Chapter 3
Analysis of Social Networks by Tensor Decomposition

Sergej Sizov, Steffen Staab, and Thomas Franz

3.1 Motivation, or Who Follows Whom

Authority ranking is a crucial component for a wide range of Social Web applications, such as thematically focused, faceted browsing or contact recommendations. Online communities (such as Twitter or Facebook) provide very limited statistics about user relations, such as the number of contacts in the users' contact list, the number of registered observers for user postings (coined "followers" on Twitter), etc. Although these counts appropriately reflect the overall users' centrality/popularity in the social community, their interpretation in a more focused context becomes very difficult. As a realistic example for this problem, we may consider two real and quite popular users in the Twitter community: *timberners_lee* (20K followers) and *parishilton* (1.7 Mio followers). In early 2010, both users have commented the launch of the novel Apple iPad technology – at the same time – by following postings (tweets), as shown in Fig. 3.1.

From the purely calculational perspective, we could draw the conclusion that the user *parishilton* is a stronger twitter authority in the sense of the topic "new technologies" than *timberners_lee*. However, deeper analysis of postings from both users immediately shows the opposite. While contributions of *timberners_lee* are clearly focused on novel research/technology aspects (like Web Science, Linked Open Data, Semantic Web), the user *parishilton* is rather devoted to themes like "celebrities," "lifestyle" or "my person."

The need for better, explicit contextualization of contributions led to various, platform-specific mechanisms, and self-organizing, emerging vocabulary extensions. The best-known form of simple content contextualization is tagging, i.e., use of short text snippets (or particular terms) for content annotation – widely known from social cites like Flickr, YouTube, Bibsonomy, and others. In the previously discussed Twitter community, a similar approach with so-called hashtags (context-indicative, characteristic words within textual content, marked by a preceding hash

S. Sizov (✉)
University of Koblenz-Landau, WeST – Institute for Web Science and Technologies
e-mail: sizov@uni-koblenz.de

B. Furht (ed.), *Handbook of Social Network Technologies and Applications*,
DOI 10.1007/978-1-4419-7142-5_3, © Springer Science+Business Media, LLC 2010

timberners_lee	*parishilton*
Following: 59	Following: 272
Followers: 20.692	Followers: 1.709.116
RT @janl: Apple:	I Love my new I-Pad.
“Preparing your web content for iPad:	So much fun!
2. Use W3C standard web technologies.” #w3c	Technology rocks!
9:46 AM Mar 21st via TweetDeck	about 2 h ago via UberTwitter

Fig. 3.1 Real postings of sample users on Twitter

sign) became quite popular. For instance, the term #*w*3*c* in the running text from Fig. 3.1 is such a hashtag. In this particular case, the community established the practice to use #*w*3*c* for indicating posting relatedness to the W3C consortium – this is a characteristic example of emergent semantics in modern social networks [10].

The baseline scenario for lightweight contextualization of user *relations* may be organized in a quite analogous manner. In particular, we may assume that some users of the social network explicitly indicate their interest on some topic (say one hashtag h). In the following, we will refer to such group of users as a h-candidate set. Authority ranking for h-candidate sets can be quite similar to the “unfocused” global setting discussed before and based on “contextualized” lists of followers. From the conceptual perspective, this approach can be seen as a special case of collaborative voting “focused” on h. In fact, we can restrict the scope of follower-lists and virtually remove all user entries that are *not* explicitly found in the h-candidate set (i.e., appear “irrelevant” in the context of h). In other words, we simply reduce the follower lists to contacts from the h-candidate set itself. The resulting cardinalities of reduced follower lists give a natural ranking for contact recommendations (whom-to-follow) in the context of the considered hashtag h.

For the discussed Twitter scenario, this functionality is really offered by an external service provider wefollow.com. During registration, users are requested to specify keywords of interest (i.e., hashtags, in our terminology) as well as their account information (ensuring access to the list of followed users from their Twitter account) to the portal provider. Subsequently, “keyword subscribers” can access a ranked list of contact recommendations (i.e., whom-to-follow suggestions). Although the exact organization of wefollow.com is not public, the generated list of recommendations can be reproduced with extremely high accuracy using h-candidate sets. Table 3.1 shows the top-20 recommendations regarding keyword *semanticweb* by wefollow.com and by collaborative voting (in the latter case, the h-candidate set of Twitter users is identical to the list of wefollow.com subscribers for *semanticweb*). Our collaborative voting implementation uses proprietary Twitter API for collecting further publicly available user information, such as follower lists. Minor differences in rank positions can be explained by the fact that Twitter profiles of few wefollow.com subscribers for *semanticweb*), namely 6 out of 242, were *not* open to public, i.e., corresponding lists of followed users were not accessible to our test application.

