap H_a^2(V) = \{a, b, c, e, f\}$. The above procedure repeats until $K_a^7 = \emptyset$. In this iteration, $K^{APX} = \{1, 2\}$ and $F^{APX} = \{a, b, c, e\} \cap H_a^2(V)$ since $\omega(K_a^5) = 0.25$ is the highest among all the $K_a^i, 1 \leq i \leq 6$. After SDSSel processes all the viewers, $K^{APX} = \{1, 2\}$ with $\omega(K^{APX}) = 0.25$. The solution $(\{1, 2\}, \{a, b, c, e\})$ is returned.

Post Processing. In the following, we propose a post-processing procedure to tailor the solution (K^{APX}, F^{APX}) returned by SDSSel with the goal to meet the social tightness constraint (i.e., F^{APX} is an h -dense group) and still maintain the approximation ratio. Given (K^{APX}, F^{APX}) , we first identify a set of *violating viewers* $\Gamma(V)$ in F^{APX} as follows. A viewer v in F^{APX} is a violating viewer if there exists at least one other viewer $u \in F^{APX}$ such that $d_G^E(u, v) > h$. Moreover, we also identify a set of *violating channels*, $\Gamma_v(C)$. A channel $q \in K^{APX}$ belongs to $\Gamma_v(C)$ if it is preferred by at most $p - 1$ viewers in F^{APX} after removing the viewer v from F^{APX} .

To tailor (K^{APX}, F^{APX}) , the post processing includes the following adjustment steps. 1) *Expanding*: a viewer $v \in (V - F^{APX})$ can be added to F^{APX} if including v will not increase $|\Gamma(V)|$ and $C(v) \cap K^{APX} \neq \emptyset$. Similarly, a channel $q \in (C - K^{APX})$ can be added to K^{APX} if adding it can increase $\omega(K^{APX})$ and $|V(q) \cap F^{APX}| \geq p$ holds. 2) *Trimming*: we remove from F^{APX} the viewer v in $\Gamma(V)$ such that $\omega(K^{APX} - \Gamma_v(C)) \geq \omega(K^{APX})$. After removing violating viewers from F^{APX} , the resulting violating channels are also removed from K^{APX} . If multiple channels can be discarded, we start with the one with the most viewers in F^{APX} that are more than h hops away from the existing viewers. Therefore, the final solution is an h -dense group and the post-processing step is able to minimize number of violating viewers.

Theoretical Analysis

In the following, we prove that SDSSel is a 2-approximation algorithm with a bounded error h , whereas the post processing fixes the error and achieves the same ratio. Let K^* and F^* denote the channel set and viewer set in the optimal so-

lution, respectively. In the following, we prove that for each channel q in K^* , the total incident diversity of channel q is not smaller than the average diversity of the optimal solution, i.e., $\Delta_q(K^*) \geq \omega(K^*), \forall q \in K^*$. This property is important for the approximation ratio because it finds the correlation of each channel's total incident diversity and the objective value. Next, we prove that if a viewer v is in the optimal solution, i.e., $v \in F^*$, the corresponding channel set of the optimal solution is always part of the $2h$ -dense channels set of v , i.e., $K^* \subseteq H_v^2(C)$. This property is crucial because it finds the correlation between viewers and channels. In the following, we first derive the performance gap between the optimal solutions (K^*, F^*) and the solution (K^{APX}, F^{APX}) obtained by SDSSel.

Lemma 1. For any channel $q \in K^*$, $\Delta_q(K^*) \geq \omega(K^*)$.

Proof. For any channel q in K^* , let $K_q^* = K^* - \{q\}$ be the set of channels obtained after removing q from K^* . Since K^* is the channel set in the optimal solution, $\omega(K_q^*) \leq \omega(K^*)$ holds⁵. This is because after q is removed from K^* , each remaining channel in K_q^* still has at least p viewers preferring it. Moreover, if q is removed from K^* and a viewer v in the corresponding F^* becomes infeasible (does not satisfy the coverage constraint), then v must prefer only the channel q in K^* . In this case, v can directly be removed from F^* , and $F^* - \{v\}$ together with K_q^* is still a feasible solution. Therefore, $\omega(K_q^*) \leq \omega(K^*)$ must hold; Otherwise, $\omega(K_q^*)$ would have been the channel set of optimal solution.

