International Journal on Artificial Intelligence Tools
Vol. 31, No. 7 (2022) 2260001 (23 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218213022600016

World Scientific
www.worldscientific.com

Higher Order Trust Ranking of LinkedIn Accounts with Iterative Matrix Methods

Georgios Drakopoulos
Department of Informatics, Ionian University
Plateia Tsirigoti 7, Kerkyra 49100, Hellas
c16drak@ionio.gr

Eleanna Kafeza
College of Innovation and Technology, Zayed University
Dubai Academic City, E-L1-008, UAE
eleana.kafeza@zu.ac.ae

Phivos Mylonas
Department of Informatics, Ionian University
Plateia Tsirigoti 7, Kerkyra 49100, Hellas
fmylonas@ionio.gr

Haseena Al Katheeri
College of Innovation and Technology, Zayed University
Abu Dhabi – Khalifa City, FF2-0-033
haseena.alaktheeri@zu.ac.ae

Received 4 April 2021
Accepted 6 January 2022
Published 18 November 2022

Trust is a fundamental sociotechnological mainstay of the Web today. There is substantial evidence about this since netizens implicitly or explicitly agree to trust virtually every Web service they use ranging from Web-based mail to e-commerce portals. Moreover the methodological framework for trusting individual netizens, primarily their identity and communications, has considerably progressed. Nevertheless, the core of fact checking for human generated content is still far from being substantially automated as most proposed smart algorithms capture inadequately fundamental human traits. One such case is the evaluation of the profile trustworthiness of LinkedIn members based on publicly available attributes available from the platform itself. A trusted profile may indirectly indicate a more suitable candidate since its contents can be easily verified. In this article a first order graph search mechanism for discovering LinkedIn trusted profiles based on a random walker is extended to higher order ranking based on a combination of functional and connectivity patterns. Results are derived for the same benchmark dataset and the first- and higher-order approaches are compared in terms of accuracy.

Keywords: Higher order metrics; attribute engineering; matrix iterative methods; stationary methods; Krylov methods; trust composition models; trust ranking; Web trust.

1. Introduction

Web trust is paramount for the proper function of a broad spectrum of online services including recommender systems, electronic auctions, e-commerce portals, and Web-based mail to name just a few. In fact, many actual service terms agreements include clauses which cannot be implemented without at least a certain degree of trust from the respective netizen base.[1,2] The converse must also hold true in a number of real world scenaria to ensure proper functionality. To this end many systems have enacted a wide range of computational trust measures such as internal access and privilege audits in enterprise operating systems or large databases, diversified social login[3] and two- or multi-factor authentication (2FA/MFA)[4] for social media or e-commerce platforms, and non-repudiation methods for e-mail systems.

The above leave open the fundamental questions of when and how can any two netizens using the same system can trust each other. A third related but underlying question is how a human trait such as trust can be expressed in digital terms and ultimately as a computational problem. In this article these questions are addressed on the basis of two principles. First, it is acknowledged that computing trust can done with human generated attributes. Second, based on established emotional models[5–8] trust can be composed from the trust disposition between netizens. As a concrete example, a methodology is presented for discovering trusted candidates for technology startups from a LinkedIn graph with startup and user profiles.

The primary research contribution of this article is the extension of an existing first order trust ranking expressed as a generic graph search scheme with two possible variants[9] to a higher order one expressed as the solution of a suitably formulated linear system. The latter is constructed from open attributes created by LinkedIn members and solved by suitable matrix iterative methods. To the best of the knowledge of the authors no such methods has been used to compute trust, which differentiates this work from previous ones. As a benchmark the same LinkedIn subgraph used to evaluate the baseline method will be also employed here.

The remaining of this article is structured as follows. In Sec. 2 the recent scientific literature regarding Web trust, blockchains, and matrix iterative methods is briefly summarized. Then Sec. 3 the higher order proposed methodology and how it differs from the baseline one are explained. The setup to conduct the experiments, their outcomes, and the related discussion are the focus of Sec. 4. The main findings as well as future research directions are given in Sec. 5. Matrices are represented by capital boldface letters and vectors by small boldface, while scalars by small letters. Vectors are always assumed to be columns, unless otherwise explicitly stated. Finally Table 1 summarizes the article notation.

Table 1. Notation of this article.

Symbol	Meaning	First in
$\triangleq$	Definition or equality by definition	Eq. (1)
$\mathrm{sign}\left(\cdot\right)$	Signum function of a scalar	Eq. (34)
$\left\vert\cdot\right\vert$	Absolute value (depending on the context)	Eq. (35)
$\left\Vert\cdot\right\Vert$	Matrix or vector norm	Eq. (17)
$\mathbf{e}_k$	Vector with 1 at the kth position and 0 everywhere else	Eq. (27)
$\mathbf{I}_n$	$n \times n$ indentity matrix	Eq. (8)
$\mathrm{radius}\left(\cdot\right)$	Matrix spectral radius	Eq. (38)
$\mathrm{diag}\left(\cdot\right)$	Diagonal part of a matrix or diagonal matrix	Eq. (14)
$\mathrm{triu}\left(\cdot\right)$	Strictly upper triangular part of a matrix	Eq. (14)
$\mathrm{tril}\left(\cdot\right)$	Strictly lower triangular part of a matrix	Eq. (14)
$\{s_1, \ldots, s_n\}$	Set with elements $s_1, \ldots, s_n$	Eq. (39)
$\left\vert\cdot\right\vert$	Set cardinality functional (depending on the context)	Eq. (20)

2. Previous Work

Trust is fundamental across a broad spectrum of Web applications for their proper functionality, including SaaS,[10] mobile payments,[11] recommender systems,[12] online news media,[13] and e-commerce platforms.[14] In the age of Semantic Web agents can be used to build trust for various entities.[15] In social media trust is a major driver for localized services.[16] LinkedIn trust can boost search for qualified startup candidates[9] and in general netizens exhibit there different trust patterns compared to Facebook including an increased openness to share professional information.[17] In Facebook trust can mitigate privacy concerns.[18] Trust in Twitter has been used among others for stock prediction,[19] rumor identification,[20] and botnet discovery.[21] In Twitter digital influence is inherently tied to trust[22–24] or more recently to community structure[25] or to social graph resilience.[27] Psychological aspects of Web trust and their applications have been recently the focus of interdisciplinary study.[28,29,26]

With the advent of blockchains[30] and later of rival technologies such as IOTA[31,32] research on both Web trust and trust limitations of IoT infrastructure was reinvigorated. Additionally, new directions have been added such as blockchain attack analysis,[33] consensus protocols,[34] and scalability.[35] Moreover, there is a broad spectrum of applications including vehicles,[36] manufacturing,[37] business disruption,[38] Industry 4.0,[39] and digital healthcare.[40]

Iterative matrix methods are frequently applied to large algebraic equations,[41] Sylvester matrix differential equations,[42] absolute matrix equations,[43] and nonnegative matrix factorization.[44] These methods have been successfully applied among other fields to computer vision,[45] multiple input-multiple output (MIMO)[46] and orthogonal frequency division multiplex (OFDM) telecommunication systems,[47] and Wiener-Hopf equations with exponential factors.[48]

3. Proposed Methodology

3.1. *Higher order linear trust composition model*

The extended trust model introduced along with the baseline method[9] and the underlying assumptions are presented here. This extension is based on a linear trust composition model inspired from established emotion composition models.[6,5] Before presenting the aforementioned model, the working definition of a trusted candidate[9] will be also repeated for clarity here in Definition 3.1. The actual values of the parameters mentioned in the following analysis are given in Table 2.

Definition 3.1. A candidate is trusted if and only if the skills listed in their LinkedIn profile can be verified directly from either itself or connected profiles.

