Applied Network Science

RESEARCH Open Access

Trajectories through temporal networks

Carolina E. S. Mattsson[1,2*] and Frank W. Takes[1]

*Correspondence:
mattsson.c@northeasternedu
[1] Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands
Full list of author information is available at the end of the article

Abstract

What do football passes and financial transactions have in common? Both are networked walk processes that we can observe, where records take the form of timestamped events that move something tangible from one node to another. Here we propose an approach to analyze this type of data that extracts the actual trajectories taken by the tangible items involved. The main advantage of analyzing the resulting trajectories compared to using, e.g., existing temporal network analysis techniques, is that sequential, temporal, and domain-specific aspects of the process are respected and retained. As a result, the approach lets us produce contextually-relevant insights. Demonstrating the usefulness of this technique, we consider passing play within association football matches (an unweighted process) and e-money transacted within a mobile money system (a weighted process). Proponents and providers of mobile money care to know how these systems are used—using trajectory extraction we find that 73% of e-money was used for stand-alone tasks and only 21.7% of account holders built up substantial savings at some point during a 6-month period. Coaches of football teams and sports analysts are interested in strategies of play that are advantageous. Trajectory extraction allows us to replicate classic results from sports science on data from the 2018 FIFA World Cup. Moreover, we are able to distinguish teams that consistently exhibited complex, multi-player dynamics of play during the 2017–2018 club season using ball passing trajectories, coincidentally identifying the winners of the five most competitive first-tier domestic leagues in Europe.

Keywords: Trajectories, Walk processes, Temporal networks, Time-respecting paths, Process-driven, Association football, Soccer, Payment systems, Mobile money

Introduction

In many areas of applied network science, researchers are interested in studying the outcomes of particular networked processes: the spread of disease, the development of consensus, the movement of people, etc. When this is the case, domain-specific research questions often center around the process rather than the network that it is unfolding over (Bockholt and Zweig 2020). In the context of such questions, it can be difficult to interpret the results of out-of-the-box network analyses (Borgatti 2005). What we need are techniques that keep the focus of analysis on some particular networked process, itself (Lambiotte et al. 2018; Schwarze and Porter 2020; Xu et al. 2016). Here, we take a process-driven approach towards analyzing observational data about networked walk

processes with the goal of devising an approach that can answer relevant domain-specific research questions.

We focus on two specific real-world walk processes: a ball passed among players during matches within seven professional football competitions and e-money transacted among mobile wallets over a single mobile money service. Association football is a hugely popular sport and data-rich analytics of sports is of growing interest (Kuper 2011; Sarmento et al. 2014). Researchers and analysts might like to know if classic findings in sports science—such as how 80% of goals are scored from short possessions—replicate using detailed spatio-temporal match data available for recent competitions (Hughes and Franks 2005; Reep and Benjamin 1968; Reep et al. 1971). As predominant styles of play have moved away from "long ball" strategies, coaches might like to know the extent to which teams benefit from developing complex multi-player tactics (Schoenfeld 2019).

With regards to the second networked process considered in this paper, mobile money is a new financial industry that has expanded rapidly across Africa, South Asia, and Southeast Asia since 2007 (GSMA Mobile Money 2015b; Suri 2017). Mobile money providers host e-money accounts and process digital transactions on behalf of users over the cellular infrastructure, which is more widely available than traditional banking infrastructure in many areas. Mobile money providers and proponents of financial inclusion are for example interested in understanding how mobile money systems are used (International Finance Corporation and Mastercard Foundation 2018; Stuart and Cohen 2011), to what extent e-money is re-used (Athique 2019; Kendall et al. 2011), and for how long e-money is saved (Blumenstock et al. 2015; Demombynes and Thegeya 2012).

The data recorded about these processes takes the form of timestamped *events* in both cases (Blumenstock et al. 2016; Economides and Jeziorski 2017; Pappalardo et al. 2019; Sarmento et al. 2014). These events are football passes and financial transactions, respectively. Individual football players (account holders) initiate and receive near-instantaneous passes (transactions) in continuous time. While we *could* choose to interpret each event as a link in a temporal network (Aslak et al. 2018; Holme and Saramäki 2012; Rocha and Masuda 2014; Taylor et al. 2017), it is unclear how this would provide answers to the questions posed above. Instead, we propose to consider each event as a record of the movement of something tangible: football passes move the ball, financial transactions move money.

There are, however, no established techniques for analyzing event data recording steps in a real-world walk process, as such. So we first identify three ways that we would want our technique to engage with domain knowledge about particular processes. First, the method should be interpretable in light of the *integrity constraints* inherent to walk processes. Players cannot kick the ball unless they have it and bookkeeping protocols prohibit accounts from spending money they do not have. Second, we want an approach that retains the meaningful *sequential information* that is implicit in the ordering of event data. Players hold onto the ball, and accounts hold onto funds, for some period of time between sequential events. Finally, we would like to incorporate contextual knowledge on fouls and throw-ins and deposits and withdrawals and other ways in which real-world processes are in fact *bounded*, i.e., there are specific events that begin, end, or re-start the process.

We propose to extract and analyze the trajectories taken by those tangible items whose movements are recorded in the event data. Extracting trajectories can be done by tracing the same football (or the same e-money) across sequences of observed events in a systematic way. In both cases we must take care to define the bounds according to the rules governing the process. Tracing the single football is then relatively straightforward. Tracing funds is more involved, in particular because there are no unique identifiers on e-money as there are on paper bills. This weighted situation requires also an informed choice on how to allocate funds to particular trajectories where this is otherwise ambiguous. Once extracted, we can analyze trajectories to answer research questions centered around the walk processes itself. In this paper, we propose a systematic approach for extracting trajectories from both unweighted and weighted processes.

Our work highlights four benefits of extracting and analyzing trajectories, each of which lets us produce a result of relevance to association football or mobile money.

First, trajectories are a particularly useful and interpretable structure because they relate directly to concepts that are already well-researched. Since at least the 1960s, researchers in sports science have studied possessions in association football; these are passing sequences with particular criteria for delineating how they begin and end (Reep and Benjamin 1968; Reep et al. 1971). We adapt the definition laid out in Hughes and Franks (2005) to trace out trajectories and produce a dataset of possessions from the 2018 FIFA World Cup that is directly comparable to theirs from the 1990 and 1994 FIFA World Cups, albeit more data-driven. Using our transparent trajectory extraction approach we reproduce their findings that over 80% of goals were made from "short" possessions with three or fewer completed passes, and that longer passing sequences produced proportionately more shots.

Second, the pattern of event attributes along the sequence of events in a trajectory can be contextually meaningful. Trajectory extraction surfaces such sequential patterns from the data and these can be used to neatly summarize the observed process. Many stand-alone use cases of mobile money involve making more than one transaction in sequence, e.g., paying a bill would mean making a cash deposit followed by a digital bill payment (Economides and Jeziorski 2017; GSMA Mobile Money 2015b; Mbiti and Weil 2013). We find that 73% of the e-money moving through this system follows a pattern that corresponds to one of several well-defined, stand-alone, use cases. Only 19.7% of e-money was re-used within the data collection window. This means that e-money is primarily single-use, in practice, even though it could be re-transacted indefinitely with little cost (and substantial benefit) to the provider.

Third, trajectories detail the location of tangible items *between* events. In the context of mobile money, this means that we can quantify the extent to which accounts use e-money for saving. "Saving" as we intuitively understand it requires building up a balance wherein some of the money entering an account remains there, undisturbed, for an appreciable length of time. We find that 21.7% of active users of this mobile money system succeeded in saving at least 5% of inflows for over 30 days at one point or another. A much larger fraction save trivial amounts for substantial periods of time and very few save larger amounts.

Finally, extracted trajectories can serve as the input for a suite of existing computational approaches for trajectory-based network analysis (LaRock et al. 2020; Peixoto

and Rosvall 2017; Rosvall et al. 2014; Scholtes 2020). It is possible, for instance, to parametrize the Markov order of a real-world walk process (Scholtes 2017). In the context of association football, "second-order" passing processes correspond to complex multiplayer dynamics where the next pass reliably depends both on who has the ball and from whom that player received the ball. We find that only a select group of *very successful* professional club football teams played with consistent second-order passing dynamics in the 2017–2018 season. This includes the four top-ranked teams in England's Premier League, the six top-ranked teams in Italy's Serie A, as well as the champions of the Spanish La Liga, the German Bundesliga, and the French Ligue 1.

The remainder of the paper is structured as follows. In the "Theory and related work" section, we review related approaches and discuss what we gain by taking a process-driven approach. This section details the network theory behind how we observe and study real-world walk processes on networks. The "Data" section describes the specific datasets analyzed in this paper and key ancillary details about the two processes. The "Methods" section introduces trajectory extraction and various ways to analyse the resulting sets of trajectories. This section details the methodology behind our work in the form of the algorithm and its computational complexity. In the "Results" section, we apply our approach to answer four domain-specific research questions. The "Conclusion" section concludes.

Theory and related work

In this section, we first note specific issues that would arise if we were to consider football passes or financial transactions as links in a temporal network. We then discuss random walks on networks, real-world walk processes, and two distinctions that can be made regarding how real-world walk processes are observed. Records of football passes and financial transactions let us observe events, or "steps", in these two real-world walk processes as they unfold over networks that we do not observe.

Temporal networks

To analyze observational data on association football or mobile money, it would be simple to interpret each pass or transaction as a link in a temporal network. Temporal network analysis is a well-developed approach with many established techniques and available computational tools (Holme and Saramäki 2012, 2019; Lambiotte and Masuda 2016; Paranjape et al. 2017). In our particular cases, however, the most common temporal network analysis techniques would involve considerable simplification of the underlying data on passes and transactions.

Existing temporal network analysis techniques do not reflect the substantive context in which this data is generated. Time-aggregation into a static network does not capture the fact that players and account holders interact with one another almost instantaneously over a continuous period of time. Temporal network techniques that use sequences of network snapshots (Rocha and Masuda 2014; Taylor et al. 2017), or multilayer networks (Aslak et al. 2018), likewise do not help us make sense of hundreds of football passes, or hundreds of millions of financial transactions, happening one at a time. At the same time, temporal network analysis techniques that treat each link separately (e.g., motif counting, subgraph matching, and reachability analysis: Badie-Modiri et al. 2020;

Boekhout et al. 2019; Bogdanov et al. 2011; Jazayeri and Yang 2020; Kovanen et al. 2011; Locicero et al. 2021; Paranjape et al. 2017; Petrovic and Scholtes 2019) do not account for the inherently sequential dependencies in how passes and transactions come to be.

Players must receive the ball to pass the ball, and accounts must have money to spend money. This can make it difficult to interpret the outputs even of *basic* temporal network analysis methods (as in: Holme and Saramäki 2012). There are many time-respecting paths through a temporal network of football passes, but in practice the ball follows only a single one. Ambiguity in how paths should be derived from networked processes makes it difficult to interpret the outputs of centrality measures and similar methods that are based on time-respecting paths (see: Saramäki and Holme 2015). Football matches also happen under a very peculiar set of rules—inter-contact times computed on 2018 FIFA World Cup match-event data would include water breaks, but only for matches played at over 32 °C (Earls 2019; Houssein et al. 2016). Such minutiae would then muddle output metrics. As an added complication, financial transactions are weighted in a way that one cannot ignore. Transactions raise or lower a node's account balance by sometimes drastically different amounts, so paths through a node , inter-event times at a node, and motifs involving a node are also—in some sense—weighted.

Walk processes on networks

Footballs and money are tangible things, and walk processes are networked processes that correspond to the movement of tangible things. Random walks have long been used as a way to explore and quantify the structure of networks; they are a pillar of network science methodology. PageRank was developed to simulate the movement of a "surfer" who moves from page to page through a hyperlink network, randomly and with probabilistic re-starts (Page et al. 1999). Infomap finds sub-network structure by minimizing the average number of bits needed to describe one step in a random walk on the network (Rosvall and Bergstrom 2008). A set of other commonly-used network analysis techniques assume the dynamics of a walk process, more or less explicitly (Backstrom and Leskovec 2010; Fouss et al. 2007; Kloumann et al. 2017; Newman 2005).