Then, we prove this lemma with contradiction. Recall that $\omega(K^*) = \frac{\Delta(K^*)}{|K^*|} = \frac{\Delta(K_q^*) + \Delta_q(K^*)}{|K_q^*| + 1}$. If $\Delta_q(K^*) < \omega(K^*)$ holds, then $\omega(K^*) = \frac{\Delta(K_q^*) + \Delta_q(K^*)}{|K_q^*| + 1} < \frac{\Delta(K_q^*) + \omega(K^*)}{|K_q^*| + 1}$. As $\omega(K_q^*) \leq \omega(K^*)$, $\Delta(K_q^*) \leq |K_q^*| \cdot \omega(K^*)$ holds. Therefore, we have $\omega(K^*) < \frac{\Delta(K_q^*) + \omega(K^*)}{|K_q^*| + 1} \leq \frac{|K_q^*| \cdot \omega(K^*) + \omega(K^*)}{|K_q^*| + 1} = \frac{(|K_q^*| + 1) \cdot \omega(K^*)}{|K_q^*| + 1} = \omega(K^*)$, leading to a contradiction. The lemma follows. \square

Lemma 2. If $v \in F^*$, then $K^* \subseteq H_v^2(C)$ holds.

Proof. Since F^* is an h -dense group, if v is in F^* , every viewer in F^* is at most h hops away from v . It implies $F^* \subseteq H_v^2(V)$. Since every channel in K^* is preferred by at least p viewers in F^* , every channel in K^* is preferred by at least p viewers in $H_v^2(V)$. By definition, $H_v^2(C)$ is the set of all the channels in C preferred by at least p viewers in $H_v^2(V)$, leading to $K^* \subseteq H_v^2(C)$. The lemma follows. \square

Recall that K_v^i is the set of channels at the beginning of iteration i when SDSSel processes v . For any viewer v in F^* , let i be the earliest iteration in the Channel-Viewer Selection step, when a channel q_i extracted from $H_v^2(C)$ is also in K^* . In other words, when SDSSel processes $H_v^2(C)$, all channels removed from $H_v^2(C)$ before iteration i (i.e., before q_i is removed) do not appear in K^* . Then we have the following lemma.

⁵Here we assume that $|K^*| > 1$, otherwise $\omega(K^*) = 0$, and the performance bound always holds.

Lemma 3. For any $v \in F^*$, if $q_i \in H_v^2(C)$ is the first channel extracted from K^* by SDSSel when processing $H_v^2(C)$, then $\Delta_q(K_v^i) \geq \omega(K^*)$ holds, $\forall q \in K_v^i$.

Proof. According to Lemma 2, $K^* \subseteq H_v^2(C)$ holds since $v \in F^*$. Therefore, if $q_i \in H_v^2(C)$ is the first channel in K^* extracted by SDSSel when processing $H_v^2(C)$, $K^* \subseteq K_v^i$. Otherwise, another channel $\hat{q} \in K^*$ would have been extracted from $H_v^2(C)$ by SDSSel before q_i . Note that $K^* \subseteq K_v^i$ also implies that $\Delta_{q_i}(K_v^i) \geq \Delta_{q_i}(K^*)$. According to Lemma 1, $q_i \in K^*$ implies that $\Delta_{q_i}(K^*) \geq \omega(K^*)$, leading to $\Delta_{q_i}(K_v^i) \geq \Delta_{q_i}(K^*) \geq \omega(K^*)$. Moreover, every channel in K_v^i has an incident diversity at least $\Delta_{q_i}(K_v^i)$ because q_i has the smallest one and is thereby extracted. Therefore, we have $\Delta_q(K_v^i) \geq \Delta_{q_i}(K_v^i) \geq \Delta_{q_i}(K^*) \geq \omega(K^*), \forall q \in K_v^i$. The lemma follows. \square

Theorem 2. SDSSel is a 2-approximation algorithm for SDSQ with a guaranteed error bound h .

