Using the Open Meta Kaggle Dataset to Evaluate Tripartite Recommendations in Data Markets

Dominik Kowald
Know-Center GmbH
Graz, Austria
dkowald@know-center.at

Matthias Traub
Know-Center GmbH
Graz, Austria
mtraub@know-center.at

Dieter Theiler
Know-Center GmbH
Graz, Austria
dtheiler@know-center.at

Heimo Gursch
Know-Center GmbH
Graz, Austria
hgursch@know-center.at

Stefanie Lindstaedt
Know-Center GmbH
Graz, Austria
slind@know-center.at

Roman Kern
Graz University of Technology
Graz, Austria
rkern@tugraz.at

Elisabeth Lex
Graz University of Technology
Graz, Austria
elisabeth.lex@tugraz.at

ABSTRACT

This work addresses the problem of providing and evaluating recommendations in data markets. Since most of the research in recommender systems is focused on the bipartite relationship between users and items (e. g., movies), we extend this view to the tripartite relationship between users, datasets and services, which is present in data markets. Between these entities, we identify four use cases for recommendations: (i) recommendation of datasets for users, (ii) recommendation of services for users, (iii) recommendation of services for datasets, and (iv) recommendation of datasets for services. Using the open Meta Kaggle dataset, we evaluate the recommendation accuracy of a popularity-based as well as a collaborative filtering-based algorithm for these four use cases and find that the recommendation accuracy strongly depends on the given use case. The presented work contributes to the tripartite recommendation problem in general and to the under-researched portfolio of evaluating recommender systems for data markets in particular.

KEYWORDS

Tripartite Recommendations; Data Markets; Recommender Systems; Collaborative Filtering; Offline Evaluation; DMA

ACM Reference Format:
Dominik Kowald, Matthias Traub, Dieter Theiler, Heimo Gursch, Stefanie Lindstaedt, Roman Kern, and Elisabeth Lex. 2019. Using the Open Meta Kaggle Dataset to Evaluate Tripartite Recommendations in Data Markets. In *Proceedings of REVEAL@RecSys'2019.* ACM, New York, NY, USA, 5 pages. https://doi.org/xx.xxx/xxxxxx.xxxxxx

REVEAL@RecSys'2019, September 16-20, 2019, Copenhagen, Denmark

ACM ISBN xxx-x-xxxx-xxxx-x/xx/xx…$15.00
https://doi.org/xx.xxx/xxxxxx.xxxxxx

1 INTRODUCTION

Data-driven services are becoming an increasingly important aspect of the modern economy, with data markets playing a key role as broker between the stakeholders of the data-driven ecosystem. Various initiatives have been started to research the requirements and dynamics of data markets. To name two examples, the "Data Market Austria" (DMA)[1] [17] is a national project in Austria, while "A European AI On Demand Platform and Ecosystem" (AI4EU)[2] aims at creating a market platform for data and artificial intelligence solutions on the European level.

For successful collaborations in data markets, the different entities need to collaborate with each other in order to create new solutions and to be able to provide innovative data products [1, 2].

Problem and objective of this work. Recommender services thereby play a crucial role in data markets, since their suggestions allow to discover potential new combinations between users, datasets and services [3]. This results in a more complex tripartite relationship comprising users, datasets and services, as well as an increased number of use cases, in comparison with a traditional recommender setting. The tripartite structure and use cases are depicted in Figure 1.

However, most of the research in recommender systems is focused on settings consisting only of users and items, like recommending new movies to viewers. Hence, these settings can be categorized as bipartite relationships. The work of [4] points out the research need for recommendations in tripartite relationship scenarios such as the data markets scenario investigated in the work at hand. Another issue is the lack of an open dataset for the evaluation of tripartite recommendations in data markets. Therefore, we propose the use of the open Meta Kaggle dataset of the well-known data science portal Kaggle.