Current models suggest that digital trust π_i in general can be composed by a number of factors operating at two distinct levels.[49,50] First, there is the first order individual component of trust. The latter in the context of LinkedIn means that trust τ_i is built on local features pertaining only to the ith LinkedIn profile l_i. Second, there is the higher order social component s_i where trust is gained or lost based on how trustworthy is the respective social environment. Again specializing to the context of this work, this component is expressed indirectly by the connectivity patterns of the LinkedIn adjacency matrix. Therefore as shown in Eq. (1):

$$\pi_i \triangleq \tau_i + s_i \tag{1}$$

Concerning the individual trust component, it will be built based on the following two open LinkedIn attributes to derive a trustworthiness metric τ_i for l_i based in part on previous approaches to digital trust[9]:

- The explicit trust τ_i^e is derived first by computing the fraction of the number c_j of skills referenced to the number c_i of skills l_i has and then by taking the arithmetic mean over all such r_i references for l_i as shown in Eq. (2):

  $$\tau_i^e \triangleq \frac{1}{r_i} \sum_{j=1}^{r_i} \frac{c_j}{c_i}, \qquad 0 \leq c_j \leq c_i \tag{2}$$

  For each r_i the number of skills c_j counts each mentioned skill only once.
- The implicit trust τ_i^m is defined as the fraction of the number c_i^* of top skills to the total number of skills c_i as shown in Eq. (3).

  $$\tau_i^m \triangleq \frac{c_i^*}{c_i} \tag{3}$$

  The top skills are determined individually for l_i as follows. The number of endorsements for all skills are sorted in descending order to obtain vector $\tilde{\mathbf{w}}_i$. Starting from the top skill of l_i at $\tilde{\mathbf{w}}\left[1\right]$ and moving one step at a time towards $\tilde{\mathbf{w}}\left[c_i\right]$ from the endorsements are progressively added until the fraction of the partial sum of endorsements to their total sum exceeds a threshold ξ^* as in Eq (4).

$$c_i^* \triangleq \min \left[k^* \mid \frac{\sum_{k=1}^{k^*} \tilde{\mathbf{w}}[k]}{\sum_{k=1}^{c_i} \tilde{\mathbf{w}}[k]} \geq \xi^* \right] \tag{4}$$

From the attribute description of the two components τ_i^e and τ_i^m it follows that they are independent. Thus they can be added in a weighted sum as shown in Eq. (5) where the hyperparameter ρ_0 indicates the relative component weight. Moreover, its addition results in more flexibility.

$$\tau_i \triangleq \left(\frac{\rho_0}{1+\rho_0} \right) \tau_i^e + \left(\frac{1}{1+\rho_0} \right) \tau_i^m \tag{5}$$

Regarding the social component s_i of Eq. (1), its formation will be based on three underlying assumptions, namely that trust as a human trait can be also derived from the social environment of l_i,[50] trust can be composed,[9] and perhaps more importantly is that this composition is linear. Any trust composition scheme is implicitly based on trust transitivity, which is essentially why higher order trust patterns can be computed. The linear composition implies Eq. (6) for l_i:

$$s_i = \sum_{j=1, j \neq i}^{n} a_{i,j} s_j, \qquad \sum_{j=1, j \neq i}^{n} a_{i,j} = 1, \quad a_{i,j} \geq 0 \tag{6}$$

The meaning of Eq. (6) is that the value s_i of the social trust component for l_i is a linear function of the respective values of connected LinkedIn profiles. The recursive nature of this equation results in s_i being computed in a higher order manner. Recasting it in a matrix-vector form yields Eq. (7):

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \Leftrightarrow \mathbf{A}\mathbf{s} = \mathbf{s} \tag{7}$$

In Eq. (7) the vector $\mathbf{s}$ contains the values of the social trust component with $\mathbf{s}[i] = s_i$ whereas the coefficients are stored in $\mathbf{A}$ with $\mathbf{A}[i,j] = a_{i,j}$. This is a normalized version of the adjacency matrix. At this point it should be highlighted that Eq. (7) can be cast as either an eigenvalue problem or a linear system, with both formulations being equally valid, as shown in Eq. (8). Depending on the available resources, either can be used to derive the trust ranking.

$$(\mathbf{I}_n - \mathbf{A})\mathbf{s} = \mathbf{0} \Leftrightarrow \mathbf{G}\mathbf{s} = \mathbf{0} \tag{8}$$

The dimension of the null space of $\mathbf{G}$ equals the number of the independent components whose linear combinations yield the solution subspace. From the definition of $\mathbf{G}$ and the Perron-Frobenius theorem there is only one such component.

3.2. Jacobi iteration

A general class of iterative methods for solving linear systems is that of the stationary methods. The starting point is the partitioning of the $n \times n$ coefficient matrix of a generic linear system as shown in Eq. (9) in order to obtain two matrices with special properties. The name of this class of methods is derived from these fundamental properties since the general objective is to reach the stationary point of Eq. (11). From the very formulation of the linear system it follows that:

$$\mathbf{M}\mathbf{x} = \mathbf{b} \Leftrightarrow (\mathbf{M}_1 + \mathbf{M}_2)\mathbf{x} = \mathbf{b} \Leftrightarrow \mathbf{x} = -\mathbf{M}_1^{-1}\mathbf{M}_2\mathbf{x} + \mathbf{M}_1^{-1}\mathbf{b} \tag{9}$$

Observe that there is full equivalence between the three parts of Eq. (9) as long as $\mathbf{M}_1$ is invertible. Notice that the inversion property is never used in practice as inverting a matrix is expensive in terms of floating point operations (FPOs) and also prone to numerical errors. Instead, a solution of a suitable linear system takes place. Eq. (9) is the basis for the generic stationary point iterative schemes of Eq. (10):

$$\mathbf{x}^{[k]} = -\mathbf{M}_1^{-1}\mathbf{M}_2\mathbf{x}^{[k-1]} + \mathbf{M}_1^{-1}\mathbf{b}, \qquad \mathbf{x}^{[k]} \in \mathbb{R}^n \tag{10}$$

Eventually and if certain conditions hold the iterative scheme of Eq. (10) will have a trajectory in the candidate solution subspace, which is itself embedded in $\mathbb{R}^n$, such that it will eventually find the stationary point of Eq. (11):

$$\mathbf{x}^{[k]} \approx \mathbf{x}^{[k-1]} \tag{11}$$

This is equivalent to Eq. (12) where the higher order function $\varphi(\cdot)$ is essentially the mechanism generating the next iteration vector.

$$\mathbf{x}^{[k+1]} = \varphi\Big(\mathbf{x}^{[k]}; \mathbf{M}_1, \mathbf{M}_2, \mathbf{b}\Big) \tag{12}$$

From Eqs. (10) and (12) it follows that the properties of $\varphi(\cdot)$ or $\mathbf{M}_1^{-1}\mathbf{M}_2$ in this case determine whether the iterative method will converge and, if yes, how fast in terms of iterations. Excluding the possible event of numerical instability caused by factors so diverse as improperly implemented FPOs, division by very small floating point numbers (FPNs), or addition of very uneven FPNs, the following characteristics are key to the successful design of stationary iterative methods:

- Matrix $\mathbf{M}_1$ should not only be invertible but also easy to invert. Along a similar line of reasoning, matrix $\mathbf{M}_1^{-1}\mathbf{M}_2$ should be easy to compute.
- The spectral radius of $\mathbf{M}_1^{-1}\mathbf{M}_2$ denoted by radius $\big(\mathbf{M}_1^{-1}\mathbf{M}_2\big)$ determines the convergence of the method.
- If the eigenvalues are clustered, even approximately, then some methods may well discover quicker the components of the stationary point.
- Matrix $\mathbf{M}_1^{-1}\mathbf{M}_2$ should be a partial contraction operator to directions perpendicular to the stationary point so that components in them vanish.

Notice that in the general case the stationary point of Eq. (12) may differ than the solution of Eq. (9) especially if $\varphi(\cdot)$ is non-linear. However, in the particular case it is linear by construction and, hence, there is only one stationary point which is also the solution of the original linear system. Thus, the selection of the starting point $\mathbf{x}^{[0]}$ is less restricted and in fact it can be any arbitrary or random vector when there is insufficient information about the stationary point.

Frequently for specific linear systems an appropriate invertible preconditioning matrix $\mathbf{P}$ is sought such that the linear system of Eq. (13) is solved instead.

$$\mathbf{PMx} = \mathbf{Pb} \tag{13}$$

The solution is equivalent with (9) but preconditions may result in faster reaching the stationary point in the solution subspace. Frequently $\mathbf{P}$ depends on the specific system. Such methods are beyond the scope of this work.

