Design and Analysis of Value Creation Networks

S. Kameshwaran Sameep Mehta Vinayaka Pandit
IBM Research - India
Email: {kameshwaran.s, sameepmehta, pvinayak}@in.ibm.com

Abstract

There are many diverse domains like *academic collaboration*, *service industry*, and *movies*, where a group of agents are involved in a set of activities through interactions or collaborations to create value. The end result of the value creation process is two pronged: firstly, there is a cumulative value created due to the interactions and secondly, a network that captures the pattern of historical interactions between the agents. In this paper we summarize our efforts towards design and analysis of *value creation networks*: 1) network representation of interactions and value creations, 2) identify contribution of a node based on values created from various activities, and 3) ranking nodes based on structural properties of interactions and the resulting values. To highlight the efficacy of our proposed algorithms, we present results on IMDB and services industry data.

Motivation

In this work, we focus on developing algorithms for domains which require different entities (people) to work together towards a common goal referred as deliverable. The motivation of this work lies in services industry in which there is high emphasis on people interacting with other people and serving customers rather than transforming physical goods in the process. However, we contend that the proposed methods are applicable to wide variety of collaboration based domains. A typical deliverable can be thought of as a *workflow* consisting of various stages of specialized tasks. For a successful execution, not only each agent has to complete the specialized task assigned to her but also interact with other nodes in workflow. The final outcome or value created of the workflow depends on both the capability of individual nodes as well as the *quality* of interaction among them. Before presenting the key challenges we briefly present two motivating examples.

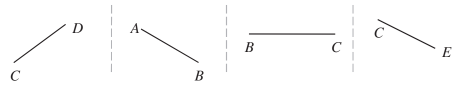
Software Support Services One of the predominant components of the software services industry is to handle maintenance and support services requests. For each request, the service delivery manager puts together a team of agents to resolve the request. The team may include developers, testers, reviewers, etc. Finally, an outcome indicating the effectiveness with which the request is resolved is recorded After serving a sufficiently large number of service requests, the delivery manager may wish to rank the agents. One might be tempted to believe that the manager would like to rank the agents based on their average effectiveness. However, our interviews with domain experts revealed that the importance of an agent is not just based on average effectiveness. Domain experts would like to take into account an agent's connection pattern in the delivery network as well due to the following reasons. Since each request requires handoffs between the agents responsible for them, an agent's connections indirectly capture the agent's influence: familiarity with the subdomains of the connected agents due to the handoffs and ability (or inability) to work with the other agents (depending on the outcomes). When a new service request arrives, estimates of the agents' influence (or rank) and their role/expertise can be utilized to make a decision on the composition of the team to resolve the new request and thereby maximizing the chance of favorable outcome.

Academic Collaboration As a second example, we consider the problem of ranking authors based on academic publications. The goal of academic publication is to disseminate new knowledge and novel insights obtained from academic research. So, the outcome of every academic publication can be measured based on parameters like, the conference or journal in which it appeared, sustained citations it gets over a period of time, awards it wins and so on. At the same time, the influence of an academic researcher is not determined merely by the outcomes of the papers. The connections an author develops and the influence of coauthors play a major role in the overall influence of an academic researcher. So, the ranking of academic researchers needs to take into account the structure of their interactions (obtained from a database like DBLP) and the impact of their papers (obtained from citations, impact ratings of conferences etc).

Consider the example shown in Figure 1(a) which models agent interactions in three different applications. There are five agents A, B, C, D, and E and four interactions involving subsets of them. In case of service delivery, outcome is measured based on customer satisfaction index, in case of academic collaboration, citation is assumed to be the outcome, and in case of movies, the average IMDB rating. The structure of their interactions is captured in Figure 1(b).