Contributions and findings. The contributions of our work are two-fold:

[1] https://datamarket.at/en/
[2] https://www.ai4eu.eu/

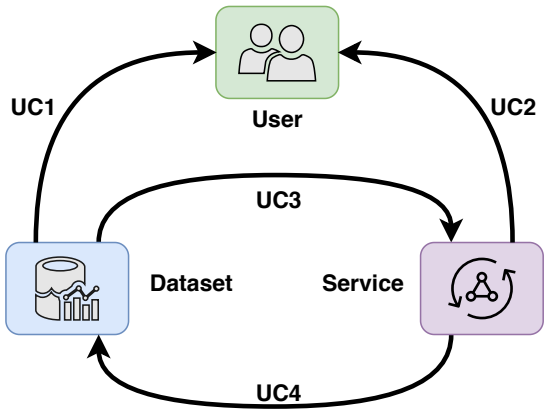

Figure 1: The tripartite relationship in a data market is spun between users, datasets and services. We identify four use cases for recommendations between these three identities, namely recommendation of datasets for users (UC1), recommendation of services for users (UC2), recommendation of datasets for services (UC3) and recommendation of services for datasets (UC4).

- We propose four use cases as well as a system architecture for recommendations in data markets (see Section 2).
- We provide evaluation results for a popularity-based as well as collaborative filtering-based algorithm for these four use cases using the open Meta Kaggle dataset (see Section 3).

Our results show that the recommendation accuracy strongly depends on the given use case. For example, in settings in which we have a limited set of candidate entities to recommend, already the simple popularity-based approach (recommending the most popular (MP) entities) provides good results. However, in more complex settings, where it is required to link services and datasets, a personalized approach such as collaborative filtering (CF) should be favored.

Taken together, our work contributes to the under-researched portfolio of recommender systems for data markets and thus, should be of interest for both researchers and practitioners in this area.

2 RECOMMENDATIONS IN DATA MARKETS

This section gives an a detailed overview of the four central data market use cases followed by the architecture of the proposed recommender system and all its components.

2.1 Use Cases

As depicted in Figure 1, data markets create a tripartite relationship between their entities users, datasets and services, thus leading to more complex recommendation problems. We identify four use cases for recommendations in the setting of data markets, investigated in more detail in the remainder of this subsection.

UC1: Recommendation of datasets for users. In the first use case, we recommend datasets to users. Thus, this one reflects a rather classic item2user recommendation problem, in which we analyze past user interactions between the target user and datasets (e. g., clicks or purchases) in order to recommend other datasets that could be interesting for the user (e. g., by using CF).

UC2: Recommendation of services for users. The second use case also reflects a classic item2user recommendation problem but this time we aim to recommend services for users of the data market. Since typically there are more services than datasets available in a data market (see Section 3.1), the set of potential candidate services is also larger, which makes this recommendation problem potentially harder than the one of UC1.

UC3: Recommendation of datasets for services. UC3 reflects a more complex recommendation problem, in which we aim to recommend datasets for services. As both entities are now item types, we do no longer have classic user interactions for CF as we have in UC1 and UC2. To overcome this, we could establish an indirect connection between a dataset and a service when a user has interacted with both, the dataset and the service (see Section 3.1).

UC4: Recommendation of services for datasets. In the fourth and final use case, we recommend services for datasets. As mentioned in UC2, we typically have more services than datasets available in a data market, which makes this use cases more complex than UC3, where the set of candidate entities to recommend is smaller. Furthermore, in UC4, we want to link services and datasets, where we do not have direct user interactions available. Thus, we believe that this use case is the most complex one and therefore, we also expect the lowest recommendation accuracy for this one (see Section 3.3).

2.2 System Architecture

The design of the system architecture of our recommender system for data markets is centred upon the scalable recommendation framework ScaR[3] [9, 10]. In Figure 2, we illustrate our main modules and how they interact with each other as well as with users and administrators of a data market. Apache ZooKeeper[4] is used for handling the communication between the modules and for load balancing (e.g., deploying multiple instances of a module).

Service Provider (SP). The SP acts as a proxy for data markets to interact with the recommender system. It provides REST-based Web services to enable users to query recommendations of datasets and services, and to add new data (e. g., user interactions, datasets or services) to the system.