Proof. Recall that the weight of each diversity edge is counted only once in $\omega(K_v^i)$. Therefore, according to Lemma 3, $\omega(K_v^i) = \frac{\sum_{q \in K_v^i} \Delta_q(K_v^i)}{2 \cdot |K_v^i|} \geq \frac{|K_v^i| \cdot \omega(K^*)}{2 \cdot |K_v^i|} = \frac{\omega(K^*)}{2}$. Since SDSSel examines every viewer $v \in V$ to extract the best channel set K^{APX} , i.e., $\omega(K^{APX}) = \max_{i \in [1, |C|], v \in V} \omega(K_v^i)$, we have $\omega(K^{APX}) \geq \frac{\omega(K^*)}{2}$, and the performance bound thereby holds.

We then prove the solution feasibility. For every $\hat{v} \in V$, if j is the iteration such that $K^{APX} = K_{\hat{v}}^j$, then $F^{APX} \subseteq H_{\hat{v}}^2(V)$ holds, thus $d_G^E(u, u') \leq 2h, \forall u, u' \in F^{APX}$. Therefore, F^{APX} is a $2h$ -dense group. Furthermore, $K_{\hat{v}}^j \subseteq H_{\hat{v}}^2(C)$, implying that every channel in $K_{\hat{v}}^j$ is preferred by at least p viewers in $H_{\hat{v}}^2(V)$. In this situation, for each $q \in K_{\hat{v}}^j$, $|V(q) \cap H_{\hat{v}}^2(V)| \geq p$ holds. We also have $(V(q) \cap H_{\hat{v}}^2(V)) \subseteq (V(K_{\hat{v}}^j) \cap H_{\hat{v}}^2(V)) = F^{APX}$. Therefore, every channel q in K^{APX} is preferred by at least p viewers in F^{APX} , and every viewer in F^{APX} belongs to $V(K^{APX})$. \square

Time Complexity. For any viewer v , the corresponding $2h$ -dense viewer set, i.e., $H_v^2(V)$, is generated by Breadth-First Search in $O(|V| + |E|)$ time. The $2h$ -dense Channel Set Generation and the Channel-Viewer Selection steps of a viewer require $O(|C|^2 + |V| + |D| + |P|)$ time. Finding the channel q with the minimum $\Delta_q(K_v^i)$ requires $O(|C|)$ time for each iteration. For all $q \in K_v^i$, finding $\Delta_q(K_v^i)$ and removing q from K_v^i take at most $O(|D|)$ times of edge scan. Finally, removing viewers in $H_v^2(V)$ related to the removed channel q takes $O(|P|)$ for all iterations. Therefore, the overall time complexity of SDSSel is $O(|V|(|V| + |E| + |C|^2 + |D| + |P|))$. Compared with $|E|$ and $|P|$, $|V|$ is usually small. Also, $O(|C|^2) = |D|$. Thus, $O(|V|(|E| + |D| + |P|))$ dominates the time complexity. Finally, in the post processing, checking the violating condition of each viewer v and finding $\Gamma_v(C)$ take $O(|V| + |E| + |C|)$ time. Therefore, post processing takes $O(|V|(|E| + |D| + |P|))$ time as well. Empirically, the experimental results manifest that most cases can be computed within seconds.

Table 2: User behavior in live streaming channels.

	Usually	Often	Rarely	Never
Twitch Viewing Frequency	38.60%	51.75%	9.65%	0.00%
Multi-Streaming	18.78%	28.73%	40.33%	12.15%

Experimental Results

To validate the ideas of SDSQ and SDSSel, we first build a prototype system on *Twitch* and conduct a user study with 432 *Twitch* users through Amazon MTurk⁶. A preference edge exists if a user follows a channel. The weights of diversity edges are acquired from cosine dissimilarity (i.e., 1-cosine similarity) of the two corresponding channels (Sarwar et al. 2001; Chen and Zhang 2017). The goal is two-fold: 1) to evaluate the SDSQ problem formulation, and 2) to evaluate the solution returned by SDSSel against the state-of-the-art baselines (described later), and users’ manual selection in terms of users’ satisfaction, and compare the solution quality of SDSSel against the users’ manual selection.

