IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

Generalized Preference Learning for Trust Network Inference

DOMENICO MANDAGLIO[1], and ANDREA TAGARELLI[1]
[1]Dept. Computer Engineering, Modeling, Electronics, and Systems Engineering (DIMES), University of Calabria, 87036 Rende (CS), Italy
(e-mail: d.mandaglio@dimes.unical.it,andrea.tagarelli@unical.it)

Corresponding author: A. Tagarelli (e-mail: andrea.tagarelli@unical.it).

ABSTRACT Trust inference is essential in a plethora of data mining and machine learning applications. Unfortunately, conventional approaches to trust inference assume trust networks are available, while in practice they must be derived from social network features. This is however a difficult task which has to cope with challenges relating to scarcity, redundancy and noise in the available user interactions and other social network features. In this work, we introduce the new problem of Trust Network Inference (TNI), that is, inferring a trust network from a sequence of timestamped interaction networks. To solve the TNI problem, we propose a principled approach based on a preference learning paradigm, under a preference-based racing formulation. The proposed approach is suitable for addressing the above challenges, moreover it is versatile (i.e., independent from the social network platform) and flexible w.r.t. the use of topological and content-based information. Extensive experimental evaluation focusing on two distinct ground-truth scenarios, has provided evidence of the meaningfulness and uniqueness of our TNI approach, which can be regarded as key-enabling for any application that requires to handle a trust network associated with a social environment.

INDEX TERMS trust computing, dynamic social networks, network inference, preference-based racing algorithm

I. INTRODUCTION

THE term trust-based social network, or simply *trust network*, commonly refers to a graph of entities (i.e., individuals) that are linked through asymmetric relationships that correspond to subjective trust statements. Given a trust network, *trust inference* is the task of predicting a new relation between two nodes, so that the locally inferred trust score can be regarded as a personalized opinion of one user (trustor) with respect to another user (trustee). Trust inference is an essential task in many data analysis and machine learning applications, from social influence propagation and opinion spreading to recommender systems and privacy preserving, whose impact extends also to peer-to-peer networks and mobile ad-hoc networks [20].

Challenges in trust inference. The conventional approach to trust inference is to compute the trust between any two non-adjacent nodes in a trust network by considering the different paths from one node to the other, as well as strategies for trust propagation and for aggregating the propagated trust values through different paths [20], [22]. Unfortunately, all existing trust-inference approaches rely on the assumption that a trust network has been already formed, while in reality *trust networks are not naturally available*. Rather, trust relations must first be determined from the available information in a social environment, e.g., the history of users' activities and their interactions.

Computing trust relations is however a particularly difficult task, because of a number of challenges that already arise at data source level (i.e., not considering the inevitable bias of the particular algorithmic solution to the problem). In fact, the amount of information representing the observed interactions and activities of users in a social network, could be *limited* in size as well as in quality. More specifically, a social network may contain a significant amount of *redundant* or irrelevant relations as well as *noise* in the information that express the strength of interaction between any two users.

Contributions. In this work, we face the above challenges by addressing a new problem we named *Trust Network Inference* (TNI). Given a sequence of timestamped interaction networks as input, the goal of TNI is to infer from this sequence a directed weighted network, whose nodes are the users in the temporal networks and links denote trust relationships with associated trust scores.

It should be emphasized that in TNI there is no dependency on existing trust relations to make predictions on trustworthiness scores or on new trust relations. Therefore, TNI emerges as a divergence from the conventional trust inference and trust link prediction problems (e.g., [7], [12], [15], [16]). Also, TNI differs from trust ranking methods (e.g., [8], [10], [17]), since in TNI the building of trust relations is extended to all nodes in a network, not only to the most trusted or reputable ones. Furthermore, our TNI problem is different from the one treated in [6], which considers trustworthiness and untrustworthiness inference through clustering all entities into two groups (i.e., good and misbehaved), under various representative attack models.

We propose to solve the TNI problem based on a generalized *preference learning* paradigm. We believe that preference learning provides key advantages in addressing all the aforementioned issues, i.e., limitedness, redundancy and noisy of the information about the users' interactions from which a trust network is to be inferred. More specifically, under a preference-based top-k selection problem, *our proposed approach aims to find a ranking of the preferential pairings that each target entity would choose to form its trust relationships.* To this purpose, we resort to an adaptive sampling strategy, and instatiate it according to three canonical ranking models that correspond to different levels of ranking pairwise preferences. One further key feature of our approach is *domain-independency*, as it does not rely on platform-specific types of user interactions. Nonetheless, our approach is designed to exploit both topological information and, when available, content information relating to the user interaction dynamics.

Evaluating inferred trust relations and associated scores is another critical aspect in research contexts related to trust computing. In this work, we also cope with such a challenge and devise two scenarios based on distinct notions of *ground-truth*: the one referring to the availability of *trust classes* (i.e., cohesive groups of mutually trusted users), and the other corresponding to the availability of a *reference trust network*. Our extensive, ground-truth-driven experimental evaluation has shown the meaningfulness of our proposed approach in both evaluation scenarios, on several dynamic interaction networks and against competing methods.

Plan of the paper. The remainder of this paper is organized as follows. Section II briefly discusses related work on trust inference — note that, given the relative novelty of the TNI problem under consideration, we shall provide a deliberately concise summary of major existing notions and approaches to trust inference, without any ambition to survey methods for trust computing. Section III introduces the problem of Trust Network Inference, and Section IV describes our proposed approach. Sections V and VI present methodology, data and results of our experimental evaluation. Finally, Section VII concludes the paper.

II. RELATED WORK ON TRUST INFERENCE

Trust inference has attracted much attention in data mining and related fields, and a variety of studies have been proposed in literature [20]. One way of interpreting the problem of trust inference is to model it in terms of either *edge feature* or *node feature*, a.k.a. "local" and "global" trust computing. In the first case, a trust relation is to be created for any two non-adjacent nodes in a network, through a mechanism of *inference*, resp. *prediction*, if the network is modeled on existing trust relations (i.e., it is a trust network) (e.g., [7], [12], [15], [16]), resp. on social network features (e.g., [2], [21]). Conversely, trust inference at node-level corresponds to computing a trust score for each node in a network, and hence it is more appropriately regarded as a trust-oriented global *ranking* of the users, which can be useful to build trust communities [17], or in general to discriminate between objectively trust and distrust entities in a network (e.g., [8], [10], [19]).

Our work refers to the local-trust computing perspective. However, as already mentioned in the Introduction, we address the trust network inference problem, for which the trust network is the output, rather than the input as in conventional trust inference approaches. Note also that our work is substantially different from previous attempts to TNI-related problems, such as [12]: in that work, a user-domain-based trusted acquaintance chain discovery algorithm is developed to make the computation of short trusted paths more efficient; however, unlike our approach, the method in [12] strongly depends on the definition of domains/categories for the content in the input social network. Also, our inference problem is different from the one considered in [6], which assumes that all entities are clustered into two groups (i.e., good and misbehaved entities), and a belief propagation method is developed to estimate that one entity belongs to different groups, simultaneously inferring its trustworthiness and untrustworthiness values, according to different attack models in interactional networks.

III. PROBLEM STATEMENT

We are given a set $\mathcal{V}$ of *entities* in a social environment (i.e., users), and a *temporal network* $\mathcal{G}$ as a series of graphs over discrete time steps $(G_1, G_2, \ldots, G_T)$, with time horizon T, where $G_t = \langle V_t, E_t, w_t \rangle$, with $1 \leq t \leq T$, is the graph at time t, with set of nodes V_t and set of directed edges E_t. Each node in V_t corresponds to a specific instance from the subset $\mathcal{V}_t$ of entities that occur at time t. Note that entities might occasionally appear and disappear in different time steps. Each edge $e = (v_i, v_j) \in E_t$ corresponds to an observed *interaction* between nodes v_i, v_j, which can be of different type depending on the specific functionalities and information available from the online social environment under consideration (e.g., mentions, answers/replies, reposts, etc). The snapshot graphs G_t are also associated with an edge weighting function $w_t(\cdot)$ to quantify the strength of each interaction; by default, the weight of an edge is set to 1.

TABLE 1: Main notations and their descriptions

symbol	*description*
$\mathcal{G}$; $G_t = \langle V_t, E_t, w_t \rangle$	series of graphs; graph at time t
$\mathcal{V}$; $\mathcal{V}_t$	set of entities or actors in $\mathcal{G}$; in G_t
$\mathcal{T} = \langle \mathcal{V}, \mathcal{E}, \omega \rangle$	trust network (to be inferred from $\mathcal{G}$)
$\mathcal{C}$; C_v	trust-context model; trust-context (induced subgraph) for entity v
o; $\mathcal{O}$	option; set of options (alternatives)
N; k	total (resp. selected) number of options
$\mathcal{R}$	ranking model
$\prec^{\mathcal{R}}$	strict preference order relation over a pair of options, according to $\mathcal{R}$
CO	Copeland's ranking model
SE	sum-of-expectations ranking model
RW	random-walk ranking model
$1-\delta$	predefined confidence (for the top-k selection problem)
$Y_{i,j}$	random variable associated to comparison of o_i with o_j
$y_{i,j}^{(t)}$	t-th observed outcome of $Y_{i,j}$
$\mathbf{Y}$	preference relation matrix
n_{max}	number of samplings for each pairwise preference probability distribution $Y_{i,j}$
$sim_S^{(t)}$; $sim_C^{(t)}$	structural (resp. content) affinity function for node comparison in G_t
α	smoothing parameter to weight $sim_C^{(t)}$ w.r.t. $sim_S^{(t)}$

We consider the Trust Network Inference (TNI) problem, that is, generating a new network from interactional dynamics observed through $\mathcal{G}$, whose nodes correspond to the entities $\mathcal{V}$ in $\mathcal{G}$ and links are inferred to denote a trust/distrust relationship between any two entities that satisfy certain relational constraints. Such constraints are meant to be specified w.r.t. a predetermined scheme of selection of **trust-context**, denoted as $\mathcal{C}$.