Data Modification Layer (DML) & Apache Solr. The DML encapsulates all database-related CRUD operations (i. e., create, retrieve, update, delete) in one module and thus, enables easy access to the underlying data backend. As shown in Figure 2, we utilize the high-performance search engine Apache Solr[5]. This data backend solution not only guarantees scalability and (near) real-time recommendations but also the support of multiple entities like the users, datasets and services we encounter here.

[3] http://scar.know-center.tugraz.at/
[4] https://zookeeper.apache.org/
[5] http://lucene.apache.org/solr/

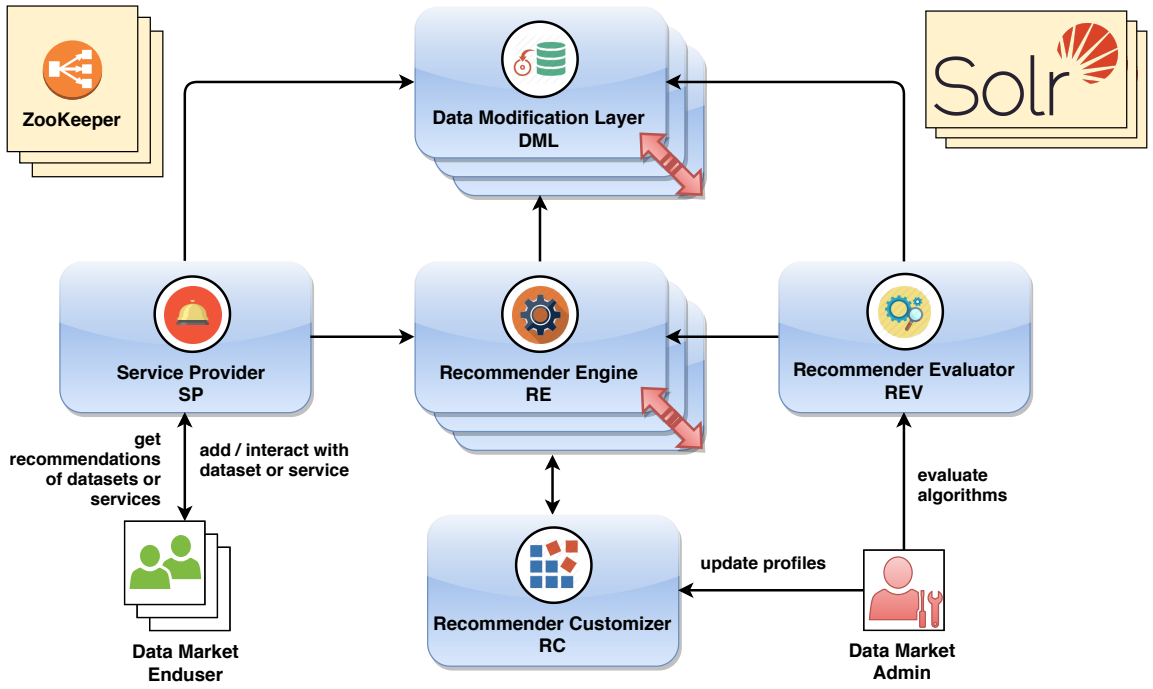

Figure 2: The system architecture of our recommender system for data markets is build upon ScaR as well as the open-source frameworks Apache Solr and ZooKeeper. The communication between the modules is handled via REST-based Web services.

Recommender Engine (RE). The RE is the main module of our recommender system for data markets as it is responsible for calculating recommendations. Here, we make use of Apache Solr's build-in data structures for efficient similarity calculations. Currently, we focus on popularity and CF-based recommendation algorithms, but the RE module could be easily extended with further algorithms as well (e. g., content-based filtering).

Recommender Customizer (RC). The RC is used to change the parameters (e. g., the number of recommended entities k) of the recommendation approaches on the fly. Thus, it holds a so-called recommendation profile for each approach, accessible and changeable by the data market administrator. These changes are then broadcast to the RE to be aware of how a specific approach should be executed.

