# Knowledge Graph based Recommendation Techniques for Email Remarketing

László Grad-Gyenge
TU Vienna
Vienna, Austria
laszlo.grad-gyenge@tuwien.ac.at

Hannes Werthner
TU Vienna
Vienna, Austria
hannes.werthner@tuwien.ac.at

Peter Filzmoser
TU Vienna
Vienna, Austria
peter.filzmoser@tuwien.ac.at

***Abstract*—This paper presents the knowledge graph, a graph based information modelling technique. The method generalizes the concept of information sources and defines a hybridization technique at the information representation level. Depending on the amount of collaborative and content-based information available, the balance of the hybridization is also discussed. The principles of the defined calculation methods, as generalization and transitivity facilitate the paradigm of relatedness. To evaluate the efficiency of the knowledge graph, graph based recommendation calculation techniques are defined and evaluated in an email remarketing activity, a real-world recommendation scenario. The results show that the spreading activation based recommendation technique is capable to increase the performance of the remarketing task.**

***Keywords–knowledge graph; recommender system; spreading activation; network science; email remarketing***

## I. Introduction

This paper presents the detailed results of our experiments on knowledge graph based recommendation techniques as introduced in our conference paper [1]. In this paper, our goal is to give a deeper insight into graph based methods and also to present the detailed evaluation results of our experiment.

Graph based recommender systems are a potential alternative to representation learning based recommendation techniques. Unlike the current trend of recommender systems finding latent factors in the "taste space", graph based methods focus more on distance like measures and relatedness [2] between the nodes in the knowledge graph. According to our past experiments, graph based methods have the potential to eliminate the cold start problem, to converge faster as rating estimators [3] and to act as the basis of relatedness based recommendation techniques.

The graph based information representation is general and flexible. It can act as a potential alternative to tensor based information modelling [4]. Recommendation techniques typically work with a previously and exactly defined information representation model. Compared to these methods, the knowledge base [3] models the information in a heterogeneous multigraph. The nodes of the graph are different entities playing role in the recommendation scenario. The edges of the graph are the various relations between the entities. An important aspect of our work is to model as much information as available in the knowledge base and develop methods capable to manage the heterogeneous information. Our basic philosophy is the more information we have, the higher is the possibility to avoid the cold start case.

Early recommender systems [5] can be treated as function estimators. Collaborative filtering estimates the rating values by calculating a weighted average on the existing rating values. Recommendation spreading [3] can be treated as the generalization of collaborative filtering for the graph based case. The advantage of the method is that it successfully alloys the information found in the rating values and the information found in the structure of the network. The most important feature of recommendation spreading is the faster calculation of the error on the rating estimation.

State of the art recommender systems are typically utilized in an information retrieval scenario. It means that instead of focusing on rating estimation, the task is to provide a list of recommended items to the user in question. The user rating on items relation can be generalized as the user interest on items, which relation type acts as the basis of the prominent, representation learning based recommendation learning techniques. On one hand, the application of state of the art techniques to generate rating estimation is quite atypical. On the other hand, state of the art methods usually derive the recommendations based on user preferences on items. Graph based methods have the potential to generalize the concept of information sources and to develop calculation methods based on the relatedness between the nodes instead of being constrained by the user preference based paradigm.

An important aspect of our work is that the information representation and the calculation methods are clearly separated. This methodology eliminates the unnecessary dependencies between the two tasks and also leads to a cleaner approach on the theoretical side.

On the representation side, an information representation method is defined to alloy the information sources of the traditional recommendation techniques as collaborative filtering, content-based filtering and some knowledge based cases. Our primary motivation to introduce this information representation method is to involve the content-based and knowledge based information sources in addition to the collaborative information in the cold start case. An important feature of the knowledge graph is that in a real-world application, before the production state, the graph contains only the content-based and the knowledge based information. If the recommendation calculation method is general and uses all the information available in the knowledge base, the recommendation system can be treated more content-based than collaborative. As the users start to interact with the system, the knowledge base will contain more collaborative information, the system acts more as a collaborative filtering technique. Thinking about the amount of information available in the aforementioned categories, the order of magnitude of collaborative information is typically higher than the order of magnitude of content-based information. To summarize it, working with general calculation method based on the knowledge graph, as the users interact with the system, the recommendations turn to be collaborative from content-based. In other words, based on the information representation method, the limes of the recommendation system is collaborative.

Looking at the calculation methods, graph based representation inherently provides the platform to involve novel calculation techniques to generate recommendations. In contrast to representation learning based techniques, graph based recommendation methods can be described as proximity calculation methods between the nodes of the graph. Various techniques can be involved to calculate the proximity measure. We think that a well elaborated proximity measure should incorporate also the topology of the graph and the length of the paths between the nodes. In other words, the parallel paths between two nodes should increase the value of the proximity measure. In addition, the results of network science can also be utilized, as network science provides several techniques to analyse a graph and methods to calculate the centrality of specific nodes.

Referring to our past results on the MovieLens 1M dataset [6], the knowledge graph based methods have the potential to significantly increase the coverage of the recommendation methods and also show faster convergence regarding the mean absolute error of the rating estimation [3]. Regarding top list recommendations, graph based methods show an increase in the quality regarding to precision, recall and NDCG [2]. In this paper we would like to focus on the performance of the graph based techniques on the email remarketing domain.

A state of the art application of recommender systems is web remarketing. The essence of remarketing is to present personalized offers to the user on media different to the origin of the offer. Remarketing is basically a post event activity, the users will be offered with items after their visit to the specific portal. Web remarketing is a prominent class of remarketing, where the promotion activity is to be done in on-line systems. Criteo [7], a prominent company in this field provides web based personalized offers based on the activity of a user.

In our experiment, the recommendation techniques are evaluated in the email remarketing system, PartnerMail [8]. Email remarketing is the remarketing activity utilizing newsletters as the media of the promotion task. The emails sent contain a list of promoted products, the products are selected by the evaluated recommendation methods. The personalization is based on the user activity collected in Booker [9], an electronic commerce portal selling books. The performance of spreading activation based methods, network science centrality measures and the human is compared. Our results show an increase in the performance of the newsletters regarding the click-through rates. We can also state that email based remarketing activity leads to a higher customer engagement based on the personalization, thus it delivers a business value.

This paper is organized as follows. Section II presents related work from the aspect of information representation and recommendation calculation techniques. This section also discusses hybridization techniques as an important, inherent feature of the knowledge graph. Section III presents the knowledge graph, our information representation technique. At first, short examples are provided about the information representation of the traditional methods and the hybridization technique of the knowledge graph. After that, the problem is reflected through an information rich example containing heterogeneous information sources. Then, the definition of the knowledge graph is provided. Finally, the balance of the collaborative and the content-based information is analysed from a more theoretical point of view. Section IV covers the calculation methods operating on the knowledge graph as collaborative filtering, spreading activation, recommendation spreading and the human expert. Having the methods defined, the paradigm of relatedness is discussed. Section V describes the dataset of the experiment. Section VI discusses the evaluation method. Section VII presents the results of the evaluation. Section VIII concludes the paper and defines the directions of the further research.

## II. Related work

As mentioned in Section I, the methodology of the information representation and the recommendation calculation is clearly separated in our work. Section II-A focuses on the evolution of recommendation techniques from the aspect of involved information sources. Section II-B discusses various calculation techniques operating on graphs. To emphasize the hybridization capability of the knowledge graph, Section II-C analyses various hybridization techniques of recommender systems.

### A. Representation

In this section we show various information representation methods developed to model the increasing amount of information sources involved into the recommendation calculation process. Related work on this topic is organized to show the evolution of information sources foreshadowing the need of a more general information representation technique.

The information necessary to perform the traditional collaborative filtering method [10], i.e., the user preferences on the items are represented in a matrix, which is typically sparse. The rows of the matrix denote the users, the columns of the matrix denote the items. The values of the matrix represent the user preferences on items. Missing values in the matrix refer the unknown or non-obtained preference values. The user preference in this case is the user rating on the specific item. The representation of the matrix of collaborative filtering is discussed in Section III-A1.

Konstas et al. [11] work on music track recommendation utilizing a graph based recommendation technique. In order to increase the recommendation quality, Konstas et al. involve tags into the recommendation scenario. The graph based knowledge base assign a type to the nodes and the edges. The approach is based on “user”, “music track” and “tag” node types and “average user play count”, “user listened to”, “user tagged” and “track is tagged” relation types. The knowledge graph is represented as a partitioned adjacency matrix. Each partition represents the appropriate relation type.

Contextual information is a very useful information source to enhance the quality of the recommendations. To involve the context into the recommendation process, a tensor based information representation technique is applied typically. Such approaches are conducted by, e.g., Hidasi et al. [12] and Adomavicius et al. [13]. Their representation approach is a three dimensional tensor, the dimensions of the tensor are "users", "items" and "context information". Looking at the mentioned approach, the tensor based representation technique can be treated as the clarification of the partitioned matrix based method.

Referring to homophily [14] and social confluence [15], the social network is a useful information source of recommendation techniques. The social network is typically represented in a graph, where the nodes of the graph represent the persons and the edges of the graph represent the social relationships. Kazienko et al. [16] introduce a multi-layered graph to represent the social network of the users in the recommendation scenario. Each layer of the graph contains homogeneous information, such as contact lists, tags, groups, favourites and opinions.

To examine the representation, there are examples of both the non-directed and the directed case. The application of the directed social relationships is more visible in the years of 2004-2006 in the trend of involving the trust network of Epinions [17] into the recommendation process [18], [19], [20]. To continue with the directed case, Guy et al. [21] improve the weighting scheme of collaborative filtering by deriving the weights also from the social ties. He et al. [22] combines user attribute values, item attribute values and social ties with a naïve Bayes approach. Yang et al. [23] apply Bayesian inference to social network in a distributed approach. Gu et al. [24] apply matrix factorization to predict user-event participation with the help of the social network.

The research results presented in this chapter show a trend of involvement of an increasing number of information sources. The evaluation of the methods show that additional information can significantly increase the recommendation quality. This collective finding motivated us to develop a generalized information representation method, so that additional information sources can be added to the knowledge base without the modification of the representation model. To cover the problem, a graph based information representation method has been elaborated, which is able to model the information need of the calculation methods mentioned in this section.

Recent works show a remarkable progress on the research of knowledge graph based recommendation techniques. Kouki et al. [25] define the knowledge graph as a bipartite graph of users and items allowing heterogeneous relations between the nodes of the graph. Hu et al. [26] utilize a heterogeneous social network to identify the leads. In their work, Hu et al. introduce label propagation and analyse various path configurations between the nodes of the graph. Catherine et al. [27] apply a probabilistic logic approach to the knowledge graph containing heterogeneous information. Burke et al. [28] present a heterogeneous network of information working with various two dimensional projections of the network.

Regarding to the current research on the knowledge graph, our intention is to define the representation method as clear and general. Our definition does not restrict the knowledge graph to be bipartite but assigns a type to each node and to each edge. Referring to the paradigm of relatedness as discussed in Section IV-B, by our definition, the paths between the nodes are not restricted by their type but are treated as relations in general. Our framework also provides a method to incorporate both the user and the item attributes. In addition, the defined calculation methods do not work with graph projections but are clearly graph oriented.

### *B. Methods*

In this section, an overview of graph based recommendation calculation techniques evolved is provided. Our goal is to illustrate the potential of the graph based information representation, as there are already various calculation methods developed utilizing the information contained in the graph. At first, random walk based methods are presented. Conducting a random walk is a straightforward method to deliver recommendations utilizing reasonable computing capacities. After that, propagation based methods are discussed in the directed and also in the non-directed case. The advantage of the propagation methods compared to the random walk method is the elimination of the stochastic behaviour. The drawback of the propagation is the computation intense implementation. Then, network science related methods are presented to introduce the non-personalized recommendation methods. Finally, neural network methods are discussed to show the potential of networks consisting of non-linear calculation units.