The trust-context is a model for inducing a subgraph of $\mathcal{G}$ from each entity v, denoted as C_v, whose structural expansion intuitively corresponds to the extent of trust that v can exert towards other entities. Note that the induced trust-context subgraphs of any two entities are not to be necessarily disjoint.

We will refer to the **trust network** inferred from $\mathcal{G}$, w.r.t. a trust-context scheme $\mathcal{C}$, as a weighted directed graph $\mathcal{T} = \langle \mathcal{V}, \mathcal{E}, \omega \rangle$, with set of trust links $\mathcal{E} = \bigcup_{v \in \mathcal{V}} \mathcal{E}_v$, where $\mathcal{E}_v$ is a set of edges between entities in the node-set of the induced subgraph C_v for v in accord with $\mathcal{C}$, and $\omega : \mathcal{E} \rightarrow [0,1]$ is a weighting function that specifies the trust level of each link, where 0 means total lack of trust (i.e., distrust) and 1 means fully trust from a source to a target node. We intuitively formulate the TNI problem as follows:

*Problem 1 (**Trust Network Inference (TNI)**):* Given a temporal network $\mathcal{G}$ built over interactions observed in a time period T between entities in a set $\mathcal{V}$, and given a trust-context scheme $\mathcal{C}$, infer a trust network $\mathcal{T}$ for all entities in $\mathcal{V}$ by exploiting the topological information available from each snapshot of $\mathcal{G}$ (along with, optionally, content-based information of the interactions) according to the trust-context scheme $\mathcal{C}$.

IV. OUR PROPOSED METHOD FOR TRUST NETWORK INFERENCE

We propose to solve the TNI problem through a generalization of the *preference-based top-k selection problem* over each entity in the input temporal network. Next, we provide background on that, then we discuss details on our proposal. Table 1 summarizes main notations used throughout the rest of the paper.

A. BACKGROUND ON PREFERENCE-BASED TOP-K SELECTION

Consider a finite set of decision *alternatives*, or *options*, $\mathcal{O} = \{o_1, \ldots, o_N\}$, for which the following assumptions hold: (i) the options in this set are pairwise comparable, (ii) there exists a finite number of samples, from an unknown pairwise-preference distribution, that provide information about whether or not an option might be preferred to another one, and (iii) the samples could be "noisy" (i.e., they could significantly vary w.r.t. the unknown distribution model).

The *preference-based top-k selection* problem is to choose the set of k options ($k < N$) that maximize the *preference* over all alternatives, which is formally equivalent to the following optimization problem:

$$\underset{S \subset \mathcal{O}, |S|=k}{\operatorname{argmax}} \sum_{o_i \in S} \sum_{o_j \in \mathcal{O} \wedge j \neq i} \mathbb{I}\{o_j \prec^{\mathcal{R}} o_i\}, \tag{1}$$

where $\prec^{\mathcal{R}}$ is a strict preference order relation according to a predefined ranking model $\mathcal{R}$, such that $o_j \prec^{\mathcal{R}} o_i$ means the option o_i is preferred to o_j, and $\mathbb{I}\{\cdot\}$ is the indicator function which is equal to 1 if the argument is true, 0 otherwise. Note also that, given that the outcomes of the pairwise comparisons could be noisy and the available number of samplings are limited, the optimality of the solution to Eq. 1 should be guaranteed with probability at least $1 - \delta$, for any predefined probability δ; typically, $\delta = 0.1$.

To quantify the pairwise preferences, the outcome of a comparison between o_i and o_j is modeled as a random variable $Y_{i,j}$, which assumes value 0 (resp. 1) if $o_i \prec o_j$ (resp. $o_i \succ o_j$), and a "neutral" 1/2 otherwise. Given a pair o_i, o_j and a set of $n_{i,j}$ realizations of their comparison $\{y_{i,j}^{(1)}, \ldots, y_{i,j}^{(n_{i,j})}\}$ of $Y_{i,j}$, assumed to be independent, the expected value $y_{i,j} := \mathbb{E}[Y_{i,j}]$ can be estimated as:

$$\bar{y}_{i,j} = \frac{1}{n_{i,j}} \sum_{l=1}^{n_{i,j}} y_{i,j}^{(l)}. \tag{2}$$

1) Ranking models

A ranking model $\mathcal{R}$ produces a complete order of the options in $\mathcal{O}$ upon the preference relation matrix $\mathbf{Y} = [y_{i,j}]_{N \times N} \in [0,1]^{N \times N}$. Following [4], we consider three different models: (i) *Copeland's ranking* (CO), (ii) weighted voting, or *sum of expectations* (SE), and (iii) *random walk* (RW) ranking.

Copeland's ranking determines that option o_i is preferred to option o_j ($o_j \prec^{CO} o_i$) if and only if $b_j < b_i$, where $b_i = |\{o_h \in \mathcal{O} \mid y_{i,h} > \frac{1}{2}\}|$, i.e., whenever o_i beats more

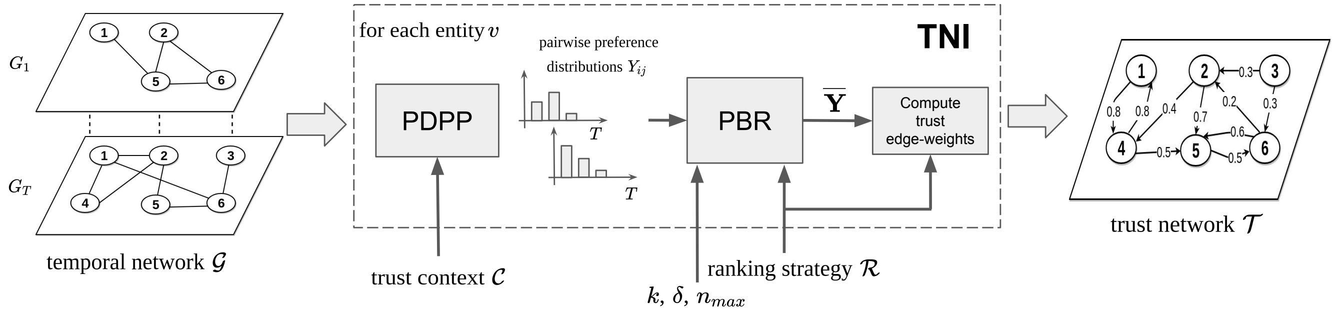

G_1
G_T
1
2
3
4
5
6
temporal network $\mathcal{G}$
for each entity v
pairwise preference distributions Y_{ij}
T
T
TNI
PDPP
PBR
$\overline{\mathbf{Y}}$
Compute trust edge-weights
trust context $\mathcal{C}$
ranking strategy $\mathcal{R}$
k, δ, n_{max}
0.8
0.8
0.4
0.7
0.2
0.3
0.3
0.6
0.5
0.5
trust network $\mathcal{T}$

FIGURE 1: Overview of our proposed framework for trust network inference

Algorithm 1 TRUST NETWORK INFERENCE($\mathcal{G} = (G_1, ., G_T), \mathcal{C}, \mathcal{R}, k, n_{max}, \delta$)

```
1: $\Upsilon \leftarrow \emptyset$
2: **for all** $v \in \mathcal{V}$ **do**
3:     $\mathcal{O}_v \leftarrow computeTrustContextOptions(\mathcal{G}, v, \mathcal{C})$
4:     $\mathbf{Y} \leftarrow$ PDPP($\mathcal{G}$, $v$, $\mathcal{O}_v$)   {*Probability distributions of pairwise preferences for* $v$}
5:     $\bar{\mathbf{Y}} \leftarrow$ PBR($\mathbf{Y}$, $v$, $\mathcal{O}_v$, $k$, $n_{max}$, $\delta$, $\mathcal{R}$)   {*Preference-based racing to compute the ranking scores*}
6:     $\Upsilon \leftarrow \Upsilon \cup \{\bar{\mathbf{Y}}\}$
7: **end for**
8: $\langle \mathcal{E}, \omega \rangle \leftarrow computeTrustEdges(\Upsilon, \mathcal{R})$
9: **return** $\mathcal{T} = \langle \mathcal{V}, \mathcal{E}, \omega \rangle$
```

options that o_j does. According to sum of expectations ranking, $o_j \prec^{SE} o_i$ holds if and only if $\sum_{h \neq j} y_{j,h} < \sum_{h \neq i} y_{i,h}$. Random walk ranking first requires a left-stochastic version $\mathbf{S} = [s_{ij}]_{N \times N}$ of the matrix $\mathbf{Y}$, such that $s_{i,j} = \frac{y_{i,j}}{\sum_{l=1}^{N} y_{l,j}}$. Then, the ranking of options is determined as the stationary probability distribution $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$ of the Markov chain underlying $\mathbf{S}$. Finally, the options are ranked according to the computed probabilities, i.e., $o_j \prec^{RW} o_i$ iff $\pi_j < \pi_i$.

B. THE TNI ALGORITHM

Given a temporal network $\mathcal{G}$, we solve the TNI problem as a generalized preference-based top-k selection problem, for each entity in $\mathcal{G}$, under constraints given by a predefined trust-context scheme $\mathcal{C}$. The model $\mathcal{C}$ is used to determine the options $\mathcal{O}$ for pairing each target entity with its "trustworthy" entities.

Our idea is to generate the edges and associated scores of the trust network to be inferred on the basis of the solution of a *preference-based racing* (PBR) algorithm applied to each target entity. PBR is a particular approach to the top-k selection problem based on an adaptive sampling strategy.