Recommender Evaluator (REV). The REV is responsible for evaluating the recommendation algorithms implemented in the RE. Hence, it can be executed to perform an offline evaluation with training/test set splits (see Section 3.2). In the future, it will also be possible to conduct online evaluations in data markets via A/B-tests.

3 EVALUATION

In this section, we present our evaluation study, in which we compare popularity-based with CF-based recommendations for all four use cases defined in Section 2.1.

3.1 Data

For our evaluation, we use the open Meta Kaggle dataset[6] (2017-11-15) of the well-known Kaggle data science portal in order to simulate a real-world data market. Here, we have 6,108 users and 45 datasets that are connected via 2,926 user/dataset interactions, where an interaction is given by a user writing about a dataset in a discussion thread. Furthermore, we have 3,334 services that are connected to the 6,108 users via 18,593 user interactions. These interactions are created by users voting for a service.

Finally, we establish a collaboration network between datasets and services (see e.g., [5]). Thus, we create a link between a dataset and a service when a user has interacted with both, the dataset and the service, which leads to 95,249 interactions. The full statistics of our dataset are summarized in Table 1.

3.2 Evaluation Method

In this section, we describe the evaluation protocol, recommendation algorithms and evaluation metrics used for our study.

Evaluation protocol. For measuring the recommendation quality in the settings of the four use cases UC1 - UC4, we follow common practice in the area of recommender systems and split our Meta Kaggle dataset into training and test splits as suggested by [6].

Specifically, we extract all entities with at least eleven interactions from whom we withhold ten interactions for the test set and

[6] https://www.kaggle.com/kaggle/meta-kaggle

Feature	#
Number of users	6,108
Number of datasets	45
Number of services	3,334
Number of user/dataset interactions	2,962
Number of user/service interactions	18,593
Number of dataset/service interactions	95,249

Table 1: Statistics of our dataset, which was collected from the Meta Kaggle platform in order to simulate the entities and interactions in a real-world data market. As the number of datasets is much smaller than the number of services, we also expect better recommendation accuracy results for UC1 and UC3 than for UC2 and UC4.

use the rest for training[7] [13]. Thus, for UC1, this results into 17 users for whom we recommend datasets; for UC2, this results into 184 users for whom we recommend services; for UC3, this results into 2,338 services for whom we recommend datasets; for UC4, this results into 44 datasets for whom we recommend services.

Recommendation algorithms. We evaluate our four uses cases for recommendations in data markets with two algorithms, namely most popular (MP) and (ii) collaborative filtering (CF). The recommendations are calculated and evaluated using the recommender system presented in Section 2.2.

MP is a non-personalized algorithm and is especially useful for new entities in a data market without any interactions so far, commonly referred as cold-start entities [16]. This approach recommends datasets or services, which are weighted and ranked by the number of interactions. As mentioned, the MP approach is non-personalized and thus, each entity will receive the same recommendations.

CF algorithms [14] analyze the interactions between users and entities, e. g., datasets and services alike. In CF methods two users are treated as similar if they have interacted with similar entities in the past. Hence, entities a similar user has interacted with in the past are candidates to recommend to a target user, who has not interacted with those entities yet. In the case of data markets, we do not only have interactions between users and entities but also between entities themselves when we consider UC3 and UC4. Here, we realize the CF approach in a similar way but instead of calculating user similarities, we calculate similarities between datasets and services, respectively.

Evaluation metrics. For measuring the accuracy of the recommendations in data markets, we use a rich set of metrics, namely Precision (P@k), F1-score (F1@k), Recall (R@k), Mean Reciprocal Rank (MRR@k), Mean Average Precision (MAP@k) and normalized Discounted Cumulative Gain (nDCG@k) [7].

We report these metrics for different numbers of recommended entities (= k), i. e., P@1 for k = 1, F1@5 for k = 5[8], R@10 for k = 10, MRR@10 for k = 10, MAP@10 for k = 10 and nDCG@10 for k = 10.

[7]We only evaluate users with a minimum of eleven interactions to ensure that we have at least one interactions for training when using ten interactions for testing.
[8]For 10 recommended entities, Precision typically reaches its highest value for k = 1 and F1 for k = 5.