Walk processes themselves can be weighted or unweighted, discrete or continuous, node-centric or edge-centic, and active or passive according to a taxonomy by Masuda et al. (2017). Football passing process and financial transaction processes both operate in continuous time; transactions are weighted while passes are not. The authors define *node-centric* processes as those where the dynamics of the process is defined in terms of the nodes. Players kick the ball. Accounts spend money. *Active* walk processes are those where "walkers" are agents stepping though the network of their own volition. In our case each pass in football is a "step" for the ball, and each financial transaction is a "step" for a certain amount of money, but neither footballs nor sums of money have agency in any sense. The processes in this study are thus examples of otherwise elusive *node-centric*, *passive* walk processes.

Real-world walk processes

It remains relatively uncommon to model and simulate real-world walk processes on networks. Examples with some presence in the literature include travellers and goods in transit (Heath et al. 2008; LaRock et al. 2020; Peixoto and Rosvall 2017; Xu et al. 2016),

packets routed over the internet (Ash 1997; Echenique et al. 2004; Fronczak and Fronczak 2009), and users surfing the web (Borges and Levene 2007; Chierichetti et al. 2012; Page et al. 1999; Xu et al. 2016). This work establishes two additional real-world examples: the passing process during football matches and the transaction process among financial accounts within a payment system. Here we consider two key features common across each of these real-world walk processes.

First, real-world walk processes maintain their integrity in practice and often occur within systems that are highly engineered to this end. Process integrity refers to the tendency of tangible items to stay where they are placed and not suddenly multiply or disappear. This is largely trivial for processes involving passengers, goods, footballs, or other physical items. Even so, there may be an authority overseeing the system who is able to intervene and fix glitches. Football matches are presided over by a team of referees who would quickly interrupt the match if a second ball were to come onto the field. Many important real-world walk processes rely on digital protocols to keep track of digital items. Packets are routed over the Internet using TCP/IP and related protocols; these have safeguards against packet loss and duplication (Forouzan 2002). Bookkeeping protocols can be decentralized (cash), centralized (checking), or algorithmic (blockchain). Payment system providers have a very strong incentive to ensure their bookkeeping is accurate, because they themselves end up on the hook for wayward funds. Exceptions to this rule are extraordinary—the president and chief executive of Liberty Bank in the United States chose to allow large ATM withdrawals in the aftermath of hurricane Katrina, for humanitarian reasons, although its flooded systems were unable to verify account balances at the time (Rivlin 2015).

Second, real-world walk processes are rarely, if ever, entirely self-contained. They are bounded in a way that is determined entirely by the real-world context. There may be complicated rules that begin and end walks, or related processes that create and destroy "walkers". These are conceptually distinct from the walk process itself and often substantively important. For traffic flow it matters greatly where people live and work. For money flow it matters greatly how people deposit and withdraw. Association football has very specific rules for when the ball enters and exits play, which are enforced (again) by the team of referees.

Observing walk processes on networks

Observational data about walk processes on networks can take many forms. Complete data would include information about the network structure underlying the process, the dynamics of this particular process, and the actual volumes involved. Most forms of data thus convey only partial information about a real-world walk process or do so piecemeal. The structure of the data is what determines which aspects of a walk process are directly incorporated, and which are left to be found, assumed, or inferred separately.

We systematically categorize different types of observational data about walk processes on networks in Table 1. Very often, data collection focuses on the network structure over which the process unfolds (Table 1, top row). In some cases, one can directly observe the relevant links, like roads (Hu et al. 2007; OpenStreetMap contributors 2017; Zhan and Noon 1998) or submarine fiber-optic cables (TeleGeography 2020). Such *network data* leaves the dynamics of the process implicit, for the

Table 1 Examples of observational data used to study walk processes on networks

	Implicit	**Explicit**
Network	*Network data*	*Path data*
	Transit network	Transit routes
	Internet connections	`traceroute` output
	Hyperlink network	Web crawler output
	Football passing network	Hypothetical plays
	Payment networks	Hypothetical flows
Process	*Event data*	*Trajectory data*
	Vessel manifests	Travel itineraries
	Router-based logs	Packet-based logs
	Hyperlink clicks	User click-streams
	Football match events	Passing sequences
	Transaction records	Flows of money

Listed processes are: travellers in transit, packets routed over the internet, users surfing the web via hyperlinks, football players passing a ball, and transactions within a payment system. The examples are organized to reflect how data can be collected (about the walk process itself or about the network over which it plays out) and that the structure of the data may or may not explicitly incorporate the integrity and bounds of the walk process

researcher to define separately. In other cases one actually defines process dynamics, explicitly, in order to query the network structure. Web crawlers (Thelwall 2002), tools such as `traceroute` (Cisco 2006), and transit apps (Kujala et al. 2018) give *path data* about the network underlying the processes they parrot. In both cases, the researcher would need to incorporate empirical data on volumes to get a complete view of the process.

Data can also be collected about walk processes themselves (Table 1, bottom row). This is often done in the form of timestamped events, such as airline flights (Guimerà et al. 2005) or hyperlink clicks (Dimitrov et al. 2017; Joachims 2002). *Event data* is similar to network data in that the dynamics of the walk process—that arriving passengers either transfer to a later flight or leave the airport—are implicit and would need to be handled separately. In some cases, however, it is possible to observe individual "walkers" as the process they are a part of unfolds. Passenger itineraries (LaRock et al. 2020; Xu et al. 2016) and user click-streams (Chierichetti et al. 2012; Paranjape et al. 2016; Scholtes 2017) are examples of such *trajectory data*. Trajectory data fully incorporates both the dynamics and the volume of the networked process, giving an exceptionally detailed observational account.

Transit processes are worth highlighting because each of the four combinations are well represented in the literature: Road networks are readily observable and used to study transit by car (Hu et al. 2007; OpenStreetMap contributors 2017; Zhan and Noon 1998). It is understood, implicitly, that road networks are used by individual cars that behave as tangible objects moving from their origin to their destination. Models of traffic flow take this into account, and generally supplement the observed network data with origin/destination records or measurements of traffic flow (Toole et al. 2015; Iqbal et al. 2014; Çolak et al. 2016). The movement of passengers via public transportation can be studied using the schedules of trains and busses. This data structure makes explicit the connections that would need to be made by individual passengers along each possible path and the associated travel times (Kujala et al.

2018). Even so, hypothetical path data must be supplemented with information on the actual usage of different routes (Sánchez-Martínez 2017). Data can also be collected about transit processes themselves, as in the case of passengers travelling by air (Guimerà et al. 2005). Flight manifests directly record distinct events in the transit process. But the fact that some passengers remain where they arrive, some travel onward, and none take *two* departing flights remains implicit within this data structure. Data in the form of individual travel itineraries sidesteps the issue by making process dynamics explicit (LaRock et al. 2020; Xu et al. 2016).

In this section we have presented a systematic categorization of observational data on real-world walk processes over networks. In the "Methods" section we present a method for extracting trajectory data from event data by leveraging process integrity and systematically incorporating detailed domain knowledge on process bounds. The resulting trajectory data encodes information about the dynamics of the process that were not accessible in the original event data.

Data

This paper considers football passing processes during matches played as a part of seven professional competitions and transaction processes facilitated by a mobile money provider. Recall from the "Real-world walk processes" section that these can be interpreted as observed walk processes and that real-world walk processes are bounded. Each record corresponds to an event that moved a football or some amount of e-money from one player or account to another.

Below, we describe both datasets in detail.

Football passing process

To study football passing processes, we can observe the on-ball events that occur during matches. Domain knowledge on the rules and aims of association football lets us specify the bounds of the observed passing processes.

Football match-event records

We analyze recent datasets of spatio-temporal match events from seven competitions collected by Wyscout and published in Pappalardo et al. (2019). This data includes all games played as a part of five first-tier professional domestic leagues (in 2017–2018) and two international competitions (in 2016 and 2018). Records describe match events corresponding to standardized actions that players often take to progress the ball during play. Each record contains information on the player, period, elapsed time within period, event type, event sub-type, position on the field, and outcome of an in-game action.

Table 2 summarizes the dataset for each competition by event type. The original event type schema is available from Wyscout in conjunction with the original data (https://apidocs.wyscout.com). We make two appreciable adjustments: introducing "Kick-off" events as a sub-type of "Free Kick" to mark the first event in each period and the first pass after a goal, and treating "Clearance" events as a sub-type of "Pass". We also rename the category "Others on the ball" to that of its main sub-type, "Touch" and treat "Offside" events as a sub-type of "Interruption". Various outcomes of events are reported in the data using standardized tags. Each action is deemed to

Table 2 Football match events

	International		Spain	Italy	England	France	Germany
Competition	**World Cup**	**Euro Cup**	**La Liga**	**Serie A**	**Premier League**	**Ligue 1**	**Bundesliga**
Year/Season	**2018**	**2016**	**2017/2018**	**2017/2018**	**2017/2018**	**2017/2018**	**2017/2018**
Observations							
Matches	64	51	380	380	380	380	306
Events	101,683	78,069	628,659	647,372	643,150	632,807	519,407
Event type							
Pass	58,081	45,197	327,572	346,771	338,665	328,130	268,797
Free kick	6232	5025	39,858	39,767	38,204	40,696	32,555
Shot	1419	1198	7978	8806	8450	8326	6898
Save attempt	523	458	3410	3561	3349	3436	2811
Touch	7351	3398	37,694	40,939	39,299	40,067	31,659
Duel	25,927	21,101	172,049	167,778	176,687	171,353	144,181
Foul	1766	1328	10,964	9994	8138	10,202	8656
Interruption	384	364	29,134	29,756	30,358	30,597	23,850

A summary of the competitions, matches, and match events in the association football data. Match events are grouped by event type and reported as counts

have been accurate (e.g., a pass reached its target) or not. Whenever there is a goal, this is included as a tagged outcome.

Boundary specification

The bounds of the observed football passing process are determined using the event types and tags supplied in the match-event datasets. The passing process is deemed to be started, interrupted, and re-started whenever the ball enters, exits, and re-enters regulation play. "Kick-off" events begin play at the start of a period and after a goal. The other sub-types of "Free Kick" (including also "Goal kick", "Corner", "Penalty", and "Throw-in") mark the re-start of the passing process after any of the various ways it can be interrupted (fouls, offsides, bringing the ball outside the field, referee whistle, etc.). Note that passes occurring outside of regulation play, such as during offside situations, do not appear in the data. Nor do other events that are not a part of the game, such as players handing a ball to a teammate for a throw-in (Pappalardo et al. 2019, Table 2).

Analysis conventions in sports science provide a second set of bounding criteria for passing processes—interruptions by the opposing team. We adapt a definition previously used for match event data from the 1990 and 1994 FIFA World Cups, which considers possessions as passing sequences that end when passes do not reach their intended target or are contacted by the opposition (Hughes and Franks 2005, p. 510). The data we use includes a more detailed accounting of match events, so we operationalize this criteria as follows: the passing process re-starts after passes, shots, and free kicks that were tagged as "inaccurate"; it also re-starts with passes made, shots made, and duels tagged as "won" by the opposing team. Non-passing events (shots, save attempts, touches, duels, fouls, and interruptions) unrelated to changes

in possession are considered a part of the passing sequence prior; these are ignored when they involve players on the team not in possession.

Financial transaction process

To study financial transaction processes, we can observe the transactions that occur within digital payment systems. Specifically, we consider transactions within a mobile money payment system. Mobile money providers operate primarily in countries with underdeveloped banking infrastructure. They host mobile wallets (i.e., e-money accounts), process transfers, and service payments for users over the cellular infrastructure (GSMA Mobile Money 2015b). These digital services are facilitated by a large cadre of on-the-ground *agents*. Mobile money agents create an interface between cash and e-money, as would a teller at a bank, often in conjunction with a retail shop (Cull et al. 2018). The domain-specific logic and language of payment systems lets us specify the boundary of the mobile money system within which we observe our financial transaction process.