In this paper, spreading activation based recommendation techniques are evaluated. The motivation behind is the following. In order to clearly see the performance of the graph based representation, our intention is to eliminate the stochastic behaviour. In addition, as our goal is to increase the coverage, non-directed methods are involved, as directed paths lead to a lower connectivity in the knowledge graph. Finally, spreading activation has the capability to take both the distance and the parallel paths in the graph into account.

*1) Random Walk:* Random walk methods treat the recommendation problem as a stochastic process on a Markov chain. To model a discrete time Markov chain with a graph based knowledge base, the states and the transition probabilities for the stochastic process are to be defined. Each node of the directed graph can be assigned to a state in the Markovian model. The transition probabilities from a specific state can be defined by normalizing the weight of each outgoing edge with the sum of the weights of the outgoing edges.

Jamali et al. [29] introduce the TrustWalker algorithm on the Epinions dataset. They define the recommender graph as the graph of the users, the edges of the graph are explicit trust statements between users. The approach is to estimate item ratings based on existing ratings of trusted or transitively trusted users. To increase the coverage of rated items, the method estimates the rating of a particular item based on the rating of similar items. The item similarity is defined with Pearson correlation. Their method is to conduct a random walk on the graph to find a trusted user. The walk is to be stopped based on a probability depending on the length of the path, the user to rate for and the item to rate. The random walk basically selects a suitable trusted user then it returns the rating of the trusted user on the item in question. If the specific item is not rated by the user, the method returns the rating of a similar item. Jamali et al. emphasize an important feature of the Trustwalker method, the ability to calculate a confidence value for the estimated rating value. They define the confidence as the variance of the rating estimation of several random walks. As a random walk is definitely a less resource exhaustive method than for example spreading activation, a straightforward method to increase its precision is to run multiple instances and return an aggregated value of the estimated ratings.

*2) Trust Propagation:* Trust propagation is a well researched area. Several publications appeared in this field. Guha et al [30] and Ziegler et al. [18] examine the propagation of trust and distrust. Hess et al. [31] propagate trust on a two layered recommendation graph. Golbeck [32] propagate trust on a film trust database. The idea behind trust propagation is to calculate recommendations based on trust relationships. A well known web portal in this area, Epinions [17] allows its users to rate items on the platform and explicitly express trust in other users. Trust in this case means trusting in other user's opinion, namely the rating. The basic idea behind trust propagation is to extend the direct trust relationship with a transitive method and generate the recommendations with the help of the extended trust metric. Applying graph related methods in trust calculations foreshadows the application of a more general, graph based knowledge base methods.

*3) Spreading Activation:* Spreading activation has been introduced by Quillian [33] and is a common method in the fields of associative networks [34], semantic networks and neural networks. The method is defined iterative, where the active nodes in the network activate their neighbouring nodes in each iteration step. The method defines a decay factor also for the node activations and for the spreading values. These decay factors has to be tuned depending on the application domain. The outcome of spreading activation is a list of graph node and activation pairs. By sorting the activated nodes on their activation in a descending order after the termination, we come to a prioritized list of the recommended items. An important property of the spreading activation algorithm is that it takes into account the topology of the paths from the source node to the recommended nodes by aggregating the spreading value of the paths in parallel.

Ziegler et al. [35] introduce the AppleSeed method on the Epinions dataset. The recommendation graph in this case is the trust network with users in its nodes and weighted trust statements in its edges. The goal of Ziegler et al. is to extend the local trust statements from neighbouring nodes to farer extent. An important feature of the AppleSeed Trust Metric is the avoidance of the dead ends. A typical problem of spreading activation methods is that nodes with zero outdegree have the opportunity to capture the spreading values by accumulation. To manage this problem, Ziegler et al. suggested backward trust propagation. When backward trust propagation is applied, a virtual edge is created from all nodes (except the source node) to the source node. The aim of these virtual edges is to trace back the activation to the source nodes. Based on the virutal edges, the dead end nodes spread their activation to the source and the accumulation of the activation can be avoided.

*4) Network Science:* Network science evolved methods and measures to analyse huge and complex networks [36]. Such networks are for example public transport networks, telecommunication networks, biological networks, sociological and semantic networks.

In order to measure the importance of the nodes of networks, network science developed various centrality measures. These measures represent the position or importance of a node in the network. Such measures are for example: degree centrality, closeness centrality, betweenness centrality and eigenvector centrality. Degree centrality counts the edges belonging to the node. Closeness centrality is the inverse of farness, which is the sum of the length of paths from the node to all other nodes. Betweenness centrality is the number of how many times a node lays on the shortest path between two nodes. Eigenvector centrality is proportional to the sum of the eigenvector centralities of its neighbours, it is a recursive definition and is calculated with the eigenvalue decomposition of the adjacency matrix of the network. PageRank [37], a well known measure for the nodes of large directed networks is basically a walk in the graph with random restart.

The mentioned centrality measures define a global measure for a network node. It means that the application of the standard centrality measures is not suitable for personalized recommendations. To illustrate the potential of network science methods, Fogarasi et al. define a personalized PageRank calculation method [38].

Jeong et al. [39] extend the traditional collaborative filtering method with network science measures. Their recommendation graph contains the social network of the users. Their dataset contains YouTube web page visiting data, which information is not incorporated into the network. In their approach, Jeong et al. extend the weighting scheme of the traditional collaborative filtering formula. The weights of the original formula are based on implicit similarity between the users, e.g., the Pearson correlation of the issued ratings on the common rated items. Jeong et al. modify this implicit similarity to a weighted sum of the implicit similarity and a network science centrality measure. In their experiment, they evaluate different weighting schemes and various centrality measures. Their results show that incorporating network science measures increase the precision of the recommendations, but also show that the recommendation quality becomes weak if higher weights are assigned to the network science measures. Jeong et al. also found that degree centrality based recommendation shows the highest performance. Referring to homophily [14] and social confluence [15], persons in a social network with high degree centrality have a high number of social network connections and can be treated as influencers.

In addition to the case introduced by Jeong et al., a potential application scenario of network science measures are the global recommendations. In the cold start case, when the actual recommendation technique is not capable to deliver recommendations, in order to provide items to the particular, a fallback method is necessary. The fallback method is inevitably non-personalized, it provides high coverage and a lower precision. Such non-personalized recommendation techniques are, e.g., recommending top selling items, most visited items or most mentioned items. Network science provides alternative methods to find top items in the set of recommendable items.

*5) Neural Network:* The common application areas of neural networks are function approximation, regression, time series prediction, classification novelty and anomaly detection. Neural networks are the general and flexible tools of artificial intelligence. According to Haykin [40], neural networks can be represented with directed graphs, which makes neural networks applicable in the field of graph based recommender systems. A possible application represents the nodes of the graph with artificial neurons and the edges of the graph with the synapses. This representation assumes that the knowledge base of the recommender system is a directed graph.

*a) Calculations:* The application of artificial neural networks is based on the calculation of the activations of the neurons. The activations are calculated with the network function. A typical network function is a simplified model of how natural neurons work. The network function basically calculates the activation of one neuron based on the neurons on its input. It can be calculated by applying the activation function to the weighted sum of the inputs of a neuron. To make calculations more general a bias is also added to the weighted sum. The network function is defined in Equation (1).

$$s_i(x) = \varphi(\sum_j w_{ij} s_j(x) + b_i) \tag{1}$$

where $s_i(x)$ denotes the network function of neuron $i$, $w_{ij}$ denotes the weight of neuron $j$ activating neuron $i$, $b_i$ denotes the bias of neuron $i$ and $\varphi$ denotes the activation function, which in most cases is a nonlinear function.

The calculation of the value of the network function is an iterative process. In the initial step, depending on the application, certain neurons are activated. The iteration steps can be calculated either synchronously or asynchronously. The synchronous method updates the activation of all the neurons at once. The asynchronous method works with a predefined order or based on random selection. The iterative process is to be stopped after a certain limit of steps is reached or the change of the activations is under a specific threshold value.

*b) Architecture, Hidden Nodes:* A neural network consists of several information processing units, of the artificial neurons. These units are simple and in most cases nonlinear. To make these information processing units able to develop more sophisticated functions or transformations, they are combined into different architectures as also described by Bengio [41]. Several architectures of artificial neural networks contain hidden nodes typically organized in hidden layers, e.g., multi-layer feed-forward networks [42]. The application of hidden nodes is to develop a more suitable mapping between input and output nodes. In the case of a representation learning problem, the role of the hidden nodes is feature extraction.

*c) Training:* The most powerful feature of artificial neural networks is the ability to learn. Artificial neural networks achieve this task by training the synapse weights between the neurons of the neural network. There are several methods to train a neural network. These methods can be classified into three main categories, as supervised learning, unsupervised learning and reinforcement learning.

Supervised learning systems are trained based on sample input and the corresponding output values, which in the case of recommender systems could mean user-item pairs, e.g., the expressed explicit or implicit interest by the user on the specific item.

Unsupervised learning in artificial neural networks can be applied to develop internal representations of sample data without any feedback to the system. Although recommender systems are based on user interaction and we can take the benefit of user feedback data to train a neural network based recommender system, there are existing approaches to involve unsupervised learning into the training process of neural network based recommender systems. Unsupervised learning systems can also be applied to develop internal or sparse representation and can be used as data compression method as a part of another recommender system architecture.

Reinforcement learning methods are applied in scenarios, where the feedback is present after a certain operation steps. For reinforcement learning methods there is no immediate input-output training data available. When a feedback is specified to a reinforcement learning system, it adjusts its internal representation according to the feedback value. An important feature of reinforcement learning methods is that the outcome of the operation is based on decisions in the past. The training should affect also those components, which were responsible for past decisions.

A prominent example of supervised learning is the error back-propagation [43]. Error back-propagation is a common training technique of multi-layered neural networks. Its mechanism is to adjust the synapse weights based on the error in the output of the network. Error is propagated back in the layers of the network by the influence of the neurons on the final error.

An example of training synapse weights in the case of unsupervised learning is Hebbian learning. Hebbian learning [44] is based on the Hebbian theory in neuroscience. In a nutshell, the rule can be explained as the synaptic binding increases between two neurons if the neurons are activated together. Equation (2) presents the formula of the Hebbian learning.

$$\Delta w_{ij} = F(y_i, x_j) \tag{2}$$

where $\Delta w_{ij}$ is the change of weight of neuron $j$ activating neuron $i$, $y_i$ is the postsynaptic signal of neuron $i$, $x_j$ is the presynaptic signal of neuron $j$ and $F$ is the learning function, e.g., $F(x, y) = \eta(xy)$.

*d) Hopfield Network:* Originally, Hopfield networks [45] are intended to be used as associative memory and noisy pattern recognition methods. Hopfield networks are basically recurrent artificial neural networks. The Hopfield network is not a feed-forward and is not a layered neural network. The two important properties of Hopfield networks are: the synaptic weights are symmetric ($w_{ij} = w_{ji}, \forall i, j$) and no unit has connection with itself ($w_{ii} = 0, \forall i$), where $w_{ij}$ denotes the synaptic influence of the neuron j to the input of neuron i. Equation (3) presents the network function applied in Hopfield networks.

$$s_i^{(H)}(x) = \begin{cases} 1 & : \sum_j w_{ij} s_j(x) \geq \theta_i \\ -1 & : \sum_j w_{ij} s_j(x) < \theta_i \end{cases} \tag{3}$$

where $s_i(x)$ is value of the network function of neuron $i$, $w_{ij}$ is the weight of neuron $j$ activating neuron $i$ and $\theta_i$.

A well known problem of Hopfield networks can be derived from its iterative training method. It is not guaranteed that Hopfield networks converge to the global optima.