A schematic depiction of the proposed framework for trust network inference is presented in Figure 1, whereas Algorithm 1 shows the pseudo-code of our TNI method. The algorithm works as follows: for each entity v in the temporal network $\mathcal{G}$, it starts with the identification of the entity-options for v according to a predefined trust-context model $\mathcal{C}$ (Line 3). Then, the probability distributions of pairwise preferences (PDPP) $\mathbf{Y}$ are computed for v based on its interaction activities observed in $\mathcal{G}$ (Line 4). Using a preference-based racing algorithmic scheme, a ranking of trust relations is computed for v according to a selected ranking model $\mathcal{R}$ (Line 5). Finally, the solutions provided by the racing procedure for all entities are used to determine both the edges and the trust scores (Line 8) to output the trust network $\mathcal{T}$. We now elaborate on each of the main steps in Algorithm 1.

1) Computing the trust-context of entities

The trust-context model $\mathcal{C}$ corresponds to the search space for the entity-options to identify as the trustworthy ones for any given target entity. One intuitive way of defining $\mathcal{C}$ is to instantiate it as the ego-network of the target entity. This notion is also supported by previous studies on trust inference which have provided evidence on that shorter paths from the trustor are more accurate to predict trust [7], and that the dilution of trust through the propagation process tends to weaken the predicted trust [13].

In the following, we will refer to the above definition of trust-context model, restricted to the out-neighborhood of any target entity v, i.e., all entities occurring as out-neighbors of v in at least one snapshot graph in $\mathcal{G}$. Clearly, the search space for the TNI problem can also be defined according to other topological structures, such as expanded ego-networks or community structures. This is *left as a further direction of research.*

2) Building the preference distributions

As previously discussed in Sect. IV-A, the true pairwise preference distributions are assumed to be unknown, however their realizations (i.e., outcomes of random variables $Y_{i,j}$) can be estimated as the observations of interaction at each snapshot G_t.

Given a target entity v in $\mathcal{G}$, every pair of entities occurring within its trust-context C_v are regarded as options for v, which can be compared at most T times. For entities v_i and v_j, we denote the outcomes of these comparisons (w.r.t. v) as $Y_{i,j} = y_{i,j}^{(1)}, \ldots, y_{i,j}^{(T)}$. To build each of the pairwise preference distributions for any entity v, we consider, for each snapshot $G_t = \langle V_t, E_t, w_t \rangle$, the set of v's outgoing nodes, denoted as $N_t(v)$, and evaluate the following outcomes for the variables $Y_{i,j}$ associated to v:

Outcome 1: $v_i \notin N_t(v) \wedge v_j \notin N_t(v)$. In this case, the two entities v_i and v_j are not comparable at time t, although, being both in C_v, they will be in some other snapshot. But at time t, v_i and v_j will not be considered to determine $y_{i,j}^{(t)}$.

Outcome 2: $v_i \in N_t(v) \vee v_j \in N_t(v)$. Let us consider a node-similarity function $sim^{(t)} : V_t \times V_t \mapsto [0,1]$ and define it as a linear combination of two functions:

- a ***structural affinity*** function $sim_S^{(t)}$: this can efficiently be computed by resorting to standard *neighborhood-based overlap* measures; for instance, Jaccard similarity, i.e., $sim_S^{(t)}(v, v_i) = \frac{|N_t(v) \cap N_t(v_i)|}{|N_t(v) \cup N_t(v_i)|}$), or Adamic-Adar index, i.e., $sim_S^{(t)}(v, v_i) = \sum_{u \in N_t(v) \cap N_t(v_i)} \log(|N_t(u)|)^{-1}$. One alternative is to consider a vector similarity function to apply to the multidimensional representations of any two nodes, which would be obtained through *node-embedding* techniques in graphs, such as, e.g., *node2vec* (see [5] for a comprehensive survey).
- a ***content affinity*** function $sim_C^{(t)}$: the edge-weighting function w_t expresses the strength of content-based interaction for any two nodes in G_t, therefore $sim_C^{(t)}(v, v_i) := w_t(v, v_i)$. To this aim, we might consider the opportunity of computing a *sentiment score* associated with the available text content (cf. Sect. V). Note that, if no content-based information is associated with the interaction between v and v_i at time t, $w_t(v, v_i)$ is assumed to be 1.

The two above functions are hence combined as follows:

$$sim^{(t)}(v, v_i) = \alpha \cdot sim_C^{(t)}(v, v_i) + (1-\alpha) \cdot sim_S^{(t)}(v, v_i), \quad (3)$$

for any pair (v, v_i), with $\alpha \in [0,1]$ (by default set to 0.5).

Finally, we compute the v's preference of choosing v_i over node v_j at time t as the probability value given by the following logistic function:

$$y_{i,j}^{(t)} := \Pr(v_i \succ v_j) = \frac{1}{1 + e^{-f(i,j) \cdot (sim^{(t)}(v, v_i) - sim^{(t)}(v, v_j))}}, \quad (4)$$

where $f(i,j)$ corresponds to the steepness of the logistic, we define as $f(i,j) = \lambda \cdot (sim^{(t)}(v, v_i) + sim^{(t)}(v, v_j))$, where λ is a scaling factor. Our motivation behind this analytical choice is twofold. First, since the similarity values range in $[0,1]$, and hence their differences range in $[-1,1]$, the full domain of values of the logistic function would not be used if the steepness value was 1. Therefore, we introduce a scaling factor to better distribute the $y_{i,j}^{(t)}$ values within $(0,1)$; for this purpose, we set λ to 10, which ensures the spanning through the interval $(0,1)$. Moreover, our definition of the steepness function and λ setting is such that the sum of similarities is considered to weight more pairwise comparisons between more similar entities than dissimilar ones. Note also that Eq. 4 is symmetric, i.e., $\Pr(v_i \succ v_j) = 1 - \Pr(v_j \succ v_i)$.

Algorithm 2 PDPP($\mathcal{G} = (G_1, \ldots, G_T), v, \mathcal{O}$)

```
Initialize $\mathbf{Y} = [Y_{i,j}]_{N \times N}$ with empty lists
for $t = 1$ to $T$ do
  for $(v_i, v_j) \in \mathcal{O} \times \mathcal{O}$, $v_i \neq v_j \neq v, v_i \neq v$ do
    if $v_i \in N_t(v) \vee v_j \in N_t(v)$ then
      Compute $sim^{(t)}(v, v_i)$ and $sim^{(t)}(v, v_j)$
      $y_{i,j}^{(t)} \leftarrow \Pr(v_i \succ v_j)$          {Using Eq. 4}
      $add(Y_{i,j}, y_{i,j}^{(t)})$
return $\mathbf{Y}$
```

Algorithm 3 PBR($\mathbf{Y}, v, \mathcal{O}, k, n_{max}, \delta, \mathcal{R}$)

```
$S = D \leftarrow \emptyset$          {Set of selected (S) and discarded (D) options}
Initialize with zeros: $\mathbf{B} = [c_{i,j}]_{N \times N}$, $\mathbf{B_u} = [u_{i,j}]_{N \times N}$, $\mathbf{B}_\ell = [l_{i,j}]_{N \times N}$          {Confidence bound matrices}
$n_{i,j} \leftarrow 0, \ \forall o_i, o_j \in \mathcal{O}$          {Sample counts}
$A \leftarrow \{(o_i, o_j) | i \neq j, 1 \leq i, j \leq |\mathcal{O}|\}$          {Set of active option pairs}
while $(n_{i,j} \leq n_{max}, \forall i \forall j) \wedge |A| > 0$ do
  for all $(o_i, o_j) \in A$ do
    $n_{i,j} \leftarrow n_{i,j} + 1$
    $y_{i,j}^{(n_{i,j})} \sim Y_{i,j}$          {Sample from the pairwise preference probability distribution}
  end for
  Update $\bar{\mathbf{Y}} = [\bar{y}_{i,j}]_{N \times N}$ with the new samples          {Using Eq. (2)}
  for $i, j = 1$ to $N$ do
    $c_{i,j} \leftarrow \sqrt{\frac{1}{2n_{i,j}} \log \frac{2N^2 n_{max}}{\delta}}$          {Update Hoeffding confidence bounds $\mathbf{B_u}, \mathbf{B}_\ell, \mathbf{B}$}
    $l_{i,j} \leftarrow \bar{y}_{i,j} - c_{i,j}$,   $u_{i,j} \leftarrow \bar{y}_{i,j} + c_{i,j}$
  end for
  $(A, S, D) \leftarrow$ SamplingStrategy($\mathcal{R}, A, \bar{\mathbf{Y}}, N, k, \mathbf{B_u}, \mathbf{B}_\ell, \mathbf{B}, D$)          {Algorithm 4}
end while
return $S, \bar{\mathbf{Y}}$
```

Algorithm 4 SamplingStrategy($\mathcal{R}, A, \bar{\mathbf{Y}}, N, k, \mathbf{B_u}, \mathbf{B}_\ell, \mathbf{B}, D$)

```
$S \leftarrow optionsToSelect(A, \mathbf{B}_\ell, \mathbf{B_u}, N, k, \mathcal{R})$
$D \leftarrow D \cup optionsToDiscard(A, \mathbf{B}_\ell, \mathbf{B_u}, N, k, \mathcal{R})$
for $(o_i, o_j) \in A$ do
  if $!isStillToUpdate((o_i, o_j), S, D, \mathbf{B}_\ell, \mathbf{B_u}, \mathbf{B}, \bar{\mathbf{Y}}, \mathcal{R})$ then
    $A = A \setminus \{(o_i, o_j)\}$
$S \leftarrow$ top-$k$ options according to $\mathcal{R}$
return $(A, S, D)$
```

Upon the above definitions, we build the pairwise preference distributions for a target node v as shown in Algorithm 2. Sampling from these distributions will correspond to randomly extracting an element from the lists $Y_{i,j}$. It should be emphasized that this sampling is important to ensure robustness of the whole approach w.r.t. noisy comparisons; we shall discuss this point later in Sect. IV-B3 The output of Algorithm 2 then becomes the input for the preference-based racing algorithm (Algorithm 3). It should be noted that our approach to the computation of pairwise preference distributions diverges from the one adopted in [4]: here, while we still do not evaluate single options quantitatively (as in value-based racing), we let any variable $Y_{i,j}$ assume values within the range $(0,1)$, to express a *degree* of preference of o_i over o_j, rather than a 0/1 (or ternary) decision (cf. Sect. IV-A).