Mobile money transaction records

We consider a large dataset of e-money transaction records from a mobile money provider in Asia covering 6 months of activity in 2016 for around 1.5 million users. The dataset contains 35 million records, each of which specifies the sender, recipient, date, amount, fee, and type of transaction along with a unique identifier.

Table 3 summarizes the dataset by transaction type. Users can deposit money onto their account via a mobile money agent (cash-dep) and via the banking system (bank-dep); users can withdraw money from their accounts via a mobile money agent or ATM (cash-wtd) and via the banking system (bank-wtd); users can transfer e-money to another user with a digital person-to-person transaction (p2p); users can also use the mobile money service to make cash payments to persons (cash-pay) and bill payments to utilities (bill-pay); finally, users can purchase pre-paid mobile calling minutes for themselves or others (mins-pay).

Mobile airtime purchases are especially numerous and orders of magnitude smaller, on average, than other transaction types. These transactions also include a timestamp

Table 3 Mobile money transaction events

Event type	Abbreviation	Transactions (%)	Tot. value (%)	Avg. value
Cash deposit	cash-dep	13.8	37.0	$237
Bank deposit	bank-dep	0.3	0.5	$143
P2P transfer	p2p	11.8	25.3	$189
Bill payment	bill-pay	4.7	4.9	$92
Cash payment	cash-pay	4.0	6.3	$138
Airtime purchase	mins-pay	56.8	1.5	$2
Cash withdrawal	cash-wtd	8.3	22.4	$238
Bank withdrawal	bank-wtd	0.4	2.0	$482

A summary of the transactions in the mobile money data, grouped by event type. The number of transactions and the total value is reported as a percentage of all transactions; the average value is reported in US Dollars at Purchasing Power Parity (PPP)

at sub-second resolution, which we use to impute more precise timestamps for transactions of other types. Transaction counts are reported in Table 3 as a share of the total number of transactions for reasons of corporate anonymity. We report amounts in US Dollars at Purchasing Power Parity (PPP).

Boundary specification

The bounds of the observed transaction process are determined using the transaction types supplied in the e-money transaction dataset. It is understood in the context of payment systems that "deposits" add money to, "withdrawals" remove money from, and "transfers" circulate money within some particular system. Considering again Table 3, the deposit transactions that place e-money into user accounts, cash-dep and bank-dep, thus start the mobile money transaction process. The corresponding withdrawal transactions, cash-wtd and bank-wtd, remove e-money from user accounts and this serves to end the transaction process. Payments and purchases likewise end the transaction process, in our particular case, as the mobile money provider handles the funds used for such transactions separately (these are: bill-pay, cash-pay, and mins-pay). P2P transfers are the only transaction type that keep funds circulating within the system among ordinary users of the system.

Methods

In this section, we present a two-step, process-driven technique for analyzing event data on a real-world walk process. The first step is to use the observed events and domain knowledge on process bounds (described for our datasets in the "Data" section) to trace out trajectories of tangible items ("Trajectory extraction" section). The second step is to conduct relevant analyses on the resulting set of trajectories ("Trajectory analysis" section). Finally, the "Experimental setup" section describes the setup used for our application of this technique to association football and mobile money, including a discussion of implementation choices and runtime.

Trajectory extraction

Below we discuss our proposed trajectory extraction method. This takes a set of events/steps and a concrete boundary specification as input. The output of the method is a set of trajectories. We define a trajectory extraction algorithm and provide its computational complexity. For the interested reader, pseudocode can be found in "Appendix: Pseudocode".

Input: event data and boundary specification

Consider a dataset D consisting of m records of events, or steps, in a real-world walk process. An event is represented as a four-tuple $d_i = (u_i, v_i, t_i, w_i)$. In the ith event (with $1 \leq i \leq m$), u_i and v_i are entities (or nodes) within some system, t_i is the timestamp at which the event occurred and w_i is a positive weight. An unweighted process is one in which just one tangible item is involved, or each item is individually identified, and hence the weight w_i is always equal to 1. Events and nodes may carry properties or attributes that characterize them. The attribute of a node n is denoted $a^{node}(n)$ and the attribute of an event $d \in D$ is denoted $a^{event}(d)$.

Note that a temporal network $G = (V, E)$ could be constructed from this event data. In this network, V is the set of nodes containing all $|V|$ entities that occur in the set of m events that form the network's edges $E = D$. Here, for reasons discussed in the "Theory and related work" section, we specifically choose *not* to focus on this temporal network, given the limitation of temporal network approaches for our desired process-driven understanding of the data. Instead we focus on how we can derive insights about the walk process that just so happened to generate this network. In particular, we extract and analyze trajectories of the tangible items involved in the process.

Extracting trajectories traces the movement of tangible items through some observed real-world system. In practice, this requires as input also a specification of the process boundary (D_{begin}, D_{end}) denoting the events that begin and end trajectories, or, equivalently, the events whereby items enter and exit the observed system. Specifying the process boundary is done by systematically identifying the events where the observed process starts, stops, and/or re-starts. In some situations this boundary can be derived directly from the event, whereas in other contexts it is dependent on the nodes involved in the event.

In a context where specific sets of event attributes (A_{begin}, A_{end}) indicate boundary events, we can simply check if event attribute $a^{event}(d_i) \in A_{begin}$ or $a^{event}(d_i) \in A_{end}$ to determine whether d_i is a boundary event. In this case, boundary $D_{begin} = \{d_i \in D \mid a^{event}(d_i) \in A_{begin}\}$ and analogously $D_{end} = \{d_i \in D \mid a^{end}(d_i) \in A_{end}\}$.

Alternatively, there may be specific sets of nodes (V_{source}, V_{sink}) that can be defined as sources and sinks. This would allow for the systematic specification of the process boundary as $D_{begin} = \{d_i = (u_i, v_i, t_i, w_i) \in D \mid u_i \in V_{source}\}$ and $D_{end} = \{d_i = (u_i, v_i, t_i, w_i) \in D \mid v_i \in V_{sink}\}$. In a similar way as for the event attributes, there may be node attributes (A_{source}, A_{sink}) against which we can check the source node attribute $a^{node}(u_i) \in A_{source}$ and target node attribute $a^{node}(v_i) \in A_{sink}$ to determine whether we are dealing with a boundary event.

Output: trajectories

A trajectory or flow f_j can be defined as a tuple $f_j = (s_j, z_j)$ containing a directed sequence of events s_j and a positive weight or size z_j representing the tangible item(s) moved by these events. In the case of an unweighted process the weight z_j is always equal to 1. The sequence of $\ell \geq 1$ events is of the form $s_j = (d_j^1, d_j^2, \ldots, d_j^\ell)$. The set of all trajectories is denoted F. Trajectories derived from a set of events satisfy a number of properties:

- *Trajectories cover the complete dataset.* Trajectories capture all item-steps in the dataset, i.e., with ℓ_j denoting the number of events in trajectory j, the sum of the weights of all steps taken in trajectories $\sum_{f_j \in F}(z_j \cdot \ell_j)$ is equal to the sum of the weights of all events $\sum_{d_i \in D} w_i$ in the dataset.
- *Trajectories are time-respecting*: in each trajectory f_j, the sequence of events over which the item(s) moved happen in that particular order, i.e., it holds that $t_j^k < t_j^{k+1}$ for all $1 \leq k < \ell$.
- *Trajectories are weight-respecting*, meaning that:

- The weight of a trajectory is always less than or equal to the weight of all events that are part of the trajectory, i.e., with w_j^k denoting the weight of event d_j^k, it holds that $z_j < w_j^k$ for all $1 \leq k \leq \ell$.
- The sum of the weights of all trajectories in which a particular event takes part (events can be, and in a weighted process often are, part of multiple trajectories) is equal to the total weight of that event, i.e., with $F(d_i)$ denoting the trajectories in which event d_i occurs, it holds that $w_i = \sum_{f_j \in F(d_i)} z_j$.

Trajectory extraction algorithm

Trajectory extraction starts with an empty set of trajectories $F := \emptyset$ and a set of partial trajectories $W := \emptyset$. The tracing procedure processes each event $d_i \in D$ in time-respecting order. For each event d_i, it is first determined whether d_i starts and/or ends a trajectory, i.e., whether $d_i \in D_{begin}$ and whether $d_i \in D_{end}$ (as defined in the "Input: event data and boundary specification" section). Then, each of the following steps are taken:

1. *Begin new trajectories*: if $d_i \in D_{begin}$ then this event is the first event in a new trajectory $f_j := (s_j, z_j)$ where $s_j := (d_i)$ and $z_j := w_i$. This new trajectory f_j is added to the working set W.
2. *Extend existing trajectories*: if $d_i \notin D_{begin}$ then this event extends at least one existing trajectory $f_k = (s_k, z_k)$ where the target node v_k^ℓ of the last event in the sequence s_k is the source node u_i of the current event d_i. The set of trajectories to choose from is $W_i = \{f_k \in W \mid v_k^\ell = u_i\}$. The weight w_i of the current event d_i is to be amassed from the trajectories $f_k \in W_i$. Let the vector $\mathbf{q}$ denote how this is collected, where q_k is the weight allocated to d_i from trajectory f_k. Maintaining a proper accounting of all item(s), the allocation must satisfy $q_k \leq z_k$ for all $f_k \in W_i$ and $w_i = \sum_{f_k \in W_i} q_k$. Finally, each trajectory f_k where $q_k > 0$ is *extended* or *partially extended*.

   - *Extension*: if $q_k = z_k$, d_i is appended to s_k.
   - *Partial extension*: if $0 < q_k < z_k$, first a new trajectory $f_j := (s_k, z_j)$ is added to set W where $z_j := z_k - q_k$. Then, f_k is extended and reduced in size; d_i is appended to s_k and $z_k := q_k$.

3. *End completed trajectories*: if $d_i \in D_{end}$ then this event is the last event in each of the trajectories of which it has been made a part, denoted $W(d_i)$. These trajectories are moved from the working set W to the result set F.

Example run: Table 4 gives a toy example. The list of five events and the boundary specification are the inputs. The set of three trajectories is the output. Our trajectory extraction algorithm proceeds in the following manner when applied to this toy example:

1. We begin with empty working and final sets of trajectories; $W := \emptyset$ and $F := \emptyset$.
2. $d_1 \in D_{begin}$ as $n_1 \in V_{source}$; this begins a new trajectory f_1 where $s_1 := (d_1)$ and $z_1 := 12$; f_1 is added to W.
3. $d_2 \notin D_{begin}$; the event d_2 can extend trajectories in W whose last event ended at node n_2; $W_2 = \{f_k \in W \mid v_k^\ell = n_2\} = \{f_1\}$; trajectory f_1 is partially extended such

Table 4 Representations of a walk process

Events

	u	v	t	w
d_1	n_1	n_2	t_1	12
d_2	n_2	n_3	t_2	2
d_3	n_2	n_4	t_3	10
d_4	n_1	n_4	t_4	20
d_5	n_4	n_5	t_5	30

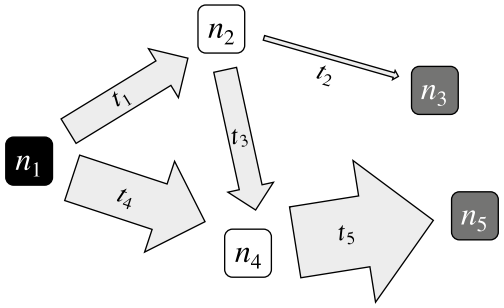
Trajectories

	s	z
f_1	(d_1, d_2)	2
f_2	(d_1, d_3, d_5)	10
f_3	(d_4, d_5)	20

Boundary

$V_{source} = \{n_1\}$ and $V_{sink} = \{n_3, n_5\}$

Temporal network



Five events recording steps in a toy walk process with the specified boundary. This process is also shown represented as a weighted, directed, temporal network with node attributes, and as a set of extracted trajectories

that $s_1 := (d_1, d_2)$ and $z_1 := 2$; the remaining weight is placed in a new trajectory f_2 where $s_2 := (d_1)$ and $z_2 := 10$.