To see a concrete example, Huang et al. [46] define a two layer artificial neural network. A layer is defined for the customers and a layer is defined for the items. The relations between the customers are calculated by demography similarity, the relations between the items are calculated by content similarity, the relations between the layers are defined by the purchase events. Huang et al. calculate the recommendations by applying the Hopfield Net algorithm. In their experiment, the network is not trained, the network is used only for calculation. Their evaluation results show increased recommendation quality.

*e) Boltzmann Machine:* A Boltzmann machine [47] is a stochastic recurrent neural network. Boltzmann machines consist of stochastic neurons. The neurons can be in either in the state 1 or 0. The Boltzmann machine defines symmetric synaptic connections between its neurons. The neurons of a Boltzmann machine are divided into two classes, as visible and hidden neurons. The visible neurons receive the training input, the hidden neurons develop the internal representation based on the training samples. The task of the hidden neurons is to explain the underlying constraints of the visible neurons. Boltzmann Machines are trained with the gradient descent method.

Salakhutdinov et al. [48] utilize the Restricted Boltzmann machine to generate recommendations. Salakhutdinov et al. create a separate Boltzmann machine for each user. The visible nodes represent recommendable items, the hidden nodes represent features developed by the Boltzmann machine. Salakhutdinov et al. apply a conditional multinomial distribution in their a pivot model, and enhance the pivot model by modelling the distribution of the hidden units with Gaussian distribution and finally introduce the Conditional Restricted Boltzmann Machine to incorporate additional information into their model. In the latter case, the model can be treated as a multilayer stochastic neural network. Salakhutdinov et al. define the calculation of recommendations as a prediction on a probabilistic graph model. To generate a prediction, they join all Restricted Boltzmann Machines to generate the estimation for each rating value. By joining the restricted Boltzmann machines developed on a per user basis, Salakhutdinov come to a large stochastic neural network, where the joint distributions are trained on an individual level. The final rating value is generated by calculating an estimated value of the estimated distribution on rating values. Referring to their results, Salakhutdinov et al. managed to reach an error rate 6% better than the score of Netflix's own system.

### C. Hybridization

The basic motivation behind hybridization techniques is to alloy different recommendation methods in order to eliminate their drawbacks and to combine their strengths. A comprehensive study about various hybridization techniques is provided by Burke et al. [49]. In their work, Burke et al. define the classic hybridization techniques as weighted, switching, mixed, feature combination, cascade, feature augmentation and meta-level hybridization. Weighted hybridization combines the preference estimations of various techniques to produce recommendations. A state of the art example of the weighted hybridization is presented by Dooms et al. [50]. Dooms et al. claim that their recommender system is also tested against real-world requirements. Switching hybridization switches between the various methods based on the context. Mixed hybridization present the combination of recommendation lists of multiple methods to the user. Feature combination merges the features found in different datasets for a particular method. Cascading hybridization follows a sequential architecture of recommendation methods, where the output of a technique acts as the input of the next technique in the pipeline. Feature augmentation based hybridization is similar to the cascade method but the output of the preceding method are features and not recommendations. In the case of the meta-level hybridization, the trained model is shared between the methods.

To summarize the classic recommendation techniques, typically, the hybridization is conducted at the calculation level. The information shared between the methods are the recommendations themselves or the trained model of the particular method. The exception is the feature combination hybridization, where the information sources are alloyed at the information representation level. This is the point where our work is related to the results of Burke et al. Our goal is to provide a general information representation technique to alloy heterogeneous information sources, thus to define the hybridization at the information representation level. This is the reason why the methods operating on the alloyed information sources are inherently hybrid methods.

The knowledge graph is capable to incorporate heterogeneous information sources and to act as the background of hybridization at the representation level. Kouki et al. [25] utilize the knowledge graph to combine multiple different sources into a single unified model. Burke et al. [28] define a heterogeneous graph calculating the recommendations with the help of two dimensional projections of the graph. In contrast to the representation method of Kouki et al., we provide a more general definition of the knowledge graph and examine the performance of the calculation methods in this more general environment. Referring to the results of Burke et al., the information is modelled in a heterogeneous knowledge graph but in our case, the recommendation methods operate clearly on the graph.

## III. Representation

This section discusses the information representation method. Our intention is to define a modelling technique, which (i) is general, (ii) is capable to model heterogeneous information sources, (iii) defines a hybridization at the representation level and (iv) has the potential to eliminate the cold start problem.

By definition, the generalization of the information representation should cover the major classic recommendation classes as collaborative filtering, content-based filtering and knowledge based approaches. The question is how the combination of the information sources is modelled in the knowledge graph. Section III-A covers this topic by discussing each classic case and also how the information representation technique alloys the different knowledge representation approaches.

An important consequence of working with generalized representation is the increase in the amount of information to be processed by the calculation methods. Referring to Section II-A, based on the experiments in the past, additional information sources typically increase the recommendation quality. Looking at the problem from the opposite point of view, the more information sources are modelled, the less the calculation methods are constrained by the lack of information. Based on an information rich knowledge base, the calculation methods have the possibility to select the useful information sources and avoid the misleading ones. Following this strategy, our intention is to represent as much information as possible, in order not to constrain the recommendation methods in achieving high coverage.

The increase in the amount of information managed by the knowledge graph also leads to a possible decrease in the number of cold start cases. Graph based calculation methods are indirectly or directly based on the existence of a path or paths between the user and the item in question. Increasing the amount of information sources leads to a more dense graph meaning a higher probability of the existence of a path between two specific nodes.

Generalizing the information sources means operating with heterogeneous information. In the beginning of our research [51] the question has been posed if is it even possible to deliver recommendations based on heterogeneous information sources. Our past results [3], [1] prove that combining heterogeneous information sources leads to an increase in the recommendation quality, as the error rate converges faster, the coverage is higher and top list recommendations are more relevant to the user than in the case of the benchmark methods.

## *A. Classic Methods & Hybridization*

A widely known categorization of recommender systems also described by Jannach et al. [52] introduces three main categories of recommender systems: collaborative filtering, content-based filtering and knowledge-based methods, which categorization we refer to as the classic methods. Jannach et al. also describe various hybridization techniques. These techniques combine different recommendation methods to raise the efficiency of the individual methods. As the main advantage of the graph based representation is the capability to model various information representation scenarios, in order to demonstrate its potential, in this section we show how the information source of classic recommendation cases are to be represented in a graph and also provide an example of alloying the discussed representation cases.

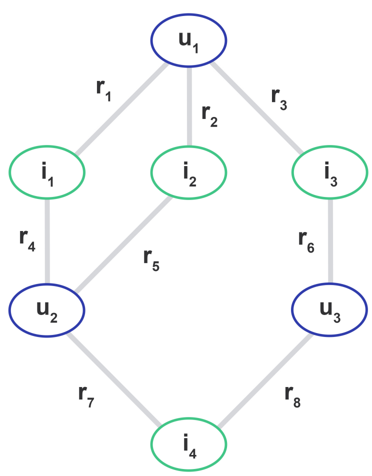


Figure 1. Collaborative Filtering Network

*1) Collaborative Filtering:* The idea behind collaborative filtering is to find people similar to the user the recommendations are generated for. The similarity is based on the rating values based on the commonly rated items. Having the similar users found, the next step is to calculate a rating estimation for the items rated only by the similar users but not rated by the user in question. The rating is a weighted average of the known rating values issued by the similar users. The weights are the similarities. An important property of collaborative filtering is that the method works with one type of relation between the entities, the rating relation. Despite its simplicity, collaborative filtering is proven to show high performance compared to other recommendation techniques. The weakness of the method is the information sparse case, when the knowledge base does not contain a sufficient number of user interactions to calculate the recommendations from.

Figure 1 presents a sample scenario of the graph based representation of the information necessary to produce recommendations utilizing the collaborative filtering technique. The graph contains users and items in its nodes and ratings in its relations. The nodes denoted with $u$ represent the users. The nodes denoted with $i$ represent the items. The relations denoted with $r$ represent the ratings of the specific user on the specific rating.

In the sample scenario we would like to find recommended items for $u_1$. In our case $u_1$ expressed interest on $i_1$, $i_2$ and $i_3$. In the graph based case, the similar taste is represented in the graph by expressed interest on the same items. As $u_1$ is interested in $i_1$ and $i_2$ and $u_2$ is also interested in $i_1$ and $i_2$, the users can be treated similar. The degree of the similarity depends on the rating values assigned to $r_1$, $r_2$, $r_4$ and $r_5$. Similarly, $u_1$ is interested in $i_3$, $u_3$ is also interested in $i_3$, the users can also be treated similar. The degree of similarity in this case depends on the value assigned to $r_3$ and $r_6$. To calculate the rating estimation for $u_1$ on $i_4$, the $r_7$ and the $r_8$ is to be averaged. The weights are determined by the degree of similarity.

In the graph based case, this relation does not necessarily have to be a rating relation. In our research, we work with the more general concept of user interest, which covers, e.g., a purchase event, a rating event or a clicked on event. The concrete type of interest depends on the application domain.

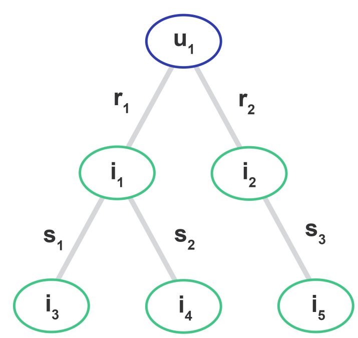


Figure 2. Content-based Filtering Network

*2) Content-based Filtering:* A general content-based filtering method operates with two types of relationships. In the first step, content-based filtering finds the items the particular user already expressed interest in. Then, in the second step, the methods recommends items, which are similar to the already known items. To accomplish this task, content-based filtering needs a similarity or distance measure between the items.

To simplify the explanation, the interest in our example is concreted as the user rating relation. Similarly to the collaborative filtering case, Figure 2 shows a sample scenario of a content-based filtering recommendation case. The graph contains users and items in its nodes and ratings and similarities in its relations. The nodes denoted with $u$ represent the users. The nodes denoted with $i$ represent the items. The relations denoted with $r$ represent the ratings of the specific user on the specific item. The relations denoted with $s$ represent the similarity of the items the relation starts and ends at.

To recommend items for $u_1$, the items the user already expressed interest in can be found by following the rating relations $r_1$ and $r_2$ and come to items $i_1$ and $i_2$. The similar items are to be found by the relations $s_1$, $s_2$ and $s_3$. Finally, the method recommends items $i_1$, $i_2$ and $i_3$. The preference order of the recommended items depends on the concrete rating and similarity values.

Looking at the scenario above, in order to generate recommendations, content-based filtering requires both user interaction and information about the items to recommend. Comparing content-based filtering to collaborative filtering, the latter method requires less user interaction in order to be able to deliver recommendations.

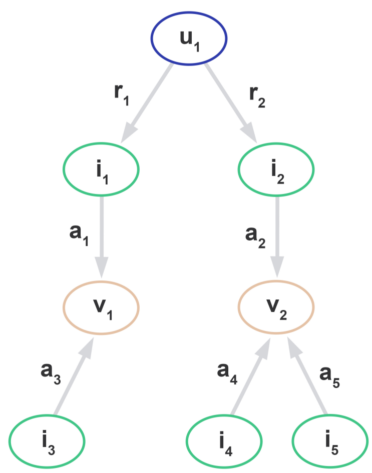


Figure 3. Knowledge-based Network

*3) Knowledge-based Recommendation:* The applications of quantitative decision support [53], reasoning [54] and decision tree [55] can also be treated as knowledge-based recommendation techniques. In order to conduct knowledge-based recommendation methods, available knowledge on the application domain should be present.

The application of knowledge-based techniques typically requires the users to specify their interest via interaction in order to avoid the cold-start problem. Neidhardt et al [56] requires the users to specify their travel destination interests by selecting multiple pictures representing the factors of travel destination representation. These methods typically also involve a user interface solution to acquire the required information. An illustrative example of user interaction is the constraint based recommendation [57], e.g., the approach of Felfernig et al. [58]. Regarding its methods, knowledge-based recommendation is a diverse and heterogeneous area. Although graphs are suitable to represent decision tree and rule based system based recommendation cases, we do not cover this direction. In our work we focus on representing user attributes, item attributes and domain knowledge.