In this work we focus on the following questions:

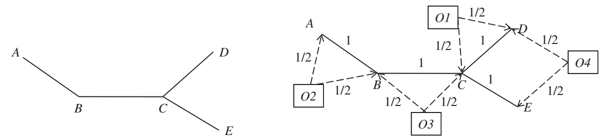
Individual Interactions



Illustrative Outcomes for Different Communities

	Service Interactions				
I.	Deal acquisition *Success*(S) / *Failure*(F)	S	S	F	S
II.	Revenue generated (in millions of USD)	20	32	28	17
	Co-authorship				
III.	Journal impact factor	1.409	2.143	1.102	0.326
IV.	Citation count	32	12	15	47
	Movie co-stars				
V.	IMDB ratings (Out of 10)	6.7	7.4	8.3	9.4
VI.	Box-office revenue (In USD)	55,808,744	70,098,308	100,003,359	107,928,162

(a)



(b) (c)

Figure 1: Example of Interactions, Values and Representation (a) Individual Interactions and Sample Values in different domains(b) Representation capturing only the interactions (c) Interaction Network capturing interactions as well as outcomes

Q1: The computer science literature is rich in modeling interaction of nodes via a directed or undirected graph. How to extend the graph based representation to capture the outcome of the interactions? In essence, a single representation which captures both the structure as well the outcome in meaningful fashion. We refer to such representation as *value creation networks*.
Q2: Given the historical interaction and outcome data for an agent, how does an agent impacts the outcome of workflows in which it is involved? A simple average based methods are not suitable for this problem, primarily due to human aspect of the problem. Consider an super agent *a* ensures that all the workflows that he is involved in succeed. Consider an agent *b* who is failure-prone when teamed with ordinary agents. Let 80% of the workflows of *b* also contain *a*. Let us further say that a large fraction of the rest (20%) of the workflows of *b* have failed. The aggregate based approach would infer that *b* is a highly successful agent. However, we would like to infer that b is prone to failures.
Q3: Finally, given the complete interaction data for all agents, how to rank the agents? The ranking should consider the structure as well as the value created by the agent.

In this paper, we summarize our work (referred as *interaction networks* in (Kameshwaran et al. 2009; 2010)) which initiates a systematic look into the analysis of interaction network while simultaneously taking into account both the structure and outcomes of the interactions. We highlight that our work adds a new dimension to traditional social network analysis which only focuses on the structural aspects of the networks arising out of interactions.

Algorithms

Basic Notations: Let $V = \{1, 2, \dots, N\}$ be the set of agents in the system, $\{1, 2, \dots, T\}$ be the set of interactions, with each interaction resulting in one of the possible outcomes $R = \{1, \dots, M\}$. Following indices are used: i and j for agents, t for interactions, and m for the outcomes. An interaction $t \in \{1, 2, \dots, T\}$ involves a subset of agents $V_t \subseteq \{1, 2, \dots, N\}$. The pattern of the interaction is given by the edge set E_t with a non-negative δ^t_{ij} denoting the weight on the edge $(i, j) \in E_t$. The weight for non-existent edges is zero. Let $R_t \in \{1, 2, \dots, M\}$ be the observed outcome of the interaction. The interaction t can thus be completely characterized by the tuples $(V_t, E_t, \{\delta^t_{ij}\}, R_t)$. Let $\rho_m \in \mathbb{R}$ denote the utility or value of an outcome $m \in R$. The set $\{\rho_m : m \in R\}$ can be cardinally ordered and if $\rho_{m'} > \rho_{m''}$, then the outcome m' is preferable to m''. Typically, the utility of the outcomes is ascertained based on domain knowledge and also varies over time. So, the representation of the data and the technique should ideally allow the user to change just the outcome values and observe the corresponding changes during analysis.
Design of Value Creation Networks: We note that eigen value based methods have been fairly successful in ranking nodes in a graph. The main reason the eigenvector based approach captures the structural aspects so well is the manner in which a node transfers part of its status to its neighbors and derives its own status as a linear combination of the status of its neighbors. We try to emulate similar logic in our design and analysis of value creation networks. Consider an interaction which involves V_t nodes and have R_t as outcome. We need a mechanism by which the utility of R_t can be transferred in parts to the nodes in V_t. At the same time, the utility of the outcome R_t itself is a *prior* and should not be affected by the status of the nodes in V_t. This suggests the following natural construction. Let the outcome of interaction t be m. The graph $(V_t, E_t, \{\delta^t_{ij}\})$ is updated as:

$$V_t \leftarrow V_t \cup \{m\} \tag{1}$$
$$E_t \leftarrow E_t \cup \{(m, i)\},\ \forall i \in V_t \tag{2}$$
$$\delta^t_{mi} \geq 0,\ \forall i \in V_t \tag{3}$$
$$\sum_{i \in V_t} \delta^t_{mi} = 1 \tag{4}$$