3) Preference-based Racing

Following [4], the preference-based racing (PBR) procedure, shown in Algorithm 3, is responsible for identifying, among the entities in the context $\mathcal{O}$ of an input target entity, the top-k trustworthy ones (or equivalently the top-k trust edges) according to a predefined ranking model $\mathcal{R}$. Besides $k, \mathcal{R}$, and the probability guarantee (δ, cf. Sect. IV-A), the algorithm requires an additional parameter, n_{max}, to control the number of samplings for each pairwise preference probability distribution (i.e., $Y_{i,j}$, with $o_i, o_j \in \mathcal{O}$).

As mentioned before, the sampling step from each of the pairwise preference probability distributions lends the algorithm more robust to the presence of "noise", i.e., irrelevant node-relations such as sporadical links and/or wrongly observed links that may occur across the input temporal network.

The algorithm also maintains a set of active pairs of options (A), i.e., options whose pairwise preference distributions need to be sampled more in order to decide which one is better, with enough high degree of *confidence*. Racing methods employ confidence intervals, typically computed through the Hoeffding bound, derived from the concentration property of the mean estimate [4]. To this purpose, Algorithm 3 maintains the estimates $y_{i,j}$ with their confidence intervals $[\ell_{i,j}, u_{i,j}]$ and iteratively samples from the pairwise preference distribution until there is enough confidence about the top-k nodes or the maximum number of samplings is reached (Line 5). According to the updates values of confidence bounds, the set of current selected options S (i.e., top-k ones) and discarded options D (not top-k) are updated. This is handled by procedure SamplingStrategy (Line 15), which is sketched in Algorithm 4.

According to [4], Algorithm 4 initially checks if some options can be included among the top-k or discarded ones with high enough probability (Lines 1 and 2). This step is performed differently according to the ranking model $\mathcal{R}$ (cf. Sect. IV-A), whereby the confidence intervals are used to decide with high probability that an option is better or worse than another. Next we provide details about the different sampling strategies.

- CO-based strategy: the aforementioned step is performed by counting, for each option o_i, the set of better options $w_i = |\{o_j \mid l_{i,j} > 1/2,\ i \neq j\}|$ and worse options $z_i = |\{o_j \mid u_{i,j} < 1/2,\ i \neq j\}|$. Then, an option o_i is among the top-k options with high probability if $|\{o_j \mid |\mathcal{O}| - z_i < w_j\}| > |\mathcal{O}| - k$ while it is to be discarded if $|\{o_j \mid |\mathcal{O}| - w_i < z_j\}| > k$.
- SE-based strategy: in this case, first the ranking score definition is applied to lower/upper bounds, i.e., for each option o_i, the averages $l_i = \frac{1}{|\mathcal{O}|-1} \sum_{j \neq i} l_{i,j}$ and $u_i = \frac{1}{|\mathcal{O}|-1} \sum_{j \neq i} u_{i,j}$ are computed. Then, similarly to the CO case, an option o_i is included among the top-k options with high probability if $|\{o_j \mid u_j < l_i\}| > |\mathcal{O}| - k$ and discarded if $|\{o_j \mid u_i < l_j\}| > k$.
- RW-based strategy: when RW is used as ranking model, selecting and discarding of option is based on the exploitation of properties of the stationary distribution of transition matrices. An upper bound on the difference between the estimated stationary distribution and the unknown true one is used in order to select the next pairwise preference distributions to sample from: the pairs selected are those whose sampling enable as much decrease as possible of this upper bound. Moreover, the same bound is exploited in order to determine the stopping criterion of the PBR procedure. Formal details are reported in *Appendix*.

For each active pair of options (o_i, o_j), a condition is checked (Line 4) to decide whether it is not necessary anymore to sample from the pairwise preference distribution of (o_i, o_j) — this holds either because with high probability o_i is better (resp. worse) than o_j or because one of the two options need to be selected (resp. discarded).

The role of k in the PBR procedure. It is worth noting that Algorithm 3 outputs the top-k trustworthy nodes together with the *whole* preference estimates $\bar{\mathbf{Y}}$, which are fed into the $computeTrustEdges$ function to finally compute the trust edge-weights in the trust network. This is done since, besides identifying the top-k trust edges (i.e., trust relationships), our goal is also to infer distrust links, which can be extracted through $\bar{\mathbf{Y}}$. In other terms, k takes the role of model parameter in the PBR procedure and only within the scope of this procedure; by contrast, in order to infer the trust network, all preference estimates may be taken into account so that each node may have more than k trust/distrust outgoing links.

4) Computing the trust edge-weights

For any given target entity v, the edge-weights in the trust network being generated are differently computed depending on the chosen ranking model and sampling strategy. For each v_i in the $\bar{\mathbf{Y}}$ matrix associated to v, using the Copeland's ranking, we set $\omega(v, v_i) = \frac{|\bar{y}_{i,j} : \bar{y}_{i,j} > \frac{1}{2}, i \neq j|}{|\mathcal{O}_v|}$. Note that the normalization is required since we want trust scores ranging in $[0, 1]$. For the other two sampling strategies, no normalization is required since the ranking scores are already in $[0, 1]$. In fact, for the SE-based strategy, we set $\omega(v, v_i) = \bar{y}_{i,j}$, whereas for the RW-based strategy, we set the edge weights to the values stored in the stationary distribution π, i.e., $\omega(v, v_i) = \pi_{v_i}$.

C. COMPUTATIONAL COMPLEXITY ASPECTS

The time complexity of TNI is determined by the cost of its two main phases: computing the preference probability distributions and preference-based racing.

Given a target entity v and its context $\mathcal{O}_v$, the time complexity of building its preference distributions (Algorithm 2) is $O(T|\mathcal{O}_v|^2 \tau_{sim})$, where τ_{sim} is the cost of similarity computation. This is explained since we need to make $|\mathcal{O}_v|(|\mathcal{O}_v| - 1)/2$ pairwise preference comparisons (through Eq. 3) between entities $v_i, v_j \in \mathcal{O}_v$ for each of the T timesteps, and each of these comparisons involves two structural similarity computations (i.e., $sim(v, v_i)$ and $sim(v, v_j)$).

The asymptotic cost of the second phase (Algorithm 3) is determined by the loop which, in the worst case, is executed n_{max} times when a satisfactory (according to δ) solution to the PBR problem cannot be found before.The cost of each iteration is $O(|\mathcal{O}_v|^2 + \tau_{SS})$, where τ_{SS} is the cost of the sampling strategy. Moreover, $\tau_{SS} = O(|\mathcal{O}_v|^2)$ for each of the sampling strategies we considered, because we need to check (in constant time) a condition for each pair of options (Line 4 in Algorithm 4). Thus, the cost of the second phase is $O(n_{max}|\mathcal{O}_v|^2)$.

The temporal cost of TNI for each entity v is $O(T|\mathcal{O}_v|^2 \tau_{sim} + n_{max}|\mathcal{O}_v|^2) = O(|\mathcal{O}_v|^2(T \cdot \tau_{sim} + n_{max}))$, and the total cost is $O(\sum_{v \in \mathcal{V}} |\mathcal{O}_v|^2 \ (T\tau_{sim} + n_{max}))$.

The spatial cost to solve TNI for each target entity v is determined by the space needed to store the pairwise preference distributions, thus its asymptotic growth is $O(T \cdot |\mathcal{O}_v|^2)$, since we need to store for each timestep the $O(|\mathcal{O}_v|^2)$ pairwise preference realizations which made up the distributions. The overall space complexity is $O(T \cdot (max_{v \in \mathcal{V}}|\mathcal{O}_v|)^2)$, since we can sequentially and independently solve the set of $|\mathcal{V}|$ PBR problems. Nonetheless, the approach is easily parallelizable by partitioning the set of entities, independently solving the PBR subproblems, then merging the results.

V. EVALUATION METHODOLOGY

We present our ground-truth-based methodology (Sect. V-A), the evaluation criteria (Sect. V-B) and datasets (Sect. V-C). Also, in Sect. V-D, we discuss the methods involved in a stage of comparative evaluation with TNI.

A. GROUND-TRUTH FOR TRUST NETWORK INFERENCE

To assess the meaningfulness of the results obtained by our TNI, we conducted different stages of evaluation based on two general, all-inclusive notions of ground-truth. These are hereinafter referred to as ***trust-class ground-truth*** and ***trust-network ground-truth***. As reported in the summary of Table 2, a ground-truth in our setting is either based on the notion of *trust class* or on the availability of a *reference trust network* for the input dynamic network.

The former corresponds to trust relations existing within a cohesive group of users in the input dynamic network, i.e., a trust class is regarded as a group of individuals whereby it is likely that they trust each other while they do not trust individuals outside the group. As exemplary domains, we recognize inferring trust network from interactions in real-life parties (i.e., contact networks) and from interactions occurring in collaborative networks (Table 2). Notably, given the relation of trust classes with the time-evolving interaction data, this ground-truth-based approach can help assess the discovery of trust/distrust relationships that are latent in the interaction data.

Trust-network ground-truth instead relies on a finer-grain type of trust relation, i.e., between pairs of users, which corresponds to the availability of a trust network that is regarded as a reference for the input dynamic network. The challenge in this case is that the ground-truth network may not be necessarily derived from interactions observed in the input time-evolving network data (i.e., two users may trust each other even though they never had a direct connection).

It should be emphasized that both the ground-truth classes and the reference trust networks were used *not to infer* a trust network, but *only for evaluation purposes*.