4. $d_2 \in D_{end}$ as $n_3 \in V_{sink}$; this ends the set of trajectories of which event d_2 is a part; $W(d_2) = \{f_1\}$ and so trajectory f_1 is moved from set W to set F.
5. $d_3 \notin D_{begin}$; $W_3 = \{f_2\}$; f_2 is extended; $s_2 := (d_1, d_3)$ while z_2 remains 10.
6. $d_4 \in D_{begin}$ as $n_1 \in V_{source}$; this begins f_3 where $s_3 := (d_4)$ and $z_3 := 20$.
7. $d_5 \notin D_{begin}$; the event d_5 has weight 30 to be collected from the trajectories in $W_5 = \{f_2, f_3\}$; the allocation $\mathbf{q} = (q_2, q_3) = (10, 20)$ is valid; f_2 and f_3 are extended such that $s_2 := (d_1, d_3, d_5)$ and $s_3 := (d_4, d_5)$.
8. $d_5 \in D_{end}$ as $n_3 \in V_{sink}$; this ends the trajectories in $W(d_5) = \{f_2, f_3\}$; f_2 and f_3 are moved from set W to set F.

Observation window: In the case where a dataset D contains an exhaustive record of a walk process, trajectory extraction would begin with an empty working set of partial trajectories ($W := \emptyset$). However, a dataset D collected about a real-world system might include only events observed over a finite period. In such a case the working set W ahead of event d_1 would not necessarily be empty. W must be initialized such that all observed events that do not begin new trajectories have existing trajectories to extend. Similarly, there may be partial trajectories left in W after event d_m. To maintain a complete accounting of items, partial trajectories eventually left in W must be moved to F with a suitably defined finalization step.

Ambiguity in allocation: In the case of an unweighted process where just one tangible item is involved there will be a single extensible trajectory $W_i = \{f_j\}$ for each $d_i \in D$. Since $z_j = w_i = 1$, z_j is fully allocated to w_i, $\mathbf{q} = (q_j) = (1)$, and trajectory f_j is extended. However, a weighted process might allow situations where nodes hold multiple extensible trajectories. In our toy example, this occurs in processing event d_5 where $W_5 = \{f_2, f_3\}$. The event weight w_5 entirely exhausts the set of trajectories to choose from, that is, $w_5 = 30 = 10 + 20 = \sum_{f_k \in W_5} z_k$. However, in other cases, an

event d_i may have a smaller weight than does the set of trajectories to choose from W_i. That is, $w_i < \sum_{f_k \in W_i} z_k$ for some $d_i \in D$. It would then be ambiguous how to amass weight w_i from the extensible trajectories $f_k \in W_i$ in Step 2 of the trajectory extraction procedure.

Allocation heuristic: An allocation heuristic resolves the aforementioned ambiguity by specifying precisely how to construct the allocation vector $\mathbf{q}$ for an event $d_i \in D$, given the event weight w_i and the set of extensible trajectories W_i. Recall that q_k denotes the amount of each trajectory $f_k \in W_i$ allocated to d_i in a way that maintains a proper accounting of all item(s). Specifically, $\mathbf{q}$ must satisfy $q_k \leq z_k$ for all $f_k \in W_i$ and $w_i = \sum_{f_k \in W_i} q_k$. Two principled options for heuristics are *last-in-first-out* and *well-mixed*. The last-in-first-out heuristic gives each node a stack to organize the items it holds at any given time. Items from incoming events are added to the node's stack on top of any items already held by that node. The items at the top of a node's stack are the first to be allocated to outgoing transactions. The well-mixed heuristic, on the other hand, gives each node a pool in which to place its items. Under this formulation, items from incoming transactions mix with existing items and no added distinction is made. Items in the sending node's pool are proportionately allocated to outgoing transactions. In either case, the weight-respecting property of trajectories as defined in the "Output: trajectories" section is retained.

Computational complexity

The elementary operation of our trajectory extraction algorithm is the extension of some trajectory. Based on this, we can infer the time complexity of different variants (unweighted, weighted, for different heuristics) of the algorithm. The basic operation with respect to the time complexity of the algorithm is also the basic operation that affects memory usage: if a trajectory is extended, the extended part should be stored. As a result, the space complexity behaves in the same ways.

The computational complexity of trajectory extraction for an unweighted walk process, where just one tangible item is involved or each item is individually identified, is $O(m)$ (recall that m is the number of events in D). The computation involves a loop over all events in D and the operations within this loop include precisely one extension of some trajectory. We give evidence that the time complexity of our implementation is linear, in practice, in the "Hardware & runtime" section.

In the weighted case, the computational complexity is determined by the chosen allocation heuristic (described in the "Trajectory extraction algorithm" section). *Last-in-first-out* has a complexity of $O(m^2)$, where one trajectory extension operation is performed on $O(m)$ partial trajectories for each of the m events in D. In practice, the expected number of partial trajectories at any one node is far from m, and as the observed process (a) takes place across many nodes and (b) is bounded in that trajectories typically end (see the "Input: event data and boundary specification" section). Practical factors are especially key in considering the feasibility of the *well-mixed* heuristic, where the time complexity can reach $O(2^{m/2})$. The worst-case is a scenario where a pair of transactions is consistently used to transfer the same items to a single new recipient; each of the $m/2$ pairs then double the number of partial trajectories held by this

recipient. We discuss the empirical runtime and memory usage of our implementation in the "Hardware & runtime" section.

Trajectory analysis

Extracted trajectories hold sequential information and details about the dynamics of the walk process that were not accessible in the original event data. In this section we detail four ways to further analyze and interpret these trajectories, using sequential patterns of attributes, summary statistics, node-level properties, and system-level process dynamics.

Sequential patterns of categorical attributes

Trajectories are sequences of events, possibly with an associated weight. With potentially hundreds of thousands up to millions of different entities, direct interpretation of the extracted trajectories may be difficult. Therefore, we propose to consider *sequential patterns* of relevant categorical attributes of the nodes or the transactions along these trajectories. For example, for trajectory f_j with sequence $s_j = (d_j^1, d_j^2, d_j^3)$ and attribute values $a(d_j^1) = a(d_j^2) = x$ and $a(d_j^3) = y$, we would find the sequential pattern (x, x, y). Sequential patterns are "higher level" in that there are much fewer unique patterns along trajectories than there are unique trajectories. Moreover, sequential patterns may be interpretable in a particular domain-specific context and thus be used to produce meaningful summary statistics.

Summary statistics

Real-world walk processes can be succinctly described using summary statistics of trajectories. In addition to sequential patterns, two important attributes for summarizing are trajectory length and duration. We define a *length* for each trajectory as the number of events in the sequence, denoted for the jth trajectory as ℓ_j. The *duration* of each trajectory can be computed from the timestamps of the first and last events, i.e., $\Delta t_j = t_j^{\ell} - t_j^1$. It is thus possible to summarize counts of trajectories with a particular sequential pattern, length, or duration. More complex summary statistics such as weighted counts, averages, and medians are also possible. As in any typical data analysis task, computed statistics can be subjected to filtering or grouping. The precise approach is context-dependent, as we will see in the "Results" section.

Node-level properties

In moving along its trajectory, each item passes through a sequence of nodes where it remains for a specific duration of time. This allows us to define properties of the nodes. We consider two in particular: holding time and turnover. The *holding time* is defined for each node along a trajectory except the first and (if defined) last. The *turnover* of each node is the total weight that passes through it. Together, turnover and holding time can be used to summarize node-level process dynamics. It is possible to define the (weighted) average holding time for each node as well as the (weighted) median. Simpler to compute, and perhaps more interpretable, is the share of a node's total turnover with a holding time greater than some cutoff duration. These values are particularly relevant

to our questions about e-money savings in mobile money systems, as we will see in the "Building up e-money savings" section.

System-level process dynamics

There exists a suite of methods based on the notion that a network can be defined specifically so that a *random* walk would produce trajectories quantifiably similar to an observed set of trajectories (Lambiotte et al. 2018; Xu et al. 2016). Researchers have used this technique to find central nodes (Pfitzner et al. 2013; Scholtes et al. 2016) and detect communities (Rosvall et al. 2014; Xu et al. 2016) on the networks so revealed by individually observed trajectories. Note that this prior work will sometimes refer to each such observation as a *path*, a word we deliberately avoid in favor of *trajectory* according to the characterization in the "Observing walk processes on networks" section.

We propose to assess the so-called Markov order of obtained trajectories in order to assess the complexity of the process. The Markov order of a networked walk process is the number of prior steps that affect the next step a walker takes. Classic random walks on weighted, directed networks are first order processes (i.e., they are Markovian). More complex dynamics can generate trajectories that deviate systematically from this expectation (i.e., non-Markovian). Second-order walks are where the prior node, as well as the current node, together determine the probabilities that model the next step a walker takes. Prominent non-Markovian dynamics have been identified in, for instance, air travel where higher orders are needed to capture recurring patterns due to return travel and regional hubs (LaRock et al. 2020). The optimal Markov order of a real-world walk process can be statistically fit from observed trajectory data (Scholtes et al. 2016), and hence also from our extracted trajectory data.

Experimental setup

This study applies trajectory extraction and trajectory analysis to the two datasets described in the "Data" section. In this section we detail the setup with which we extract possessions from football match events and extract flows of money from mobile money transactions. Software and implementation details are noted as well as the specific hardware used and the algotihm runtimes.

Software & implementation

The passing process that plays out during football matches is unweighted; there is only one ball in play at any given moment during a match. For unweighted trajectory extraction (as described in the "Trajectory extraction algorithm" section) as well as for describing and summarizing the resulting trajectories (as described in the "Sequential patterns of categorical attributes" and "Summary statistics" sections) we use `pandas` (Reback et al. 2020). Neither an allocation heuristic nor an accommodation for a finite observation window are needed in this case. To compute the optimal Markov order of the observed passing dynamics (as in the "System-level process dynamics" section) we use `pathpy` (Scholtes 2020). Our plots are produced using `matplotlib` (Caswell et al. 2019). The code required to reproduce our trajectory extraction and analysis is made available in Additional file 1.

The transaction process that plays out within a digital payment system is weighted. We use `follow-the-money`, a computational implementation developed for weighted trajectory extraction (as described in the "Trajectory extraction algorithm" section) on transaction data from digital payment systems. This software can be found at https://github.com/carolinamattsson/follow-the-money and is openly available under a GNU Affero General Public License version 3 (Mattsson 2020). The two required inputs are the transaction data file and a configuration file containing the details required to define the payment system boundary. The boundary specification is described in detail in the "Financial transaction process" section and the configuration file itself is made available in Additional file 4.

Given that the process is weighted, we must also select an allocation heuristic. This choice affects the extracted trajectories and should be made deliberately. Figure 1 visually describes the heuristics from the "Trajectory extraction algorithm" section with respect to a financial transaction process. We select the *last-in-first-out* (LIFO) allocation heuristic over the *well-mixed* option for this work as it is the most attractive with respect to algorithmic complexity (see: "Computational complexity") and interpretable within our specific context.

The LIFO heuristic has several advantages in the context of payment systems, specifically. First, it is intuitive. An account that receives a $100 transfer and promptly pays rent will generate a straightforward $100 trajectory from whoever sent them the transfer, through their account, and on to their landlord. Moreover, under LIFO this person paying their rent creates the same $100 trajectory irrespective of whether they have $10 in their account or $10,000. Finally, LIFO introduces a stylized representation of savings into the system because it parallels a particular way of conceptualizing how people save money. This common, colloquial understanding of "savings" is as the funds that accumulate at the bottom of an account until the account holder needs to "dip into" them.