Afterward, we evaluate the effectiveness and the efficiency of SDSSel on three large-scale real datasets, namely *Yelp* (Yelp 2016), *Douban*, and *Twitch*. *Yelp* is a location-based social network with 4.1M reviews, 1.5M users, and the corresponding check-in records (regarded as preferred channels here). *Douban* includes 5M vertices and 86M edges, where users may recommend and like books, movies, and music (regarded as preferred channels here). *Twitch* includes 6M viewers and 250 crawled channels. As *Twitch* does not provide APIs to crawl the social network, *Twitch* channels are coupled with a Twitter social network dataset (R. Zafarani and H. Liu 2009), and the preference edges are generated following Watts-Strogatz model (Watts and Strogatz 1998). The weights of diversity edges are acquired from cosine dissimilarity.

We compare SDSSel with 4 state-of-the-art baselines: GD-LM-MIN (Roy, Lakshmanan, and Liu 2015), SSGSelect (Yang et al. 2012), TrustSVD (Guo, Zhang, and Yorke-Smith 2015), and the Integer Linear Programming formulation (ILP) that finds the optimal solution of SDSQ. GD-LM-MIN forms groups of users based on their preferences toward channels. Users are allocated in the same group if their top- k preferred channels are the same without considering their social relations. SSGSelect forms groups of users based on their social tightness and their preferences towards the channels without considering the channel diversity. TrustSVD recommends the channels according to the preference and trust of friends, and any two friends are in the same group if they are recommended the same channel. ILP is the Integer Linear Programming formulation for small instances of SDSQ. Together with IBM CPLEX, it finds the optimal solution for SDSQ. All algorithms are implemented on an HP DL580 server with 4

⁶For the user study, we implement a Chrome plugin to crawl, upon agreement from the participants, the user profiles, their friends, the channels they follow and the description, the channels their friends follow and their friends of friends.

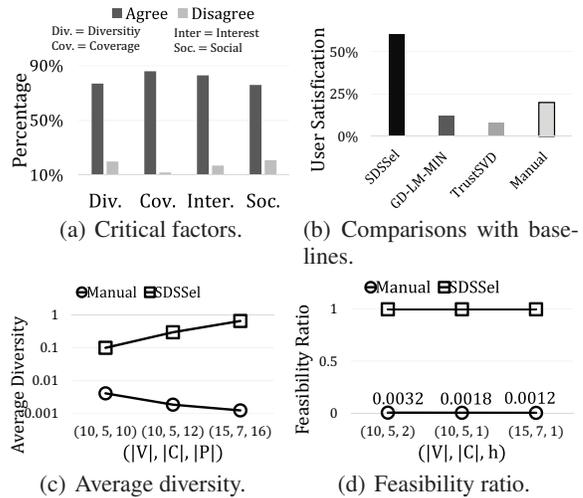


Figure 2: User study results.

Intel Xeon E7-4870 2.4 GHz CPUs and 1 TB RAM. Each result is averaged over 50 samples.

User Study

Table 2 first summarizes the behavior of the 432 *Twitch* users, indicating that 90% of users are frequent (i.e., answering as “usually” or “often”) live streaming users, and 56% of users watch multiple live streaming channels at the same time because simultaneously viewing different angles can indeed improve the user satisfaction, especially for sports, games, and news (Haimson and Tang 2017; Hamilton et al. 2016; Mostafa et al. 2016). Figure 2(a) then shows the correlation between user satisfaction and the four factors considered in SDSQ: (i) channel diversity, (ii) coverage constraint, (iii) minimum interest constraint, and (iv) social tightness. Participants are surveyed with a questionnaire regarding their satisfaction with those factors (e.g., “Do you prefer sharing your streaming experience with friends or with random people?”, “Do you prefer a list of diverse streaming content while you are multi-streaming?”, “Do you form special chat groups while watching a live stream?”). The result manifests that live streaming users are more satisfied with diversified channels (channel diversity) that they are interested in (i.e., the coverage constraint) and sharing the viewing experience with friends (social tightness and minimum interest constraint), instead of random people in the same chat room. The analysis of real datasets (i.e., *Twitch*, *Youtube Live*, and *bilibili*) also manifests that the four factors are cornerstones for live streaming.