Interest in the methods of Eq. (12) has been reinvigorated since they are used in many current graph neural network (GNN) algorithms, where graph topological properties are also considered besides the functional ones.[51]

The Jacobi method can be obtained from the template of Eq. (10) by observing that any rectangular matrix $\mathbf{M}$ can be uniquely partitioned to three constituent parts, each with its own interpretation, as shown in Eq. (14). In Eq. (14) the matrices at the right hand side are the diagonal part, the strictly lower triangular, and the strictly upper triangular part of the original matrix $\mathbf{M}$.

$$\mathbf{M} = \mathrm{diag}\left(\mathbf{M}\right) + \mathrm{tril}\left(\mathbf{M}\right) + \mathrm{triu}\left(\mathbf{M}\right) \tag{14}$$

Combining Eqs. (14) and (10) the Jacobi method is derived for the options:

$$\mathbf{M}_1 \triangleq \mathrm{diag}\left(\mathbf{M}\right) \qquad \text{and} \qquad \mathbf{M}_2 \triangleq \mathbf{M} - \mathbf{M}_1 = \mathrm{triu}\left(\mathbf{M}\right) + \mathrm{tril}\left(\mathbf{M}\right) \tag{15}$$

The interpretation of Eq. (15) is that the ith equation of Eq. (8) is solved for the ith unknown $\mathbf{s}\left[i\right]$ and updated at each iteration, perhaps after rearrangement of equations or unknowns to avoid zero coefficients $\mathbf{G}\left[i,i\right]$. This results in Eq. (16):

$$\mathbf{s}^{[k]}\left[i\right] = -\frac{1}{\mathbf{G}\left[i,i\right]} \sum_{j=1 j\neq i}^{n} \mathbf{G}\left[i,j\right] \mathbf{s}^{[k-1]}\left[j\right] \tag{16}$$

Under Jacobi scheme each unknown $\mathbf{s}\left[i\right]$ is updated in the same rate. This may be desirable if the components of the initial solution $\mathbf{x}^{[0]}$ are almost uniformly far from the true solution or, alternatively, if each vertex of the graph has approximately the same number of neighbors. In these cases, the trust metric value of each vertex is updated approximately at the same rate. This reduces the probability that the Jacobi process is delayed by only a few components.

Regarding the termination criteria, there is a number of conditions which can be used. In this work the failsafe criterion of a maximum number of iterations T will be used in conjunction with the condition of Eq. (17) which checks

Algorithm 1 The Jacobi iteration

```
Require: Matrix $\mathbf{M}$, vector $\mathbf{b}$, initial guess $\mathbf{x}^{[0]}$, iterations $T$, and threshold $\xi_0$
Ensure: An approximate solution to Eq. (9) is computed
partition $\mathbf{M}$ as in Eq. (15) to obtain $\mathbf{M}_1$ and $\mathbf{M}_2$
repeat
    compute $\mathbf{x}^{[k]}$ from Eq. (10)
until $T$ is exceeded or Eq. (17) is satisfied
return last $\mathbf{x}^{[k]}$
```

whether the overall difference in the magnitude of two solutions drops below a threshold ξ_0.

$$\left\|\mathbf{x}^{[k]} - \mathbf{x}^{[k-1]}\right\|_2 \leq \xi_0 \tag{17}$$

Algorithm 1 summarizes the Jacobi iteration.

3.3. *Gauss-Seidel iteration*

An improvement over the Jacobi iteration is Gauss-Seidel scheme. The latter can be seen as solving the ith equation for the ith unknown $\mathbf{s}[i]$ but now the new $i-1$ updates for unknowns $\mathbf{s}[1]$ up to $\mathbf{s}[i-1]$ are used. This leads to Eq. (18):

$$\mathbf{s}^{[k]}[i] = -\frac{1}{\mathbf{G}[i,i]}\left(\sum_{j=1}^{i-1}\mathbf{G}[i,j]\,\mathbf{s}^{[k]}[j] + \sum_{j=i+1}^{n}\mathbf{G}[i,j]\,\mathbf{s}^{[k-1]}[j]\right) \tag{18}$$

In matrix form the Jacobi iteration is derived from the general matrix partitioning scheme of Eq. (14) as shown in Eq. (19):

$$\mathbf{M}_1 = \mathrm{diag}(\mathbf{M}) + \mathrm{triu}(\mathbf{M}) \qquad \text{and} \qquad \mathbf{M}_2 = \mathrm{tril}(\mathbf{M}) \tag{19}$$

In this case each component of $\mathbf{x}^{[k]}$ is updated at a different rate with those components in the first places of this vector using more components of $\mathbf{x}^{[k-1]}$. On the contrary, the last components of $\mathbf{x}^{[k]}$ rely almost exclusively on updated values. Therefore, it is advisable to rearrange the vertex location in the system of Eq. (8) in increasing degree order so that the update of the trust value of a vertex with a rather large neighborhood will use a big number of updated values from the current iteration. Alternatively, a topological ordering of the vertices can yield an update sequence with a minimal number of update conflicts or update delays, depending on the optimality criterion. In this work the first option was used.

With the exceptions of the preparatory vertex rearrangement step and the different partitioning of matrix $\mathbf{M}$ the Gauss-Seidel iteration is identical to the Jacobi iteration and hence its algorithmic layout is similar to Algorithm 1.

3.4. *GMRES*

Another general approach for iteratively solving the linear system of Eq. (8) is the class of Krylov based iterative methods. They first try to place any linear system in the context of a suitably generated Krylov space and then devise an iterative method based on an optimality criterion in that space. A Krylov space is the linear subspace of $\mathbb{R}^n$ generated by any pair of matrix $\mathbf{V} \in \mathbb{R}^{n \times n}$ and vector $\mathbf{y} \in \mathbb{R}^n$. They create a set V of n vectors as in Eq. (20) whose linear span is the Krylov space.

$$V \triangleq \{\mathbf{y}, \mathbf{V}\mathbf{y}, \ldots, \mathbf{V}^{n-1}\mathbf{y}\}, \quad |V| = n \tag{20}$$

The Generalized Minimum RESidual (GMRES) method relies heavily on the Krylov space spanned by matrix $\mathbf{G}$ of Eq. (8) and the residual $\mathbf{r}^{[0]}$ which is obtained from the starting point $\mathbf{s}^{[0]}$, which can very well be in initial guess or a random vector when no other information is available, as shown in Eq. (21).

$$\mathbf{r}^{[0]} \triangleq -\mathbf{G}\mathbf{s}^{[0]} \tag{21}$$

In this case the Krylov space the sought solution lies in is spanned by the n vectors of Eq. (22). Note that they ordinarily do not constitute a basis for that particular space. Instead, they are the starting point for finding one.

$$V \triangleq \left\{\mathbf{r}^{[0]}, \mathbf{G}\mathbf{r}^{[0]}, \ldots, \mathbf{G}^{n-1}\mathbf{r}^{[0]}\right\} = \left\{-\mathbf{G}\mathbf{s}^{[0]}, -\mathbf{G}^2\mathbf{s}^{[0]}, \ldots, -\mathbf{G}^n\mathbf{s}^{[0]}\right\} \tag{22}$$

Observe the special structure of V since the right hand side of Eq. (8) is zero. Also, the starting point $\mathbf{s}^{[0]}$ cannot be zero since in that case the span degenerates to zero.

In each iteration GMRES constructs progressively a base for V and evaluates how the residual norm behaves. Since the subspace obtained in each step is properly included in the next one, each inclusion of a new component will at best decrease the residual norm. This happens as the exact solution $\mathbf{s}$ may not have components in each of the n directions of the base vectors of V. Each new basis vector $\mathbf{q}_i$ is computed through the Arnoldi iteration, which is a specialization of the Gram-Schmidt orthogonalization process tailored for Krylov spaces.

Assume the n base vectors for the specific Krylov space of Eq. (22) are $\mathbf{q}_i$, $1 \leq i \leq n$. The solution $\mathbf{s}$ is a linear combination of these vectors for some coefficients c_i as shown in Eq. (23). Expressing its right hand side in matrix-vector form yields:

$$\mathbf{s} = \sum_{i=1}^{n} c_i \mathbf{q}_i = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \ldots & \mathbf{q}_n \end{bmatrix} \begin{bmatrix} c_1 & c_2 & \ldots & c_n \end{bmatrix}^T = \mathbf{Q}\mathbf{c} \tag{23}$$

In Eq. (23) the columns of the orthonormal matrix $\mathbf{Q}$ are the basis vectors $\mathbf{q}_i$, whereas vector $\mathbf{c}$ contains the coefficients c_i. Note that $\mathbf{s}$ may depend on only n' vectors, in which case GMRES terminates in n' iterations with the exact solution.

