Mixed-Initiative Feature Engineering Using Knowledge Graphs

Martin Atzmüller
Tilburg University (CSAI)
Tilburg, Netherlands
matzmuller@uvt.nl

Eric Sternberg
University of Kassel (ITeG)
Kassel, Germany
est@cs.uni-kassel.de

ABSTRACT
This paper proposes a mixed-initiative feature engineering approach using explicit knowledge captured in a knowledge graph complemented by a novel interactive visualization method. Using the explicitly captured relations and dependencies between concepts and their properties, feature engineering is enabled in a semi-automatic way. Furthermore, the results (and decisions) obtained throughout the process can be utilized for refining the features and the knowledge graph. Analytical requirements can then be conveniently captured for feature engineering – enabling integrated semantics-driven data analysis and machine learning.

CCS CONCEPTS
• Human-centered computing → Visualization; • Computing methodologies → Knowledge representation and reasoning; • Machine learning;

KEYWORDS
knowledge graph, feature engineering, machine learning

ACM Reference Format:

1 INTRODUCTION
For many areas of machine learning and data analysis such as predictive modeling, anomaly detection, or pattern mining, the majority of the applied methods still considers the data in isolated fashion. In contrast, an effective integrated approach is given by constructing a knowledge graph. Here, the data is integrated in a comprehensive knowledge structure capturing the relations between concepts and their properties in an explicit way. Then, this structure can be exploited in order to facilitate machine learning and data analysis. However, the knowledge graph mainly focuses on the structuring of the concepts and their relations, while specific modeling tasks, as well as data characteristics (e.g., distributions, correlations) are typically not captured. This second step is the main focus of this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

K-CAP 2017, December 4–6, 2017, Austin, TX, USA © 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.
ACM ISBN 978-1-4503-5553-7/17/12... $15.00
https://doi.org/10.1145/3148011.3154473

2 RELATED WORK
Existing works utilizing semantic structures in data mining [10, 14, 16, 19] focuses on applying ontologies in the data mining step. However, so far the approaches only apply a "shallow" coupling. Here, advanced knowledge-rich representations like knowledge graphs [4, 17, 20] are a prominent research direction, e.g., [7, 13]. In contrast to existing approaches for feature engineering, e.g., [1–3, 20], this work proposes to use knowledge graphs for feature engineering in a mixed-initiative approach: Utilizing the formalized relations, a semi-automatic visualization method enables advanced feature engineering, knowledge capture, and refinement.

For visualization and browsing graphs and network structures [12], there are a variety of techniques, including visualizations for showing densely connected subgroups (as sets of nodes) [11], special cluster visualizations [8], and interactive techniques for analyzing connected graph structures [9]. In contrast, this paper focuses on feature engineering using knowledge graphs, visualizing feature dependencies to support feature selection and construction.

For feature engineering also explanation-awareness plays an important role. In particular, if explanations for the complete models, or parts thereof can be provided, then the acceptance can often be significantly improved, e.g., [6]. In the proposed approach, this is enabled by inspecting the underlying data and formalized relations at the relevant representational level and dimensions.
3 METHOD
Below, we first outline and summarize the mixed-initiative approach for feature engineering using knowledge graphs. After that, we describe the novel visualization method in detail.

3.1 Overview
The proposed mixed-initiative process is depicted in Figure 1. We start with heterogeneous complex datasets that are integrated into a knowledge graph structure. After that, this knowledge structure can be applied for machine learning and data analysis. We do not describe the initial construction of the knowledge graph in detail. Instead, we focus on the subsequent feature engineering step, e.g., enabling feature selection and feature construction.

In this paper, we propose a mixed-initiative approach, such that first a set of relevant (initial) features is compiled (feature generation) using the concepts and relations modeled in the knowledge graph. This can happen, e.g., by selecting concepts of specific types, such as parts of specific machinery in industrial applications, and attributes describing their properties, as encoded by relations in the knowledge graph. Then, a semi-automatic process is initiated (i.e., for feature engineering), guided by interactive visualization as described below (interactive analysis). Finally, the engineered features are provided for data analysis and machine learning (analysis and mining). In addition, decisions taken during the feature engineering step and/or the analysis and mining step can be utilized for refinement: They can, for example, be (re-)integrated into the knowledge graph, such as formalizing/adding/extend annotations of features (represented by concepts) or refining the respective relations, e.g., based on the information that one feature is strongly correlated with another one. Furthermore, the knowledge graph structure itself, as well as the set of generated and/or selected features can be iteratively adapted based on the obtained insights.

We provide examples of such actions in the real-world case study sketched below. Finally, then also the knowledge graph itself can be applied for providing additional context regarding the results of the analysis and mining step, as well as during the feature engineering step, e.g., to provide explanations [6, 18]. This is facilitated by the links to the originating concepts and the covered data records, respectively, which can then be exploited, e.g., by exploring a node’s neighborhood in the knowledge graph, or by browsing and/or inspecting the respective sets of objects [5].