Figure 2(b) compares the user satisfaction of SDSSel, GD-LM-MIN, TrustSVD, and Manual (i.e., users manually select a set of viewers and a set of channels). Each user is asked to choose the solution that generates the best user experience. The results manifest that SDSSel outperforms the other baselines because it effectively incorporates all the four crucial factors in live streaming, whereas the others are designed for different scenarios, and the manual selection does not find a nice solution. Figure 2(b) manifests that the

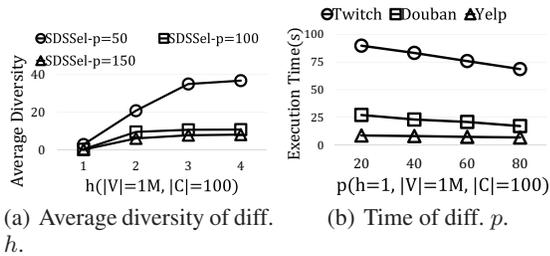


Figure 3: Effectiveness and efficiency.

requirements in live streaming are different from those in traditional video streaming (e.g., Youtube) since live streaming systems allow the viewing of multiple preferred streams simultaneously (diversity factor) and enable real-time interactions among friends (social and interest constraint).

For the manual selection, Figure 2(c) shows that the average diversity drops when the graph size increases, and Figure 2(d) shows that the *feasibility ratio* (the ratio of the feasible solutions to SDSQ) of manual selection is very low. Therefore, it is difficult for human to properly choose both the channels and group members at the same time, even with a small graph of no more than 15 individuals and 16 channels. Nevertheless, users indeed prefer the solutions specific for live streaming even though they are not able to find them, as shown in Figure 2(b). The analysis of the manual selection results shows that users prefer to have at least 4.71 friends watching the same channels together, who are usually within 1.57 hops in their social networks. This result provides good guidance to decide proper p and h values for SDSQ.

Sensitivity Tests on Large Datasets

Figure 3 evaluates `SDSSel` on a large-scale dataset that randomly samples 1M users from *Douban*. Figure 3(a) manifests that when the social constraint h increases (i.e., a looser group is allowed), the average diversity of the selected channels improves because more distant viewers who prefer diverse channels are included. Moreover, a larger p (i.e., a smaller group is not allowed) results in a smaller average diversity since the number of channels satisfying the minimum interest constraint decreases. Also, the execution time in Figure 3(b) decreases because the minimum interest constraint effectively discards the channels with fewer than p viewers. Furthermore, `SDSSel` performs significantly faster on *Yelp* and *Douban* because they both have smaller densities ($\leq 4.78 \cdot 10^{-6}$) than *Twitch* ($5.56 \cdot 10^{-6}$). As a result, the 2h-dense sets on *Yelp* and *Douban* include much fewer candidates. However, the execution time on *Twitch* is still smaller than 90 seconds, demonstrating the high efficiency of `SDSSel` on different datasets.