Because of the special structure of V if $\mathbf{G}$ multiplies $\mathbf{Q}_i$, namely the matrix with the first i columns of $\mathbf{Q}$, then the result will be a product of $\mathbf{Q}_{i+1}$ by an upper Hessenberg matrix $\mathbf{H}_i$ acting as a weight as shown in Eq. (24). This special structure of $\mathbf{H}_i$ is attributed both to the structure of V as well as to the independence of $\mathbf{q}_i$.

$$\mathbf{G}\mathbf{Q}_i = \mathbf{Q}_{i+1}\mathbf{H}_i \tag{24}$$

To compute the candidate solution $\mathbf{s}^{[k]}$ during the kth iteration first it is written as the sum of the starting point $\mathbf{s}^{[0]}$ plus a correction vector $\mathbf{z}^{[k]}$ as shown in Eq. (25).

$$\mathbf{s}^{[k]} = \mathbf{s}^{[0]} + \mathbf{z}^{[k]} \tag{25}$$

Since $\mathbf{z}^{[k]}$ is in the Krylov space and in particular in its subspace spanned by the first k basis vectors, then $\mathbf{z}^{[k]}$ can be written as a linear combination of $\mathbf{q}_1$ up to $\mathbf{q}_k$ for some coefficients y_1 to y_k or as shown in matrix-vector form in Eq. (26):

$$\mathbf{z}^{[k]} = \sum_{i=1}^{k} y_i \mathbf{q}_i = \mathbf{Q}_k \mathbf{y}_k \tag{26}$$

At the heart of GMRES is the selection of the coefficient vector $\mathbf{y}_k$. The latter is computed so that the current candidate solution $\mathbf{s}^{[k]}$ minimizes the length of the residual $\mathbf{r}^{[k]}$ over the current basis subset of the Krylov space as in Eq. (27).

$$\left\|\mathbf{r}^{[k]}\right\|_2 = \left\|-\mathbf{G}\left(\mathbf{s}^{[0]} + \mathbf{z}^{[k]}\right)\right\|_2 = \left\|-\mathbf{G}\mathbf{s}^{[0]} - \mathbf{G}\mathbf{Q}_k\mathbf{y}_k\right\|_2 = \|\beta_0\mathbf{e}_1 - \mathbf{H}_k\mathbf{y}_k\|_2 \tag{27}$$

This is a linear least squares problem for $\mathbf{y}_k$. A selection of a different norm can lead to another optimization problem which is potentially non-convex. Also it has a Hessenberg coefficient matrix which means its solution has a much lower computational complexity compared to the general case. This can be seen from the fact that Eq. (27) is almost in the form required by the standard QR factorization approach.

From Eq. (27) it follows that the constant β_0 has the value of Eq. (28).

$$\beta_0 = \left\|-\mathbf{G}\mathbf{s}^{[0]}\right\|_2 = \left\|\mathbf{r}^{[0]}\right\|_2 \tag{28}$$

Regarding the termination of GMRES, there is a number of criteria. One is to set an iterations limit $T(n)$. Also here is checked whether the difference between two successive candidate solutions drops below a threshold ξ_1 as shown in Eq. (29).

$$\left\|\mathbf{s}^{[k]} - \mathbf{s}^{[k-1]}\right\|_2 \leq \xi_1 \tag{29}$$

The GMRES method is summarized in Algorithm 2.

Algorithm 2 The GMRES iterative method

```
Require: Matrix $\mathbf{G}$, initial guess $\mathbf{s}^{[0]}$, number of iterations $T(n)$, and threshold $\xi_1$
Ensure: An approximate solution to $\mathbf{G}\mathbf{s} = \mathbf{0}$ is found
compute the initial residual $\mathbf{r}^{[0]}$
repeat
    compute another $\mathbf{q}_k$ from the Arnoldi process
    compute the linear least squares of Eq. (27)
until $T(n)$ is exceeded or Eq. (29) is satisfied
return $\mathbf{s}^{[k]}$
```

4. Results

4.1. *Experimental setup*

Table 2 has the actual value for every parameter mentioned in the preceding analysis. Notice that the values for ρ_0 and ξ^* are the same with those used in the experiments of the baseline method for fairness reasons.

The LinkedIn dataset[9] which was used in the experiments has the properties shown in Table 3. Observe that it is a relatively sparse graph which despite that has a quite connected critical mass as more than half of the vertices are within a

Table 2. Experimental setup.

Parameter	Value
Relative importance hyperparameter ρ_0 in Eq. (5)	1
Top skill endorsement threshold ξ^* in Eq. (4)	0.75
Threshold ξ_0 for Jacobi termination in Eq. (17)	0.05
Jacobi and Gauss-Seidel maximum iterations T in Algorithm 1	250
GMRES correction length threshold ξ_1 in Eq. (29)	0.05
GMRES maximum iterations $T(n)$ in Algorithm 2	Eq. (40)
Levenberg-Marquart parameter correction length threshold ξ_2 in Sec. 4.3	0.05
Levenberg-Marquart regularization parameter λ_0 in Eq. (46)	0.5

Table 3. LinkedIn dataset properties.

Parameter	Value	Parameter	Value
Number of startup profiles v_s	47	Graph diameter	10
Number of candidate profiles v_c	6391	Fraction of vertices with distance 6	15.66%
Edges	1512 388	Fraction of vertices with distance 7	12.33%
Triangles	31 4003	Fraction of vertices with distance 8	7.81%
Squares	109 863	Fraction of vertices with distance 9	4.68%
Connected components	1	Fraction of vertices with distance 10	1.15%

distance which is at most half the graph diameter. This was identified as a critical factor for the success of the baseline method.

4.2. *Performance metrics*

For the stationary method template of Eq. (10) two error metrics which are readily accessible are the correction magnitude $u^{[k]}$ and angle $\vartheta_r^{[k]}$. They are computed between any two successive iterations as shown respectively in Eqs. (30) and (32).

$$u^{[k]} \triangleq \left\| \mathbf{x}^{[k]} - \mathbf{x}^{[k-1]} \right\|_2 \tag{30}$$

In Table 4 the average Δu over all iterations is given.

In Eq. (30) $u^{[k]}$ can be thought of as a measure of speed the solution moves with as successive iterations of the template of Eq. (10) take place. It can be further analyzed using the recursive nature of the iteration generator of Eq. (10) as follows.

$$\begin{aligned} \mathbf{x}^{[k]} - \mathbf{x}^{[k-1]} &= \left(-\mathbf{M}_1^{-1}\mathbf{M}_2\mathbf{x}^{[k-1]} + \mathbf{M}_1^{-1}\mathbf{b} \right) - \left(-\mathbf{M}_1^{-1}\mathbf{M}_2\mathbf{x}^{[k-2]} + \mathbf{M}_1^{-1}\mathbf{b} \right) \\ &= -\mathbf{M}_1^{-1}\mathbf{M}_2 \left(\mathbf{x}^{[k-1]} - \mathbf{x}^{[k-2]} \right) \\ &= \left(-\mathbf{M}^{-1}\mathbf{M}_2 \right)^k \left(\mathbf{x}^{[1]} - \mathbf{x}^{[0]} \right) \\ &= \mathbf{S}_m \cdot \operatorname{diag}\left(\mu_1^k, \ldots, \mu_n^k \right) \cdot \mathbf{S}_m^T \cdot \left(\mathbf{x}^{[1]} - \mathbf{x}^{[0]} \right) \end{aligned} \tag{31}$$

The last line of Eq. (31) comes directly from the spectral decomposition of the iteration matrix $-\mathbf{M}_1^{-1}\mathbf{M}_2$. From this follows immediately the important role of its eigenstructure, especially its spectral radius denoted by radius $\left(-\mathbf{M}_1^{-1}\mathbf{M}_2\right)$, to the convergence and the rate thereof of the template of Eq. (10). The rate of convergence depends heavily on the ratio of the largest to the smallest eigenvalues and, thus, matrices $\mathbf{M}_1^{-1}$ and $\mathbf{M}_2$ can be accordingly designed to minimize it.

The correction angle of Eq. (32) moves along a similar line of reasoning by computing the local curvature of the solution path. Large changes may indicate a realignment of the solution trajectory towards the right direction, which can possibly be attributed to the discovery of an as yet unknown solution component. On the contrary, small changes may indicate trajectory convergence. The derivation of Eq. (32) was based on the Cauchy-Schawrz identity plus the observation that in Eq. (10) when $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{b}$, and $\mathbf{x}^{[0]}$ are real, then $\mathbf{x}^{[k]}$ are real.

$$\vartheta_r^{[k]} \triangleq \arccos\left(\frac{\left(\mathbf{x}^{[k]}\right)^T \mathbf{x}^{[k-1]}}{\left\|\mathbf{x}^{[k]}\right\|_2 \left\|\mathbf{x}^{[k-1]}\right\|_2} \right) \tag{32}$$

In Table 4 the average $\Delta\vartheta_r$ over all iterations is given.

Additional metrics based on or inspired of Eqs. (30) and (32) are the acceleration $a^{[k]}$ of Eq. (33) and the angle change $\vartheta_c^{[k]}$ or Eq. (34). The former roughly corresponds to the acceleration of the solution as it moves along the solution path in $\mathbb{R}^n$ generated by the mechanism of Eq. (10).

$$a^{[k]} \triangleq \frac{1}{2}\left\|\mathbf{x}^{[k]} - 2\mathbf{x}^{[k-1]} + \mathbf{x}^{[k-2]}\right\|_2 \tag{33}$$

In Table 4 the average Δa is given.