Figure 3 illustrates a sample scenario of a knowledge-based recommendation method. The graph contains nodes representing users, items and attribute values. The node denoted with $u$ represent the user. The nodes denoted with $i$ represent the items. The nodes denoted with $v$ represent the attribute values. Our representation technique assigns a node to each attribute value. We refer to these nodes as attribute nodes. Attribute value nodes represent the concrete values of attribute nodes. For example, in order to represent cloth sizes, the knowledge base should contain a node for size “S”, “M” and “L”, respectively. The relations denoted with $r$ represent the ratings of the specific user on the specific item. The relations denoted with $a$ represent that the specific item has a the specific attribute value.

In the sample scenario we would like to find recommended items for $u_1$. As the figure shows, $u_1$ expressed interest in $i_1$ and $i_2$. Items $i_1$ and $i_3$ share the attribute value $v_1$. Items $i_2$, $i_4$ and $i_5$ share the attribute value $v_2$. The preference order of the recommended items depends on the concrete rating values and also on the calculation method whether it takes into consideration the number of outgoing edges of attribute nodes.

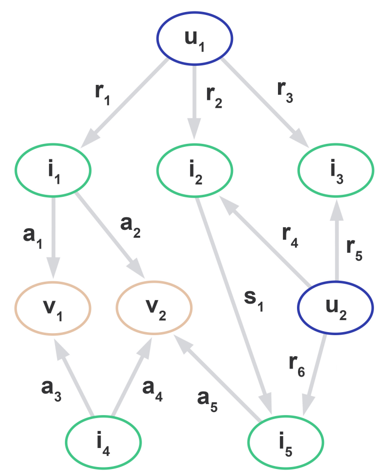


Figure 4. Hybrid Network

*4) Hybridization:* The primary goal of the hybridization techniques is to combine different recommendation methods to achieve better performance and to avoid the pitfalls, e.g., the cold-start effect. To accomplish this task, various hybridization techniques were evolved in the past decades of recommender systems research. To mention some techniques, the scoring of the source engines can be combined by for example weighting, switching or mixing. More advanced techniques are feature combination, cascading, feature augmentation or meta-level hybridization [49]. In our research, we work with a hybridization technique, which incorporates the existing information at the information representation level. We assume that more general calculation methods can be developed by moving the hybridization technique to the information representation level.

Figure 4 presents a sample network representing heterogeneous information. In this example we show how to merge the information types presented in the previous 3 sections. The graph contains nodes representing users, items and attribute values. The nodes denoted with $u$ represent the users. The nodes denoted with $i$ represent the items. The nodes denoted with $v$ represent the attribute values. The edges denoted with $r$ represent the ratings of the specific user on the specific item. The edges denoted with $a$ represent that the specific item has a the specific attribute value. The edges denoted with $s$ represent the similarity of the items the relation starts and ends at.

In the example, we would like to recommend items for $u_1$. The user expressed interest in items $i_1$, $i_2$ and $i_3$. Items $i_1$ and $i_4$ share the attribute values $v_1$ and $v_2$. Items $i_1$ and $i_5$ share the attribute value $v_2$. Users $u_1$ and $u_2$ share their interest on items $i_2$ and $i_3$. The similarity of $i_2$ and $i_5$ is represented with $s_1$.

The paths between $u_1$ and $i_4$ ($r_1, a_1, a_3$ and $r_1, a_2, a_4$) represent a knowledge-based recommendation case. The paths between $u_1$ and $i_5$ are more heterogeneous. The path $r_1, a_2, a_5$ represents a knowledge-based case. The path $r_2, s_1$ represents a content-based case. The paths $r_2, r_4, r_6$ and $r_3, r_5, r_6$ represent two collaborative filtering cases. The example shows a recommendation scenario based on heterogeneous information. Moreover, the graph incorporates the information types necessary to conduct classic recommendation techniques.

### B. An Information Rich Example

The previous section covered the information sources of the classic methods. In this section we would like to present an imaginary scenario, the recommendation case of perfumes. The example presents a more complex case illustrating the possibilities of the graph based representation.

Due to personal reasons, Eve wants to try a new perfume in order to replace her current one, Orienta. She already did a research. The two possible candidates are Pinky and Fracca. Asking her boyfriend, Peter, he recommends Pinky. She also asks her friend, Irene but she has no experience with the mentioned fragrances. To help Eve, Irene asks her friends, Petra and Sarah. Petra recommends Pinky. Sarah prefers Fracca. As Eve wants to make a substantial decision, she further analyses the products to find out that Fracca is produced in Paris and Orienta is produced in Angers. To generalize it, these perfumes are produced in France. In addition to other components, Orienta contains Musk and Amber. Eve prefers Musk to Amber. Musk can also be found in Pinky and Amber can also be found in Fracca. [51]

Figure 5 visualizes the perfume scenario. The recommendation case contains various types of information. The dark blue edges represent the social network of the example. The social relationships can also be detailed, as the “in relation” and “friends” relation types show. The persons involved are Eve, Peter, Irene, Petra and Sarah. Referring to social influence, the relations between Eve and Irene and then Irene and Petra illustrate the transitivity of the social relations. Relations of type “like” and “dislike” illustrate the user-item interactions, which is an example of the collaborative case. The relations of type “component of”, “produced in” and “located in” stand for the example of knowledge based recommendations. The “ARM” type stands for the association rule mining, illustrating that aggregated information can also be involved into the recommendation process. To conclude it, the example presents a combination of information sources of social networks, collaborative methods, knowledge-base methods and association rule mining.

### C. Definition

Based on the issues discussed in the previous sections, this section presents the definition of the information representation method, as we refer to it, the knowledge graph. Depending on the application scenario, the definition is provided for both the directed and the undirected case. The knowledge graph is a labelled, weighted, restricted multi-graph.

*1) The Directed Case:* Equation (4) presents the definition of the directed knowledge graph.

$$\mathcal{K}_d = (T_n, T_e, N, E_d, t_n, t_{e_d}, r_{e_d}), \tag{4}$$

where $N$ denotes the set of nodes existing in the graph. $E_d \subseteq \{(u, v)|u \in N \wedge v \in N \wedge u \neq v\}$ denotes the set of directed edges between the nodes. $T_n$ denotes the set of node types. $T_e$ denotes the set of edge types. The function $t_n \subset N \times T_n$ assigns a node type to each node. The function $t_{e_d} \subset E_d \times T_e$ assigns an edge type to each edge. The partial function $r_{e_d} \subset E_d \times \mathbb{R}$ assigns a rating value to some of the edges.

Equation (5) introduces the set of directed edges of type rating.

$$E_{d,\texttt{Rating}} = \{e \in E_d | t_{e_d}(e) = \texttt{Rating}\} \tag{5}$$

*2) The Undirected Case:* Equation (6) presents the definition of the undirected knowledge graph.

$$\mathcal{K}_u = (T_n, T_e, N, E_u, t_n, t_{e_u}, r_{e_u}), \tag{6}$$

where $E_u \subseteq \{\{u, v\}|u \in N \wedge v \in N \wedge u \neq v\}$ denotes the set of undirected edges between the nodes. The function $t_{e_u} \subset E_u \times T_e$ assigns an edge type to each edge. The partial function $r_{e_u} \subset E_u \times \mathbb{R}$ assigns a rating value to some of the edges.

Equation (7) introduces the set of directed edges of type rating.

$$E_{u,\texttt{Rating}} = \{e \in E_u | t_{e_u}(e) = \texttt{Rating}\} \tag{7}$$

*3) Further Definitions:* Equation (8) introduces the set of nodes of type `Person`. Equation (9) introduces the set of nodes of type `Item`.

$$N_{Person} = \{n \in N | t_n(n) = \texttt{Person}\} \tag{8}$$

$$N_{Item} = \{n \in N | t_n(n) = \texttt{Item}\} \tag{9}$$

At the moment, type assignments do not influence the final recommendation result and are introduced for completeness and further research.

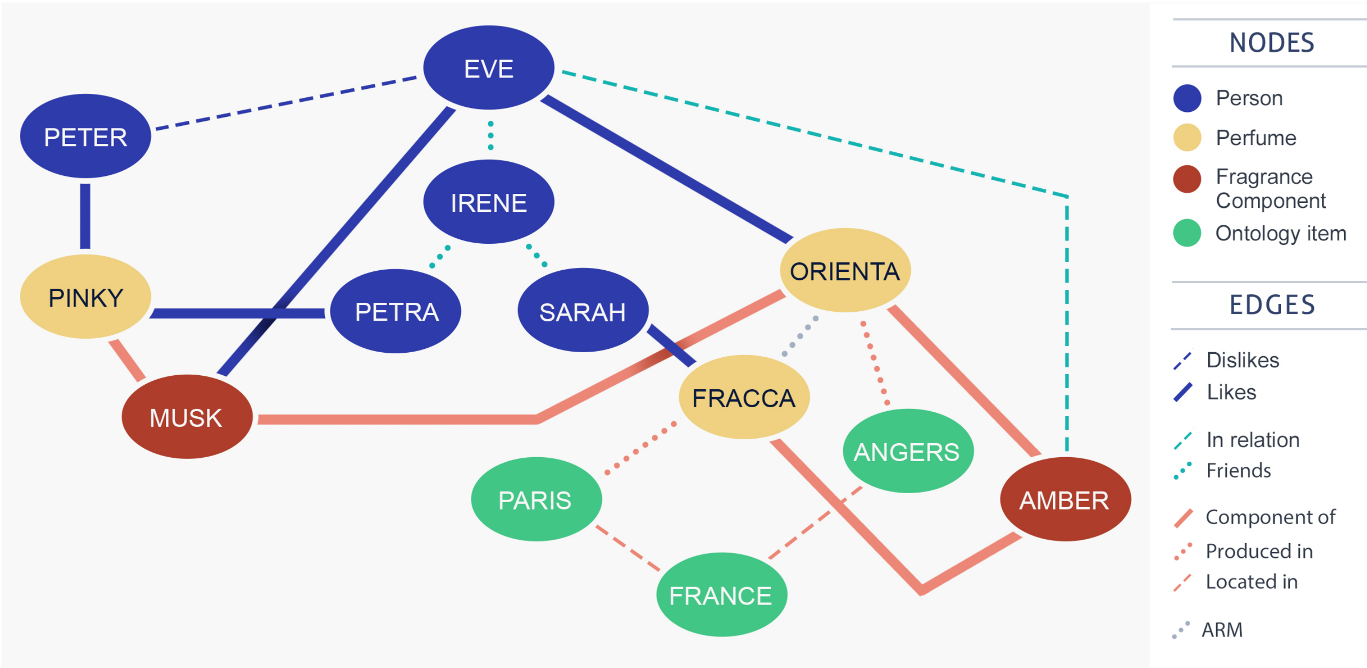


Figure 5. A Heterogeneous Recommendation Scenario

### D. The Balance of the Hybridization

To summarize the previous sections, the knowledge graph is capable to represent the information necessary to conduct both collaborative filtering, content-based filtering and knowledge based recommendation. The hybridization of the information sources is conducted at the representation level.

Looking at the origin of the information, the information sources can be classified as static and user generated. Static information is defined as the content-based and the knowledge based information. This information is typically supplied by knowledge workers of the software system the recommender system is attached to. The user generated information is described as the information derived from the interaction of the users with the software system (in most cases the items).