The outcome m is added as a node (and is called as *outcome-node*). A directed edge is added from m to each of the other agents that participated in the interaction. The weights on the newly added edges that are given by (3) and (4), captures the relative contribution of agents in realizing the outcome m. The weights on the non-existent edges are zero: $\delta^t_{im} = 0$, $\forall i$ and $\delta^t_{mi} = 0$, $\forall i \notin V_t$. The status or the influence of the outcome node is later used in the algorithm to transfer it to the participating nodes via the directed edges.

Without loss of generality, we can assume that each of the outcomes is realized in at least one of the interactions. The aggregation of the graphs augmented with outcomes is:

$$V \leftarrow V \cup \{1, 2, \ldots, M\} \quad (5)$$

$$E \leftarrow E \cup \{(m, i) : \exists t,\ R_t = m \wedge i \in V_t\} \quad (6)$$

$$\delta_{mi} = \sum_t \omega^t \delta^t_{mi},\ \forall i, m \quad (7)$$

$$\delta_{im} = 0 \quad (8)$$

The outcomes $\{1, 2, \ldots, M\}$ are added as nodes to the interaction network and an edge from outcome m to a node i exists if i had been a part of at least one interaction with outcome m. The weights on the outcome-agent edges are taken as linear combination of the corresponding weights in the individual interactions. We call the above network as the *interaction network* and the corresponding edge weight matrix as the *agent-outcome interaction matrix* Δ of order $(N + M)$. For the network in figure 1(b) with outcome *II* of figure 1(a), the agent-outcome interaction network is shown in figure 1(c). The four outcomes $O1, O2, O3, O4$ correspond to the outcomes of the four interactions. Assuming equal contribution from each agent, weight on the directed edge from an outcome to an agent is $1/2$.

The matrix Δ captures both the inter-agent interactions and agent-outcome interactions. It is asymmetric. The overall intended effect of the directed construction is to let the outcome-nodes transfer their utilities to the agents and the utilities of the outcome-nodes are not altered. The matrix Δ, however does not take into account the utilities of the outcomes $\{\rho_m\}$. We treat the utilities as exogenous status of the outcomes and combine with Δ to obtain the ranking.

Outcome Aware Node Ranking: The traditional eigenvector ranking has a limitation that it can only be applied to non-negative, symmetric matrices. Thus, if the underlying graph is asymmetric, especially with *unchosen* nodes (nodes with zero in-degree), the ranking provided by eigenvector centrality is inconsistent. The unchosen nodes receive no status from the other nodes and hence contribute nothing to the nodes to which they are connected. But, in our construction the zero in-degree nodes play a critical role of transferring the status of outcomes to the nodes.

Let e be the vector that captures the status of the outcomes nodes. That is, $e[i]$ for agent nodes are made identical and for outcomes nodes, it will be a function of their values. Let x be the vector that an iterative technique computes as the final influence of the nodes. In other words, the difference vector $(x - e)$ should be expressible in an eigenvector-like form with respect to a scaling parameter α and the incident matrix Δ of the augmented network.

$$(x - e) = \alpha \Delta^T x \quad (9)$$

In other words, we are looking for an x such that,

$$x = \alpha \Delta^T x + e \quad (10)$$

Observe that the Equation (10) captures the effect of interactions and the values associated with the outcomes nodes. Our measure of influence of the nodes is similar to a centrality measure used by Bonacich (Bonacich and Lloyd 2001) in finding influence of nodes in unweighted, directed graphs in very small social networks coming from marriage data among elite families. In what follows we establish some important properties of our approach.