B. ASSESSMENT CRITERIA

Given a trust network $\mathcal{T} = \langle \mathcal{V}, \mathcal{E}, \omega \rangle$ inferred from a dynamic interaction network $\mathcal{G}$, we define a ground-truth trust classification as a partitioning Γ of the set of entities $\mathcal{V}$ into disjoint trust classes. Also, we denote with $\Gamma(v)$ the trust-class of entity v. We considered the following trust-class ground-truth based assessment criteria, for each entity v:

- *Binary preference* (Bpref) [3], which measures how many judged relevant candidates Rel are retrieved (i.e., occur in $\mathcal{T}$) ahead of judged irrelevant candidates $notRel$:

$$bpref(v) = \frac{1}{|Rel|} \sum_{v_r \in Rel} 1 - \frac{|rankedHigher(v_r)|}{|Rel|},$$

  where v_r is a relevant retrieved candidate, v_i is a member of the first $|Rel|$ irrelevant retrieved candidates, and $rankedHigher(v_r) = \{v_i \in notRel \mid \omega(v, v_i) > \omega(v, v_r)\}$. We define Rel (resp. $notRel$) as the set of out-neighbors of v in $\mathcal{T}$ such that $\Gamma(u) = \Gamma(v)$ (resp. $\Gamma(u) \neq \Gamma(v)$). The *global bpref* of a trust network is computed as the average entity bpref, i.e., $bpref(\mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} bpref(v)$.
- *Average intra-class trust*, as the average trust amount settled by v towards individuals within the same trust-class:

$$\Omega_{\Gamma}(v) = \frac{1}{|Rel|} \sum_{v_r \in Rel} \omega(v, v_r).$$

- *Average extra-class trust*, as the average trust amount settled by v towards individuals outside the v's trust-class:

$$\Omega_{\neg\Gamma}(v) = \frac{1}{|notRel|} \sum_{v_i \in notRel} \omega(v, v_i).$$

For the second type of ground-truth-based evaluation, given the availability of a reference trust network, we used it for a network similarity evaluation task, measuring *Precision*, *Recall* and *F1-score*. For any given $\mathcal{T}$ produced by TNI and reference network $\mathcal{T}^*$, both with set of nodes $\mathcal{V}$, precision (resp. recall) corresponds to the fraction of edges in $\mathcal{T}$ (resp. $\mathcal{T}^*$) shared with the other network, whereas F1-score is the harmonic mean of precision and recall.

TABLE 2: Ground-truth based evaluation types

	trust relation	*explicit network information*	*domain type*	*case studies*
Trust-Class	within-group links	no yes	Inferring trust network from interactions in real-life parties Inferring trust network from interactions in online collaborative system	Political parties Wikipedia editing
Trust-Network	individual pairs	yes	Inferring trust network from interactions in profit-based circles	Product rating

TABLE 3: Main structural features of our evaluation network datasets

	#entities $(\|\mathcal{V}\|)$	*#edges*	*#time steps* (T)	*avg. density*
DKpol, *DKpol-c*	490	1 821	30	0.074
DKpol-exp, *DKpol-exp-c*	490	288 680	32	0.047
WikiEdit *WikiEdit-exp*	1 115	33 304	49	0.06
CiaoDVD *CiaoDVD-c*	17 615	348 791	27	0.0174

C. CASE STUDIES AND DATASETS

We used 3 real-world, publicly available datasets: *DKpol* [11], *WikiEdit*,[1] and *CiaoDVD* [9]. According to Table 2, the former two were used for the trust-class ground-truth evaluation, the latter for the trust-network ground-truth evaluation. Table 3 provides a summary of structural characteristics of our evaluation networks. Also, we considered content-based variants, for a *total of 8 networks used in our evaluation*.

1) DKpol: Trust inference for political parties

DKpol contains Twitter following and activity data (i.e., tweets, retweets and replies) originally collected from the profiles of Danish politicians during the month leading to the parliamentary election in 2015. The profiled 494 politicians are distributed across 10 parties, each of which was regarded as one trust-class, i.e., politicians who are affiliated to the same party are supposed to trust each other, while distrusting politicians of other parties. By aggregating the user interactions on a daily basis, we extracted 30 directed networks such that, in the t-th snapshot, an edge from u to v is drawn if, at time t, u mentioned v, retweeted a v's tweet, or replied to a v's tweet. Starting from *DKpol*, we built a weighted network variant, dubbed *DKpol-c*, whereby the tweet contents are subjected to tool for sentiment analysis in Danish texts [18]. Each edge (u, v) in *DKpol-c* is weighted with a float value in $[0, 1]$ corresponding to the highest mood-score computed by the tool for the text of the tweet(s) posted by v and mentioned/replied/retweeted by u.

In order to stress our approach, we added noise to the data by simulating a *multicast propagation* of tweets/retweets made by a user towards her/his followers. In this scenario, which resulted in the *DKpol-exp* network, a tweet/retweet of user u triggers a set of links from u's followers to u. In addition, we built a content-based weighted variant, *DKpol-exp-c*, whose follower links are weighted with the neutral score of 0.5. Our rationale is that such links correspond to weak ties and, therefore, they would not be considered for the direct linkage contribution in Eq. 3 (i.e., $sim_C^{(t)}$ set to zero), however they are still considered in the structural similarity computations.

2) WikiEdit: Trust inference for a collaboration system

Our second case study concerns the context of Wikipedia page editing, which normally gives rise to either controversy or agreement among the editors. Our goal was to infer a trust network by observing the editing activities made by a set of users over a selection of pages of VIPs (from politics, sport, and other categories). The possible edit events are 'add', 'delete' or 'restore' content. The amount of text involved in each edit is quantified by the number of used words. Based on this information, we built the temporal network *WikiEdit* by considering the edits related to 10 among the top-edited pages and aggregating the events on a monthly basis. The *WikiEdit* network was obtained by modeling each edit event (of any type) made by a user u at time t as a set of edges in the t-th snapshot directed from u to each other user involved in the edit. In particular, the 'add' event involves only the active user (who performs the edit) while each 'delete' or 'restore' event is also annotated with the target user (the one who previously added/deleted the text). Each interaction e between two users is also labeled with a sign: 'positive' if they agree with the edit corresponding to the interaction, 'negative' otherwise. We exploit this additional information, together with the number of words nw_e involved in the edit, in order to compute the weight w_e of the interaction by means the following logistic function:

$$w_e = (1 + e^{-sign(e)\cdot\log_{10}(1+wc_e)})^{-1},$$

where $sign(e) = +1$ if e is a positive interaction, -1 otherwise. Note that positive (resp. negative) interactions will have weights higher (resp. lower) than 0.5.

We also considered an expanded version of *WikiEdit*, dubbed *WikiEdit-exp*. In this case, for each 'add' edit to page p made by user u at time t, we created weak ties (with neutral weight 0.5) from u to each other user that added content to p before t in order to represent a weak form of agreement of u towards the past 'add' edits made to p.

We created a graph where nodes are the page editors and links correspond to positive interactions between editors. On this graph, we applied the well-known Louvain community detection method [1] to obtain a partitioning of nodes that we consider as ground-truth communities for the evaluation.

[1] It will be made available at http://people.dimes.unical.it/andreatagarelli/data/.

3) Inferred trust network vs. reference trust network

For the trust-network ground-truth evaluation task, we considered the *CiaoDVD* dataset where users provide movie ratings (from 0 to 5) and can define their own local trust network by adding other users to their trust circle. The latter is considered as the ground-truth trust network for our evaluation.

We derived two temporal networks, *CiaoDVD* and *CiaoDVD-c*, where we aggregated the ratings on a monthly basis and extracted an edge from node u to v, in the t-snapshot, if there is at least one movie rated by both users in that month and v rated it before u. The rating similarity of the users is exploited to quantify the strength of interaction in the weighted version of the network, dubbed *CiaoDVD-c*. More specifically, given two users u and v and a set of M movies rated by both users at time t, and let $\mathbf{r}_u = [r_{u,1} \dots, r_{u,M}]$ and $\mathbf{r}_v = [r_{v,1}, \dots, r_{v,M}]$ be the associated ratings vectors, we quantify the strength of the interaction as:

$$w(u,v) = 1 - \frac{1}{M}\sum_{i=1}^{M}\left|\frac{r_{u,i}}{5} - \frac{r_{v,i}}{5}\right|.$$

D. COMPETING METHODS

We finally considered a comparative evaluation stage with a twofold goal: comparing the trust network inferred by our TNI w.r.t. a trust network built by (i) a data-driven baseline and (ii) a local-trust inference method (cf. Sect. II).

Our defined *data-driven baseline* (DDB) infers a trust network by aggregating the interactions observed in an input temporal network over all timesteps. In particular, for DKpol and CiaoDVD networks, the trust score of an edge (u, v) is computed as $W_{u,v}/W_u$, where $W_{u,v}$ here denotes the sum of weights of the interactions from u to v over all timesteps and W_u is the total sum of weights of interactions of u with any other node. For WikiEdit networks, the trust score of an edge (u, v) is computed as $W^+_{u,v}/(W^+_{u,v} + W^-_{u,v})$ where $W^+_{u,v}$ is the sum of weights of positive edits between u and v, while $W^-_{u,v}$ is the sum of the complement-one values of the weights of negative edits. This is explained to balance the numerical contributions given by positive interactions (i.e., edge weights above 0.5) vs. negative interactions (edge weights below 0.5). For example, suppose node u has one positive interaction with node v with weight 0.9 and five negative interactions all with weight 0.02: without complementing the negative interaction weights, the trust score of u to v would be $0.9/(0.9 + 5 * 0.02) = 0.9$ (i.e., high trust, which is counterintuitive); otherwise, it would be $0.9/(0.9+5*0.98) = 0.155$, which is more reasonable since it likely denotes a distrust relation rather than a trust one.