Approach	P@1	F1@5	R@10	MRR@10	MAP@10	nDCG@10
UC1: MP	**0.823**	**0.470**	**0.717**	**0.217**	**0.597**	**0.729**
UC1: CF	0.705	0.431	0.611	0.192	0.484	0.635
UC2: MP	0.103	0.050	0.066	0.023	0.026	0.072
UC2: CF	**0.137**	**0.086**	**0.114**	**0.037**	**0.054**	**0.121**
UC3: MP	1.000	0.411	0.707	0.232	0.580	0.750
UC3: CF	**1.000**	**0.636**	**0.934**	**0.281**	**0.925**	**0.948**
UC4: MP	0.000	0.000	0.000	0.000	0.000	0.000
UC4: CF	**0.022**	**0.006**	**0.006**	**0.003**	**0.004**	**0.009**

Table 2: Evaluation results of our four use cases for recommendations in data markets. While we get the best recommendation accuracy results for the unpersonalized MP algorithm in UC1, the personalized CF approach provides the best results for the more complex UC2, UC3 and UC4. The poor results for UC4 indicate that we need more sophisticated algorithms than MP and CF in this setting. Please note that bold numbers indicate the best results for a use case.

Please note that we set the maximum number of k to 10, which is a common value for the evaluation of recommender systems [15].

3.3 Results

In this section, we present the results of our evaluation with respect to our four use cases. Table 2 holds the resulting numbers achieved in our experiments.

UC1: Recommendation of datasets for users. This use case reflects the least complex one as we recommend from a quite limited set of candidate entities (i.e., 45 datasets) with a small number of connections to the target entities (i. e., 2,962 user interactions). This is also reflected in the recommendation accuracy results presented in Table 2 as the unpersonalized MP approach provides better results than the personalized CF one. This behavior of MP outperforming CF can only be observed in this use case, which shows that personalized approaches are not always necessary.

UC2: Recommendation of services for users. When recommending services for users, we face a more complex problem since we have a much larger set of candidate entities (i.e., 3,334 services). Thus, the accuracy results in UC2 are much lower than the ones in UC1. Furthermore, in this case, the CF approach, which analyzes the 18,593 interactions between users and services in a personalized manner, provides better results than MP.

UC3: Recommendation of datasets for services. Similar to UC1, in UC3, we also recommend datasets but this time for services instead of users. For this use case, we also have a large set of 95,249 interactions between datasets and services available, leading to the overall best results for CF across all four use cases. Interestingly, both MP and CF provide a perfect score for P@1 of 1.000, which indicates that both algorithms rank a highly-connected dataset on the first position that is relevant for all 2,338 evaluated services.

UC4: Recommendation of services for datasets. UC4 reflects the most complex of our use cases since we have a large set of 3,334 candidates services available, which are linked via 95,249

interactions to a small set of 44 datasets being the evaluated entities. This is reflected in the results shown in Table 2 as both algorithms, MP and CF, provide the worst results across all use cases. Here, the unpersonalized MP approach even reaches a recommendation accuracy of 0.000 for all metrics, thus not recommending a single relevant service.

3.4 Discussion

Our evaluation results show that there is no one-size-fits-all solution for recommendations in data markets. One particular finding of us is, that in cases having a limited set of candidate entities available like in UC1, popularity-based methods such as MP provide good results. Another finding is that personalized methods such as CF should be favored when the use cases get more complex, for example if we have a larger set of candidate entities as it is the case in UC2. The same holds for the recommendations of entities to other entities, like datasets to services in UC3.

However, our results also show that both MP and CF provide poor results for UC4 being the most complex use case. For such a setting, we need more sophisticated methods that incorporate also other data sources, e. g., content-based filtering (CBF) approaches [11]. For overcoming sparsity problems, these approaches could also be combined with word embeddings [8, 12].