Firstly, it is easy to see that, it reduces to the eigenvector ranking when the relationships are symmetric and the value of all the outcomes are identical (say zero). In the general setting as above, the value of x is given by

$$x = (I - \alpha \Delta^T)^{-1} e \quad (11)$$

The vectors e and x, and identity matrix I are of order $(N + M)$. We can show that the method works best when α is in the range $(0, 1/\lambda)$ where λ is the largest eigenvalue of Δ. We refer to this algorithm as *Outcome Aware Ranking Algorithm* (OARA). The choice and effect of free parameters α and e of OARA are described in (Kameshwaran et al. 2010).

Inferring Individual Node's Impact: We measure the impact of an agent by assigning a weight, w_a in the range of $[0, 1]$ that is indicative of the agent's contribution towards success/failure of workflows. Given these weight assignments and a specific workflow W, let $w_{avg} = \frac{\sum_{a \in W} w_a}{|a \in W|}$ be the average weight of the agents belonging to the workflow. One way to explain the outcome of the workflow is to compare the average weight to certain thresholds associated with the outcomes. For simplicity, let us assume that the utility of outcomes R_ts are either 0 (failure) or 1 (success). Our approach extends directly to the continuous case. Let S_t and F_t be two thresholds in the range $[0, 1]$ corresponding to successful and failure workflows respectively. The assignment of weights is said to explain the outcome of a successful workflow W if $w_{avg} > S_t$. Aggregate based method of assigning weights would average the outcomes of the agent's workflows. For an agent a, $w_a = \frac{\sum_{t: a \in W_t} R_t}{|\{t: a \in W_t\}|}$. Let f be the fraction of workflows that are explained by the aggregation based approach. Our goal is to significantly improve the fraction of explained workflows in comparison to f. We start with any valid assignment of weights to agents. For each workflow that is not explained by the current assignment, update the weights of the agents belonging to the workflow in small quantities ϵ in such a way that the gap between the threshold and its average decreases. If a workflow's outcome is 1 and it is not explained, we increment the weight of each of its agents by ϵ. Similarly, if the workflow's outcome is 0 and it is unexplained, then, we reduce the weights of each of its agents by the same quantity ϵ. While updating the weights in this fashion, we restrict them to the $[0, 1]$ range. The procedure is terminated when the fraction of the explained workflows is above a threshold F (say 0.95) or if the fraction of the explained workflows in the last L rounds does not increase by a minimum threshold.

Experimental Results

Ranking in IMDB Data: For a given set of movies, we construct the interaction network as follows: each actor who

1.Marlon Brando	2. Al Pacino	3.Robert De Nero	4. Sean Bean	5.Jean Reno	6.Don Cheadle	7. John Travolta	8. Hugh Jackman
9. George Clooney	10.Casey Affleck	11. Brad Pitt	12. Matt Damon	13.Dan Fredenburgh	14.Bill Nighy	15. Johnny Depp	16. Orlando Bloom
17. Jack Davenport	18.Lee Arenberg	19.Tom Hollander	20.Jude Law	21.Anthony Hopkins	22.Sean Penn	23.Samuel L. Jackson	24.Kevin Bacon
25.Tom Hanks	26.Steve Buscemi	27.Clive Owen	28.Nicolas Cage				

Table 1: List of actors used in experiments

R_1 = OARA(Δ, e)	23 9 25 12 10 7 21 15 3 6 20 8 24 16 28 2 22 26 27 11 1 4 14 18 5 17 19 13
R_2 = SVD(Δ)	9 23 25 12 7 10 21 15 3 20 8 6 24 28 16 2 22 26 27 11 4 1 14 5 17 18 19 13
R_3 = OARA(Δ', e)	25 28 23 9 12 10 7 21 15 3 16 20 6 8 24 2 22 26 27 11 14 1 4 18 17 5 19 13
R_4 = SVD(Δ')	9 23 25 12 7 10 21 15 3 20 8 6 24 28 16 2 22 26 27 11 1 4 14 5 17 18 19 13

Table 2: Rankings of actors under different conditions

appears in any of the movies is an agent, each movie represents an interaction that is incident on all its main actors (we choose first 6 listed names) and the outcome associated with the interaction is the average user ratings for the movie. Table 1 shows the actor list used for experimentation.