We chose the classic *TidalTrust* [7] (TT in short) as a representative local-trust inference method. This is designed to exploit the topological information in an input trust network for predicting a trust score for each pair of nodes that do not have a direct connection. The choice of selecting the shortest path derives from the hypothesis that reliability of trust values progressively decays proportionally to their distance from the source node. The trust between non-adjacent nodes is inferred by considering only shortest paths through trusted neighbors. The trust from a source to a destination node is calculated by calling a recursive trust function on the trusted neighbors, which terminates when the destination is reached. When the trust is back propagated to the source, it is averaged and rounded among the different trusted paths. Also, a path-pruning threshold is set to the maximum of the lowest trust values in each individual path from source to destination node. We used TT as follows: From the trust network obtained through DDB, repeatedly remove one edge at a time from the baseline network, then apply TT to compute its trust score, until all edges in the network are examined.

VI. RESULTS

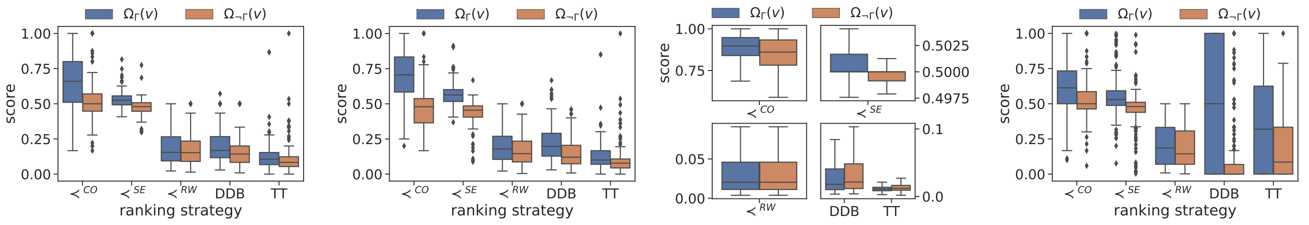
We present our main experimental results for each of the ground-truth-based evaluation stages (Sects. VI-A–VI-B). In this regard, note that a major goal of our experimental analysis is the assessment of TNI by varying the setting of its main parameters; nonetheless, unless otherwise specified, we will present results that correspond to default settings for parameters δ (0.1), α (0.5), k ($|\mathcal{O}_v|/2$, for any v), n_{max} (100), and Jaccard similarity as topological overlap function. In Sect. VI-C, we also discuss TNI efficiency aspects. Finally, in Sect. VI-D, we summarize main experimental findings.

A. TRUST-CLASS GROUND-TRUTH EVALUATION

Trust score distributions. For each entity (i.e., user in the input temporal network), we analyzed the distribution of its trust values among entities in the same ground-truth class and in the other classes. More specifically, we analyzed the boxplots of the distributions of $\Omega_\Gamma(v)$ and $\Omega_{\neg\Gamma}(v)$ values, over all entities in a network, for various sampling strategies and varying α. For the sake of presentation, we report here results corresponding to the default, balanced setting of α (i.e., 0.5) in Fig. 2, while results for $\alpha \in \{0.15, 0.85\}$ can be found in *Appendix*.

One important remark that supports the effectiveness of TNI is that, on average, an entity v tends to assign higher trust scores to entities in its ground-truth class ($\Gamma(v)$) than entities outside. This particularly holds, reagrdless of α in non-noisy networks (i.e., *DKpol*, *DKpol-c*, *DKpol-exp*) and for strategies CO and SE, which allow much clearer separation of the two distributions than the RW strategy. By contrast, it is worth emphasizing that the competitors can have an opposite trend, as in *DKpol-exp*, or even an overly positive-bias, as in *WikiEdit*.

Moreover, considering the effect on the distributions by varying α (Figs. 5–6, in *Appendix*), while negligible differences can be observed between the corresponding cases, for each network and method, we also found no monotonic behavior in the distribution overlap by progressively varying α; for instance, the default value of 0.5 (cf. Fig. 2b) ensures



(a) *DKpol* (b) *DKpol-c* (c) *DKpol-exp* (d) *WikiEdit*

FIGURE 2: Trust-class ground-truth evaluation: Boxplots of the distributions of the average intra-class trust ($\Omega_\Gamma(v)$) and of the average extra-class trust ($\Omega_{\neg\Gamma}(v)$) values

TABLE 4: Trust-class ground-truth evaluation: Global $bpref$ results. Bold text refers to the best values per dataset

	TNI with $\prec^{CO}$			TNI with $\prec^{SE}$			TNI with $\prec^{RW}$			DDB	TidalTrust
	α=0.85	α=0.5	α=0.15	α=0.85	α=0.5	α=0.15	α=0.85	α=0.5	α=0.15		
DKpol	0.531	**0.532**	0.493	0.484	0.484	0.499	0.418	0.423	0.401	0.248	0.406
DKpol-c	0.576	0.635	0.579	0.603	**0.644**	0.582	0.438	0.456	0.456	0.566	0.463
DKpol-exp	0.433	0.436	**0.522**	0.177	0.194	0.209	0.155	0.168	0.178	0.234	0.266
DKpol-exp-c	0.445	**0.548**	0.522	0.195	0.210	0.213	0.163	0.175	0.174	0.239	0.272
WikiEdit	0.524	0.402	0.391	**0.554**	0.46	0.441	0.443	0.386	0.386	0.392	0.293
WikiEdit-exp	**0.378**	0.354	0.352	0.1	0.1	0.1	0.142	0.138	0.14	0.02	0.286

better separation between the distribution boxplots than the other settings of α in *DKpol-c*, whereas for a network like *WikiEdit-exp* which was built on content-based collaborative editing, $\alpha = 0.85$ (cf. Fig. 5d) might be preferred to other settings.

Bpref analysis. Table 4 shows $bpref$ results obtained by different variants of TNI and by competing methods. Several remarks stand out. First, concerning the sampling strategies, CO and SE models generally lead to better performance of TNI than in the RW case, on every dataset and regardless of the α setting. In particular, CO improves upon SE especially in the noisy (i.e., expanded) networks, while SE prevails over CO in content-based networks (*DKpol-c* and *WikiEdit*). Second, TNI performance always increases when the network information is combined with content information to determine the preference probabilities (i.e., *DKpol-c* vs. *DKpol*, and *DKpol-exp-c* vs. *DKpol-exp*), regardless of the sampling strategy and α setting. Third, concerning the impact of parameter α, the balanced setting (i.e., $\alpha = 0.5$) leads to performance results that are comparable or better than for $\alpha = 0.85$ in the DKpol networks, while an opposite tendency is observed for WikiEdit networks, which are indeed more content-oriented than DKpol ones; analogously, $\alpha = 0.15$ may behave better than the balanced setting on noisy structure-oriented networks like *DKpol-exp*. Fourth, TNI significantly outperforms the competitor methods, at least when equipped with the CO strategy. DDB can behave better than TT (*DKpol-c* and *WikiEdit*), but the opposite holds on the noisy networks: this happens since TT is able to exploit the rich connectivity of expanded networks for inferring new trust links, while DDB considers the local interactions only.

Effect of k and n_{max}. Besides investigating the roles of the sampling stategy and of the α parameter, we also evaluated the impact of k and n_{max} on the TNI performance. To this end, we devised two stages: i) we varied n_{max} from the default 100 up to 500, while keeping k fixed to the default of half of the trust-context size, and ii) we varied k for different percentages of the trust-context size, with n_{max} fixed to 100. α was set to the default 0.5.

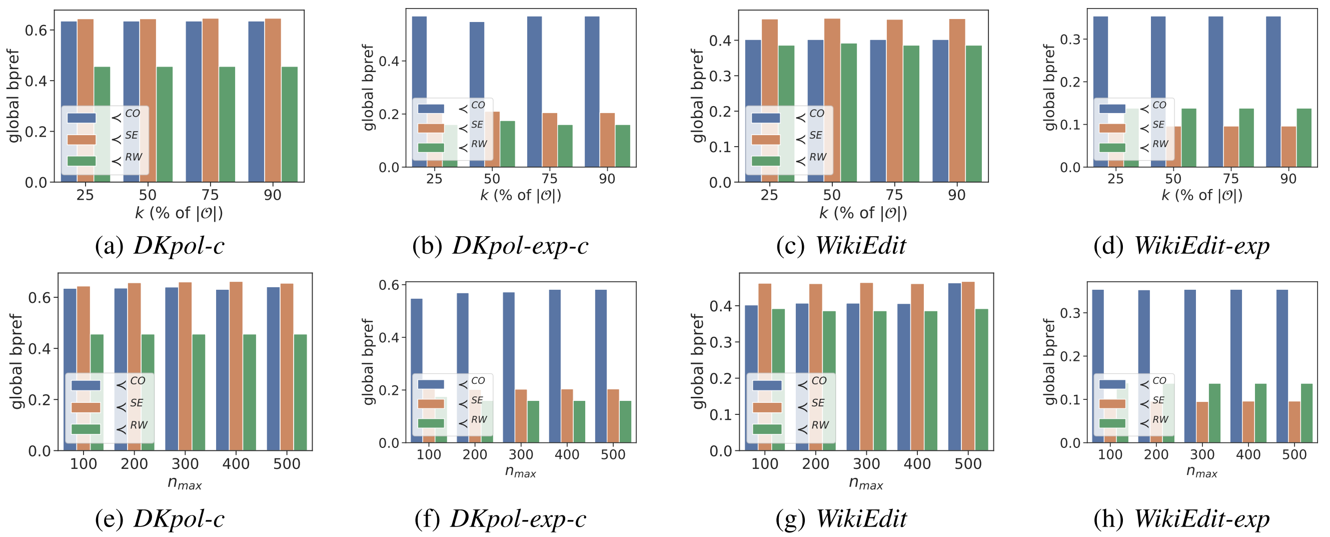
Figure 3 shows $bpref$ results obtained for various sampling strategies. At first sight, it stands out that, in both evaluation stages and for each network dataset, the relative differences between the sampling strategies follow the same trend when varying k (Fig. 3a-d) and n_{max} (Fig. 3e-h), respectively. Also, our choice of default settings of the two parameters turns out to correspond to $bref$ results that are very close to the performance peaks. Overall, this not only suggests relative robustness of TNI to variations of k and n_{max}, but also that the computational burden due to an increase of the values of the parameters can be avoided, since no particularly significant performance gain is guaranteed above specific values (i.e., default values).