The angle change of Eq. (34) is the absolute value of the difference of the correction angles between two successive iterations. It is an additional metric of trajectory curvature spanning, albeit a somewhat less crude one that the correction angle as it spans over three iterations. Thus, $\vartheta_c^{[k]}$ gives more than the instantaneous curvature.

$$\vartheta_c^{[k]} \triangleq \left|\vartheta_r^{[k]} - \vartheta_r^{[k-1]}\right| \tag{34}$$

In Table 4 the average $\Delta\vartheta_c$ is given.

An alternative definition for Eq. (34) is that of Eq. (35) which quantizes the change in the correction angle at the range of $\{\pm 1\}$. Because of this, the alternative definition will not be used here.

$$\tilde{\vartheta}_c^{[k]} \triangleq \operatorname{sign}\left(\vartheta_r^{[k]} - \vartheta_r^{[k-1]}\right) \tag{35}$$

Another way to assess the performance of the template of Eq. (10) when applied to the linear system of Eq. (9) is to measure the distance between the left hand side vector $\mathbf{b}$ and the result of the application of a candidate solution $\mathbf{x}^{[k]}$ to the coefficient matrix $\mathbf{M}$. This is called the residual and is defined as shown in Eq. (36):

$$\mathbf{r}^{[k]} \triangleq \mathbf{b} - \mathbf{M}\mathbf{x}^{[k]} \tag{36}$$

Given the above definition performance metrics similar to those in Eqs. (30), (32), (33), and (34) can be constructed to assess the convergence of the iterative method in the range of $\mathbf{M}$ through $\mathbf{r}^{[k]}$ as opposed to the domain of $\mathbf{M}$ which is evaluated through $\mathbf{x}^{[k]}$. To evaluate how a change in the range of $\mathbf{M}$ is reflected in its domain the recursive nature of Eq. (10) will be used as follows in Eq. (37).

$$\begin{aligned}\mathbf{r}^{[k]} - \mathbf{r}^{[k-1]} &= -\mathbf{M}\left(\mathbf{x}^{[k]} - \mathbf{x}^{[k-1]}\right)\\ \Rightarrow \left\|\mathbf{r}^{[k]} - \mathbf{r}^{[k-1]}\right\|_2 &\leq \|\mathbf{M}\|_2\left\|\mathbf{x}^{[k]} - \mathbf{x}^{[k-1]}\right\|_2\end{aligned} \tag{37}$$

Thus, convergence to a stationary point also drives the difference in the range of $\mathbf{M}$ to zero, provided that $\|\mathbf{M}\|_2$ is finite. Therefore, the nullspace of $\mathbf{M}$ must have a dimension of zero, meaning in the context of this work that there are no isolated components in the original graph topology and so each vertex contributes to the proposed trust model of Eq. (8). Moreover, the lower its spectral radius is, the tighter the above bound is. The role of the spectrum of $\mathbf{M}$ to the convergence of the stationary point methods becomes clearer in light of Eq. (38):

$$\|\mathbf{M}\|_2 \triangleq \frac{\max\{|\lambda|, \lambda \in \Lambda\}}{\min\{|\lambda|, \lambda \in \Lambda\}} = \frac{\operatorname{radius}(\mathbf{M})}{\min\{|\lambda|, \lambda \in \Lambda\}} \tag{38}$$

Eq. (38) is in fact a special form since $\mathbf{M}$ is square and real. Also recall that in Eq. (38) Λ is the spectrum of $\mathbf{M}$ defined as in Eq. (39):

$$\Lambda \triangleq \{\lambda, \mathbf{Mg} = \lambda\mathbf{g}\} \tag{39}$$

In any case, since there is an equivalence up to a constant to the decay rate in the domain and the range of $\mathbf{M}$, then it suffices to monitor only one of these spaces and not both. This also justifies the termination criteria presented earlier.

Regarding GMRES its convergence to the exact solution $\mathbf{s}$ depends heavily on the order the Arnoldi iteration discovers the basis vectors for the Krylov space. Since the optimal order cannot be known in advance, keeping track of the residual and monitoring its magnitude is a feasible termination criterion. For matrices with clustered eigenvalues in practice the convergence is satisfactory and requires much less than n iterations. Also since n is typically a very large number of steps with a prohibitively computational cost, a function sublinear in n is frequently selected for $T(n)$. In this article the specific choice was that of Eq. (40).

$$T(n) = \left\lceil n^{\frac{2}{3}} \right\rceil \tag{40}$$

Other than that, the performance metrics for GMRES are those described earlier for the stationary point methods.

In Figure 1 the spectrum of $\mathbf{G}$ of Eq. (8) is given in logarithmic scale. Usually such information is unavailable and has to be estimated by system designers. From

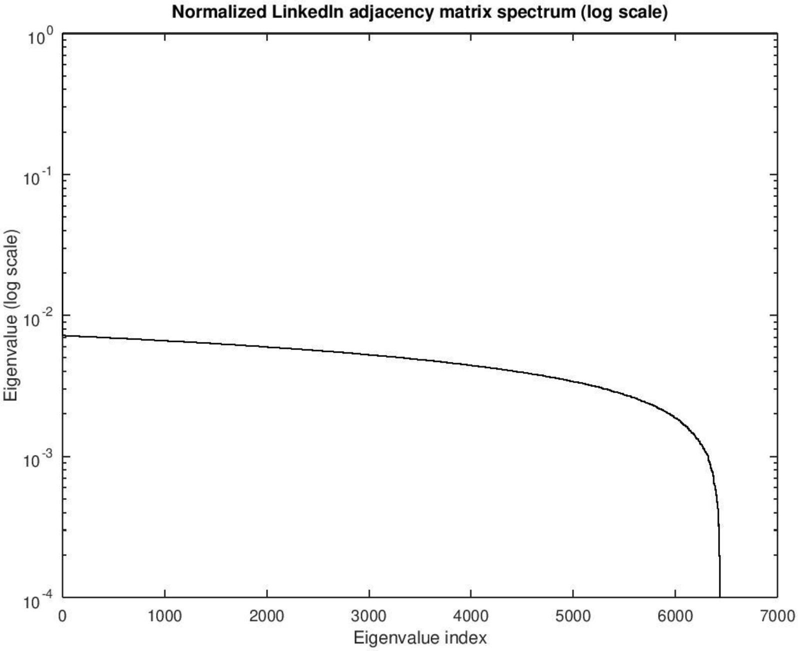

Fig. 1. Spectrum of $\mathbf{G}$ (log scale).

the eigenvalue decay rate it can be seen that after the maximum eigenvalue there is a big gap with the second largest eigenvalue being almost two orders of magnitude lower. This indicates a graph which can be significantly densified, since the difference between these eigenvalues is a metric of graph expansion potential and a heuristic for the respective Cheeger (or isoperimetric) number. Moreover, the heavy tail indicates there is a considerable number of eigenvalues in the region between 10^{-2} and 10^{-3} with a small fraction of eigenvalues below 10^{-3}. This is in accordance with the observation that there is a connected heavy mass and a relatively small periphery.

4.3. *Ranking properties*

The accuracy of the proposed ranking methodology will be the same as with that of the baseline methodology, namely that it will be defined as the overall ratio I_0 of the number of employees e'_k found to the total number of employees e_k who have a LinkedIn profile for the kth startup.[9] Assuming v_s startups exist in the dataset, Eq. (41) gives the definition for I_0. Its probabilistic properties have been examined in the work introducing the baseline methodology. For fair comparison reasons I_0 will be used here as well to evaluate ranking accuracy.

$$I_0 \triangleq \frac{1}{v_s} \sum_{k=1}^{v_s} \frac{e'_k}{e_k} \tag{41}$$

The resulting rankings are of interest not only in terms of accuracy but also of robustness. The latter means that small numerical perturbations in the process are not expected to have significant effects on the ranking itself. One way to see this is based on the observation that both the baseline and the proposed methodology yield respectively the rankings $\mathbf{s}_b$ and $\mathbf{s}_p$ whose logarithmic scree plot, namely the plot of the frequency of the components of $\mathbf{s}_b$ and $\mathbf{s}_p$ versus their ranking is close to a straight line as shown in Fig. 2. This strongly implies a Zipf or power law type ranking where the ith candidate has a score as shown in Eq. (42):

$$\mathbf{r}_b\left[i\right] = \alpha_b i^{-\gamma_b} \qquad \text{and} \qquad \mathbf{r}_p\left[i\right] = \alpha_p i^{-\gamma_p} \tag{42}$$

In Eq. (42) the normalizing constants ensure that the elements of the respective rankings add to one. The exponents play a crucial role as they determine the decay rate with a bigger exponent denoting a sharper decay and, thus, a more robust ranking for the top elements since the distance between any two consecutive ranking scores is also bigger. This situation is reversed for the bottom elements which are cramped at the ranking tail but this effect is of no concern here.