Comparisons with Baseline Algorithms

Figure 4 compares `SDSSel` with various baselines on *Yelp*. Because `SSGSelect` and `GD-LM-MIN` do not return the solutions in 24 hours when $|V| > 500$, we compare `SDSSel` with them by downsizing *Yelp* into a smaller net-

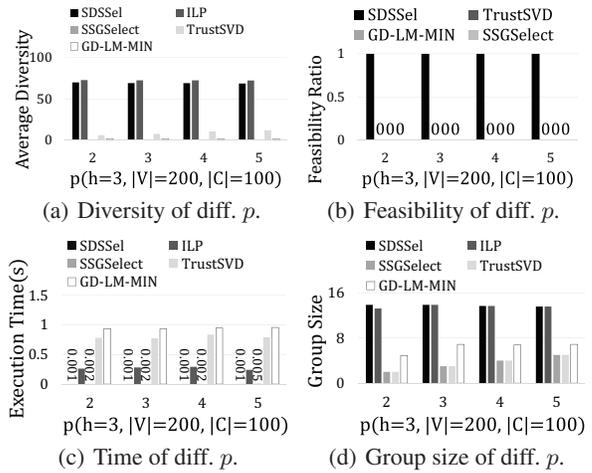


Figure 4: Comparisons with baselines on small datasets.

work ($|V| = 200$ and $|C| = 100$). Figure 4(a) shows that `SDSSel` outperforms the baselines. For a larger p , the average diversity of `SDSSel` drops because the number of channels satisfying the interest constraint (p) decreases. Nevertheless, the average diversity of `SDSSel` is still very close to the optimal solution (`ILP`). In contrast, the average diversity of `TrustSVD` increases as p grows because more channels are included in the solution. However, all solutions returned by `TrustSVD` are infeasible because the viewer groups formed by `TrustSVD` with $p \geq 2$ do not satisfy the social constraint, as shown in Figure 4(b), where *Feasibility Ratio* denotes the ratio of the number of acquired feasible solutions to the total number of tested SDSQ instances. By contrast, `SDSSel` always acquires feasible solutions, yet `TrustSVD` and `GD-LM-MIN` do not even find one feasible solution when $p \geq 2$.

For the efficiency, Figure 4(c) demonstrates that `SDSSel` (< 0.001 s) outperforms all the baselines. Although the execution time of `SSGSelect` is comparable to `SDSSel`, the solution quality of `SSGSelect` is very poor, as shown in Figures 4(a). Moreover, `TrustSVD` and `GD-LM-MIN` do not consider all important factors in live streaming, and the results confirm that the conventional preference-based approaches are not applicable to our scenario. Finally, Figure 4(d) indicates that the number of viewers selected by `SDSSel` is much more than those selected by `SSGSelect`, `TrustSVD`, and `GD-LM-MIN` because their results usually do not satisfy the interest constraint.

Conclusion

In this paper, we consider the important factors for the organization of online soirees in a live multi-streaming scenario. We formulate a new query, namely *Social-aware Diverse and Preferred Live Streaming Channel Query (SDSQ)*, to jointly find a set of diverse and preferred channels along with a group of socially tight viewers. We analyze the hardness and inapproximability of SDSQ and develop a 2-approximation algorithm, namely `SDSSel`, with a guaranteed error bound. We implement SDSQ on *Twitch* and conduct a

user study and large-scale experiments. The results demonstrate that those factors are cornerstones of live streaming, and SDSSel outperforms the baseline algorithms and manual selections by the users.

Acknowledgments

This work was supported in part by Ministry of Science and Technology (MOST), Taiwan, under MOST 105-2218-E-007-032-MY2, MOST 106-3114-E-468-001-, MOST 105-2218-E-002-035-, MOST 106-2221-E-001-015-MY2, and MOST 106-2221-E-007-115-MY2.