Moreover, as already shown in Table 4, CO turns out to be the winner strategy for noisy networks (i.e., *DKpol-exp-c*, *WikiEdit-exp*), while SE prevails on other situations.

B. TRUST-NETWORK GROUND-TRUTH EVALUATION

For the second stage of evaluation, we filtered out the edges with trust scores below a certain threshold, which was set for each entity v as the 25-th percentile of the trust score of entities linked to v. Then, we derived an unweighted trust network to enable comparison with the unweighted reference networks of the CiaoDVD dataset.

Table 5 shows precision, recall and F1-score values w.r.t. the ground-truth trust network, for TNI (equipped with different sampling strategies and varying α) and competitors. Looking at the table, we observe that the best scores are



(a) *DKpol-c* (b) *DKpol-exp-c* (c) *WikiEdit* (d) *WikiEdit-exp*

(e) *DKpol-c* (f) *DKpol-exp-c* (g) *WikiEdit* (h) *WikiEdit-exp*

FIGURE 3: Trust-class ground-truth evaluation: Global $bpref$, (a)-(d) varying k (with fixed n_{max}) and (e)-(h) varying n_{max} (with fixed k)

TABLE 5: Trust-network ground-truth evaluation: Precision, recall and F1-score results. Bold text refers to the best values per dataset, for each criterion

		Precision					Recall					F1-score				
		$\prec^{CO}$	$\prec^{SE}$	$\prec^{RW}$	DDB	TT	$\prec^{CO}$	$\prec^{SE}$	$\prec^{RW}$	DDB	TT	$\prec^{CO}$	$\prec^{SE}$	$\prec^{RW}$	DDB	TT
CiaoDVD	$\alpha = 0.85$	0.231	0.235	**0.236**	0.232	0.231	0.804	0.842	**0.849**	0.796	0.812	0.359	0.367	**0.369**	0.359	0.36
	$\alpha = 0.5$	0.231	0.231	**0.236**			0.804	0.822	**0.85**			0.358	0.36	**0.37**		
	$\alpha = 0.15$	0.22	0.231	**0.234**			0.743	0.82	**0.844**			0.34	0.36	**0.367**		
CiaoDVD-c	$\alpha = 0.85$	**0.239**	0.238	0.238	0.236	0.226	0.838	0.847	**0.87**	0.807	0.793	0.372	0.372	**0.374**	0.365	0.352
	$\alpha = 0.5$	0.233	0.237	**0.238**			0.826	0.859	**0.899**			0.363	0.371	**0.377**		
	$\alpha = 0.15$	0.228	**0.237**	0.234			0.828	0.871	**0.899**			0.358	**0.372**	**0.372**		

always obtained by TNI, mostly with the RW strategy; this shows higher recall than the other strategies, while all three lead to similar performance in terms of precision and F1-score. Concerning precision in particular, the gap between TNI and the competitors is relatively small, and all achieve quite low values in both *CiaoDVD* networks. This is explained since the ground-truth network of CiaoDVD indeed was not derived from interaction data (i.e., a user may trust another one without interacting with her/him), thus the inferred trust network may not be in accord with the ground-truth knowledge. Mid-high values of recall are instead obtained on both networks, with the RW strategy outperforming SE and CO. In particular, TNI with RW or SE (along with CO in *CiaoDVD-c*) outperforms the two competitors, despite their bias in producing high trust scores for most edges.

C. EFFICIENCY EVALUATION

Table 6 shows the execution times of TNI, broken down into the procedures PDPP and PBR, using the default settings. It can be noted that, regardless of the particular network and strategy, most of the total running time is due to the PDPP procedure. Moreover, the RW strategy tends to yield better time performance of TNI, though of the same order of magnitude as for the other two strategies.

We also analyzed the time efficiency of TNI by varying the maximum number of samplings n_{max} from 100 to 500. Figure 4 shows time performances on DKpol and WikiEdit networks.[2] In accord with the computational complexity analysis (Sect. IV-C), the execution time for SE and CO strategies grows linearly with n_{max}. By contrast, the RW execution time grows much slower or even negligibly: this is explained since the value of n_{max} is checked by the RW strategy to decide if a pair of options does not need to be sampled anymore (Line 4 in Algorithm 4), and this leads the random walk to convergence faster than the other two strategies.

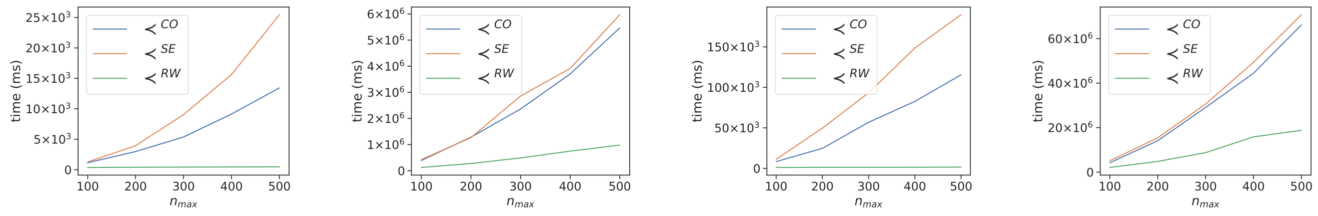
D. DISCUSSION

We coped with the task of assessing our proposed TNI by designing ground-truth-based stages of evaluation. We believe this design is remarkable as it allowed us to define an all-inclusive approach to the exploitation of ground-truth knowledge for evaluation purposes. Our stages of evaluation of TNI have indeed been defined upon either a notion of trust-class (i.e., cohesive group of mutually-trusted users) or on the availability of a reference trust-network for the input dynamic network. As a consequence,

[2] Platform Linux (Mint 18), with 2.6 GHz Intel Core i7-4720HQ, 16GB RAM

TABLE 6: TNI execution times (in seconds)

	TNI with $\prec^{CO}$			TNI with $\prec^{SE}$			TNI with $\prec^{RW}$		
	PDPP	PBR	total	PDPP	PBR	total	PDPP	PBR	total
DKpol	0.93	0.24	1.17	0.92	0.35	1.27	0.89	0.05	0.94
DKpol-c	0.98	0.2	1.18	1.02	0.26	1.28	1.01	0.06	1.07
DKpol-exp	420.25	44.32	464.57	420.05	66.44	486.49	420.63	1.32	421.95
DKpol-exp-c	409.75	52.38	462.13	410.83	86.52	497.35	423.07	2.4	425.47
WikiEdit	4.02	2.32	6.34	4.74	2.4	7.138	4.77	0.32	5.09
WikiEdit-exp	4172.50	110.8	4283.30	4173.96	230.24	4404.20	4176.44	12.32	4188.76
CiaoDVD	38452.02	310.60	38762.62	38454.04	521.52	38975.56	38501.02	43.48	38544.50
CiaoDVD-c	38545.02	322.10	38867.12	38540.53	543.85	39084.38	38543.52	53.48	38597



(a) *DKpol-c* (b) *DKpol-exp-c* (c) *WikiEdit* (d) *WikiEdit-exp*

FIGURE 4: TNI runtime performance by varying n_{max}

we identified three representative application domains for TNI, with corresponding case studies that refer to relevant scenarios in network analysis.

Upon these premises, experimental results have revealed important findings about the meaningfulness and effectiveness of our proposed method. TNI is capable of inferring a trust network where each entity (i.e., user) observed in the input time-evolving network is associated with higher trust scores to entities in its ground-truth class than to entities of other classes. By contrast, competing methods fail in having this behavior, showing sometimes an opposite trend or even an overly positive-bias (i.e., much more trust links than expected).

Despite having a number of parameters, TNI has shown to be surprisingly robust to their variation; particularly, the sampling strategies (i.e., ranking models) follow similar trends when varying the top-k trusted options for every target entity, and the number of samplings for each pairwise preference probability distribution (n_{max}). Also, using a balanced setting for α (i.e., the smoothing parameter that controls the contributions of structural and content information from the input network to determine the preference probabilities) has shown to be an appropriate default choice.

From an efficiency viewpoint, TNI running time grows linearly with the number of samplings. Overall, considering a trade-off between impact on the efficiency and impact on effectiveness of TNI, the sampling strategy based on the Copeland's ranking model turned out to be the best choice.

VII. CONCLUSIONS

We introduced the Trust Network Inference problem and proposed a preference-learning-based approach to solve it. Our approach can be regarded as key-enabling for any application that needs to build a trust network associated with a social environment from user interactions observed over time, in order to exploit the inferred trust relatioships in a variety of mining tasks.

Several aspects in our approach are worthy to be further investigated. Different definitions of trust-context and of structural/content affinity functions could easily be integrated into our proposed TNI framework; for instance, as we mentioned earlier in the paper, the trust-context model could be defined according to various topological structures, such as expanded ego-networks or community structures. Another aspect of interest is to extend our method to build a trust network incrementally in online tasks, i.e., inferring and maintaining/updating a trust network over a stream of interaction networks.

To encourage further development of our work, we make available to the community the preprocessed data used in the evaluation and the source code of TNI, at: *http://people.dimes.unical.it/andreatagarelli/tni/*.

REFERENCES

[1] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. J. Stat. Mech., 10:P10008, 2008.

[2] Z. Borbora, M. A. Ahmad, K. Z. Haigh, J. Srivastava, and Z. Wen. Exploration of robust features of trust across multiple social networks. In Proc. IEEE Conf. on Self-Adaptive and Self-Organizing Systems (SASOW), pp. 27–32, 2011.