The two power law models of Eq. (42) are fit to the respective actual sorted scores $\{(i, \mathbf{s}_b\left[i\right])\}$ and $\{(i, \mathbf{s}_p\left[i\right])\}$. Since the procedure is virtually identical, the subscripts from the rankings will be dropped in the following analysis.

The Levenberg-Marquardt algorithm used in this work is described in Algorithm 3. Notice that the parameter vector $\mathbf{t}$ is updated in each iteration by solving a linear system in order to compute a correction vector $\delta\mathbf{t}^{[k]}$ added to it.

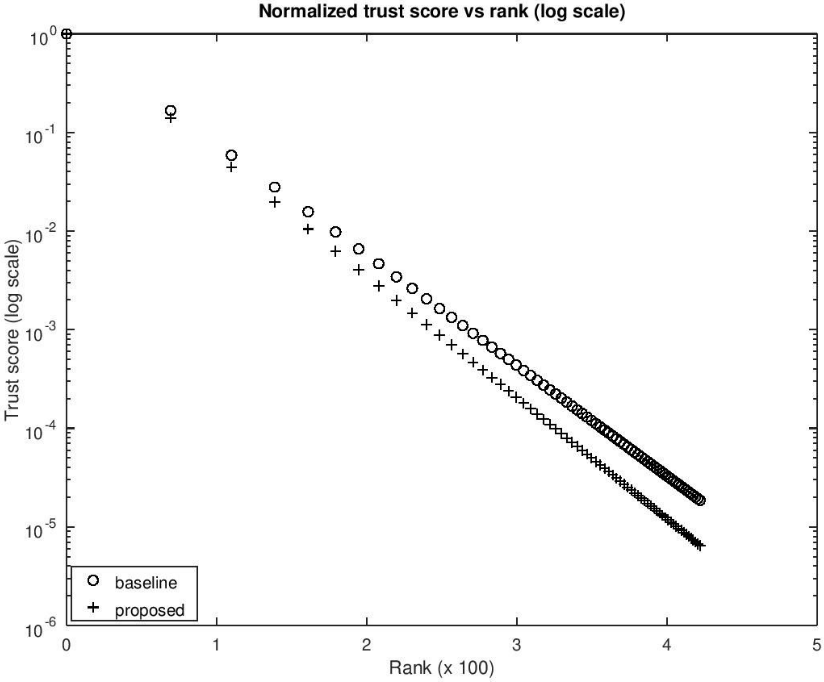

Fig. 2. Trust scoring (log scale).

Algorithm 3 Levenberg-Marquardt algorithm for power law fitting

```
Require: Power law model $r$, scoring $\mathbf{s}$, and regularization parameter $\lambda_0$
Ensure: A power law model is fit to the scores
generate the starting point $\mathbf{t}$
repeat
    construct $\left(\mathbf{J}^T\mathbf{J} + \lambda_0 \operatorname{diag}\left(\mathbf{J}^T\mathbf{J}\right)\right)^{-1}$ as in Eq. (48) and $\mathbf{J}^T(\mathbf{s}-\mathbf{r})$ as in Eq. (47)
    solve regularized system of Eq. (46) to compute $\delta\mathbf{t}^{[k]}$
    update parameter vector $\mathbf{t} = \mathbf{t}^{[0]} + \delta\mathbf{t}^{[k]}$
until $\left\|\delta\mathbf{t}^{[k]}\right\|_2$ drops under $\xi_2$
return updated version of $\mathbf{t}$
```

The vector $\mathbf{t}^{[0]}$ contains the initial values for the two parameters to be estimated $\alpha^{[k]}$ and $\gamma^{[k]}$ is defined as in Eq. (43). Also let $\delta\mathbf{t}$ be the current value estimates and $\mathbf{t}^{[k]}$ be the correction computed during the kth iteration.

$$\mathbf{t}^{[0]} \triangleq \begin{bmatrix}\alpha^{[0]} & \gamma^{[0]}\end{bmatrix}^T \tag{43}$$

At each iteration once $\delta\mathbf{t}^{[k]}$ is computed the estimated is updated as in Eq. (44).

$$\mathbf{t} = \mathbf{t}^{[0]} + \delta\mathbf{t}^{[k]} \tag{44}$$

The Jacobian matrix for the parameters α and γ is given in Eq. (45). By construction each row contains the first order partial derivative of the model to be fit to the actual measurements evaluated at each of the data point abscissas i, $1 \leq i \leq n$.

$$\mathbf{J} \triangleq \begin{bmatrix} \left.\frac{\partial \mathbf{r}}{\partial \alpha}\right|_{i=1} & \cdots & \left.\frac{\partial \mathbf{r}}{\partial \alpha}\right|_{i=n} \\ \left.\frac{\partial \mathbf{r}}{\partial \gamma}\right|_{i=1} & \cdots & \left.\frac{\partial \mathbf{r}}{\partial \gamma}\right|_{i=n} \end{bmatrix} = \begin{bmatrix} i^{-\gamma}\,|_{i=1} & \cdots & i^{-\gamma}\,|_{i=n} \\ -\gamma\alpha i^{-\gamma-1}\,|_{i=1} & \cdots & -\gamma\alpha i^{-\gamma-1}\,|_{i=n} \end{bmatrix} \tag{45}$$

Based on Eq. (45) the first order approximation $\mathbf{J}^T\mathbf{J}$ of the true Hessian matrix is constructed by evaluating the Jacobian matrix of Eq. (45) at the latest respective values of $\mathbf{t}$. Recall that the Hessian matrix contains the second derivatives of $\mathbf{r}$ with respect to the two parameters α and γ evaluated at the data points absicssas i and it is the computational heart of the multiparameter Gauss-Newton method.

At each iteration the linear system of Eq. (46) is solved in order to compute $\delta\mathbf{t}^{[k]}$. The first order matrix is augmented with a diagonal correction which amounts to multiplying diagonal elements by $1 + \lambda_0$. Typically λ_0 is either a small positive constant or a decaying positive variable. In this work the former option was used. Although this regularization strategy, similar to those used in ill conditioned inverse problems,[52] leads to an inexact solution, the system is more stable numerically.

$$\left(\mathbf{J}^T\mathbf{J} + \lambda_0 \operatorname{diag}\left(\mathbf{J}^T\mathbf{J}\right)\right)\delta\mathbf{t}^{[k]} = \mathbf{J}^T(\mathbf{s} - \mathbf{r}) \tag{46}$$

In Eq. (46) during each iteration the Jacobian matrix $\mathbf{J}$ has to be reconstructed using the latest parameter values stored in $\mathbf{t}$. This is tantamount to moving the search to a new point closer to the true parameter location and computing a new update. The vectors $\mathbf{s}$ and $\mathbf{r}$ contain respectively the actual ranking scores sorted in descending order and the predicted values from the power law model.

The right hand side of Eq. (46) has the elements shown in Eq. (47). They are approximations of the total differential of r for the respective parameters.

$$\mathbf{J}^T(\mathbf{s} - \mathbf{r}) = \begin{bmatrix} \sum_{i=1}^{n} (\mathbf{s}[i] - \mathbf{r}[i]) \left.\frac{\partial r}{\partial \alpha}\right|_i & \sum_{i=1}^{n} (\mathbf{s}[i] - \mathbf{r}[i]) \left.\frac{\partial r}{\partial \gamma}\right|_i \end{bmatrix}^T \tag{47}$$

Note that r refers to the model to be fit, which is a differentiable function, whereas $\mathbf{r}$ is the vector of the values of r at the abscissas i, $1 \leq i \leq n$.

In this particular case the coefficient matrix of Eq. (46) is 2 × 2. Therefore, its inverse $\mathbf{\Delta}$ is given by the closed form of Eq. (48). The upper part is the formula for the normalizing factor δ^*, whereas the lower part is the formula for $\mathbf{\Delta}$ itself.

$$\delta^* = \frac{1}{(1+\lambda_0)^2 \sum_{i=1}^{n} \left(\left.\frac{\partial r}{\partial \alpha}\right|_i\right)^2 \sum_{i=1}^{n} \left(\left.\frac{\partial r}{\partial \gamma}\right|_i\right)^2 - \left(\sum_{i=1}^{n} \left.\frac{\partial r}{\partial \alpha}\right|_i \left.\frac{\partial r}{\partial \gamma}\right|_i\right)^2}$$

$$\boldsymbol{\Delta} = \delta^* \begin{bmatrix} (1+\lambda_0)\sum_{i=1}^{n} \left(\left.\frac{\partial r}{\partial \gamma}\right|_i\right)^2 & -\sum_{i=1}^{n} \left.\frac{\partial r}{\partial \alpha}\right|_i \left.\frac{\partial r}{\partial \gamma}\right|_i \\ -\sum_{i=1}^{n} \left.\frac{\partial r}{\partial \alpha}\right|_i \left.\frac{\partial r}{\partial \gamma}\right|_i & (1+\lambda_0)\sum_{i=1}^{n} \left(\left.\frac{\partial r}{\partial \alpha}\right|_i\right)^2 \end{bmatrix} \tag{48}$$

This iterative process terminates when the magnitude of the correction vector in the kth iteration $\left\|\delta \mathbf{t}^{[k]}\right\|_2$ drops under a threshold ξ_2.