Table 3.1 Simple strategies for to-follow recommendations, h = #*semanticweb*

	WeFollow	h-set	#followers:		H-set	#followers:	
POS	recommends	recommends	total	in h-set	recommends	total	in H-set
1	tommyh	PaulMiller	2,215	82	timberners_lee	20,694	252
2	jahendler	jahendler	909	76	timoreilly	1,428,425	200
3	ivan_herman	tommyh	738	76	jahendler	910	185
4	PaulMiller	ivan_herman	680	69	LeeFeigenbaum	273	176
5	opencalais	opencalais	1,797	68	danbri	1,781	160
6	danja	danja	1,313	59	kidehen	1,806	159
7	CaptSolo	juansequeda	988	57	ivan_herman	680	153
8	juansequeda	CaptSolo	1,224	52	PaulMiller	2,216	150
9	sclopit	gothwin	685	50	tommyh	737	142
10	gothwin	robocrunch	3,679	49	novaspivack	7,896	139
11	robocrunch	alexiskold	4,321	48	johnbreslin	1,933	128
12	kristathomas	kristathomas	1,499	48	w3c	10,672	128
13	kendall	kendall	1,694	45	mimasnews	198	127
14	bobdc	andraz	2,065	44	iand	1,094	123
15	phclouin	sclopit	513	42	rww	1,045,511	123
16	brown2020	cjmconnors	466	39	terraces	632	119
17	alexiskold	gkob	399	37	mhausenblas	449	118
18	andraz	dorait	2,541	35	opencalais	1,799	112
19	cjmconnors	phclouin	266	35	ldodds	621	110
20	ontoligent	openamplify	1,387	34	semanticnews	843	102

The apparent drawback of the proposed strategy is its limitation to proactive users that exploit the new service and explicitly "subscribe" for selected hashtags of interest. Unsubscribed users are not part of the h-candidate set and thus cannot be recommended, disregarding their (real) importance in the context of h. This limitation can be avoided with alternate strategies of constructing the candidate set. In particular, we can consider "active" users in the sense of h (e.g., by finding h in their recent postings). In the following, we will refer to a group of candidates that actively used the desired hashtag h within a certain timeframe (e.g., 4 weeks) as a H-candidate set. In practice, the H-candidate set can be directly obtained by keyword-based search (for h as query) through the common API of many social platforms. The second part of Fig. 3.1 demonstrates corresponding follow-recommendations for the hashtag h = #*semanticweb* on Twitter. It can be observed that the the H-candidate set provides substantially higher recall by including highly relevant users in the field of h = #*semanticweb* that have no explicit registration at wefollow.com.

From the conceptual perspective, the introduced strategies allow for mapping user relations in common Social Web applications onto graph structures, where nodes represent users and edges correspond to relations that link users to each other. Consequently, graph-based authority ranking algorithms known from Web retrieval, such as PageRank [7], HITS [17] or SALSA [20], can be adopted for the Social Web setting, too. Instead of ratings for Web pages they will then output ratings for users in a social environment, with respect to one or more criteria, e.g., hub and authority scores in HITS. These scores reflect the centrality/importance of particular users in the social network and thus can be exploited for relevance estimation, e.g., in contact/follower recommendation scenarios.

Two important observations can be made about the authority ranking for social graphs. On one hand, the computational models of standard algorithms for Web analysis only consider structural information, i.e., the connectivity of graph nodes. Additional link semantics, e.g., knowledge about different types of relations, is not used. On the other hand, there are many cases of overlapping, redundant, and conflicting vocabulary describing similar problems. Therefore, we may expect redundancies like the co-existence of different hashtags/themes with highly similar (or coherent) meaning, such as h_1 = *#semanticweb*, h_2 = *#RDF* and h_3 = *#ontology*. Common authority ranking algorithms provide no support for finding such groups of semantically coherent items.