4 CONCLUSION AND FUTURE WORK

In this paper, we presented our initial steps for providing and evaluating recommendations in data markets. Therefore, we first provided four potential use cases, which included recommendation of datasets for users (UC1), recommendation of services for users (UC2), recommendation of datasets for services (UC3), and recommendation of services for datasets (UC4). Then, we proposed a system architecture for a recommender system for data markets based on the scalable recommendation framework ScaR.

Finally, we provided an evaluation of these four uses using the Meta Kaggle dataset and our proposed recommender system. Here, we find that the unpersonalized most popular approach (MP) provides the best results for UC1 and the personalized collaborative filtering approach (CF) provides the best results for the more complex use cases UC2, UC3 and UC4.

Limitations and future Work. One limitation of our evaluation is that we have simulated a real-world data market using the Meta Kaggle dataset. Although, this dataset provides all relevant entities of data markets (i. e., users, datasets and services), we plan to also conduct evaluation studies in real-world data markets such as the ones created in the DMA and AI4EU initiatives.

Furthermore, so far, we have only evaluated the two algorithms MP and CF. Thus, we also plan to extend our study with more recommendation approaches such as content-based filtering (see Section 3.4).

Acknowledgments. This work was supported by the Know-Center GmbH, the FFG flagship project Data Market Austria (DMA) and the H2020 project AI4EU (GA: 825619). The Know-Center GmbH is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency (FFG).

REFERENCES

[1] José María Cavanillas, Edward Curry, and Wolfgang Wahlster (Eds.). 2016. *New Horizons for a Data-Driven Economy.* Springer, Cham, Switzerland. DOI: http://dx.doi.org/10.1007/978-3-319-21569-3
[2] Edward Curry. 2016. The big data value chain: definitions, concepts, and theoretical approaches. In *New horizons for a data-driven economy*. Springer, Cham, 29–37.
[3] Ernesto Damiani, Paolo Ceravolo, Fulvio Frati, Valerio Bellandi, Ronald Maier, Isabella Seeber, and Gabriela Waldhart. 2015. Applying recommender systems in collaboration environments. *Computers in Human Behavior* 51 (2015), 1124–1133.
[4] Daniela Godoy and Alejandro Corbellini. 2016. Folksonomy-Based Recommender Systems: A State-of-the-Art Review. *International Journal of Intelligent Systems* 31, 4 (2016), 314–346. DOI: http://dx.doi.org/10.1002/int.21753 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/int.21753
[5] Ilire Hasani-Mavriqi, Dominik Kowald, Denis Helic, and Elisabeth Lex. 2018. Consensus dynamics in online collaboration systems. *Computational Social Networks* 5, 1 (2018), 2.
[6] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.
[7] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
[8] Tom Kenter and Maarten De Rijke. 2015. Short text similarity with word embeddings. In *Proc. of CIKM'15*. ACM, 1411–1420.
[9] Emanuel Lacic, Dominik Kowald, Denis Parra, Martin Kahr, and Christoph Trattner. 2014. Towards a scalable social recommender engine for online marketplaces: The case of apache solr. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 817–822.
[10] Emanuel Lacic, Matthias Traub, Dominik Kowald, and Elisabeth Lex. 2015. ScaR: Towards a Real-Time Recommender Framework Following the Microservices Architecture. In *Proceedings of LSRS2015 Workshop at RecSys 2015.* .
[11] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.
[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints* (Jan 2013), arXiv:1301.3781. arXiv:cs.CL/1301.3781
[13] Denis Parra and Peter Brusilovsky. 2009. Collaborative filtering for social tagging systems: an experiment with CiteULike. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 237–240.
[14] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. *Introduction to recommender systems handbook*. Springer.
[15] Alan Said and Alejandro Bellogín. 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 129–136.
[16] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 253–260.
[17] Matthias Traub, Heimo Gursch, Elisabeth Lex, and Roman Kern. 2017. Data Market Austria *(Institute of Systems Sciences, Innovation and Sustainability Reports)*, Romana Rauter, Martina Zimek, Aisma Linda Kiesnere, and Rupert J. Baumgartner (Eds.). Institute of Systems Sciences, Innovation and Sustainability, University of Graz, 353–363.