TABLE I. Count of graph edges by type in the MovieLens dataset

| Edge type | Count |
|---|---|
| PersonAgeCategory | 6 040 |
| PersonGender | 6 040 |
| PersonOccupation | 6 040 |
| PersonZipCodeRegion | 6 040 |
| ItemGenre | 6 408 |
| ItemYearOfPublishing | 3 883 |
| ItemRating | 1 000 209 |

To illustrate the amount of information the calculation methods operate on, Table I presents the amount of edges by edge type in the MovieLens 1M dataset [6] represented in the knowledge graph as described in one of our previous papers [3]. The total number of static edges is 34 451. The total number of user generated edges is 1 000 209. The example shows that the magnitude of the amount of the user generated edges is significantly higher than the magnitude of the amount of the static edges, which case is typical in the case of real-world applications.

Referring to one of our principles, a properly defined calculation method should treat the information sources as general. The method should not distinguish the information source types at the algorithmic level. If the calculation method meets this condition, the behaviour of the method depends on the magnitude of the amount of information found in the knowledge graph. In the information sparse case, when the knowledge graph is sparse on user generated information and relatively rich on static information, the calculation method tends to be more content-based or knowledge-based. As the users start to interact with the system, the user generated content will be emphasized and the calculation methods will behave more like the collaborative methods.

The main advantage of the knowledge graph can be described from this aspect. If there is not enough user generated information available, the methods automatically focus on the static information. In the case of a vast of user interaction in the graph, the behaviour of the methods become more collaborative like.

## IV. Methods

This section provides definition of the methods in the knowledge graph based case. The experiments conducted are organized to evaluate the spreading activation, network science based methods and the performance of the human expert. The methods collaborative filtering and recommendation spreading are also covered, in order to illustrate the paradigm of relatedness [2]. Some methods deliver personalized recommendations, some methods are non-personalized. The involvement of non-personalized methods is discussed in Section VI.

### A. Definitions

*1) Collaborative Filtering:* Figure 6 illustrates the collaborative filtering method [59] in the graph based case.

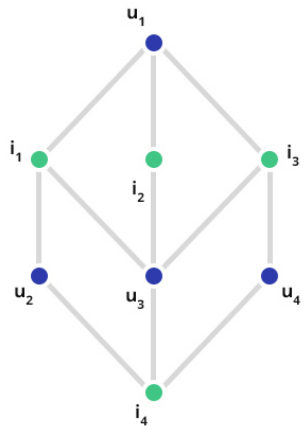


Figure 6. Collaborative Filtering

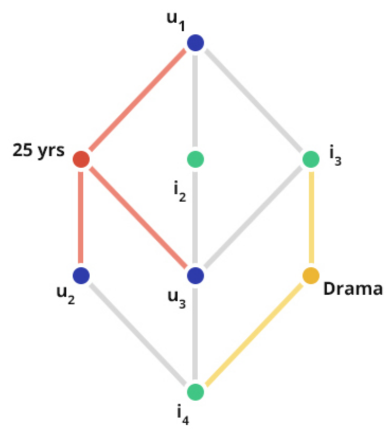


Figure 7. Spreading Activation

The graph contains users and items. Nodes denoted with $u$ represent the users. Nodes denoted with $i$ represent the items. Edges represent the user ratings on the items. As collaborative filtering derives its calculations based on rating values, only the edges of type `Rating` are necessary to calculate the rating estimation. These edges are denoted with grey colour.

In the graph based case, collaborative filtering can be calculated in two phases. In the first phase, the users having a common rated item with the user in question should be selected. In the second phase, the rating estimations are to be calculated for the items rated by the selected users but not by the user in question.

Equation (10) provides the definition of the rating estimation formula of the collaborative filtering method in the case of the undirected knowledge graph, $\mathcal{K}_u$. The rating estimation is provided for user $u$ on item $i$.

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{e \in E_{u,\mathtt{Rating}}, \{v,i\}=e, u \neq v} (r(e) - \bar{r_v}) s_{u,v}}{\sum_{e \in E_{u,\mathtt{Rating}}, \{v,i\}=e, u \neq v} s_{u,v}}, \quad (10)$$

where $\hat{r}_{u,i}$ denotes the estimated rating value for user $u, u \in N_{\mathtt{Person}}$ on item $i, i \in N_{\mathtt{Item}}$. $\bar{r}_u$ denotes the average of the already issued ratings by user $u$, $u \in N_{\mathtt{Person}}$. $s_{u,v}$ denotes the similarity between user $u$ and $v$, $u \in N_{\mathtt{Person}}$, $v \in N_{\mathtt{Person}}$ and is defined as the Pearson similarity of the issued ratings by the users on the common rated items.

To summarize it, collaborative filtering is a rating estimation method. Its essence is to calculate a weighted average of the known rating values, where the weight of a specific rating is defined as the similarity of the issuer of the rating to the user in question. Another important feature of the collaborative filtering method is that it averages the rating values after subtracting the average of the ratings issued by the user from each rating value.

*2) Spreading Activation:* Figure 7 illustrates the spreading activation method [33]. The example graph contains users, user attributes, items and item attributes. Nodes denoted with $u$ represent the users. Nodes denoted with $i$ represent the items. The node with caption "25 yrs" illustrates a concrete user attribute value. The node with caption "Drama" illustrates a concrete item attribute value. Edges of colour gray represent the user ratings on the items. Edges of colour red represent that the specific user has the specific user attribute value. Edges of colour yellow represent that the specific item has the specific item attribute value. In contrast to collaborative filtering, spreading activation utilizes all edges in the knowledge graph regardless of their respective type.

As already mentioned in Section II-B3, spreading activation operates on a graph. To calculate recommendations with the method on the undirected knowledge graph ($\mathcal{K}_u$), an activation value is calculated for the nodes of the graph. Equation (11) defines the activation score function.

$$a_{(k)} \subset N \times \mathbf{R}, \quad (11)$$

where $k$ denotes the iteration step.

In the initial step of the calculation, the activation of the node representing the user the recommendations are to be generated for is set to 1. The activation of all the other nodes is set to 0 as described in Equation (12).

$$a_{(0)}(n) = \begin{cases} 1 & : n = n_s \\ 0 & : n \neq n_s \end{cases} \quad (12)$$

where $n_s$ denotes the node represents the user to generate the recommendations for. This node is to be referred as the source node.

Spreading activation is an iterative method. In each iteration step, a part of the activation is kept at the node, and another part of the activation is propagated to its neighbours. The former amount is determined by the `activation relax` ($r_a$) parameter. The latter amount is determined by the `spreading relax` ($r_s$) parameter. The propagated activation is divided among the neighbour nodes equally. Equation (13) defines the propagation of the activation.

$$a_{(k+1)}(n) = r_a a_{(k)}(n) + r_s \sum_{m \in M_n} \frac{a_{(k)}(m)}{z_m}, \quad (13)$$

where $k$ denotes the iteration step, $k > 0$. $M_n$ denotes the neighbour nodes of $n$, $M_n = \{m | \{m,n\} \in E_u\}$. $z_m$ denotes the count of neighbours of $m$, $z_m = |\{p | \{m,p\} \in E_u\}|$.

The termination criteria is step based. If the iteration reaches the specified `step limit` ($c$), the propagation stops.

To summarize it, spreading activation basically calculates a proximity value in the graph. As the activation is relaxed with the length of the path between two nodes, the activation value of a node depends on its distance from the source node. As the parallel paths between two nodes result in the accumulation of the activation spreading to the node, spreading activation takes the structure of the graph also into consideration.

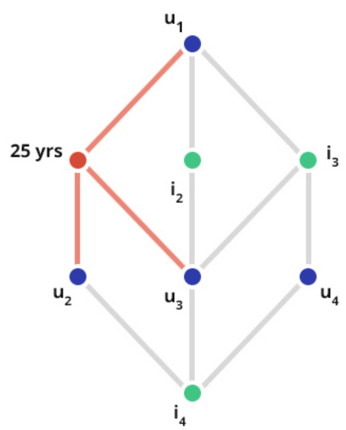


Figure 8. Recommendation Spreading

*3) Recommendation Spreading:* Figure 8 illustrates the recommendation spreading method introduced by Grad-Gyenge et al. [3]. The example graph contains users, user attributes and items. Nodes denoted with $u$ represent the users. Nodes denoted with $i$ represent the items. The node with caption "25 yrs" illustrates a concrete user attribute value. Edges of colour gray represent the user ratings on the items. Edges of colour red represent that the specific user has the specific user attribute value. In contrast to spreading activation, recommendation spreading utilizes all types of edges in the knowledge graph with one constraint. The last edge of the path leading to the recommended item should have the type `Rating`.

Recommendation spreading operates on the undirected knowledge graph ($\mathcal{K}_u$). The method is basically the generalization of the collaborative filtering for the graph based case. The method modifies the classic collaborative filtering formula by defining the weights of the averaged rating values based on the structure of the graph. Recommendation spreading is a spreading activation based method running the same iteration to propagate the activation values. During the iteration, the activation flown through each edge is summarized. Equation (14) defines the weights of the rating values.

$$A_e = \sum_{i \in [0,c-1], n \in e, n \in N_{\texttt{Person}}} r_s \frac{a_{(i)}(n)}{z_n}, \tag{14}$$

where $e$ denotes the edge to summarize the flown through activations for, $e \in E_{u,\texttt{Rating}}$.

To estimate the rating values, recommendation spreading uses the sum of the flown through activations as the weight of the rating value of the corresponding edge. Equation (15) provides the formula of recommendation spreading.

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{e \in E_{u,\texttt{Rating}}, \{v,i\}=e, u \neq v} (r(e) - \bar{r_v}) A_e}{\sum_{e \in E_{u,\texttt{Rating}}, \{v,i\}=e, u \neq v} A_e}. \tag{15}$$

*4) Network Science:* In the cold-start case, when personalized recommendation is not possible due to the lack of information about the user in question, non-personalized recommendation is necessary. Non-personalized recommendation can be conducted e.g, by presenting top selling or hand-picked items. Network science centrality measures provide a possible alternative to the mentioned methods.

Network science [36] developed several centrality measures to assign a number to the nodes of a graph representing the importance of each node based on its position. In our experiments, degree centrality, closeness centrality, betweenness centrality and eigenvector centrality based recommendations are analysed. The mentioned measures are introduced in Section II-B4.

To calculate non-personalized recommendations, the involved network science centrality measures are calculated on the directed knowledge graph ($\mathcal{K}_d$). The relevance order of the nodes is determined by the centrality score of the nodes.

*5) Human Expert:* Human expert based recommendations are also involved in the experiment, to act as the baseline method. As human expert based personalization is not feasible due to capacity and financial constraints, the method is managed as non-personalized. As described in Section VI, the evaluation of the methods is organized into campaigns. The human expert provided a list of recommended items, thus the same list of items is offered to all users in the same campaign. The recommended items are selected based on publicly available best seller lists, domain knowledge and market experience.

## B. The Paradigm of Relatedness

Looking at Figure 6, Figure 8 and Figure 7, collaborative filtering, recommendation spreading and spreading activation can be compared from the aspect of constraints on the paths between the nodes representing the entities of the recommendation scenario. To illustrate the constraints on the paths, the edges of different types are denoted with different colours.

Collaborative filtering essentially operates only with relations of type `Rating`. In the case of recommendation spreading, the edges in the path between the users and the items are less constrained. There is no constraint on the type of the edges except for the last edge of the path, which edge has to be the type of `Rating`. Spreading activation defines no constraint on the type of the edges in the paths.

The reason behind the existence of these constraints can be found in the different purpose of the methods. The goal of collaborative filtering and recommendation spreading is to generate rating estimations. To generalize the concept "user ratings", we introduce the concept "user interest". User interest can be any explicit or implicit user interaction on the items. Looking at representation learning (e.g., SVD) based methods, these methods find the latent factors of the user interest space. In other words, these methods are based on and are constrained by the paradigm of user interest.