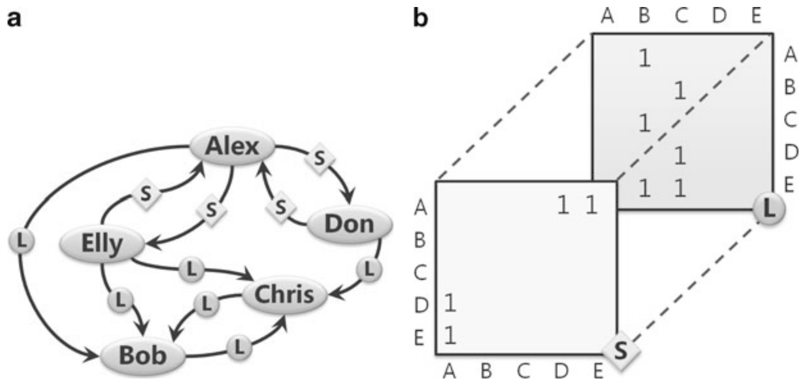
3.2 The Social Web as a Tensor

This section introduces the advanced TweetRank approach for authority ranking in Social Web communities. In doing so, we refine the formal notion of social graphs and tensors, introduce tensor factorization for faceted authority ranking, and show realistic examples of framework outputs.

3.2.1 The TweetRank Model

We define a Social Web graph as a graph $G = (V, L, E, \mathit{linkType})$ where V is the set of users in the community, L is the set of literals (e.g., hashtags), and E is the set of relations between users in V. Additionally, the function $\mathit{linkType} : E \rightarrow L$ returns the annotation from L that relates two users. Following our Twitter application scenario introduced in previous section, Fig. 3.2a shows an (over)simplified Social Web graph that contains five users (Alice = A, Bob = B, Chris = C, Don = D, Elly = E), two hashtag-like literals (*lifestyle* = L, *semanticweb* = S) and ten relations of two different types: *follow-lifestyle* and *follow-semanticweb*. The precise semantics of such links remains application-specific; in our sample case we assume that user X links to user Y by edge of type Z iff a) X follows Y (in the common sense of Twitter) and b) both X and Y have recently used the hashtag Z in their own postings/tweets. For instance, the graph expresses that Alex follows Bob regarding *lifestyle*.

We represent Social Web graphs by a 3-dimensional tensor **T** where each of its slices represents an adjacency matrix for one relation type from L. Figure 3.2b illustrates the tensor resulting from the transformation of the sample graph shown



Sample Social Web Graph

Tensor Representation

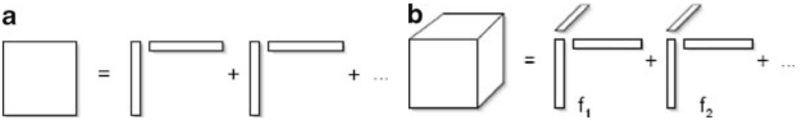
Fig. 3.2 Modeling example

in Fig. 3.2a. The first adjacency matrix $\mathbf{T}\,(:,:,1)$[1] models linkage by the property *semanticweb*. An entry >0 corresponds to the existence of a link by this property, empty entries are considered as zeroes. The second matrix $\mathbf{T}\,(:,:,2)$ models links by the property *lifestyle*. For instance, the fact that Alex follows Bob regarding *lifestyle* results in $\mathbf{T}\,(1,2,2) = 1$ in tensor representation.

3.2.2 PARAFAC for Authority Ranking

The Social Web graph can be described by an adjacency matrix. For a network graph matrix M the well known authority ranking methods like HITS [17] can be applied. HITS defines the authority ranking problem through mutual reinforcement between so-called hub and authority scores of graph nodes (community users, in our case). The authority (relevance) score of each node is defined as the sum of hub scores of its predecessors. Analogously, the hub (connectivity) score of each node is defined as a sum of the authority scores of its successors. By applying the singular value decomposition (SVD) to the adjacency matrix, we obtain hub and authority scores of graph nodes for each singular value of M, which can be interpreted as rankings regarding different themes or latent topics of interest. Formally, by this method, some arbitrary matrix $M \in \mathbf{R}^{k\times l}$ is splitted into three matrices $U \in \mathbf{R}^{k\times m}$, $S \in \mathbf{R}^{m\times m}$, $V \in \mathbf{R}^{l\times m}$. U and V represent the outlinks and inlinks with respect to the principal factor contained in S. Corresponding to our notation, M can be written as sum of rank-one-matrices by $M = \sum_{k=1}^{m} S^k \cdot U^k \circ V^k$. This 2-way decomposition yields authority and hub scores (cf. Fig. 3.3a) [18].

[1] Throughout this chapter we use the common Matlab-notation for addressing entries in tensors and vectors.



Matrix Decomposition [18] Tensor Decomposition [18]

Fig. 3.3 (**a**) Matrix decomposition (**b**) Tensor decomposition

Modeling several link types by separate matrices results in very sparse and not connected matrices. Instead, the tensor model applied by TweetRank enables the representation of all adjacency matrices including information about the connections between link types. Tensor decomposition methods like PARAFAC can then detect further hidden dependencies.