Given that the mobile money transaction data was recorded over a 6-month period, we must also contend with a finite observation window (as defined in the "Trajectory extraction algorithm" section). This we do by handling initial and final balances separately, employing the time-window functionality of `follow-the-money`. Our computation is initialized by inferring the existence of a prior transaction that brings the balance of each account up to the level it would need to maintain a positive balance

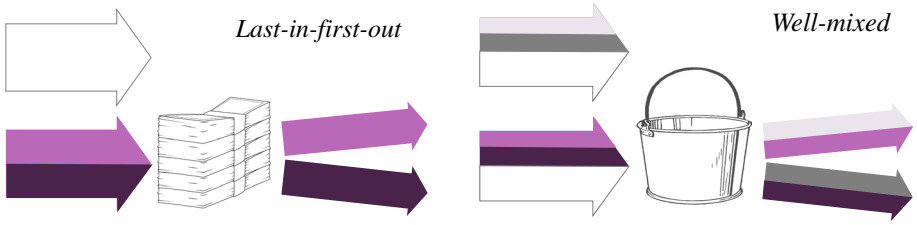


Fig. 1 Allocation heuristic. An illustration of the allocation outcomes for a simple series of transactions involving a highlighted account. This account is represented by a stack for last-in-first-out allocation, and a pool for well-mixed allocation. The account receives two $100 transactions, and later sends $50 to two different accounts. The last-in-first-out heuristic uses the most recent incoming transaction to fund the outgoing transactions, creating two $50 trajectories. The well-mixed heuristic pulls evenly from both incoming transactions, creating four $25 trajectories. Empty arrows are funds that are not yet allocated to an outgoing transaction

throughout the 6-month period. Similarly, we close out the system by inferring the existence of a later transaction that brings the balance of each account down to zero. Instead of incomplete trajectories, then, `follow-the-money` produces trajectories that begin or end with transactions of type "inferred" and these can be analyzed alongside the complete trajectories.

We also employ the functionality provided in `follow-the-money` to account for the transaction fees charged on some transactions and to avoid continuing to trace trajectories that become smaller than one unit of the local currency for reasons of, e.g., floating point arithmetic. The scripts used to run the computations are made available in Additional file 3. Summarizing and analyzing the resulting trajectories is done using `follow-the-money`, `pandas`, and `matplotlib`.

Finally, note that in running trajectory extraction we make three specific assumptions about each of the datasets. The first is completeness; we assume these datasets include a record of all events during the observation window. The second is that these datasets are correctly ordered in time. And, lastly, we assume no observed events violate process integrity as defined in the "Real-world walk processes" section. Passes and transactions that are disallowed for reasons of process integrity should not happen, be thus impossible to observe, and not end up in the data.

Hardware & runtime

Unweighted trajectories were extracted from the football match-event datasets and boudary specification described in the "Football passing process" section. Trajectory extraction was run on a machine with a 2.3 GHz quad-core processor and 16 GB memory. Table 5 presents the runtime of the algorithm and the resulting number of possessions. For convenience, the number of matches and events are restated from Table 2. The runtime increases approximately 29.5 seconds per 100, 000 events (sample standard deviation $s = 0.77$), which is linear, precisely as theoretically shown in the "Computational complexity" section.

Weighted trajectories were extracted from the mobile money transaction dataset and boundary specification described in the "Financial transaction process" section. This was run on one computing core with a 2.2 GHz processor. This computation utilized 1 h 9 min and 49 s of CPU time, required 5.89 GB of memory, and resulted in a dataset of 33 million trajectories. Note that our case is far from the worst-case with respect to the time-complexity of the algorithm for a weighted walk process under LIFO as discussed in the "Computational complexity" section. There are a large number of active accounts

Table 5 Possessions extracted from each football match-event dataset and the runtime of this step

	International		Spain	Italy	England	France	Germany
Competition	**World Cup**	**Euro Cup**	**La Liga**	**Serie A**	**Premier League**	**Ligue 1**	**Bundesliga**
Year/season	**2018**	**2016**	**2017/2018**	**201720/18**	**2017/2018**	**2017/2018**	**2017/2018**
Matches	64	51	380	380	380	380	306
Events	101,683	78,069	628,659	647,372	643,150	632,807	519,407
Possessions	25,470	20,765	162,583	163,817	169,926	164,651	138,042
Runtime	29 s	24 s	187 s	191 s	189 s	191 s	153 s

(around 1.5 million) and events that end trajectories (i.e., payments and withdrawals) are prominent (see Table 3), two aspects that, as stated in the "Computational complexity" section, ensure the number of partial trajectories ending at any one node is a very small number compared to m, realizing very feasible running times in practice.

Results

In this section, we share four results obtained using trajectories extracted from the match event and transaction data. First, we use trajectories to replicate classic findings from sports science on possession lengths in association football. Second, we summarize how account holders use mobile money services by grouping trajectories of e-money into meaningful categories. Third, we quantify the extent to which account holders build up savings in e-money. Lastly, we demonstrate that passing play of a higher Markov order distinguishes exceptional club teams in five top association football leagues.

Passing sequences, shots, and goals

Passing sequences or possessions have been used to study association football at the professional level since long before the current age of plentiful, detailed, data on games. Reep and Benjamin (1968) observed that around 80% of goals are scored from short possessions, meaning those with three or fewer completed passes, and that it takes around 10 shots to score one goal (see also: Reep et al. 1971). Hughes and Franks (2005) confirm these findings using match data from the FIFA World Cup tournaments in 1990 and 1994. They contend that so many goals are scored from short passing sequences simply because these are more numerous; longer passing sequences are *more likely* to lead to shots and goals.

The aim of our analysis is to establish if these classic findings in sports science can be replicated decades later using the kinds of detailed spatio-temporal match data that have become available for recent competitions, specifically the 2018 FIFA World Cup. Tracing out trajectories lets us delineate possessions using systematic and transparent criteria similar to those described in Hughes and Franks (2005) (see the "Football passing process" section). This makes our data directly comparable to that used in the earlier work. From the 101,683 spatio-temporal match events we extract 25,470 trajectories that each correspond to a possession (see Table 5). As described in the "Sequential patterns of categorical attributes" section, we can compute specific features of these trajectories using the attributes of the match events that make them up. Specifically, we designate the number of "accurate" passes (including set pieces that are not themselves shots) as the possession length and note whether each sequence of events led to a shot and/or a goal.

Counting the trajectories at each length in each outcome category lets us produce Fig. 2 which reproduces key figures from Hughes and Franks (2005, Figures 1-3, and 6 on pgs. 510–512). The top panels show the length-distribution of possessions and of the subset that led to goals. These are strikingly similar to the equivalent plots in the prior work, with perhaps more of the very longest possessions and more goals from zero-length possessions (in our case, these are predominantly direct shots from set pieces and goals from rebounds). By our count, 82.2% of goals during the 2018 FIFA World Cup resulted from possessions with passing sequences of length three or less. The lower panels in Fig. 2 show, as do the prior authors, that longer passing sequences were more

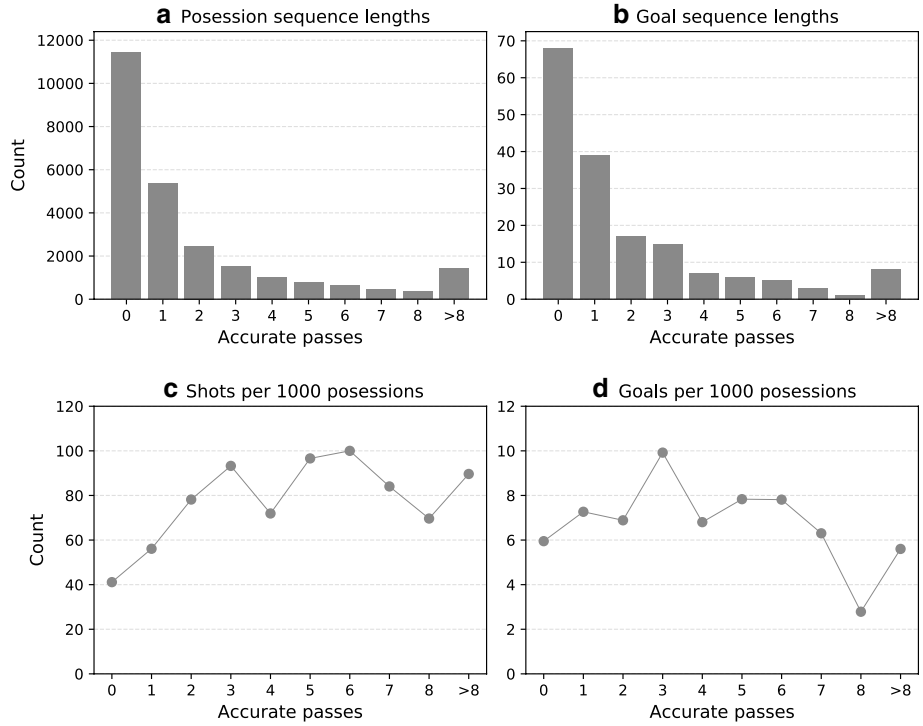


Fig. 2 Football passing sequences. The length distribution of **a** possession sequences during the 2018 FIFA World Cup and **b** possession sequences that ended in a goal. Also the conversion ratio of possessions into **c** shots and **d** goals at each length. Possession length is the number of "accurate" passes (including free kicks that are not themselves shots) during that sequence of events. Penalty shootouts are excluded

likely to result in shots; and it still takes around 10 shots to score one goal. What is less clear from this more recent data, however, is whether longer passing sequences continue to have a higher conversion ratio into goals. This could be due to changes in the game over the intervening decades. Regardless, the finding that around 80% of goals are scored from short possessions appears to hold in 2018 just as it did 28 years earlier.

Use and circulation of mobile money

Mobile money is relatively new and potentially revolutionary digital financial infrastructure (GSMA Mobile Money 2018). Understanding how account holders use these systems is of great interest to mobile money providers and proponents of financial inclusion (Almazan and Lynn 2015; Cull et al. 2018; International Finance Corporation and Mastercard Foundation 2018; Stuart and Cohen 2011). However, user behavior can be difficult to relate to raw transaction data. For instance, neither survey takers nor the users they poll tend to consider deposits and withdrawals as separate services (Intermedia 2016). Account holders wishing to accomplish a particular task will often need to make more than one transaction to get their money where they need it to go. This makes it especially difficult to measure the extent to which e-money is being meaningfully re-used, as opposed to trivially re-transacted in completing a single task. Consistent re-use of e-money is a precondition for reaching

the grandest goals of some mobile money proponents, who envision a world where e-money comes to replace cash (Athique 2019; Kendall et al. 2011).

Here, we summarize how account holders use a mobile money service by meaningfully grouping the e-money trajectories extracted from the providers' own transaction records. Tracing out weighted trajectories using a relevant boundary specification and allocation heuristic (see the "Financial transaction process" section) lets us follow e-money across multiple sequential transactions, each of which has a particular type. Sequential patterns of transaction types are readily interpretable as stand-alone use cases of mobile money: the prototypical use case is a digital transfer that involves a cash deposit, then a person-to-person digital transaction, and finally a cash withdrawal (Mbiti and Weil 2013). Paying a bill or purchasing pre-paid mobile minutes (i.e., airtime) would entail making a cash deposit followed by the payment transaction. Where providers offer a formal over-the-counter service, as in our case, e-money from a cash deposit can also be used to pay another person in cash (GSMA Mobile Money 2015b). Mobile money systems are also used for money storage and savings wherein cash is deposited only to be withdrawn again sometime later. Economides and Jeziorski (2017) describe this transaction pattern as a means to avoid carrying cash while travelling and to avoid keeping cash at home over the short to medium term. It would also occur when users are maintaining e-money in their mobile money accounts for a longer period of time as a form of savings (Jack and Suri 2011). Person-to-person transactions that are *not* subsequently withdrawn are special; they keep e-money "circulating" within the mobile money system where they can be meaningfully re-used. As described in the "Summary statistics" section, we can group trajectories by these contextually relevant transaction patterns to produce meaningful summary statistics about the use of this mobile money system.