The goal of spreading activation is to assign preference scores to the nodes. To achieve this task, spreading activation operates on the generalized knowledge base, where different types of relations are represented at the same generalization level. Looking at the entities of the recommendation scenario represented as the nodes of the graph, we introduce the concept of relatedness, the relatedness of the entities of the recommendation scenario. The concept of relatedness incorporates two important features, the generalization and the transitivity. The generalization has already been discussed. To illustrate the transitivity of the relatedness with concrete examples, an item can be recommended because it is similar to an already bought item or because the friend of the user likes it. As spreading activation is general and is also transitive regarding its calculations, it is based on the paradigm of relatedness. To provide further explanation, Grad-Gyenge et al. [2] demonstrate the paradigm of relatedness on the MovieLens dataset [6].

## V. Dataset

During the evaluation period, the portal offered 117 367 books to its visitors. The electronic commerce system is based on a relational database, thus its data is to be transferred and transformed into the knowledge graph. To fulfil this task, the portal is integrated with PartnerMail via its web based API. The PartnerMail API provides methods to manage the knowledge graph by inserting, updating and deleting the nodes and the relations of the graph. Based on the integration, the user data, item data and user interactions are transferred to the knowledge graph in real-time.

The database of Booker [9] contains users, books, user attributes, book attribues, user interactions and the book category tree. The attributes of the users are hometown and birth year. These attributes are specified by the user and are not mandatory fields. Book attributes are author, publisher, year of publishing, number of pages and price. These attributes are specified by the distributor of the books and are typically specified. The books are organized into a tree-like category structure by the knowledge workers of the portal. The portal also contains a wish-list feature. With the help of the wish-list, users can maintain a list of books intending to buy.

To model the available information in the knowledge graph, each user and each item is represented with a node. As also discussed in Section III-A2, a node is created for each attribute value. The nodes representing the users and the items are connected to these nodes representing attribute values with an edge of the appropriate type. In some cases, an item can have multiple attribute values. In this case, the item is connected to multiple attribute nodes. Such case is for example a book having multiple authors. The representation of the category tree is straightforward in the graph based case. A node is created for each category and each subcategory. The nodes are connected then with the edges of the appropriate type.

In order to cover also the user interactions, different kind of relations between the users and the books are stored in the knowledge base. The user interactions can be classified as expressing explicit or implicit interest. For example, if a user visits the web page containing the description of a book, an edge is inserted into the knowledge graph to represent the implicit interest. If a user puts items to their wish-list, the inserted edge represent an explicitly expressed interest.

The information is represented in the knowledge base as defined in Section III. Table II presents the node types introduced to store the entities of the recommendation scenario and also the occurrence counts of the nodes. Nodes of type `Person` represent the users of the system, as customers and persons signed up to the newsletter. Nodes of type `HomeTown` represent the hometown of the persons. A node is created for each hometown. Nodes of type `BirthYear` represent the birth year of the persons, e.g., `1978`. A node is created for each birth year. Nodes of type `Item` represent the books offered to the users, e.g., `Manga` and `Hieronymus Bosch`. Nodes of type `Author` represent the authors of the books, e.g., `Kurt Vonnegut` and `John Updike`. A book can have multiple authors. Nodes of type `Publisher` represent the publisher of the bookes, e.g., `Osiris Publishing` and `A & C Black`. Nodes of type `YearOfPublishing` represent the year of publishing of the books, e.g., `2007`. Based on the consultations with the experts of Booker, the number of pages of the books is organized into categories. The following categories are defined `0-60`, `61-100`, `101-200`, `201-500` and `501-1000`. Nodes of type `NumberOfPages` these categories. Similarly to page number categories, price categories are also defined as `0-1000`, `0-3000`, `1001-3000`, `3001-6000` and `6001-10000`. As the defined categories are overlapping, a book can belong to multiple categories, similarly like in the case of the author. Nodes of type `ItemCategory` represent the main and subcategories defined by the knowledge engineers of Booker. Such categories are for example `travel`, `art` and `religion`.

TABLE II. Node Types and Occurrences.

| *Type* | *Count* |
|---|---:|
| `Person` | 17 134 |
| `HomeTown` | 105 |
| `BirthYear` | 7 |
| `Item` | 117 367 |
| `Author` | 45 918 |
| `Publisher` | 6 351 |
| `YearOfPublishing` | 67 |
| `NumberOfPages` | 5 |
| `PriceCategory` | 5 |
| `ItemCategory` | 598 |

TABLE III. Edge Types and Occurrences.

| *Type* | *Count* |
|---|---:|
| `PersonBirthYear` | 8 |
| `PersonHomeTown` | 175 |
| `ItemAuthor` | 127 613 |
| `ItemCategory` | 30 800 |
| `ItemNumberOfPages` | 112 524 |
| `ItemPriceCategory` | 212 473 |
| `ItemPublisher` | 116 746 |
| `ItemYearOfPublishing` | 96 653 |
| `BoughtItem` | 22 064 |
| `OnWishList` | 2 972 |
| `ItemVisited` | 4 590 |
| `SubCategory` | 486 |

Table III presents the edge types introduced to store the relations between the entities of the recommendation scenario and also the occurrence counts of the edges.

Relations of type `PersonBirthYear` between nodes of type `Person` and nodes of type `PersonBirthYear` connects the specific persons to their birth year. Relations of type `PersonHomeTown` between nodes of type `Person` and nodes of type `HomeTown` connects the specific persons to their hometown. Relations of type `ItemAuthor` between nodes of type `Item` and nodes of type `Author` connects the specific book to its author or authors. Relations of type ItemCategory between nodes of type Item and nodes of type ItemCategory connects the books to their appropriate category or categories. Relations of type ItemNumberOfPages between nodes of type Item and nodes of type NumberOfPages connects the items to their appropriate page category or categories. Relations of type ItemPriceCategory between nodes of type Item and nodes of type PriceCategory connects the items to their appropriate price category or categories. Relations of type ItemPublisher between nodes of type Item and nodes of type Publisher represent connects the books to their publisher. Relations of type ItemYearOfPublishing between nodes of type Item and nodes of type YearOfPublishing connects the books to their year of publishing. Relations of type SubCategory between nodes of type ItemCategory connects the subcategories to their higher level category. Relations of type BoughtItem between nodes of type Person and nodes of type Item represent the purchase interaction. Relations of type OnWishList between nodes of type Person and nodes of type Item represent the wish-list interaction. Relations of type ItemVisited between nodes of type Person and nodes of type Item represent the interaction the user visited the web page of the book.

To interpret the amount of edges presented in Table III, in can be read that the knowledge base is rich on item attributes but is sparse on person attributes. The reason for it is the different source of the information. The item attributes are presented by the publishing companies, while the user attributes are described by the users. The wish-lists are also densely populated. The number of edges of type ItemAuthor is higher than the number of nodes of type Item. The difference illustrates that certain books have multiple authors. The difference between the count of edges of type ItemNumberOfPages and ItemPublisher shows that the item attributes are not specified for all items. The higher number of ItemPriceCategory is caused by the overlapped PriceCategory intervals.

An important drawback of the dataset is the relatively low amount of edges of type ItemVisited. The reason of this problem can be found in the organization of the purchase work-flow on the portal. In order to increase the click-through rates, Booker requires the users to log-in only at the end of the purchase process. This is the reason why users typically anonymously browse the content of the portal and the user interest of this type can not be logged.

As already mentioned, the electronic commerce system forwards the information to the knowledge graph in real-time. During the experiment, the amount of nodes and edges in the knowledge graph constantly increased. The amount of nodes and edges presented in Table II and Table III were recorded on 23 January, 2015.

## VI. Evaluation

The evaluation of the methods is based on the email based remarketing activity of the web portal Booker. The essence of remarketing is to provide offers to the visitors of the electronic commerce portal after their visit. In our experiment, the offered items are selected by the recommendation techniques and are sent in personalized emails to the visitors. Depending on the actual method, the list of the items can be personalized or non-personalized. The emails are organized into several newsletter campaigns.

As already mentioned in Section V, the electronic commerce system constantly updates the knowledge graph. The recommendation techniques operate on this knowledge graph and are invoked at the point of time a list of recommended items is required from the email sending and personalizing software PartnerMail. During the evaluation period, the click-through events of the users are logged. The evaluation of the methods is essentially based on the click-through rates.

### A. Newsletters Sent

The evaluation period has been conducted between 16 Jul, 2014 and 14 Jan, 2015. The newsletters are organized into 16 campaigns. To summarize the amount of evaluation cases, during the experiment, 241 062 emails have been sent of which 35 229 emails have been opened.

The campaigns can be classified as Recommender system based campaigns and Human expert based campaigns. In the case of a Recommender system based campaign, a personalized recommendation technique is invoked for each user. If the method is not able to generate a sufficient number of recommendations due to the cold-start case, a non-personalized technique is involved as a fall-back method. This technique ensures that an email can be sent to all contacts. In the case of the Human expert based case, the non-personalized, human expert based recommendation method is utilized.

TABLE IV. Newsletter send dates.

| *Type* | *Date sent* |
|---|---|
| Recommender System | 2014-07-16 |
| Human Expert | 2014-07-23 |
| Recommender System | 2014-07-26 |
| Recommender System | 2014-08-01 |
| Human Expert | 2014-08-06 |
| Human Expert | 2014-08-27 |
| Recommender System | 2014-08-29 |
| Recommender System | 2014-09-12 |
| Human Expert | 2014-09-22 |
| Recommender System | 2014-09-26 |
| Human Expert | 2014-10-02 |
| Human Expert | 2014-10-09 |
| Recommender System | 2014-10-15 |
| Human Expert | 2014-10-22 |
| Recommender System | 2014-10-31 |
| Recommender System | 2014-12-14 |

Table IV summarizes the newsletter campaigns. Column *type* presents the type of the campaign, column *Date sent* contains the send date. The date of the last campaign is 2014-12-14 and the end of the experiment is 2015-01-14. The additional month is involved in order to be able to measure the click-through events of the users.

### B. Methods

As the knowledge graph does not contain user rating on items, some of the recommendation techniques described in Section IV could not be involved into the evaluation process. To generate personalized recommendations, the spreading activation, as described in Section IV-A2, is utilized. The fall-back methods in this case are the centrality measure based techniques, as defined in Section IV-A4. The human expert based method is described in Section IV-A5. In this case no fall-back method is necessary, as the method is non-personalized.

Based on our past results [3], we focused on the different settings of the `step limit` parameter setting both the `spreading relax` and the `activation relax` to a constant value, 0.5. Human expert and network science based methods require no further configuration.

In the assembling phase of a campaign, the concrete method configurations have to be specified. It means that in a campaign only the specified method or methods are evaluated.

### C. Recommendation Lists

The methods described in Section IV assign a preference value to the nodes of the graph. To assemble a list of recommended items, the nodes are sorted in descending or by their preference value. Nodes with no preference value are not included in the list. The preference value is defined (i) as the rating estimation in the case of collaborative filtering and recommendation spreading, (ii) as the activation in the case of spreading activation and (iii) as the centrality value in the case of network science methods.

A relatively simple filtering technique is applied to the list of recommended items in order in increase user satisfaction. The rules are defined by the human expert and are summarized as (i) at most 2 books can be present from the same author, (ii) books are excluded from consecutive newsletters for two months and (iii) already purchased books are excluded. Having the list generated, the first $n$ items are selected and presented to the user.

### D. Evaluation Measures

User interaction is measured by various click-through events. These events are specific to email remarketing and basically model the work-flow of the purchase process. The following steps are modelled during the evaluation: sending a newsletter, opening a newsletter, clicking on an item in a newsletter, ordering an item and paying for the item. The states according to the process are identified as `Sent`, `Opened`, `Clicked`, `Ordered` and `Paid`.

In order to preserve the computational resources, the emails sent by PartnerMail do not embed the images into the content of the email but rather contain a reference to the images. A typical email client software does not download these referred images due to security and privacy concerns. The displaying of these images requires user interaction. The user can instruct the email client to download and display the images in the email. As the images are served by our server, PartnerMail records this event in the evaluation log.