These methods are regarded as higher-order equivalents to matrix decompositions. The PARAFAC tensor decomposition has the advantage of robustness and computational efficiency. These advantages are due to its uniqueness up to scaling and permutation of the produced component matrices [14]. By PARAFAC input tensors are transformed to so called Kruskal tensors, a sum of rank-one-tensors. Consequently, in TweetRank we derive authority and hub scores for particular latent aspects (topics) of the analyzed data from particular rank-one-tensors of the decomposition. In the context of this chapter we focus on three-mode-tensors that represent connectivity between graph nodes (users) together with semantics of user relations.

Formally, a tensor $\mathbf{T} \in \mathbf{R}^{k \times l \times m}$ is decomposed by n-Rank-PARAFAC into components matrices $U_1 \in \mathbf{R}^{k \times n}$, $U_2 \in \mathbf{R}^{l \times n}$, $U_3 \in \mathbf{R}^{m \times n}$ and n principal factors (pf) λ_i in descending order. Via these $\mathbf{T}$ can be written as a Kruskal tensor by $\mathbf{T} \approx \sum_{k=1}^{n} \lambda_k \cdot U_1^k \circ U_2^k \circ U_3^k$ where λ_k denotes the kth principal factor, U_i^k the kth column of U_i and $\circ$ the outer product [18]. U_i yields the ratio of the ith dimension to the principal factors. So, similar to SVD, PARAFAC derives hidden dependencies related to the pfs and expresses the dimensions of the tensor by relations to the pfs. Depending on the number of pfs PARAFAC decomposition can be loss-free. For a third-mode-tensor $\mathbf{T} \in \mathbf{R}^{k \times l \times m}$ a weak upper bound for this rank is known: $rank(\mathbf{T}) \leq \min\{kl, lm, km\}$ [18]. There is no proper way for estimating the optimal number of pfs for an appropriate decomposition but several indicators like residue analysis or core consistency exist [2]. The PARAFAC decomposition of a tensor derives authority and hub scores plus additional scores for the relevance of link types (cf. Fig. 3.3b). The tensor $\mathbf{T}$ in Sect. 3.2.1 combines information about who follows whom with our explanations of the follow-links. So the PARAFAC decomposition would yield U_1 with subject-pf relation, U_2 with object-pf relation and U_3 with property-pf relation. In other words U_1 keeps the hub scores as relevance of the users to the pfs, U_2 the authorities scores as relevance of the users to the pfs and U_3 scores of the relevance of literals (hashtags) to the pfs. In line with HITS the largest entry of U_1^1 corresponds to the best hub for the first pf and the largest entry of U_2^1 to the best authority.

Table 3.2 PARAFAC vs. HITS results

PARAFAC				HITS		indegree	
Score	Hashtag	Score	User	Score	User	Degree	User
Group 1				0.62	Chris	3	Bob
1.00	lifestyle	0.71	Bob	0.56	Bob	3	Chris
–	–	0.70	Chris	0.50	Alex	2	Alex
Group 2				0.16	Don	1	Don
1.00	semanticweb	0.70	Don	0.16	Elly	1	Elly
0.001	lifestyle	0.70	Alex				
–	–	0.10	Elly				

3.2.3 Ranking Example

Applying the above factorization and analysis to the graph illustrated by Fig. 3.2 yields the results shown by the first four columns of Table 3.2. Two groups are identified, one where the hashtag *lifestyle* has a high score, and one where *semanticweb* is scored highly. The authoritative resources for each group differ from each other. Bob and Chris have high scores with respect to *lifestyle*. Don and Alex are the top authorities with respect to *semanticweb*. The application of HITS results in the ranking shown by column 5 and 6. The HITS ranking corresponds to a ranking based on the indegrees of the resources. Notably, the rankings produced by the PARAFAC analysis are different from the HITS/indegree results as they provide rankings with respect to different knowledge aspects in the data.

3.3 Implementation

Having introduced the theoretical background behind TweetRank, we present the implementation into an applicable system in Java.[2] We describe the three core components of the TweetRank architecture, which encapsulate a 3-step process, namely (1) the collection of data and its transformation to a tensor model, (2) data preprocessing, and (3) analysis.