We find that mobile money is primarily single-use. Table 6 presents a detailed summary of this trajectory data, showing the five stand-alone patterns and the corresponding "circulating" patterns that include at least one meaningful re-use. The 35

Table 6 Sequential patterns in mobile money use

Use case	Transaction type motif	Number (%)	Value (%)	Duration (h)
Storage/savings	cash-dep → cash-wtd	5.9	23.1	21
Digital transfer	cash-dep → p2p → cash-wtd	5.9	23.5	24
Circ. digital transfer	cash-dep → [p2p] → p2p → cash-wtd	3.6	8.8	67
Cash payment	cash-dep → cash-pay	4.0	12.5	1.5
Circ. cash payment	cash-dep → [p2p] → cash-pay	2.5	4.2	59
Bill payment	cash-dep → bill-pay	4.0	8.8	0.1
Circ. bill payment	cash-dep → [p2p] → bill-pay	3.3	4.2	77
Airtime purchase	cash-dep → mins-pay	40.3	2.4	165
Circ. airtime purchase	cash-dep → [p2p] → mins-pay	21.2	1.2	276

Trajectories observed when tracing funds using the last-in-first-out heuristic, grouped by their sequential transaction patterns. The number and value of trajectories is reported as a percentage of the number and value of all trajectories that begin with a deposit made in the first 5 months of data collection. The median duration is weighted by value. Brackets denote that consecutive person-to-person transactions have been consolidated. Not shown are sequential transaction patterns reflecting niche actions (i.e., below 1% of trajectories) or incomplete trajectories (i.e., money that remains in the system at the end of the finite data collection window). For this reason, the percentages do not add up to 100%

million transactions worth a total of $3.1 billion (PPP) become 33 million trajectories totaling $1.7 billion (PPP). Considering those trajectories where the e-money was deposited within the first 5 months of our data collection window, stand-alone use cases amount to 73% of activity. Only 19.7% of e-money was observed "circulating" within the system before exiting (7.3% remained in the system at the end of the finite data collection window). Across use cases, the median unit of e-money that is re-used remains in the system for considerably longer than its single-use counterpart. In a relatable anecdote, most of the e-money used for bill payments moves through the system in under an hour perhaps because many bills are paid last-minute.

Building up e-money savings

The possibility that mobile money could promote personal savings in countries with under-developed banking sectors has been raised (Demombynes and Thegeya 2012; Jack and Suri 2011) and is often touted by development agencies (Global Development Program 2012). However, the causal effect of mobile money on savings is inconclusive (Blumenstock et al. 2015; Aker et al. 2016) and uptake of savings-specific services offering a rate of return has been low (GSMA Mobile Money 2015a; Suri 2017). This is perhaps not unexpected—the act of saving requires a person to leave some amount of money undisturbed for a long period of time and those in precarious financial situations face many challenges to building up savings, in general (Banerjee and Duflo 2012).

The aim here is to quantify the extent to which mobile money users build up savings as e-money in their accounts. Our trajectories note the length of time that e-money spends in each visited account, and as explained in the "Software & implementation" section our choice of heuristic gives this a direct interpretation in the context of saving. Money that recently entered an account is used first and only when more recent funds are exhausted do older, longer-saved funds get used. As described in the "Node-level properties" section, we can find the total turnover for each account and the share of this that the account holder left undisturbed for longer than a given duration. From this we calculate the fraction of accounts that left undisturbed over 20%, 10%, 5% and 0% of their turnover for longer than some length of time. One might consider 30 days as an appropriate cutoff for some balance to have been successfully "saved".

We find that mobile money is *rarely* used to build up sizeable savings balances. Figure 3 shows the percentage of accounts with more than three incoming transactions that managed to accumulate a savings balance over a period up to and above 30 days. Our count shows that 21.7% of such users succeeded in saving 5% of incoming funds for over 30 days at some point during the period that we observed. The majority did not save even 1% of incoming funds for that long. Many accounts do maintain small balances for long periods of time, as indicated by the slow decline of the curve for any non-zero balance. The other curves decline faster as fewer accounts maintained more sizeable balances over the same lengths of time.

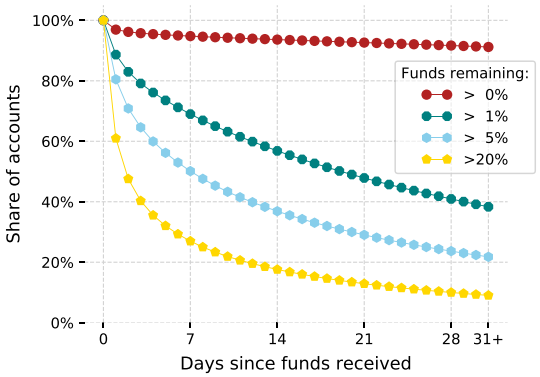


Fig. 3 Mobile money savings. The share of accounts that left savings balances undisturbed for longer than a period of time up to and above 30 days. The curves denote savings balances that correspond to 0%, 1%, 5%, and 20% of the total amount the account received during the first 5 months of data collection. These curves decline as fewer accounts maintained such balances over longer periods of time. Funds entering accounts before or after the first 5 months of data collection are not included; this ensures we could track all funds for at least 31 days. Accounts with fewer than three incoming transactions are excluded from the count

Multi-player tactics in association football

Association football is an intensely competitive environment where teams employ different strategies and tactics in seeking some advantage over an opposing team. Here we use trajectories to investigate whether or not teams consistently demonstrate complex multi-player tactics. Specifically, the concept of *Markov order* described in the "System-level process dynamics" section can be used to quantify the sustained complexity of a team's passing play. First-order passing processes are best described by a network—players routinely pass the ball to particular teammates. We would expect teams that consistently execute complex multi-player tactics in their play to generate passing dynamics with a Markov order greater than one, meaning they go beyond what is captured by a simple network model. Whether from ingrained practice of multi-player tactics, less interference from the opposing team, or exceptional situational awareness, the players would appear to take into account who they received the ball *from* in who they pass the ball *to*.

In our dataset of five first-division domestic competitions described in the "Football passing process" section, we find that a small but exceptionally successful subset of teams generated complex passing dynamics over the 2017–2018 season. Each of the champions of our five domestic leagues played with second-order passing dynamics. So did the next three top-ranked teams in England's Premier League and the next *five* top-ranked teams in Italy's Serie A. Figure 4 gives the full rankings, where teams generating second-order passing dynamics are shown in bold. Of these 23 teams: 17 finished in the top 5, 3 more in the top 10, and 3 in the bottom half of their respective leagues. We consider this to be evidence of complex multi-player tactics at the top echelon of professional club teams in association football.

On the other hand, passing play during international competitions consistently corresponds to a first-order process. That passing is Markovian holds for all teams who played in the 2018 FIFA World Cup and the 2016 UEFA European Cup. None of the national teams that we observe had players engaging in complex multi-player tactics with enough

Rank	Spain La Liga	Italy Serie A	England Premier League	France Ligue 1	Germany Bundesliga
1	**Barcelona**	**Juventus**	**Man. City**	**Paris S-G**	**Bayern**
2	Atlético Madrid	**Napoli**	**Man. United**	Monaco	Schalke
3	**Real Madrid**	**Roma**	**Tottenham**	**Lyon**	Hoffenheim
4	Valencia	**Inter Milan**	**Liverpool**	**Marseille**	**Dortmund**
5	Villarreal	**Lazio**	Chelsea	Rennes	**Leverkusen**
6	Real Betis	**Milan**	Arsenal	Bordeaux	**RB Leipzig**
7	Sevilla	Atalanta	Burnley	St-Étienne	VfB Stuttgart
8	Getafe	Fiorentina	Everton	Nice	Eintracht Frankfurt
9	Eibar	Torino	Leicester City	Nantes	Mönchengladbach
10	Girona	**Sampdoria**	Newcastle	Montpellier	Hertha
11	Espanyol	Sassuolo	Crystal Palace	Dijon	Werder
12	Real Sociedad	Genoa	**Bournemouth**	**Guingamp**	Augsburg
13	**Celta de Vigo**	Chievo	West Ham	Amiens SC	Hannover 96
14	Alavés	Udinese	Watford	Angers	Mainz
15	Levante	Bologna	Brighton	Strasbourg	SC Freiburg
16	Athletic Club	Cagliari	Huddersfield	Caen	Wolfburg
17	Leganes	SPAL	Southampton	Lille	Hamburger SV
18	Deportivo	Crotone	Swansea	Toulouse	FC Köln
19	Las Palmas	Verona	Stoke City	Troyes	
20	Málaga	Benevento	West Brom	Metz	

Fig. 4 League rankings. The final 2017–2018 season rankings of five professional leagues for association football. In bold are the 23 teams that demonstrated second-order passing dynamics during play over the course of the season

consistency for a second-order process to better fit the trajectory data. The chance that the ball moved from one player to the next aligns better with statistical expectations *without* taking into account multi-player sequential combinations.

Conclusion

This paper has demonstrated a new approach for analyzing networked walk processes. We systematically characterized observational data about real-world walk processes on networks, noting that event data is common but has properties that prohibit the use of standard approaches from temporal network analysis. We then proposed a trajectory extraction technique that respects integrity constraints, incorporates domain-specific process bounds, and retains inherent sequential information. This method was applied to mobile money and association football by considering transactions and passes as records of events in the respective real-world walk process.

Regarding football, trajectories let us replicate classic findings on possessions from sports science, demonstrating that several findings about the game in 1990 still hold in 2018. We also demonstrated that passing play is a first order Markovian process among most teams, while exceptional league teams show non-Markovian dynamics. Higher-order passing dynamics let us identify the top teams in the most competitive European leagues. In the domain of mobile money, trajectories let us summarize use of a system and quantify the extent to which account holders build up e-money savings; both are of top concern for the payment industry as they help better understand the system's clients.

Proponents of financial inclusion, and perhaps also regulators, might use these new metrics to compare and monitor mobile money systems.

Within both domains this work opens up considerable avenues for further research. Regarding football, the question of *why* many top league teams play with second-order dynamics is deserving of study. Event data from other team sports may also benefit from analysis as unweighted walk processes with bounds delineated by the rules of the game. Regarding mobile money, it is of likely interest whether providers offering different services (or operating under different regulatory frameworks) are used similarly. Our approach is also applicable to transaction records from other systems including app-based, intra-bank, and large-value payment systems.

Taking a methodological perspective on potential future work, each of our results is an empirical finding that could serve to better parametrize walk-based models of the observed network processes. We would like realistic models of real-world walk processes to reproduce basic features of empirical trajectories; this is already the logic underlying multi-order network representations of complex systems (Lambiotte et al. 2018; Xu et al. 2016). There is every opportunity for future work to incorporate meaningful process bounds, weighted walks, and a notion of continuous time into these types of frameworks.

Appendix: Pseudocode

The approach presented in the "Trajectory extraction algorithm" section outlines the generic procedure of trajectory extraction, regardless of the chosen allocation heuristic. In Algorithm 1 we present pseudocode that implements this trajectory extraction procedure using the LIFO heuristic, which we also employ in our experiments in the "Results" section.

Recall from the "Input: event data and boundary specification" section that the input consists of the event data D and process boundary D_{begin}. For readability purposes, we assume here that there are no events containing self-loops, e.g., events $d_i = (u_i, v_i, t_i, w_i) \in D$ for which $u_i = v_i$. For initialization of book-keeping array E, we also input the number of nodes n, which is simply the number of unique entities u_i or v_i over all events in D, and generally known. The output is a set of trajectories F made to contain all trajectories, both partial and complete. This ensures that we need not include D_{end} as a part of the input nor define a finalization step. In this way, Algorithm 1 outputs trajectories that precisely satisfy the completeness, weight-respecting and time-respecting properties outlined in the "Output: trajectories" section. It works as follows.