[3] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In Proc. ACM SIGIR, pp. 25–32, 2004.

[4] R. Busa-Fekete, B. Szörényi, W. Cheng, P. Weng, and E. Hüllermeier. Top-k Selection based on Adaptive Sampling of Noisy Preferences. In Proc. ICML, pp. 1094–1102, 2013.

[5] H. Cai, V. W. Zheng, and K. C.-C. Chang. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Trans. Knowl. Data Eng.*, 30(9):1616–1637, 2018.

[6] X. Fan, D. He, and J. Bi. Trustworthiness and untrustworthiness inference with group assignment. In Proc. ICWS, pp. 389–404, 2018.

[7] J.A. Golbeck. Computing and Applying Trust in Web-based Social Networks. PhD thesis, College Park, MD, USA, 2005.

[8] F. C. Graham, A. Tsiatas, and W. Xu. Dirichlet PageRank and Ranking Algorithms Based on Trust and Distrust. Internet Mathematics, 9(1):113–134, 2013.

[9] G. Guo, J. Zhang, D. Thalmann, and N. Smith. ETAF: An Extended Trust Antecedents Framework for Trust Prediction. In Proc. IEEE/ACM ASONAM, pp. 540–547, 2014.

[10] Z. Gyöngyi, H. Garcia-Molina, and J. O. Pedersen. Combating Web Spam with TrustRank. In Proc. VLDB, pp. 576–587, 2004.

[11] O. Hanteer, L. Rossi, D. Vega D'Aurelio, and M. Magnani. From Interaction to Participation: The Role of the Imagined Audience in Social Media Community Detection and an Application to Political Communication on Twitter. In Proc. IEEE/ACM ASONAM, pp. 531–534, 2018.

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

[13] A. Jøsang, E. Gray, and M. Kinateder. Simplification and analysis of transitive trust networks. Web Intelligence and Agent Systems, 4(2):139–161, 2006.

[14] S. Kumar, W. L. Hamilton, J. Leskovec, and D. Jurafsky. Community interaction and conflict on the web. In Proc. WWW, pp. 933–943, 2018.

[15] H. Liu, E. Lim, H. W. Lauw, M. Le, A. Sun, J. Srivastava, and Y. A. Kim. Predicting trusts among users of online communities: an Epinions case study. In Proc. ACM Conf. on Electronic Commerce (EC), pp. 310–319, 2008.

[16] P. Massa and P. Avesani. Controversial Users Demand Local Trust Metrics: An Experimental Study on Epinions.com. In Proc. AAAI, pp. 121–126, 2005.

[17] S. Nepal, W. Sherchan, and C. Paris. STrust: A Trust Model for Social Networks. In Proc. TrustCom, pp. 841–846, 2011.

[18] F. Å. Nielsen. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In Proc. ESWC2 Workshop on 'Making Sense of Microposts', 2011.

[19] F. J. Ortega, J. A. Troyano, F. L. Cruz, C. G. Vallejo, and F. Enríquez. Propagation of trust and distrust for the detection of trolls in a social network. Comp. Netw., 56:2884–2895, 2012.

[20] W. Sherchan, S. Nepal, and C. Paris. A survey of trust in social networks. ACM Comput. Surv., 45(4):47:1–47:33, 2013.

[21] J. Tang, H. Gao, H. Liu, and A. Das Sarma. eTrust: understanding trust evolution in an online world. In Proc. ACM KDD, pp. 253–261, 2012.

[22] Y. Yao, H. Tong, F. Xu, and J. Lu. Subgraph extraction for trust inference in social networks. Ency. of Soc. Netw. Anal. Min., 2nd Ed., 2018.

[23] J.A. Aslam and S.E. Decatur. General bounds on statistical query learning and PAC learning with noise via hypothesis boosting. Inf. Comput., 141(2):85–118, 1998.

[24] R.E. Funderlic and C.D. Meyer. Sensitivity of the stationary distribution vector for an ergodic markov chain. Linear Algebra and its Applications, 76:1–17, 1986.

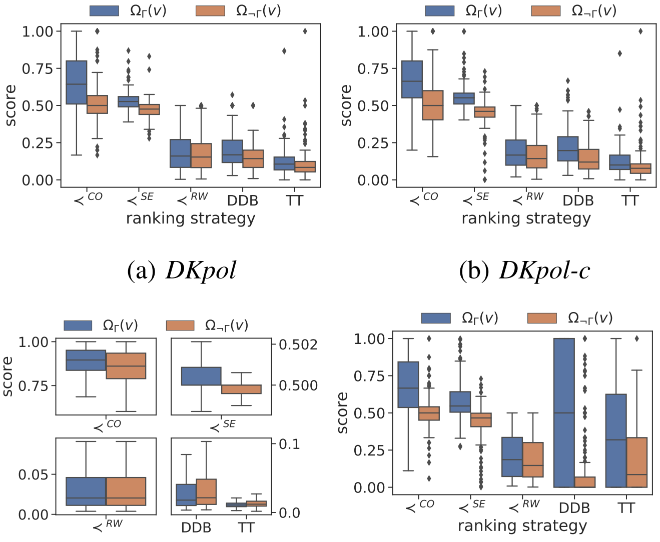
APPENDIX

DETAILS OF THE RANDOM-WALK-BASED SAMPLING STRATEGY

Random walk ranking first requires a left-stochastic version $\bar{\mathbf{S}} = [s_{ij}]_{N \times N}$ of the matrix $\bar{\mathbf{Y}}$, such that $s_{i,j} = \frac{\bar{y_{i,j}}}{\sum_{l=1}^{N} \bar{y_{l,j}}}$. Given confidence intervals for the entries of matrix $\bar{\mathbf{Y}}$, denoted with matrix $\mathbf{B} = [c_{i,j}]_{N \times N}$, confidence intervals for elements in $\bar{\mathbf{S}}$, denoted with matrix $\tilde{\mathbf{B}} = [\tilde{c}_{i,j}]_{N \times N}$, are computed (by applying a result in [23]) as:

$$\tilde{c}_{ij} = \frac{N}{3} \max_{k} c_{k,j} \sum_{l} \bar{y}_{l,j} \tag{5}$$

Note that the elements of a particular column of $\tilde{\mathbf{B}}$ are equal to each other, thus $\|\tilde{\mathbf{B}}\|_1 = \max_j \sum_i |\tilde{c}_{i,j}| = \frac{N^2}{3} \max_{k,j} c_{k,j} \sum_l \bar{y}_{l,j}$.



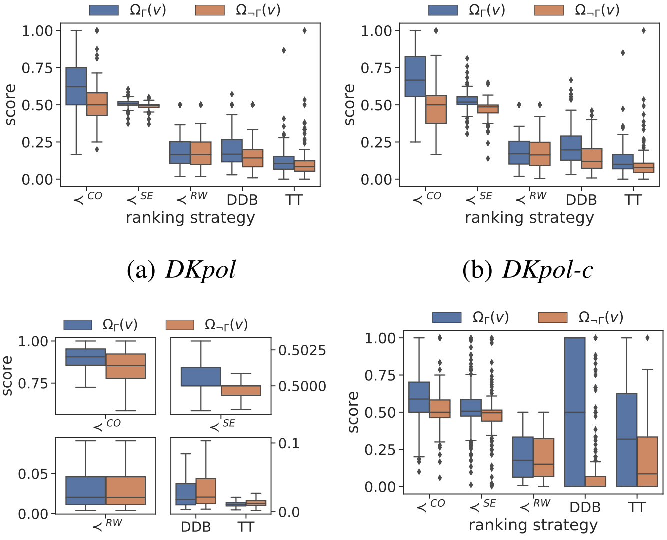
(a) *DKpol*

(b) *DKpol-c*

(c) *DKpol-exp*

(d) *WikiEdit*

FIGURE 5: Trust-class ground-truth evaluation: Boxplots of the distributions of the average intra-class trust ($\Omega_\Gamma(v)$) and of the average extra-class trust ($\Omega_{\neg\Gamma}(v)$) values, with $\alpha = 0.85$



(a) *DKpol*

(b) *DKpol-c*

(c) *DKpol-exp*

(d) *WikiEdit*

FIGURE 6: Trust-class ground-truth evaluation: Boxplots of the distributions of the average intra-class trust ($\Omega_\Gamma(v)$) and of the average extra-class trust ($\Omega_{\neg\Gamma}(v)$) values, with $\alpha = 0.15$

Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$ and $\bar{\boldsymbol{\pi}} = (\bar{\pi}_1, \ldots, \bar{\pi}_N)$ be the stationary distributions of S and $\bar{S}$ respectively. Then, by applying the result of [24], it follows that:

$$\|\boldsymbol{\pi} - \bar{\boldsymbol{\pi}}\|_{max} \leq \|\tilde{\mathbf{B}}\|_1 \|\bar{\mathbf{A}}^*\|_{max} \tag{6}$$

where $\bar{\mathbf{A}}^* = [\bar{a}^*_{ij}]_{N\times N} = (I - \bar{\mathbf{S}} + \mathbf{1}\boldsymbol{\pi}^T)^{-1} - \mathbf{1}\boldsymbol{\pi}^T$. Notice that, in order to obtain better estimates of the preferences, the bound in Eq. 6 suggests the minimization of $\|\tilde{\mathbf{B}}\|_1$ which can be performed by sampling pairs $(i,j) = \mathrm{argmax}_{i,j}\, c_{i,j} \sum_l \bar{y}_{l,j}$. At each time, the pairs of options that satisfy this condition are maintained as set of active options to be sampled next.

ADDITIONAL RESULTS FOR SECT.VI-A

Figures 5 and 6 show the boxplots of the distributions of the average intra-class trust and of the average extra-class trust, with $\alpha = 0.85$ and $\alpha = 0.15$, respectively.

• • •