3.3.1 Data Collection and Transformation

The first process step for the ranking of Social Web data is its collection. The objective of this stem is to construct the graph of semantic relations between relevant

[2] The framework implementation is available as public domain open source package: http://west.uni-koblenz.de/Research.

users of the platform $G = (V, L, E, linkType)$ (cf. Sect. 3.2.1). For many Social Web platforms, gathering of relevant data for constructing the candidate set of users V can be directly implemented on top of existing platform-specific API functions. For instance, Twitter API support is available for all major programming languages (including Java). This API can be used by an arbitrary account holder of the platform, with certain performance limitations (in terms of the hourly request rate). Large-scale access to API functions can be granted upon individual request.

The process of data collection starts with specifying custom terms(s) of interest. These terms are forwarded to the platform-specific API function for keyword-based search, which returns matching postings together with metadata (usually including author ID). The author IDs are then extracted and added to V. Subsequently, predecessors and successors of these "root" users – in terms of content related relations – are also retrieved and added to V. This step is usually supported by platform API functions for user profile details. For instance, Twitter API provides explicit functions for finding predecessors and successors of the given user: a followers-list (i.e., users that observe postings of the given person) and a following-list (other users that are monitored by the given person). Further expansion of V can be achieved by traversing following-relations and follower-relations transitively, up to a certain maximum depth (i.e., finding successors of successors, etc.).

In the next step, for each user from $u \in V$ a number of his recent postings is retrieved. In Twitter API this functionality is explicitly offered as part of the comprehensive search support. Each posting is then decomposed into terms $t \in L$. For constructing edges in E, we assume that user $u_1 \in U$ links to user $u_2 \in U$ by edge $e \in E$ labeled $linkType(e) = t \in L$, if a) u_1 is known to be a predecessor of u_2 in the platform sense (e.g., in terms of Twitter, u_1 follows u_2) and both u_1 and u_2 have frequently used term t in their postings retrieved so far. In the particular Twitter case, a common practice of lightweight content annotation is the use of hashtags. For this reason, we add the preceding hash sign to all the terms of interest before sending the initial query to Twitter, and do not consider non-hashtag terms in returned postings. Finally, the graph is transformed into the tensor representation for factorization analysis.

Important tuning parameters of the framework include the number of necessary postings for userlist initialization, the maximum number of predecessors/successors to be considered for each user, the number of recent postings per user to be processed, and filtering criteria for removing potentially irrelevant users and terms. For Twitter we instantiated our framework with the following settings (empirically estimated in series of comparative experiments): 300 users for the intitial "seed set," up to 500 direct predecessors/successors per user (dropping down by randomly removing superfluous entries, when necessary), 100 recent postings per user to analyze, each hashtag $t \in L$ should be used by at least ten different users, each user $u \in V$ should use at least three hashtags from L in his postings.

A further pre-processing step is the weighting of collected user relations to further remedy the negative effects of domination. We amplify relations based on their hashtag frequency so that statements with less frequent (i.e., selective) hashtags are amplified stronger than more common relations. As an effect, the adjacency indicators in the tensor have the following property:

$$\mathbf{T}(x,y,z) = \begin{cases} 1 + log\frac{\alpha}{links(z)}, & x,y \in V, \\ & links(z) = |\{e \in E|linkType(e) = z\}|, \\ & \alpha = links(x)|\forall t \in L, links(x) \geq links(t) \\ 0, & \text{else} \end{cases}$$

The value α denotes the number of relations in which the most dominant hashtag participates. The function $links(t)$ ($links : L \rightarrow \mathbb{N}_0$) returns the number of relations in E induced by hashtag $t \in L$.

We remark that the implemented pre-processing steps are valuable for generating ranking analyses in general. Notably, simple methods for authority ranking, e.g., the counting of inlink scores per resource and predicate, benefit more from such pre-processing than more complex methods like PARAFAC.

3.3.2 Analysis

The analysis step implements the PARAFAC decomposition of the tensor, as modeled and created by the previous process steps. We have integrated existing software packages [4] for this purpose. As indicated in Sect. 3.2.2, the number of factors for the PARAFAC decomposition is crucial for the quality of the results of the analysis. The determination of the optimal number of factors is a case of open research. However, heuristics for determining a suitable number of factors have been published, e.g., the core consistency diagnostic (CORCONDIA) [2]. The factor determination applied in TweetRank builds upon such research.