The links in the emails contain a unique identifier and point to the PartnerMail server. If a user clicks on a link, the PartnerMail server logs the click-through event based on the unique identifier and forwards the user to the web page of the book on the Booker portal.

The books are to be ordered by the users on the web page of the specific book on the Booker portal. As the portal is integrated with PartnerMail, the order events are immediately forwarded to the knowledge graph. If the user cancels the current work-flow but returns later and finalizes the order, PartnerMail still records the appropriate click-through event.

Depending on the type of the payment method, the pay events can be separated from the order events. In the case of a credit card based payment, the payment immediately follows the order. In the case of the cash based payment, the customer orders the book and pays for it personally when physically picks up the books. In the latter case, a mentionable amount of work-flows do not reach the `paid` state, as it involves additional resources from the customer.

## VII. Results

In this section, the results of the evaluation described in Section VI are presented. At first, a broad overview of the remarketing emails is provided in numbers. Then, the conversion rates are analysed for each recommendation technique. Finally, the performance of the method configurations is discussed.

TABLE V. Count of newsletters per state and recommendation technique

| State | Spreading Activation | Network Science | Human Expert |
|---|---|---|---|
| Sent | **66 148** | **72 884** | 102 030 |
| Opened | 11 700 | 9 206 | 14 323 |
| Clicked | 1 265 | 260 | 772 |
| Ordered | 24 | 0 | 17 |
| Paid | **17** | 0 | **6** |

Table V presents the number of remarketing emails sent during the evaluation period. The rows of the table represent the state of the emails as described in Section VI-D. The columns of the table represent the recommendation technique as described in Section VI-B. The values of the table are the count of emails in the particular state generated with the particular recommendation technique.

Our primary finding is the increase in the performance of the emails in the case of the `Spreading Activation`. Comparing the number of emails in the `Paid` state of the `Spreading Activation` to the `Human Expert`, the `Spreading Activation` shows a higher number of the `Paid` cases despite of having a lower amount of `Sent` cases. In this broad overview, `Network Science` based methods show a poor performance, as the number of `Paid` recommendations in this case is 0.

The high number of cold start cases can be determined by analysing the number of the fall-back method based recommendation cases. Also looking at our past results [3], comparing the number of sent `Spreading Activation` cases to the number of sent `Network Science` cases, a relatively high portion (52%) is identified. Also consulting with the experts of the electronic commerce system, the reason behind the high amount of cold start cases can be found in the dataset. Several users signed up only for the newsletter. In their case, the knowledge graph does not contain sufficient information to generate the recommendations, as spreading activation is not able to find a path between the user and any of the items.

To further analyse the performance of the methods, the subtables of Table VI present the conversion rates of each technique. The rows of the tables contain the source states. The columns of the tables contain the destination states. The values of the table contain the measured conversion rates of the states of the emails.

Conversion rate based comparison provides a more clean picture. Comparing the performance of the `Spreading`

TABLE VI. Conversion rates of the recommendation techniques

(a) Spreading activation

| | Opened | Clicked | Ordered | Paid |
|---|---|---|---|---|
| Sent | 17.688% | 1.912% | 0.036% | 0.026% |
| Opened | | 10.812% | 0.205% | 0.145% |
| Clicked | | | 1.897% | 1.344% |
| Ordered | | | | 70.833% |

(b) Network Science

| | Opened | Clicked | Ordered | Paid |
|---|---|---|---|---|
| Sent | 12.631% | 0.357% | 0.000% | 0.000% |
| Opened | | 2.824% | 0.000% | 0.000% |
| Clicked | | | 0.000% | 0.000% |
| Ordered | | | | 0.000% |

(c) Human Expert

| | Opened | Clicked | Ordered | Paid |
|---|---|---|---|---|
| Sent | 14.038% | 0.757% | 0.017% | 0.006% |
| Opened | | 5.390% | 0.119% | 0.042% |
| Clicked | | | 2.202% | 0.777% |
| Ordered | | | | 35.294% |

Activation to performance of the Human Expert, 0.026% of the personalized emails resulted in a purchase event, while the ratio is 0.006% in the case of the Human Expert. Looking at the individual conversion rates, the Spreading Activation shows a higher performance in all cases, except the Clicked to Ordered conversion. To conclude it, personalized email remarketing has the potential to outperform the human expert based remarketing, thus the personalization has the potential to increase the business value.

Analysing the conversion rate from the Sent to the Opened state, the performance of the methods (17.688%, 12.631%, 14.038%) is different but the difference is significantly smaller than in the next conversion step. This is the step, where there users do not have the sufficient information to make the decisions based on the content of the emails. The conversion in this step is more based on the engagement of the users to the brand, the content has a low influence to this decision.

The Opened to Clicked conversion rate is influenced more by the content of the items. The state transition rates in this step (10.812%, 2.824%, 5.390%) better represent the performance of the methods, as the difference in the conversion rates is more visible. In this conversion, the Spreading Activation shows the highest performance, Human Expert shows approximately the half of its performance and Network Science delivers the lowest conversion rate.

The web page of a book provides the most detailed information about a recommended item, thus the Clicked to Ordered state transition is at most influenced by the information about the books. In this conversion step, the performance of the Network Science methods drops to 0 showing the low relevance of the items selected by this technique. The Collaborative Filtering shows a bit better performance than the Spreading Activation.

The last conversion step is the Ordered to the Paid transition. As the Network Science does not reach this conversion step, the performance of the Spreading Activation (70.833%) and the performance of the Human Expert (35.294%) is to be compared in this step. Similarly to the Opened to Clicked conversion, the performance of the Spreading Activation is two times the performance of the Human Expert. This conversion rate is the last step where the payment is to be performed. Regarding the decision making process, this conversion step does not involve information about the products from the user. The decision to perform at this point is emotion based driven by the finances.

To conclude the results, the spreading activation based recommendation technique outperforms the human expert in the conversion steps, where the human decision making is not analytical but is more emotional. In other words, spreading activation is to be applied in the recommendation cases, where a more emotional decision making is necessary.

TABLE VII. Conversion rates of the method configurations

| Method Configuration | Opened | Clicked | Ordered | Paid |
|---|---|---|---|---|
| Spreading Activation 3 | 15.3982% | 1.6581% | 0.0000% | 0.0000% |
| Spreading Activation 4 | 17.7506% | 1.7468% | 0.0483% | 0.0345% |
| Spreading Activation 5 | 16.9413% | 1.9600% | 0.0068% | 0.0000% |
| Spreading Activation 6 | 17.8073% | 1.9073% | 0.0404% | 0.0180% |
| Spreading Activation 7 | **21.0223%** | **2.4182%** | **0.0962%** | **0.1099%** |
| Betweenness Centrality | 12.7597% | 0.3514% | 0.0000% | 0.0000% |
| Closeness Centrality | 12.5700% | 0.3010% | 0.0000% | 0.0000% |
| Degree Centrality | **13.2988%** | 0.3879% | 0.0000% | 0.0000% |
| Eigenvector Centrality | 10.5723% | **0.4365%** | 0.0000% | 0.0000% |

Table VII presents the performance of the method configurations. The rows represent the method configurations. In the case of Spreading Activation, the number following the name of the method indicates the value of the step limit parameter. Network science based methods are not configurable. The columns of the table present the destination states. The values of the column present the conversion rates from the Sent state to the state denoted in the column title of the cell.

Analysing the configurations of the Spreading Activation method, the step limit setting 7 leads to the highest conversion rate. The step limit settings 3 and 5 lead to a poor performance, as the recommendations based on this technique did not lead to a purchase event. The detailed results show that fine-tuning the step limit parameter is important, as the performance of the methods strongly depends on this setting.

Analysing network science methods, involving degree centrality leads to the highest Sent to Opened conversion and eigenvector centrality leads to the highest Sent to Clicked conversion.

## VIII. Conclusion and Future Work

This paper presented the knowledge graph, an information representation technique capable to model heterogeneous information sources and also to alloy the information sources of the classic recommendation methods as collaborative filtering, content-based filtering and knowledge-based methods. The knowledge graph defines a hybridization method at the information representation level and also generalizes the concept of information sources, as discussed in Section III.

As our approach separates the information representation and the calculation methods, the methods are expected to manage the information in the knowledge graph as general. Looking at the static and the user generated information sources, the balance of the hybridization has been examined at the theoretical level in Section III-D. A principle of the information representation methods of the knowledge graph is the generalization of the information sources. The generalization at the representation level induced generalization and transitivity also at the calculation level as discussed in Section IV-B.

The evaluation of the methods has been conducted in a real-world experiment, as the conversion rates of an email remarketing scenario have been measured. In the experiment, spreading activation, network science and human expert based methods have been compared. The evaluation results show that the personalization of the emails have the potential to increase the performance of the email remarkating regarding to the conversion rates and the business value. Network science based methods proven to deliver low quality recommendations. A more theoretical result is the spreading activation based methods are intended to be used in an emotion based decision making scenario.

In the future, we plan to evaluate the methods in a web remarketing scenario. To achieve this, several optimization and GPU techniques should be involved in order to be able to generate the recommendations with a reasonable time need. On the theoretical side, we would like to alloy network science and representation learning based recommendation methods.

## References


[1] L. Grad-Gyenge and P. Filzmoser, "Recommendation Techniques on a Knowledge Graph for Email Remarketing," in *eKNOW 2016, The Eighth International Conference on Information, Process, and Knowledge Management*. eKNOW (2016): IARIA XPS Press, 2016, pp. 51–56.

[2] ——, "The Paradigm of Relatedness," in *AKTB 2016 : 8th Workshop on Applications of Knowledge-Based Technologies in Business*, 2016, pp. 13–20.

[3] L. Grad-Gyenge, P. Filzmoser, and H. Werthner, "Recommendations on a Knowledge Graph," in *MLRec 2015 : 1st International Workshop on Machine Learning Methods for Recommender Systems*, 2015, pp. 13–20.

[4] B. Hidasi and D. Tikk, "Fast ALS-Based Tensor Factorization for Context-Aware Recommendation from Implicit Feedback." in *ECML/PKDD (2)*, ser. Lecture Notes in Computer Science, P. A. Flach, T. D. Bie, and N. Cristianini, Eds., vol. 7524. Springer, 2012, pp. 67–82. [Online]. Available: http://dblp.uni-trier.de/db/conf/pkdd/pkdd2012-2.html#HidasiT12

[5] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining Association Rules between Sets of Items in Large Databases." in *SIGMOD Conference*, P. Buneman and S. Jajodia, Eds. ACM Press, 1993, pp. 207–216.

[6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. New York, NY, USA: ACM, 1999, pp. 230–237. [Online]. Available: http://doi.acm.org/10.1145/312624.312682

[7] Criteo. (2016, Dec.) Criteo. [Online]. Available: http://www.criteo.com/

[8] CreoGroup. (2016, Dec.) PartnerMail. [Online]. Available: http://www.partnermail.eu/

[9] aLap. (2016, Dec.) Booker.hu Online Bookstore - Order a Book. [Online]. Available: http://www.booker.hu/

[10] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. New York, NY, USA: Cambridge University Press, 2011. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0521493366

[11] I. Konstas, V. S. 0002, and J. M. Jose, "On social networks and collaborative recommendation." in *SIGIR*, J. Allan, J. A. Aslam, M. Sanderson, C. Zhai, and J. Zobel, Eds. ACM, 2009, pp. 195–202. [Online]. Available: http://dblp.uni-trier.de/db/conf/sigir/sigir2009.html#KonstasSJ09

[12] B. Hidasi and D. Tikk, "Fast ALS-Based Tensor Factorization for Context-aware Recommendation from Implicit Feedback," in *Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ser. ECML PKDD'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 67–82. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33486-3_5

[13] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems." in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, pp. 217–253. [Online]. Available: http://dblp.uni-trier.de/db/reference/rsh/rsh2011.html#AdomaviciusT11

[14] D. Easley and J. Kleinberg, "Homophily," in *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010. [Online]. Available: http://www.cs.cornell.edu/home/kleinber/networks-book;http://www.bibsonomy.org/bibtex/29b041bdba459fdc749a0b13de3d343d3/mstrohm

[15] J. Tang, S. Wu, and J. Sun, "Confluence: conformity influence in large social networks." in *KDD*, I. S. Dhillon, Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, R. L. Grossman, and R. Ut hurusamy, Eds. ACM, 2013, pp. 347–355.