Algorithm 1: TrajectoryExtraction-LIFO

Input: Dataset D of events $d_i = (u_i, v_i, t_i, w_i) \in D$ sorted by timestamp t_i, boundary $D_{begin} \subseteq D$, number of nodes n
Output: Set F of trajectories $f_j = (s_j, z_j) \in F$

```
1  F ← ∅;
2  E[n] ← new [];
3  foreach d_i = (u_i, v_i, t_i, w_i) ∈ D do                    // ordered by timestamp
4      if d_i ∈ D_begin then                                    // start a new trajectory
5          s ← [d_i];    z ← w_i;    f ← (s, z);
6          F ← F ∪ {f};
7          E[v_i] ← E[v_i] + [f];
8      else                                                     // extend existing trajectories
9          foreach f_j = (s_j, z_j) ∈ E[u_i] do                 // in reverse order for LIFO
10             if z_j ≤ w_i then                                // simple extension
11                 s_j ← s_j + [d_i];
12                 E[u_i] = E[u_i] − [f_j];    E[v_i] = E[v_i] + [f_j]
13             else                                             // partial extension
14                 s ← s_j + [d_i];    z ← w_i;    f ← (s, z);
15                 F ← F ∪ {f};
16                 E[v_i] = E[v_i] + [f];
17                 z_j ← z_j − w_i;
18             end
19             w_i = w_i − z_j;
20             if w_i ≤ 0 then        // stop if no more weight needs to be distributed
21                 break;
22             end
23         end
24     end
25 end
26 return F;
```

The algorithm starts with initialization of the result set of (partial) trajectories F and the node-indexed array of lists E used for storing references to (partial) trajectories ending at that node (lines 1–2). Then, the main loop defined in line 3 of the algorithm iterates over all $m = |D|$ events, which are ordered by timestamp. If an event (based on its type, or based on its nodes, as determined in the "Input: event data and boundary specification" section) is part of the starting boundary, a new trajectory is started (lines 4–7). This consists of initializing it based on the current event and its weight (line 5), adding it to the set of trajectories (line 6) and storing that the new trajectory ends at the target node of the current event (line 7).

If the current event $d_i = (u_i, v_i, t_i, w_i)$ is not in the starting boundary, we loop over the (partial) trajectories in $E[u_i]$, containing all the trajectories that currently end in u_i and could possibly be extended by the current event. Crucially, this is done in reversed order of the list $E[u_i]$, to ensure that the LIFO heuristic is implemented. If the current weight w_i is equal to the weight z_j of the trajectory under consideration (line 10), that trajectory is extended by the current event d_i (line 11) and bookkeeping is done to track that this trajectory now ends at v_i (line 12). The same lines 10–12 are executed if the current trajectory has a weight z_j smaller than the current weight w_i. Then, the remaining weight, as set later in line 19, ensures that the loop initiated in line 9 continues to look for trajectories to which the current event can be appended. Alternatively, if $z_j > w_i$, lines 13–17 apply, ensuring that the trajectory under consideration is partially extended. A copy of the current trajectory is made, given precisely the weight w_i left to be distributed, and extended by the current event (lines 14–16) while the remaining weight $z_j - w_i$ is left behind in the current

trajectory (line 17). Next, the weight left to be distributed is decremented by the weight of the trajectory that was just extended (line 20), and a check is done to see if the procedure should be terminated because all the weight has been distributed (lines 21–22). Finally, the set of trajectories is returned (line 27) and the algorithm terminates.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s41109-021-00374-7.

Additional file 1. This file is a Jupyter Notebook containing trajectory extraction and analysis code to reproduce the results in the "Passing sequences, shots, and goals" and "Multi-player tactics in association football" sections.

Additional file 2. This is a comma-separated data file containing the trajectories extracted from the 2018 FIFA World Cup football match-event data as detailed in the "Experimental setup" section.

Additional file 3. This text file contains the program execution scripts used to extract weighted trajectories from the mobile money transaction data as detailed in the "Experimental setup" section.

Additional file 4. This JSON file is the configuration file used in extracting weighted trajectories from the mobile money transaction data as detailed in the "Experimental setup" section.

Additional file 5. This HTML file displays the analysis code that produced the results in the "Use and circulation of mobile money" and "Building up e-money savings" sections.

Acknowledgements
We thank Geoff Canright, Kenth Engø-Monsen, and David Lazer for establishing institutional support. We thank Brennan Klein, Soodabeh Milanlouei, Guy Stuart, Soren Heitmann, Alessandro Vespignani, and Sean P. Cornelius for useful comments and discussion.

Authors' contributions
CESM performed the analysis and wrote the manuscript. Both authors developed the theory, formalized the methods, discussed the results, and edited the final manuscript. The authors read and approved the final manuscript.

Availability of data and materials
The association football datasets analyzed in this study are available at https://figshare.com/collections/Soccer_match_event_dataset/4415000/2 from Pappalardo et al. (2019). The data that support the findings of the mobile money portion of this study are available from Telenor Research but restrictions apply to the availability of these data, which were used under a Collaborative Research and Data Use Agreement with Northeastern University for the current study.

Availability of software and code
All software used during this study are available under an open-source licence: https://pandas.pydata.org/ (Reback et al. 2020), https://matplotlib.org/ (Caswell et al. 2019), https://www.pathpy.net/ (Scholtes 2020) and https://github.com/carolinamattsson/follow-the-money (Mattsson 2020). The configuration files, program execution scripts, and analysis code used during this study are included in this published article and its Additional files 3, 4, 5.

Declarations

Competing interests
Publication of this manuscript may affect the value of US Provisional Patent 62/809,359 filed by Northeastern University; CESM would benefit financially from its commercialization. Other authors declare that they have no competing interests.

Human subjects
This study was ruled Exempt, Category #4 under Northeastern University IRB# 18-07-16, requiring safeguards against attempts at re-identification.

Author details
[1]Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands. [2]Network Science Institute, Northeastern University, Boston, MA, USA.

Received: 2 December 2020 Accepted: 14 April 2021
Published online: 03 May 2021

References

Aker JC, Boumnijel R, McClelland A, Tierney N (2016) Payment mechanisms and anti-poverty programs: evidence from a mobile money cash transfer experiment in Niger. Center for Global Development Working Paper 268. Accessed 12 Oct 2016

Almazan M, Lynn E (2015) OTC & mobile money: making sense of the data. https://www.gsma.com/mobilefordevelopment/programme/mobile-money/otc-mobile-money-making-sense-of-the-data. Accessed 13 Nov 2018
Ash GR (1997) Dynamic routing in telecommunications networks, 1st edn. McGraw-Hill Professional, New York
Aslak U, Rosvall M, Lehmann S (2018) Constrained information flows in temporal networks reveal intermittent communities. Phys Rev E. https://doi.org/10.1103/PhysRevE.97.062312
Athique A (2019) A great leap of faith: the cashless agenda in Digital India. New Media Soc 21(8):1697–1713. https://doi.org/10.1177/1461444819831324
Backstrom L, Leskovec J (2010) Supervised random walks: predicting and recommending links in social networks. arXiv:1011.4071 [physics, stat]. Accessed 14 Sept 2020
Badie-Modiri A, Karsai M, Kivelä M (2020) Efficient limited-time reachability estimation in temporal networks. Phys Rev E 101(5):052303. https://doi.org/10.1103/PhysRevE.101.052303
Banerjee A, Duflo E (2012) Chapter 8: Saving brick by brick. In: Poor economics: a radical rethinking of the way to fight global poverty, Reprint edition edn. PublicAffairs, New York
Blumenstock JE, Callen M, Ghani T, Koepke L (2015) Promises and pitfalls of mobile money in Afghanistan: evidence from a randomized control trial. In: Proceedings of the seventh international conference on information and communication technologies and development. ICTD '15. Association for Computing Machinery, New York, pp 1–10. https://doi.org/10.1145/2737856.2738031. Accessed 15 Sept 2020
Blumenstock JE, Eagle N, Fafchamps M (2016) Airtime transfers and mobile communications: evidence in the aftermath of natural disasters. J De Econ 120:157–181. https://doi.org/10.1016/j.jdeveco.2016.01.003
Bockholt M, Zweig KA (2020) Towards a process-driven network analysis. Appl Netw Sci 5(1):56. https://doi.org/10.1007/s41109-020-00303-0
Boekhout HD, Kosters WA, Takes FW (2019) Efficiently counting complex multilayer temporal motifs in large-scale networks. Comput Soc Netw 6(1):8. https://doi.org/10.1186/s40649-019-0068-z
Bogdanov P, Mongiovì M, Singh AK (2011) Mining heavy subgraphs in time-evolving networks. In: 2011 IEEE 11th international conference on data mining, pp 81–90. https://doi.org/10.1109/ICDM.2011.101
Borgatti SP (2005) Centrality and network flow. Soc Netw 27(1):55–71. https://doi.org/10.1016/j.socnet.2004.11.008
Borges J, Levene M (2007) Evaluating variable-length Markov chain models for analysis of user Web Navigation Sessions. IEEE Transn Knowl Data Eng 19(4):441–452. https://doi.org/10.1109/TKDE.2007.1012
Caswell TA, Droettboom M, Hunter J, Firing E, Lee A, Klymak J, Stansby D, Andrade ESd, Nielsen JH, Varoquaux N, Root B, Hoffmann T, Elson P, May R, Dale D, Lee J-J, Seppänen JK, McDougall D, Straw A, Hobson P, Gohlke C, Yu TS, Ma E, Vincent AF, Silvester S, Moad C, Katins J, Kniazev N, Ariza F, Ernest E (2019) matplotlib/matplotlib v3.1.0. Zenodo. https://doi.org/10.5281/zenodo.2893252
Chierichetti F, Kumar R, Raghavan P, Sarlos T (2012) Are web users really Markovian? In: Proceedings of the 21st international conference on World Wide Web. WWW '12. Association for Computing Machinery, New York. https://doi.org/10.1145/2187836.2187919, pp 609–618. Accessed 14 Sept 2020
Çolak S, Lima A, González MC (2016) Understanding congested travel in urban areas. Nat Commun 7(1):1–8
Cisco: Understanding the Ping and Traceroute Commands. Cisco (2006). https://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html. Accessed 14 Sept 2020
Cull R, Gine X, Harten S, Heitmann S, Rusu AB (2018) Agent banking in a highly under-developed financial sector: evidence from Democratic Republic of Congo. World Dev 107:54–74. https://doi.org/10.1016/j.worlddev.2018.02.001
Demombynes G, Thegeya A (March 2012) Kenya's mobile revolution and the promise of mobile savings. SSRN Scholarly Paper ID 2017401, Social Science Research Network, Rochester, NY http://papers.ssrn.com/abstract=2017401. Accessed 03 Oct 2016
Dimitrov D, Singer P, Lemmerich F, Strohmaier M (2017) What makes a link successful on Wikipedia? In: Proceedings of the 26th international conference on World Wide Web. WWW '17. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, pp 17–926. https://doi.org/10.1145/3038912.3052613. 23 Feb 2021
Earls M (2019) As the World Heats Up, Soccer Must Adapt. Scientific American. Accessed 14 Sept 2020
Echenique P, Gómez-Gardeñes J, Moreno Y (2004) Improved routing strategies for Internet traffic delivery. Phys Rev E 70(5):056105. https://doi.org/10.1103/PhysRevE.70.056105
Economides N, Jeziorski P (2017) Mobile money in Tanzania. Mark Sci 36(6):815–837. https://doi.org/10.1287/mksc.2017.1027
Forouzan BA (2002) TCP/IP protocol suite, 2nd edn. McGraw-Hill Inc., New York
Fouss F, Pirotte A, Renders J-M, Saerens M (2007) Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans Knowl Data Eng 19(3):355–369. https://doi.org/10.1109/TKDE.2007.46
Fronczak A, Fronczak P (2009) Biased random walks in complex networks: the role of local navigation rules. Phys Rev E 80(1):016107. https://doi.org/10.1103/PhysRevE.80.016107
Global Development Program (2012) Financial Services for the Poor. Bill & Melinda Gates Foundation. http://www.gatesfoundation.org/What-We-Do/Global-Development/Financial-Services-for-the-Poor. Accessed 18 Oct 2016
GSMA Mobile Money: State of the Industry 2015 (2015a) Mobile Insurance, Savings, and Credit. Industry Report, GSMA. http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2016/03/SOTIR_2015.pdf
GSMA Mobile Money: State of the Industry 2015 (2015b) Mobile Money. Industry Report, GSMA. http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2016/03/SOTIR_2015.pdf
GSMA Mobile Money (2018) 2017 State of the Industry Report on Mobile Money. Industry Report, GSMA. https://www.gsma.com/mobilefordevelopment/programme/mobile-money/2017-state-industry-report-mobile-money. Accessed 13 Nov 2018
Guimerà R, Mossa S, Turtschi A, Amaral LN (2005) The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles. Proc Natl Acad Sci 102(22):7794–7799. https://doi.org/10.1073/pnas.0407994102