The result of the analysis is a Kruskal (cf. Sect. 3.2.2) tensor [18] that approximates the original tensor. As illustrated in Fig. 3.4, the resulting vectors for the first (row), second (column), and third dimension are represented by three matrices. The columns of each of the matrices correspond to the scores calculated for the different factors $pf_1 \ldots pf_n$. Analogue to the SVD, entries in the column vectors correspond to authority scores, i.e., indicating the relevance of a resource with respect to its

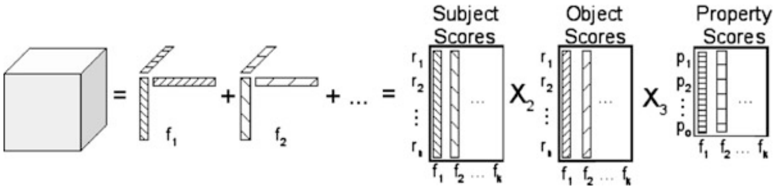


Fig. 3.4 Result of the analysis

in-degree. Entries in the row vectors correspond to hub scores, i.e., indicating the relevance of a resource with respect to its out-degree. We refer to [17] for a thorough analysis of the correspondence between SVD and its interpretation for link analysis. Entries of the vectors in the third dimension indicate the relevance of a term with respect to the hub and authority users. Based on this notion, we interpret hub scores as indicative for the relevance of a user as a "follower" (i.e., observer of other users). Vice versa, authority scores indicate the relevance of a user as subject of observation (whom-to-follow recommendations). As we modeled posting terms by the third dimension, their relevance for particular factors can be looked up in the vectors of the third dimension.

3.3.3 Use Case Example

To demonstrate the functionality of the framework presented so far, we consider the request *semanticweb* from our run-through Twitter scenario. We initiate the construction of the thematically focused social graph by sending to Twitter the keyword-based search request $q = \#\mathit{semanticweb}$. After the expansion step (adding predecessors and successors, downloading postings from all users collected so far) and common preprocessing as described before, we obtain a social graph $G = (V, L, E, \mathit{linkType})$ with $|V| = 1{,}323$ users, $|L| = 175$ hashtags, and $|E| = 17{,}190$ user relations.

Subsequently, the tensor decomposition with $f = 15$ PARAFAC factors provides user authority, user hub, and hashtag relevance scores for each factor. Table 3.3 shows most relevant top-5 hashtags and users for some factors of this decomposition (ordered by hashtag relevance and user authority, respectively).

Some important observations can be made about the results shown in Fig. 3.3. First, the authority scoring in the prevalent factor "semanticweb" is closely related to results produced by simpler ranking mechanisms (e.g., H-candidate set recommendation as discussed in Sect. 3.1). The results are entirely based on recent user postings and current user relations, so any side effects of long-term user profiling (such as temporary user activity in a certain topic, but long time before) will have no influence on current contact recommendations. Beyond the mainstream factor (say *semanticweb core*) the decomposition captures a number of second-order themes related to *semanticweb* and reflected in Twitter postings, such as *web tools*, *multimedia*, *security & privacy*, *social media*, or *programming*. In this sense, the diversity and structuring of recommendation results are substantially increased. As a result, the user can better identify the actual target (sub-)topic of his personal interest related to *semanticweb*, and then follow best-scored users in the context of this particular, thematically focused theme.

Table 3.3 TweetRank results for query "semanticweb" on Twitter

Score	Hashtag	Score	User	Score	Hashtag	Score	User
Factor 1 ("semanticweb")				Factor 4 ("programming")			
0.147	semantics	0.238	timberners_lee	0.111	programming	0.183	cjmconnors
0.125	business	0.143	PEPublishing	0.053	analytics	0.178	DublinCore
0.106	lod	0.142	jahendler	0.047	semantic	0.108	spirinet
0.091	semweb	0.138	timoreilly	0.046	microdata	0.098	AskAaronLee
0.054	semanticweb	0.097	semanticnews	0.040	startups	0.096	GeoffWigz
Factor 2 ("web tools")				Factor 5 ("securiry&privacy")			
0.266	java	0.419	SCMagazine	0.163	security	0.285	BLSocSci
0.251	php	0.143	HTML5watcher	0.130	web20	0.278	socialwendy
0.220	http	0.128	opencalais	0.083	privacy	0.162	pedantic_web
0.214	joomla	0.102	hadoop	0.067	apps	0.154	drthinkmore
0.199	javascript	0.097	LSIstorage	0.054	china	0.136	linuxhoundhost
Factor 3 ("multimedia")				Factor 6 ("social media")			
0.049	music	0.265	Beyond15	0.212	foaf	0.484	jwolfnbaa
0.044	video	0.185	junglejar	0.172	socialmedia	0.165	rdfQuery
0.032	semweb	0.172	CSS3	0.118	facebook	0.130	CreativeCustoms
0.023	iphone	0.134	davidstack	0.109	rdfa	0.078	websciencetrust
0.021	innovation	0.125	emtacl	0.054	webscience	0.064	virtualrooms

3.4 Related Work

From the conceptual perspective, two topics can be seen as closely related to our TweetRank approach: authority ranking for Web contents and graph-based relevance ranking for semi-structured data. This section gives a short overview of these areas and distinguishes TweetRank from other existing solutions.