[16] P. Kazienko, K. Musial, and T. Kajdanowicz, "Multidimensional Social Network in the Social Recommender System," *CoRR*, vol. abs/1303.0093, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1303.html#abs-1303-0093

[17] Epinions. (2016, Dec.) Epinions. [Online]. Available: http://www.epinions.com/

[18] C.-N. Ziegler and G. Lausen, "Propagation Models for Trust and Distrust in Social Networks," *Information Systems Frontiers*, vol. 7, no. 4-5, pp. 337–358, Dec. 2005. [Online]. Available: http://dx.doi.org/10.1007/s10796-005-4807-3

[19] P. Massa and P. Avesani, "Trust-Aware Collaborative Filtering for Recommender Systems." in *CoopIS/DOA/ODBASE (1)*, ser. Lecture Notes in Computer Science, R. Meersman and Z. Tari, Eds., vol. 3290. Springer, 2004, pp. 492–508. [Online]. Available: http://dblp.uni-trier.de/db/conf/coopis/coopis2004-1.html#MassaA04

[20] A. Jøsang, S. Marsh, and S. Pope, "Exploring Different Types of Trust Propagation." in *iTrust*, ser. Lecture Notes in Computer Science, K. Stølen, W. H. Winsborough, F. Martinelli, and F. Massacci, Eds., vol. 3986. Springer, 2006, pp. 179–192. [Online]. Available: http://dblp.uni-trier.de/db/conf/itrust/itrust2006.html#JosangMP06

[21] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. Ofek-Koifman, "Personalized recommendation of social software items based on social relations." in *RecSys*, L. D. Bergman, A. Tuzhilin, R. D. Burke, A. Felfernig, and L. Schmidt-Thieme, Eds. ACM, 2009, pp. 53–60. [Online]. Available: http://dblp.uni-trier.de/db/conf/recsys/recsys2009.html#GuyZCRUYO09

[22] J. He and W. W. Chu, "A Social Network-Based Recommender System (SNRS)," in *Data Mining for Social Network Data*, 2010, pp. 47–74.

[23] X. Yang, Y. Guo, and Y. Liu, "Bayesian-Inference-Based Recommendation in Online Social Networks." *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 642–651, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/tpds/tpds24.html#YangGL13

[24] H. Gu, M. Gartrell, L. Zhang, Q. Lv, and D. Grunwald, "AnchorMF: Towards Effective Event Context Identification," in *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, ser. CIKM '13. New York, NY, USA: ACM, 2013, pp. 629–638. [Online]. Available: http://doi.acm.org/10.1145/2505515.2505548

[25] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, and L. Getoor, "HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems," in *Proceedings of the 9th ACM Conference on Recommender Systems*, ser. RecSys '15. New York, NY, USA: ACM, 2015, pp. 99–106. [Online]. Available: http://doi.acm.org/10.1145/2792838.2800175

[26] Q. Hu, S. Xie, J. Zhang, Q. Zhu, S. Guo, and P. S. Yu, "HeteroSales: Utilizing Heterogeneous Social Networks to Identify the Next Enterprise Customer," in *Proceedings of the 25th International Conference on World Wide Web*, ser. WWW '16. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, pp. 41–50. [Online]. Available: http://dx.doi.org/10.1145/2872427.2883000

[27] R. Catherine and W. Cohen, "Personalized Recommendations Using Knowledge Graphs: A Probabilistic Logic Programming Approach," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys '16. New York, NY, USA: ACM, 2016, pp. 325–332. [Online]. Available: http://doi.acm.org/10.1145/2959100.2959131

[28] R. Burke, F. Vahedian, and B. Mobasher, "Hybrid recommendation in heterogeneous networks," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2014, pp. 49–60.

[29] M. Jamali and M. Ester, "TrustWalker: a random walk model for combining trust-based and item-based recommendation." in *KDD*, J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. Zaki, Eds. ACM, 2009, pp. 397–406. [Online]. Available: http://dblp.uni-trier.de/db/conf/kdd/kdd2009.html#JamaliE09

[30] R. Guha and P. Raghavan, "Propagation of Trust and Distrust," *null*, pp. 403—412, 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.3157

[31] C. Hess, K. Stein, and C. Schlieder, "Trust-enhanced visibility for personalized document recommendations," in *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2006, pp. 1865–1869. [Online]. Available: http://portal.acm.org/citation.cfm?id=1141277.1141718

[32] J. Golbeck, "Semantic Web Interaction through Trust Network Recommender Systems," in *End User Semantic Web Interaction Workshop at the 4th International Semantic Web Conference*, 2005.

[33] M. R. Quillian, "Semantic memory," in *Semantic Information Processing*, M. Minsky, Ed. Cambridge, MA: MIT Press, 1968, pp. 227–270.

[34] N. V. Findler, Ed., *Associative Networks: The Representation and Use of Knowledge of Computers*. Academic Pr, 1979. [Online]. Available: http://www.amazon.com/Associative-Networks-Representation-Knowledge-Computers/dp/0122563808%3FSubscriptionId=13CT5CVB80YFWJEPWS02&tag=ws&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0122563808

[35] C.-N. Ziegler and G. Lausen, "Spreading Activation Models for Trust Propagation," in *Proc. IEEE International Conference on e-technology, e-commerce, and e-service (EEE)*, Taipei, Taiwan, March 2004, pp. 83–97, best paper award for EEE2004.

[36] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010. [Online]. Available: http://www.cs.cornell.edu/home/kleinber/networks-book

[37] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford University, Tech. Rep., 1999.

[38] D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós, "Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments," *Internet Mathematics*, vol. 2, no. 3, pp. 333–358, 2005. [Online]. Available: http://scholar.google.de/scholar.bib?q=info:FrrSSl4mADMJ:scholar.google.com/&output=citation&hl=de&as_sdt=0&ct=citation&cd=1

[39] J. H. Jeong and J. W. Kim, "Personalized Recommendation Based on Collaborative Filtering with Social Network Analysis," in *International Conference on Information and Computer Applications*, ser. ICICA 2012, vol. 24. IACSIT Press, 2012, pp. 67–71.

[40] S. Haykin, "Neural Networks Viewed as Directed Graphs," in *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999, pp. 702–718. [Online]. Available: http://www.bibsonomy.org/bibtex/2e13a49ec715f4e52f1dcca0d4c5c8b2c/chrmina

[41] Y. Bengio, "Learning Deep Architectures for AI." Tech. Rep. 1, 2009. [Online]. Available: http://dblp.uni-trier.de/db/journals/ftml/ftml2.html#Bengio09

[42] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43–62, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169743997000610

[43] S. Haykin, "Back-Propagation Algorithm," in *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999, pp. 183–195. [Online]. Available: http://www.bibsonomy.org/bibtex/2e13a49ec715f4e52f1dcca0d4c5c8b2c/chrmina

[44] ——, "Hebbian Learning," in *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999, pp. 77–80. [Online]. Available: http://www.bibsonomy.org/bibtex/2e13a49ec715f4e52f1dcca0d4c5c8b2c/chrmina

[45] ——, "Hopfield Models," in *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999, pp. 702–718. [Online]. Available: http://www.bibsonomy.org/bibtex/2e13a49ec715f4e52f1dcca0d4c5c8b2c/chrmina

[46] Z. Huang, W. Chung, T.-H. Ong, and H. Chen, "A Graph-based Recommender System for Digital Library," in *Proceedings of the 2Nd ACM/IEEE-CS Joint Conference on Digital Libraries*, ser. JCDL '02. New York, NY, USA: ACM, 2002, pp. 65–73. [Online]. Available: http://doi.acm.org/10.1145/544220.544231

[47] S. Haykin, "Boltzmann Machine," in *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999, pp. 584–591. [Online]. Available: http://www.bibsonomy.org/bibtex/2e13a49ec715f4e52f1dcca0d4c5c8b2c/chrmina

[48] R. Salakhutdinov, A. Mnih, and G. E. Hinton, "Restricted Boltzmann machines for collaborative filtering." in *ICML*, ser. ACM International Conference Proceeding Series, Z. Ghahramani, Ed., vol. 227. ACM, 2007, pp. 791–798. [Online]. Available: http://dblp.uni-trier.de/db/conf/icml/icml2007.html#SalakhutdinovMH07;http://doi.acm.org/10.1145/1273496.1273596;http://www.bibsonomy.org/bibtex/2537c88cc0b98318ebbad1f76bfd0c451/dblp

[49] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, Nov. 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1021240730564

[50] S. Dooms, T. Pessemier, and L. Martens, "Online Optimization for User-specific Hybrid Recommender Systems," *Multimedia Tools Appl.*, vol. 74, no. 24, pp. 11 297–11 329, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1007/s11042-014-2232-7

[51] L. Grad-Gyenge and H. Werthner, "Network Science @ Recommender Systems," in *Effective, Agile and Trusted eServices Co-Creation*. TUCS Lecture Notes 19 (2013): Turku Centre for Computer Science, 2013, pp. 67–77. [Online]. Available: http://publik.tuwien.ac.at/files/PubDat_219604.pdf

[52] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems An Introduction*. Cambridge University Press, 2011.

[53] H. K. Bhargava, S. Sridhar, and C. Herrick, "Beyond Spreadsheets: Tools for Building Decision Support Systems," *Computer*, vol. 32, no. 3, pp. 31–39, Mar. 1999. [Online]. Available: http://dx.doi.org/10.1109/2.751326

[54] R. Reiter, "A Theory of Diagnosis from First Principles," *Artif. Intell.*, vol. 32, no. 1, pp. 57–95, Apr. 1987. [Online]. Available: http://dx.doi.org/10.1016/0004-3702(87)90062-2

[55] A. Gershman, A. Meisels, K.-H. Lüke, L. Rokach, A. Schclar, and A. Sturm, "A Decision Tree Based Recommender System." in *IICS*, ser. LNI, G. Eichler, P. G. Kropf, U. Lechner, P. Meesad, and H. Unger, Eds., vol. 165. GI, 2010, pp. 170–179. [Online]. Available: http://dblp.uni-trier.de/db/conf/iics/iics2010.html#GershmanMLRSS10

[56] J. Neidhardt, R. Schuster, L. Seyfang, and H. Werthner, "Eliciting the Users' Unknown Preferences," in *Proceedings of the 8th ACM Conference on Recommender Systems*, ser. RecSys '14. New York, NY, USA: ACM, 2014, pp. 309–312. [Online]. Available: http://doi.acm.org/10.1145/2645710.2645767

[57] A. Felfernig and R. Burke, "Constraint-based Recommender Systems: Technologies and Research Issues," in *Proceedings of the 10th International Conference on Electronic Commerce*, ser. ICEC '08. New York, NY, USA: ACM, 2008, pp. 3:1–3:10. [Online]. Available: http://doi.acm.org/10.1145/1409540.1409544

[58] A. Felfernig, "Koba4MS: Selling Complex Products and Services Using Knowledge-Based Recommender Technologies." in *CEC*. IEEE Computer Society, 2005, pp. 92–100. [Online]. Available: http://dblp.uni-trier.de/db/conf/wecwis/cec2005.html#Felfernig05;http://doi.ieeecomputersociety.org/10.1109/ICECT.2005.57;http://www.bibsonomy.org/bibtex/268088334dec36badb214253e5ec8812d/dblp

[59] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '94. New York, NY, USA: ACM, 1994, pp. 175–186. [Online]. Available: http://doi.acm.org/10.1145/192844.192905