We used the ranking according to the left singular vector corresponding to the largest singular value of interaction graph as baseline algorithm. Intuitively, this heuristic does not address the subtle technical point we covered in depth: that of ensuring that the outcomes are used to influence the ranking of agents and not vice versa. The experiments were conducted under following settings:

Setting 1 R_1 and R_2 are ranking by OARA and SVD based baseline respectively.

Setting 2 We pick two highly ranked actors in both the rankings, say A_1 and A_2. For each of the movies in which either of them appears, we artificially reduce the averaging rating by 2. We then pick the two middle-ranked actors, A_3 and A_4. We increase the ratings of those movies in which either of them appears by 2. Let R_3 and R_4 be the ranking generated on the modified data by OARA and baseline.

The different rankings obtained are as in Table 2 (With $A_1 = 9, A_2 = 23, A_3 = 28, A_4 = 16$). The rankings are given in the ascending order of ranks; the actor with the first rank appears first and so on. One would expect the rankings of A_1 and A_2 to go down and those A_3 and A_4 to go up. Notice (by comparing R_1 and R_3) the rankings of the both top actors (number 23 and 9) have gone down while those of the two chosen mid-ranked actors A3 and A4 (number 28 and 16) have gone up. Notice (by comparing R_2 and R_4) that there is hardly a noticeable change in the SVD rankings before and after modification. This shows that our formulation is takes into account the changes in outcomes or value.

Individual Impact Analysis in Services Data: We used the simulation framework to carry out our experiments. We generated instances of service value creation networks by first creating a network of agents and then pushing a certain number of workflows through them. The outcome of the workflows was decided by the average of the weights of the assigned agents. The iterative analysis was run on the resulting service value creation networks. The efficacy of our approach was measured based on the how it could improve the fraction of the workflows that were explained by the final weight assignment as opposed to the fraction explained by the simple aggregation method. Tables 3 and 4 present the comparison under different settings. Moreover,

#Agents	#Workflows	% of Ag. Apr.	% of Iter. Apr.
250	500	74	90
250	2000	64	81
250	10000	57	77
250	60000	62	78
250	80000	46	61
125	500	67	82
125	1500	62	81
125	5000	63	79
125	10000	66	78
125	30000	44	52

Table 3: Comparison of the two approaches as number of requests is increased while keeping agents constant

#Agents	#Workflows	% of Ag. Apr.	% of Iter. Apr.
6	10	77	90
10	10	73	91
100	100	69	84
125	1000	62	81
125	5000	63	79
250	500	71	92
250	5000	56	79
250	100000	38	50

Table 4: Comparison of the two approaches for different choices of service requests and agents.

the overall assignment was not influenced in a big fashion by introducing *success prone* or *failure prone* agents which points towards stability of our assignments generated using iterative techniques.

Conclusion

In this article, we presented the key construct of value creation networks which capture structure as well as the outcome of the interactions. Diverse problem domains can be modeled using such an approach. We presented a novel update algorithm for characterizing the nodes so as to explain individual outcomes. We also presented a new algorithm for ranking the nodes based on the structural and value creation aspects of the network. We point interested readers to our previous works (Kameshwaran et al. 2009; 2010) for elaborate details.

References

Bonacich, P., and Lloyd, P. 2001. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks* 23:191–201.

Kameshwaran, S.; Mehta, S.; Pandit, V.; Parija, G.; Singh, S.; and Viswanadham, N. 2009. Analyses for service interaction networks with applications to service delivery. In *SIAM International Conference on Data Mining (SDM)*.

Kameshwaran, S.; Pandit, V.; Mehta, S.; Viswanadham, N.; and Dixit, K. 2010. Outcome aware node ranking in interaction networks. In *Conference of Information and Knowledge Management (CIKM)*.