3.4.1 Rating Web Pages

PageRank [7], HITS [17] and SALSA [20] are prominent algorithms for ranking Web pages based on link analysis. PageRank builds upon a model of a random walk among Web pages, where the stationary probability of passing through a certain page is interpreted as measure of its importance. HITS is based on the notion of a mutual re-enforcement between importance (authority) and connectivity (hub) scores of Web pages. SALSA can be seen as a more complex hybrid solution that integrates ideas of PageRank and HITS by combination of both link traversing directions (i.e., forward and backward) for constructing graph models. The conceptual generalization for this kind of methods is given in [12]. Unlike TweetRank, this family of methods provides no natural mechanisms for expressing and exploiting semantics of links/relations.

The contextualization of graph models can be achieved through different customizations of the mentioned models. Possible adaptations include various custom weightings of graph edges (e.g., based on appearance of particular terms in Web documents [22, 24], content classification [11, 15], structural properties like in-domain vs. out-domain linking [6], etc.) or joint probabilistic modeling for content and connectivity of Web pages [9]. In contrast to TweetRank, these solutions are designed for the Web setting and do not introduce distinguished link semantics. The solution presented in [19] uses for Web authority ranking the higher-order representation of the hyperlink graph by labeling the graph edges with the anchor text of the hyperlinks. This method is closely related to TweetRank, but addresses a fully different problem setting (links and anchors in the Web graph vs. user relations in Social Web).

Another kind of contextualization for authority ranking models can be observed in the area of search personalization. For instance, Eirinaki and Vazirgiannis present a modification of the PageRank algorithm to compute personalized recommendations of Web pages given a path of visited pages [13]. Their approach requires access to web server logs that provide statistics about the paths browsed by other users. BrowseRank [21] is a further example of a page ranking approach that requires to collect statistics on user behavior such as the time spent on a web page. The generalized algorithm for personalized authority ranking is described in [16].

Our TweetRank approach is designed for a different scenario of context-oriented contact recommendation in Social Web environments. The presented approach is conceptually more general and does not rely on user profiles and query logs. As when browsing the Web, detailed statistical information about prior user interactions is often not available through proprietary APIs of Social Web portals (especially for privacy protection reasons). However, this information can be easily integrated with TweetRank, if necessary.

3.4.2 Rating (Semi-)Structured Data

ObjectRank [5] adds authority transfer weights for different types of links to the PageRank algorithm. Such weights influence the *random walk* of prospective users and are to be assigned by domain experts. Beagle++ [8] is an extension for the Beagle desktop search engine that applies ObjectRank to RDF meta data about desktop objects to improve their ranking in desktop search scenarios. TweetRank also considers the semantics of relations, however, it is an approach for computing ranks for users and user groups on-the-fly, as an answer to a hashtag-based query. It does not rely on manually assigned link weights, and is based on the generalized HITS algorithm instead of PageRank.

Anyanwu and Sheth present a framework for query answering with respect to so called semantic associations [3]. A semantic association represents semantic similarity between paths connecting different resources in an RDF model. Aleman-Meza et al. [1] presented and evaluated methods for ranking semantic associations. As a continued work of [3], the presented methods target the identification of similar resources to apply it in scenarios like terror-prevention. Their approach involves ranking criteria considering graph structure, and user context. User context is defined statically by selecting ontology concepts that are considered as representative for a user's context. Ramakrishnan et al. present heuristics for weighting graph patterns connecting two nodes in a graph considering the differences of edges given by RDF graphs that include schema information encoded as RDFS ontologies [23]. Prior approaches on graph pattern analysis presented methods assuming that only one type of edge exists. Next to a presentation of the heuristics, they present an evaluation of them targeting the question which heuristic results in higher quality patterns.

3.5 Conclusion

In this chapter we presented TweetRank, a novel approach for authority ranking in Social Web communities. Conceptually, TweetRank is a correspondent to authority ranking methods known from Web retrieval, such as PageRank or HITS. Our approach exploits the novel representational model for social graphs, based on 3-dimensional tensors. This allows us to exploit in the natural way the available semantics of user relations. By applying the PARAFAC tensor decomposition we identify authoritative sources in the social network as well as groups of semantically coherent terms of interest. Therefore, TweetRank can be seen as a next step towards efficient and effective search/recommendation technology for the Social Web.