References

- Asahiro, Y.; Miyano, E.; and Samizo, K. 2010. Approximating maximum diameter-bounded subgraphs. In *Proceedings of the 9th Latin American Symposium (LATIN): Theoretical Informatics*, 615–626.
- Chen, K., and Zhang, Z. 2017. Learning to classify fine-grained categories with privileged visual-semantic misalignment. *IEEE Trans. Big Data* 3(1):37–43.
- Covington, P.; Adams, J.; and Sargin, E. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198.
- Drosou, M., and Pitoura, E. 2012. Disc diversity: result diversification based on dissimilarity and coverage. *PVLDB* 6(1):13–24.
- Facebook. 2016. Facebook newsroom. <https://newsroom.fb.com/news/2016/04/introducing-new-ways-to-create-share-and-discover-live-video-on-facebook/>.
- FISU. 2017. 29th summer universiade. <http://www.fisu.net/events/summer-universiades-events>.
- Guo, G.; Zhang, J.; and Yorke-Smith, N. 2015. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 123–129.
- Haimson, O. L., and Tang, J. C. 2017. What makes live events engaging on facebook live, periscope, and snapchat. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 48–60.
- Hamilton, W. A.; Tang, J. C.; Venolia, G.; Inkpen, K.; Zillner, J.; and Huang, D. 2016. Rivulet: Exploring participation in live events through multi-stream experiences. In *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video (TVX)*, 31–42.
- Hamilton, W. A.; Garretson, O.; and Kerne, A. 2014. Streaming on twitch: fostering participatory communities of play within live mixed media. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1315–1324.
- Hurley, N., and Zhang, M. 2011. Novelty and diversity in top-n recommendation - analysis and evaluation. *ACM Trans. Internet Techn.* 10(4):14:1–14:30.
- Jain, A.; Sarda, P.; and Haritsa, J. R. 2004. Providing diversity in k-nearest neighbor query results. In *Proceedings of the 8th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, 404–413.
- Koren, Y., and Sill, J. 2013. Collaborative filtering on ordinal user feedback. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 3022–3026.
- Lappas, T.; Liu, K.; and Terzi, E. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 467–476.
- Lu, Z.; Dou, Z.; Lian, J.; Xie, X.; and Yang, Q. 2015. Content-based collaborative filtering for news topic recommendation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 217–223.
- Mashable. 2017. Facebook live with chat with friends. <http://mashable.com/2017/05/23/facebook-live-with-chat-with-friends>.
- Mostafa, A. E.; Inkpen, K.; Tang, J. C.; Venolia, G.; and Hamilton, W. A. 2016. Socialstreamviewer: Guiding the viewer experience of multiple streams of an event. In *Proceedings of the 19th International Conference on Supporting Group Work*, 287–291.
- Park, C.; Kim, D. H.; Oh, J.; and Yu, H. 2016. Trecso: Enhancing top-k recommendation with social information. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, 89–90.
- R. Zafarani and H. Liu. 2009. Social computing data repository at ASU. <http://socialcomputing.asu.edu>.
- Roy, S. B.; Lakshmanan, L. V. S.; and Liu, R. 2015. From group recommendations to group formation. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 1603–1616.
- Sarwar, B. M.; Karypis, G.; Konstan, J. A.; and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference (WWW)*, 285–295.
- Shen, C.; Yang, D.; Huang, L.; Lee, W.; and Chen, M. 2016. Socio-spatial group queries for impromptu activity planning. *IEEE Trans. Knowl. Data Eng.* 28(1):196–210.
- Shuai, H.; Yang, D.; Yu, P. S.; and Chen, M. 2013. Willingness optimization for social group activity. *PVLDB* 7(4):253–264.
- Twitch. 2016a. How group chat works. <https://help.twitch.tv/customer/portal/articles/1500268-how-group-chat-works>.
- Twitch. 2016b. How to use the friends feature. <https://help.twitch.tv/customer/portal/articles/2418761>.
- Watts, D. J., and Strogatz, S. H. 1998. Collective dynamics of small-world networks. *Nature* 440–442.
- Williamson, D. P., and Shmoys, D. B. 2011. *The Design of Approximation Algorithms*. Cambridge University Press.
- Yang, D.; Shen, C.; Lee, W.; and Chen, M. 2012. On socio-spatial group query for location-based social networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 949–957.
- Yelp. 2016. Yelp dataset challenge. <https://www.yelp.com/dataset/challenge>.
- Zhang, C.; Zhang, Y.; Zhang, W.; Lin, X.; Cheema, M. A.; and Wang, X. 2014. Diversified spatial keyword search on road networks. In *Proceedings of the 17th International Conference on Extending Database Technology (EDBT)*, 367–378.
- Zhao, T.; McAuley, J. J.; and King, I. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM)*, 261–270.