Table 4 has the exponent values γ for the baseline and proposed methods.

4.4. *Results and discussion*

In Table 4 are presented the values for the performance metrics discussed in the remainder of this Section. The results about the baseline method are the best scores for the respective metrics. Since the baseline counts steps in a much different way than the iterations of the three methods presented here, the comparison will be only in terms of accuracy I_0 of Sec. 4.2 and the ranking exponent γ of Sec. 4.3.

From the entries of Table 4 it follows that the higher order method clearly outperforms the first order one. This is attributed to the added information which is available to the former through the adjacency matrix of the dataset. The relatively small increase in the accuracy I_0 can be attributed to the fact that the baseline method starts from a high point. Moreover, since all three iterative methods solve the same linear system, it follows that they result in the same value for the exponent γ. The latter is lower for the proposed methodology, indicating a sharper ranking decay and consequently a clear difference between top ranking profiles and the rest. Notice that the ranking is derived from Eq. (1) for the proposed methodology. In all cases the starting vector $\mathbf{s}^{[0]} = \mathbf{1}_n$ was used which denotes a uniform initial ranking as there is no a priori reason to reward or demerit any profiles. As it was

Table 4. Results.

	Baseline[9]	Jacobi	Gauss-Seidel	GMRES
Accuracy I_0	0.8739	0.9311	0.9311	0.9311
Exponent γ	−2.5814	−2.8322	−2.8322	−2.8322
Iterations	N/A	124	103	79
$\Delta\vartheta_c$	N/A	58.13°	58.11°	35.6°
$\Delta\vartheta_r$	N/A	32.11°	28.42°	18.92°
Δu	N/A	8.3321	9.5217	14.3331
Δa	N/A	12.8992	17.3333	25.5023

seen in the preceding analysis, the selection of the starting point does not influence the convergence of the iterative methods used in this work.

Concerning the comparison of the three iterative methods, all the performance metrics favor GMRES with Gauss-Seidel being a very distant second with performance almost similar to that of Jacobi. On average the candidate solutions of the stationary point methods have a slower and more convoluted trajectory as denoted by low speed and acceleration in conjunction with wider turns as indicated by the larger angles. On the contrary, GMRES yields a smoother trajectory. This can be attributed to the fact that GMRES progressively constructs a solution during each iteration which is at least as good as the previous one since it finds the best projection to a new basis vector of the Krylov space containing the true solution.

In general, algorithmic efficiency plays a central role in large scale problems such as this one. For the class of the iterative algorithms the total complexity is divided to two possibly competing factors, namely the total number of iterations and the complexity per iteration. The cases of the Jacobi and GMRES methods are very insightful regarding on how the two above factors are interrelated. In GMRES each iteration is a linear least squares problem of linearly increasing dimensions. Their special structure allows them to be solved in an number of FPOs comparable to those of the Jacobi iteration. However, because of the different strategy of GMRES each correction leads closer to the exact solution as denoted by the lower number of iterations of Table 4. Thus, GMRES in this context utilizes more efficiently the available computational resources to find a better candidate solution.

The proposed approach requires only one hyperparameter ρ_0 which determines the relative importance between the implicit and the explicit parts of the individual trust component in Eq. (5). Since there is no evidence suggesting either should be preferred, ρ_0 was set to 1 so that both components have the same weight. Additionally, this was the value used in the experiments for the baseline methodology, contributing in this way to a fair comparison. The latter requires two more hyperparameters, namely ρ_1 and α_0, to give weights to competing factors elsewhere in the decision algorithm. Therefore, the proposed methodology has less complexity when it comes to hyperparameters, making it thus easier to understand.

The baseline method besides being a first order one was also based on the implicit assumption that the profiles of trusted candidates would be clustered so they can be efficiently reached with short sequences of local jumps. A substantial amount of empirical evidence appears to corroborate the working assumption that social graphs tend to densify over time[53] which in the LinkedIn case can be attributed to invitations being frequently accepted out of professional courtesy or best professional practices. Therefore, professional with similar skillsets are often connected, favoring thus local searches. On the other hand, the proposed method benefits from a dense graph as it has more information for each profile from neighboring ones.

Concerning the limitations of the proposed methodologies, perhaps the most basic one is the question whether trust composition is linear. Even if trust is shown to be composed non-linearly, a similar framework for systems of non-linear equations can be designed. Moreover, since social graphs tend to be frequently updated incrementally, inserting new profiles is bound to influence trust ranking. Although neither the baseline nor the proposed methods have been designed with that in mind, they can be naturally extended. The baseline method can visit new profiles, whereas the iterative methods can use the existing trust scoring as a starting point.

5. Conclusions and Future Work

The focus of this article is the extension of first order trust metrics, as computed by readily available attributes of LinkedIn, to higher order ones. The key for the latter is a system of equations based on the assumption that trust for a given LinkedIn profile can be linearly composed from the trust values assigned to neighboring ones. Additionally, three iterative methods, namely the Jacobi, the Gauss-Seidel, and the GMRES, are used to compute the trust scores from the proposed metric. The performance of these methods has been assessed with metrics including the number of iterations, the average velocity and angle of the trajectory of the candidate solutions, and the exponent of the ranking decay. Moreover, a tailored Levenberg-Marquardt algorithm was used to fit a power law to the ranking derived by the baseline and the proposed methodologies through a sequence of regularized linear systems.

Possible research directions include extending tests to more benchmark LinkedIn graphs with various characteristics such as varying density, diameter, or average degree. Moreover, trust composition models warrant further investigation based on findings from social media analysis, computer science, and cognitive sciences. Specifically, the homophily connectivity patterns between trusted profiles can serve as a starting point. The application of eigenvalue computation algorithms to the trust computation is another possible option. The inclusion of more LinkedIn attributes to the proposed trust metric should finally be researched.

Acknowledgments

This research has been funded by Research Initiative Fund Grant (RIF) 18056, Zayed University, UAE.

References

1. S. Kaushik and C. Gandhi, Multi-level trust agreement in cloud environment, in *IC3* (IEEE, 2019), pp. 1–5.
2. W. D. Du and J.-Y. Mao, Developing and maintaining clients' trust through institutional mechanisms in online service markets for digital entrepreneurs: A process model, *The Journal of Strategic Information Systems* **27**(4) (2018) 296–310.
3. Y. Madani, J. Bengourram and M. Erritali, Social login and data storage in the big data file system HDFS, in *Int. Conf. on Compute and Data Analysis* (IEEE, 2017), pp. 91–97.