3.2 Visualization
For inspecting and assessing the generated features, their characteristics and dependencies, we propose a visualization called the feature matrix plot, see Figure 2: It consists of several (sub-)plots, for each pair of features denoted by the corresponding row and column labels. As an example, we consider the feature pair A1/A3 (indicated by the red rectangle). The yellow subplot at (1,1) provides information about the domain of A1 via a histogram, accordingly the subplot at (3,3) for A3. The gray subplot at (1,3) shows a heat map, while the subplot at (3,1) provides information about missing values and the (Spearman) correlation between A1 and A3. A detailed view on two exemplary heat map subplots for the feature pairs A2/A4 and A2/A5 is given to the right of Figure 2: Each subplot shows a different grid, derived from nominal (or appropriately discretized intervals) of the two feature domains; the vertical partition comes from the “row” feature (here A2) and the horizontal partition from the “column” feature (here A4 resp. A5). Therefore, all partitionings of a subplot represent all possible value realizations for the given feature pair. The hotter the color, the larger the number of individuals in that partition. Each heat map is normalized so the highest occurrence is red and the lowest blue, the color gradient shows all possible pixel colors. The heat map can also be understood as an 2D Histogram (from above) in which the bins are represented by the single grid elements; the pixel color represents their height. For numeric features, we basically focus on the other subplots (below).

The subplots shown in gray depict three different types of information. First the background indicates the proportion between individuals used for generating the subplots and the individuals that have least one missing value for one of the paired features (indicated in yellow) – for data quality assessment and cross-checks. The colored oval in the center of the subplot displays the correlation between the paired features whereby the extent on the x-axis shows the correlation between the discretized intervals; the extent on the y-axis shows the correlation between the original (undiscretized) domains. Furthermore the oval’s color provides information about the direction of a possible correlation (green=positive, red=negative); the color inside stands for the direction of the discretized intervals. The outer ovals’ color shows the direction of the original domain. A gray color indicates that at least one of the paired features is nominal, whereas numerical and ordinal features provide richer visualizations. Overall, the idea here is to provide a fast visual method to detect, e.g., data quality and discretization problems via strongly skewed ovals and strongly correlated features by big circles.
Mixed-Initiative Feature Engineering Using Knowledge Graphs

4 CASE STUDY

The introduced semi-automatic approach utilizing the matrix plots for interactive analysis has been applied in an (anonymized) industrial case study, where data analysis was applied to productive logistic data. One major problem that occured in the iterative workflow was the large amount of concepts/features contained in the knowledge graph which had then to be “condensed” to a set of meaningful features that are relevant for the analysis. For that purpose, we applied the proposed approach including the novel visualization method. In particular, this allowed us to perform feature engineering and selection in an iterative process considering knowledge graph construction, feature construction, and refinement. By interpreting the visualization the set of features can be optimized, e.g., by removing strongly correlated attributes and detect/fix problematic domain discretizations before running costly data analysis tasks. If it is necessary to resign on features the plot can be utilized to focus on the most informative and available attributes. In that way, also the experts could be involved into this interactive visual process in order to provide feedback on the constructed feature set. Furthermore, important feedback could be derived from the visualizations regarding the source data that was used for generating the knowledge graph. For that, basically a complex knowledge graph data structure was evolved from plain relational logistic data. The structure was a composition from two graphs that, e.g., represent the material flows, and composition of different processes and products. As a particularly important case, the 1D histogram was used to refine a central KPI feature, which was used for data analysis and further machine learning approaches. This KPI feature was constructed from the graphs’ structure itself capturing the experts’ beliefs. Then, using the visualization the relations, dependencies and correlations to other concepts from the knowledge graph could be checked, and expectations of the domain specialists could be verified. This allowed us to both construct the feature according to their domain knowledge and exploiting/ refining the knowledge graph at the same time, e.g., by adding relations that were missing, or by removing incorrect relations and dependencies. Figure 3 shows a feature matrix plot of an already improved feature set from a knowledge graph where 18 attributes have been selected. It is easy to see that the domain distributions are quite

Figure 2: Schematic breakdown of the feature matrix plot for five attributes. The feature matrix plot is made up of three individual subplot types that provide valuable information, e.g., for feature engineering, knowledge graph optimization, discretization and spotting datasource problems.
exotic. Here, we also include some information on the "evolution" of some features. As an example, consider an earlier version of an important feature denoted by A16. Here, the distributions and correlations were not well expressed according to domain knowledge. The final refined version is given by A11; here, we can also note the absence of noise which directly visualizes the improved mapping of experts knowledge. A5 shows a discretization problem as it can be seen in all subplots A5 is involved. Here, all correlation ovals are extremely stretched on the x-axis, also the corresponding heatmaps show only a resolution of one “logical” pixel on the y-axis. The problem was that all individuals fall in only one discretization interval. The final feature set was improved by replacing the numeric A5 by an adapted nominal version (and by removing feature A16, as discussed above). Altogether, the proposed approach proved crucial in feature engineering and in also refining the knowledge graph in order to be applicable for data analysis and machine learning.

Our results indicate the efficacy of the proposed approach. The domain specialists could utilize the proposed approach very well. Guided by the visualization, they could easily interpret its results. Working together with the data scientists in the feature engineering phase, they were effectively supported using the proposed approach, especially applying the presented visualization method. Furthermore, the knowledge graph was also iteratively adapted and refined according to the outcomes of the modeling and analysis phases.

For future work, we aim to further extend explanation-aware approaches utilizing the semantic structures formalized in the knowledge graph in order to provide ad-hoc knowledge in context. This also connects to further visualization approaches for supporting detailed inspection of knowledge graph components, and its refinement [15]. Further interesting directions concern scalable machine learning and data analysis methods on complex knowledge graphs targeting predictive as well as descriptive approaches.

REFERENCES