References

1. Boanerges Aleman-Meza, Christian Halaschek-Wiener, Ismailcem B. Arpinar, Cartic Ramakrishnan, and Amit P. Sheth. Ranking complex relationships on the semantic web. *IEEE Internet Computing*, 9(3):37–44, 2005
2. Claus A. Andersson and Rasmus Bro. The n-way toolbox for matlab. *Chemometrics and Intelligent Laboratory Systems*, 52(1):1–4, 2000
3. Kemafor Anyanwu and Amit P. Sheth. The p operator: Discovering and ranking associations on the semantic web. *SIGMOD Record*, 31(4):42–47, 2002
4. Brett W. Bader and Tamara G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32(4):635–653, 2006
5. Andrey Balmin, Vagelis Hristidis, and Yannis Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, pages 564–575, 2004
6. Krishna Bharat and Monika Rauch Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In *21st Annual International ACM SIGIR Conference, Melbourne, Australia*, pages 104–111, 1998
7. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*, 1998
8. Paul Alexandru Chirita, Stefania Ghita, Wolfgang Nejdl, and Raluca Paiu. Beagle++ : Semantically enhanced searching and ranking on the desktop. In *ESWC*, 2006

9. David A. Cohn and Thomas Hofmann. The missing link – a probabilistic model of document content and hypertext connectivity. In *13th Conference on Advances in Neural Information Processing Systems (NIPS), Denver, USA*, pages 430–436, 2000
10. Klaas Dellschaft and Steffen Staab. An epistemic dynamic model for tagging systems. In *19th ACM Conference on Hypertext and Hypermedia (Hypertext 2008), Pittsburgh, USA*, pages 71–80, 2008
11. Michelangelo Diligenti, Marco Gori, and Marco Maggini. Web Page Scoring Systems for Horizontal and Vertical Search. In *11th International World Wide Web Conference (WWW), Honolulu, USA*, pages 508–516, 2002
12. Chris H. Q. Ding, Xiaofeng He, Parry Husbands, Hongyuan Zha, and Horst D. Simon. PageRank, HITS and a Unified Framework for Link Analysis. In *25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland*, pages 353–354, 2002
13. Magdalini Eirinaki and Michalis Vazirgiannis. Usage-based pagerank for web personalization. *Data Mining, IEEE International Conference on*, pages 130–137, 2005
14. Richard A. Harshman and Margaret E. Lundy. Parafac: Parallel factor analysis. *Computational Statistics and Data Analysis*, 18(1):39–72, 1994
15. Taher H. Haveliwala. Topic-sensitive PageRank. In *11th International World Wide Web Conference (WWW), Honolulu, USA*, pages 517–526, 2002
16. Glen Jeh and Jennifer Widom. Scaling Personalized Web Search. In *12th International World Wide Web Conference (WWW), Budapest, Hungary*, pages 271–279, 2003
17. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999
18. Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3), 2009 (to appear)
19. Tamara G. Kolda, Brett W. Bader, and Joseph P. Kenny. Higher-Order Web Link Analysis Using Multilinear Algebra. In *5th IEEE International Conference on Data Mining (ICDM), Houston, USA*, pages 242–249, 2005
20. Ronny Lempel and Shlomo Moran. SALSA: the Stochastic Approach for Link-Structure Analysis. *ACM Transactions on Information Systems (TOIS)*, 19(2):131–160, 2001
21. Yu-Ting Liu, Bin Gao, Tie-Yan Liu, Ying Zhang, Zhiming Ma, Shuyuan He, and Hang Li. Browserank: letting web users vote for page importance. In *SIGIR*, pages 451–458, 2008
22. Davood Rafiei and Alberto O. Mendelzon. What is this Page known for? Computing Web Page Reputations. *Computer Networks*, 33(1–6):823–835, 2000
23. Cartic Ramakrishnan, William H. Milnor, Matthew Perry, and Amit P. Sheth. Discovering informative connection subgraphs in multi-relational graphs. *SIGKDD Explor. Newsl.*, 7(2):56–63, 2005
24. Matthew Richardson and Pedro Domingos. The Intelligent surfer: Probabilistic Combination of Link and Content Information in PageRank. In *14th Conference on Advances in Neural Information Processing Systems (NIPS), Vancouver, Canada*, pages 1441–1448, 2001