Heath MF, Vernon MC, Webb CR (2008) Construction of networks with intrinsic temporal structure from UK cattle movement data. BMC Vet Res 4(1):11. https://doi.org/10.1186/1746-6148-4-11
Holme P, Saramäki J (2012) Temporal networks. Phys Rep 519(3):97–125. https://doi.org/10.1016/j.physrep.2012.03.001
Holme P, Saramäki J (2019) Temporal network theory. Springer, Berlin
Houssein M, Lopes P, Fagnoni B, Ahmaidi S, Yonis SM, Leprêtre P-M (2016) Hydration: the New FIFA World Cup's challenge for referee decision making? J Athl Train 51(3):264–266. https://doi.org/10.4085/1062-6050-51.3.04
Hu J, Razdan A, Femiani JC, Cui M, Wonka P (2007) Road network extraction and intersection detection from aerial images by tracking road footprints. IEEE Trans Geosci Remote Sens 45(12):4144–4157. https://doi.org/10.1109/TGRS.2007.906107
Hughes M, Franks I (2005) Analysis of passing sequences, shots and goals in soccer. J Sports Sci 23(5):509–514. https://doi.org/10.1080/02640410410001716779
Intermedia: Financial Inclusion Insights Survey, Wave 3. Intermedia (2016). http://finclusion.org/data_fiinder/
International Finance Corporation, Mastercard Foundation (2018) Digital Access: The Future of Financial Inclusion in Africa. Technical report, Partnership for Financial Inclusion https://www.ifc.org/wps/wcm/connect/REGION__EXT_Content/IFC_External_Corporate_Site/Sub-Saharan+Africa/Priorities/Financial+Inclusion/za_ifc_partnership_financial_inclusion_publications. Accessed 07 Jan 2020
Iqbal MS, Choudhury CF, Wang P, González MC (2014) Development of origin-destination matrices using mobile phone call data. Transp Res Part C Emerg Technol 40:63–74
Jack W, Suri T (2011) Mobile money: The economics of M-PESA. Technical report, National Bureau of Economic Research. http://www.nber.org/papers/w16721. Accessed 03 Oct 2016
Jazayeri A, Yang CC (2020) Motif discovery algorithms in static and temporal networks: a survey. J Complex Netw. https://doi.org/10.1093/comnet/cnaa031
Joachims T (2002) Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data Mining. KDD '02. Association for Computing Machinery, New York, pp 133–142 https://doi.org/10.1145/775047.775067. Accessed 23 Feb 2021
Kendall J, Maurer B, Machoka P, Veniard C (2011) An emerging platform: from money transfer system to mobile money ecosystem. Innov Technol Gov Glob 6(4):49–64. https://doi.org/10.1162/INOV_a_00100
Kloumann IM, Ugander J, Kleinberg J (2017) Block models and personalized PageRank. Proc Natl Acad Sci 114(1):33–38. https://doi.org/10.1073/pnas.1611275114
Kovanen L, Karsai M, Kaski K, Kertész J, Saramäki J (2011) Temporal motifs in time-dependent networks. J Stat Mech Theory Exp 11:11005. https://doi.org/10.1088/1742-5468/2011/11/P11005
Kujala R, Weckström C, Darst R (2018) A collection of public transport network data sets for 25 cities. Zenodo
Kuper S (2011) A football revolution. Financial Times. Accessed 14 Sept 2020
Lambiotte R, Masuda N (2016) A guide to temporal networks, vol 4. World Scientific, Singapore
Lambiotte R, Rosvall M, Scholtes I (2018) Understanding complex systems: from networks to optimal higher-order models. arXiv:1806.05977 [cond-mat, physics:physics]. Accessed 24 July 2018
LaRock T, Nanumyan V, Scholtes I, Casiraghi G, Eliassi-Rad T, Schweitzer F (2020) HYPA: efficient detection of path anomalies in time series data on networks. In: Proceedings of the 2020 SIAM international conference on data mining. Proceedings, pp. 460–468. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611976236.52. Accessed 03 Sept 2020
Locicero G, Micale G, Pulvirenti A, Ferro A (2021) TemporalRI: a subgraph isomorphism algorithm for temporal networks. In: Benito RM, Cherifi C, Cherifi H, Moro E, Rocha LM, Sales-Pardo M (eds) Complex networks & their applications IX. Studies in Computational Intelligence, Springer, Cham, pp 675–687. https://doi.org/10.1007/978-3-030-65351-4_54
Masuda N, Porter MA, Lambiotte R (2017) Random walks and diffusion on networks. Phys Rep 716–717:1–58. https://doi.org/10.1016/j.physrep.2017.07.007
Mattsson C (2020) carolinamattsson/follow-the-money v0.2.0. https://github.com/carolinamattsson/follow-the-money
Mbiti I, Weil DN (2013) The home economics of E-money: velocity, cash management, and discount rates of M-Pesa users. Am Econ Rev 103(3):369–374. https://doi.org/10.1257/aer.103.3.369
Newman MEJ (2005) A measure of betweenness centrality based on random walks. Soc Netw 27(1):39–54. https://doi.org/10.1016/j.socnet.2004.11.009
OpenStreetMap contributors: Planet dump. OpenStreetMap (2017). https://planet.osm.org
Page L, Brin S, Motwani R, Winograd T (November 1999) The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-166, Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/
Pappalardo L, Cintia P, Rossi A, Massucco E, Ferragina P, Pedreschi D, Giannotti F (2019) A public data set of spatio-temporal match events in soccer competitions. Sci Data 6(1):236. https://doi.org/10.1038/s41597-019-0247-7
Paranjape A, West R, Zia L, Leskovec J (2016) Improving Website Hyperlink Structure Using Server Logs. In: Proceedings of the ninth ACM international conference on web search and data mining. WSDM '16. Association for Computing Machinery, New York, NY, USA, pp 615–624. https://doi.org/10.1145/2835776.2835832. Accessed 23 Feb 2021
Paranjape A, Benson AR, Leskovec J (2017) Motifs in temporal networks. In: Proceedings of the tenth ACM international conference on web search and data mining, pp 601–610
Peixoto TP, Rosvall M (2017) Modelling sequences and temporal networks with dynamic community structures. Nat Commun 8(1):582. https://doi.org/10.1038/s41467-017-00148-9
Petrovic LV, Scholtes I (2019) Counting causal paths in Big Times Series data on networks. arXiv:1905.11287 [physics]. Accessed 14 Sept 2020
Pfitzner R, Scholtes I, Garas A, Tessone CJ, Schweitzer F (2013) Betweenness preference: quantifying correlations in the topological dynamics of temporal networks. Phys Rev Lett 110(19):198701. https://doi.org/10.1103/PhysRevLett.110.198701
Reback J, McKinney W, jbrockmendel Bossche JVd, Augspurger T, Cloud P, gfyoung Sinhrks Klein A, Roeschke M, Hawkins S, Tratner J, She C, Ayd W, Petersen T, Garcia M, Schendel J, Hayden A, MomIsBestFriend Jancauskas V, Battiston P, Seabold S, chris-b1 h-vetinari Hoyer S, Overmeire W, alimcmaster1 Dong K, Whelan C, Mehyar M (2020) pandas-dev/pandas: Pandas v0.24.2. Zenodo. https://doi.org/10.5281/zenodo.3509134

Reep C, Benjamin B (1968) Skill and chance in association football. J R Stat Soc Ser A (Gen) Ser A (Gen) 131(4):581. https://doi.org/10.2307/2343726
Reep C, Pollard R, Benjamin B (1971) Skill and chance in ball games. J R Stat Soc Ser A (Gen) 134(4):623–629. https://doi.org/10.2307/2343657
Rivlin G (2015) Why New Orleans's Black Residents are still underwater after Katrina. The New York Times. Chap. Magazine. Accessed 14 Sept 2020
Rocha LEC, Masuda N (2014) Random walk centrality for temporal networks. N J Phys 16(6):063023. https://doi.org/10.1088/1367-2630/16/6/063023
Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci 105(4):1118–1123. https://doi.org/10.1073/pnas.0706851105
Rosvall M, Esquivel AV, Lancichinetti A, West JD, Lambiotte R (2014) Memory in network flows and its effects on spreading dynamics and community detection. Nat Commun 5:5630. https://doi.org/10.1038/ncomms5630
Saramäki J, Holme P (2015) Exploring temporal networks with greedy walks. Eur Phys J B 88(12):334. https://doi.org/10.1140/epjb/e2015-60660-9
Sarmento H, Marcelino R, Anguera MT, Campaniço J, Matos N, Leitão JC (2014) Match analysis in football: a systematic review. J Sports Sci 32(20):1831–1843. https://doi.org/10.1080/02640414.2014.898852
Schoenfeld B (2019) How Data (and Some Breathtaking Soccer) Brought Liverpool to the Cusp of Glory. The New York Times. Chap. Magazine. Accessed 14 Sept 2020
Scholtes I (2017) When is a network a network? Multi-order graphical model selection in pathways and temporal networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. KDD '17. Association for Computing Machinery, New York, pp 1037–1046. https://doi.org/10.1145/3097983.3098145
Scholtes I (2020) IngoScholtes/pathpy. https://github.com/IngoScholtes/pathpy Accessed 14 Sept 2020
Scholtes I, Wider N, Garas A (2016) Higher-order aggregate networks in the analysis of temporal networks: path structures and centralities. Eur Phys J B 89(3):61. https://doi.org/10.1140/epjb/e2016-60663-0
Schwarze AC, Porter MA (2020). Motifs for processes on networks. arXiv:2007.07447 [physics]. Accessed 16 July 2020
Stuart G, Cohen M (2011) Cash in, cash out Kenya: the role of M-PESA in the lives of low-income people. The Financial Services Assesment project. Microfinance Opportunities, http://wwww.gsmworld.com/mobilefordevelopment/wp-content/uploads/2012/06/cash_in_cash_out_kenya.pdf. Accessed 30 July 2015
Suri T (2017) Mobile money. Annu Rev Econ 9(1):497–520. https://doi.org/10.1146/annurev-economics-063016-103638
Sánchez-Martínez GE (2017) Inference of public transportation trip destinations by using fare transaction and vehicle location data: dynamic programming approach. Transp Res Rec 2652(1):1–7. https://doi.org/10.3141/2652-01
Taylor D, Myers S, Clauset A, Porter M, Mucha P (2017) Eigenvector-based centrality measures for temporal networks. Multiscale Model Simul 15(1):537–574. https://doi.org/10.1137/16M1066142
TeleGeography (2020) Submarine Cable Map. TeleGeography, Oceans https://www.submarinecablemap.com/. Accessed 14 Sept 2020
Thelwall M (2002) Methodologies for crawler based Web surveys. Internet Res 12(2):124–138. https://doi.org/10.1108/10662240210422503
Toole JL, Colak S, Sturt B, Alexander LP, Evsukoff A, González MC (2015) The path most traveled: travel demand estimation using big data resources. Transp Res Part C Emerg Technol 58:162–177
Xu J, Wickramarathne TL, Chawla NV (2016) Representing higher-order dependencies in networks. Sci Adv 2(5):1600028. https://doi.org/10.1126/sciadv.1600028
Zhan FB, Noon CE (1998) Shortest path algorithms: an evaluation using real road networks. Transp Sci 32(1):65–73. https://doi.org/10.1287/trsc.32.1.65

Publisher's Note