4. C. Jacomme and S. Kremer, An extensive formal analysis of multi-factor authentication protocols, *TOPS* **24**(2) (2021) 1–34.
5. H. R. Conte and R. Plutchik, A circumplex model for interpersonal personality traits, *Journal of Personality and Social Psychology* **40**(4) (1981) 701.
6. P. Ekman and W. V. Friesen, Constants across cultures in the face and emotion, *Journal of Personality and Social Psychology* **17**(2) (1971) 124.
7. A. Egges, S. Kshirsagar and N. Magnenat-Thalmann, A model for personality and emotion simulation, in *Int. Conf. on Knowledge-Based and Intelligent Information and Engineering Systems* (Springer, 2003), pp. 453–461.
8. Z.-T. Liu, Q. Xie, M. Wu, W.-H. Cao, Y. Mei and J.-W. Mao, Speech emotion recognition based on an improved brain emotion learning model, *Neurocomputing* **309** (2018) 145–156.
9. G. Drakopoulos, E. Kafeza, P. Mylonas and H. Al Katheeri, Building trusted startup teams from LinkedIn attributes: A higher order probabilistic analysis, in *ICTAI* (IEEE, 2020), pp. 867–874.
10. M. Kuciapski, P. Lustofin and P. Soja, Examining the role of trust and risk in the Software-as-a-Service adoption decision, in *Hawaii Int. Conf. on System Sciences* (IEEE, 2021).
11. X. Gong, K. Z. Zhang, C. Chen, C. M. Cheung and M. K. Lee, What drives trust transfer from web to mobile payment services? The dual effects of perceived entitativity, *Information & Management* **57**(7) (2020).
12. G. Deepak, B. Shwetha, C. Pushpa, J. Thriveni and K. Venugopal, A hybridized semantic trust-based framework for personalized Web page recommendation, *International Journal of Computers and Applications* **42**(8) (2020) 729–739.
13. J. Strömbäck, Y. Tsfati, H. Boomgaarden, A. Damstra, E. Lindgren, R. Vliegenthart and T. Lindholm, News media trust and its impact on media use: Toward a framework for future research, *Annals of the International Communication Association* **44**(2) (2020) 139–156.
14. G. Lăzăroiu, O. Neguriţă, I. Grecu, G. Grecu and P. C. Mitran, Consumers' decision-making process on social commerce platforms: Online trust, perceived risk, and purchase intentions, *Frontiers in Psychology* **11** (2020).
15. C. Brogan and J. Smith, *Trust Agents: Using the Web to Build Influence, Improve Reputation, and Earn Trust* (John Wiley & Sons, 2020).
16. K. H. Kwon, C. Shao and S. Nah, Localized social media and civic life: Motivations, trust, and civic participation in local community contexts, *Journal of Information Technology & Politics* **18**(1) (2021) 55–69.
17. S. E. Chang, A. Y. Liu and W. C. Shen, User trust in social networking services: A comparison of Facebook and LinkedIn, *Computers in Human Behavior* **69** (2017) 207–217.
18. E. W. Ayaburi and D. N. Treku, Effect of penitence on social media trust and privacy concerns: The case of Facebook, *International Journal of Information Management* **50** (2020) 171–181.
19. Y. Ruan, A. Durresi and L. Alfantoukh, Using Twitter trust network for stock market analysis, *Knowledge-Based Systems* **145** (2018) 207–218.
20. B. Rath, W. Gao, J. Ma and J. Srivastava, From retweet to believability: Utilizing trust to identify rumor spreaders on Twitter, in *Int. Conf. on Advances in Social Networks Analysis and Mining* (IEEE/ACM, 2017), pp. 179–186.
21. G. Lingam, R. R. Rout and D. Somayajulu, Detection of social botnet using a trust model based on spam content in Twitter network, in *ICIIS* (IEEE, 2018), pp. 280–285.

22. I. Pentina, L. Zhang and O. Basmanova, Antecedents and consequences of trust in a social media brand: A cross-cultural study of Twitter, *Computers in Human Behavior* **29**(4) (2013) 1546–1555.
23. M. J. Park, D. Kang, J. J. Rho and D. H. Lee, Policy role of social media in developing public trust: Twitter communication with government leaders, *Public Management Review* **18**(9) (2016) 1265–1288.
24. F. Riquelme and P. González-Cantergiani, Measuring user influence on Twitter: A survey, *Information Processing & Management* **52**(5) (2016) 949–975.
25. D. Drakopoulos, K. C. Giotopoulos, I. Giannoukou and S. Sioutas, Unsupervised discovery of semantically aware communities with tensor Kruskal decomposition: A case study in Twitter, in *SMAP* (IEEE, 2020).
26. G. Drakopoulos, Y. Voutos, P. Mylonas and S. Sioutas, Motivating item annotations in cultural portals with UI/UX based on behavioral economics, in *IISA* (IEEE, 2021).
27. G. Drakopoulos, E. Kafeza, P. Mylonas and S. Sioutas, A graph neural network for fuzzy Twitter graphs, in *CIKM Companion Volume*, eds. G. Cong and M. Ramanath **3052** (CEUR-WS.org, 2021).
28. C. Ye, C. F. Hofacker, J. Peloza and A. Allen, How online trust evolves over time: The role of social perception, *Psychology & Marketing* **37**(11) (2020) 1539–1553.
29. L. Fell, A. Gibson, P. Bruza and P. Hoyte, Human information interaction and the cognitive predicting theory of trust, in *Conf. on Human Information Interaction and Retrieval* (ACM, 2020), pp. 145–152.
30. H. Huang, J. Lin, B. Zheng, Z. Zheng and J. Bian, When blockchain meets distributed file systems: An overview, challenges, and open issues, *IEEE Access* **8** (2020) 50574–50586.
31. G. De Roode, I. Ullah and P. J. Havinga, How to break IOTA heart by replaying?, in *IEEE Globecom Workshops* (IEEE, 2018), pp. 1–7.
32. S. Popov and Q. Lu, IOTA: Feeless and free, *IEEE Blockchain Technical Briefs* (2019).
33. M. Mirkin, Y. Ji, J. Pang, A. Klages-Mundt, I. Eyal and A. Juels, BDoS: Blockchain Denial-of-Service, in *Computer and Communications Security* (ACM, 2020), pp. 601–619.
34. G. Drakopoulos, E. Kafeza and H. Al Katheeri, Proof systems in blockchains: A survey, in *SEEDA-CECNSM* (IEEE, 2019).
35. Q. Zhou, H. Huang, Z. Zheng and J. Bian, Solutions to scalability of blockchain: A survey, *IEEE Access* **8** (2020) 16440–16455.
36. P. C. Bartolomeu, E. Vieira and J. Ferreira, IOTA feasibility and perspectives for enabling vehicular applications, in *IEEE Globecom Workshops* (IEEE, 2018), pp. 1–7.
37. A. Raschendorfer, B. Mörzinger, E. Steinberger, P. Pelzmann, R. Oswald, M. Stadler and F. Bleicher, On IOTA as a potential enabler for an M2M economy in manufacturing, *Procedia CIRP* **79**(1) (2019) 379–384.
38. J. Frizzo-Barker, P. A. Chow-White, P. R. Adams, J. Mentanko, D. Ha and S. Green, Blockchain as a disruptive technology for business: A systematic review, *International Journal of Information Management* **51** (2020).
39. U. Bodkhe, S. Tanwar, K. Parekh, P. Khanpara, S. Tyagi, N. Kumar and M. Alazab, Blockchain for Industry 4.0: A comprehensive review, *IEEE Access* **8** (2020) 79764–79800.
40. S. Tanwar, K. Parekh and R. Evans, Blockchain-based electronic healthcare record system for healthcare 4.0 applications, *Journal of Information Security and Applications* **50** (2020).
41. Z. Liu, Z. Li, C. Ferreira and Y. Zhang, Stationary splitting iterative methods for the matrix equation $AXB = C$, *Applied Mathematics and Computation* **378** (2020).

42. K. Nouri, S. P. A. Beik, L. Torkzadeh and D. Baleanu, An iterative algorithm for robust simulation of the Sylvester matrix differential equations, *Advances in Difference Equations* **2020**(1) (2020) 1–13.
43. M. Dehghan and A. Shirilord, Matrix multisplitting Picard-iterative method for solving generalized absolute value matrix equation, *Applied Numerical Mathematics* **158** (2020) 425–438.
44. J. Liu, X. Wang, Z. Chen, Y. Tan, X. Li, Z. Zhang and L. Wang, An iterative method for identifying essential proteins based on non-negative matrix factorization, *IEEE Access* **8** (2020) 226685–226696.
45. P. Li, J. Xie, Q. Wang and Z. Gao, Towards faster training of global covariance pooling networks by iterative matrix square root normalization, in *CVPR* (IEEE, 2018), pp. 947–955.
46. M. Albreem, M. Juntti and S. Shahabuddin, Efficient initialisation of iterative linear massive MIMO detectors using a stair matrix, *Electronics Letters* **56**(1) (2019) 50–52.
47. X. Liu, X. Zhang, J. Xiong, F. Gu and J. Wei, An enhanced iterative clipping and filtering method using time-domain kernel matrix for PAPR reduction in OFDM systems, *IEEE Access* **7** (2019) 59466–59476.
48. M. J. Priddin, A. V. Kisil and L. J. Ayton, Applying an iterative method numerically to solve $n \times n$ matrix Wiener–Hopf equations with exponential factors, *Philosophical Transactions of the Royal Society A* **378**(2162) (2020) 20190241.
49. L. Wei, J. Wu, C. Long and B. Li, On designing context-aware trust model and service delegation for social Internet of Things, *IEEE Internet of Things Journal* (2020).
50. G. Drakopoulos, Y. Voutos and P. Mylonas, Annotation-assisted clustering of player profiles in cultural games: A case for tensor analytics in Julia, *BDCC* **4**(4) (2020).
51. G. Drakopoulos, I. Giannoukou, P. Mylonas and S. Sioutas, A graph neural network for assessing the affective coherence of Twitter graphs, in *IEEE Big Data* (IEEE, 2020), pp. 3618–3627.
52. D. Smyl, T. Tallman, D. Liu and A. Hauptmann, An efficient quasi-Newton method for nonlinear inverse problems via learned singular values, *IEEE Signal Processing Letters* (2021).
53. T. Kobayashi and M. Génois, The switching mechanisms of social network densification, *Scientific reports* **11**(1) (2021) 1–11.