Speculative Execution of Similarity Queries: Real-Time Parameter Optimization through Visual Exploration

T. Spinner[1], U. Schlegel[1], M. Schall[1,2], F. Sperrle[1], R. Sevastjanova[1], B. Gobbo[3], J. Rauscher[1], M. El-Assady[1], D. Keim[1]

[1] University of Konstanz [2] University of Applied Sciences Konstanz [3] Politecnico di Milano

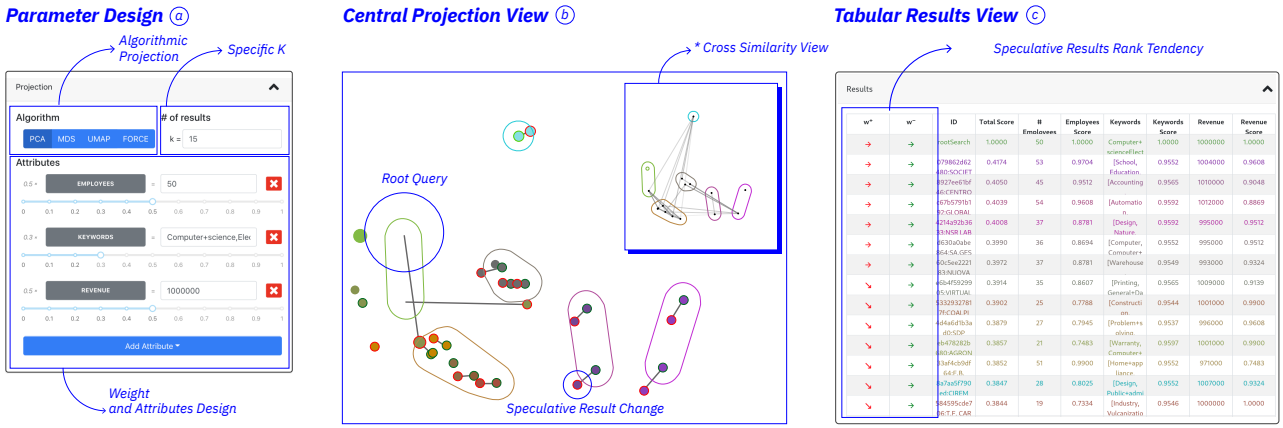

Figure 1: The SimSearch workspace, built around the central projection view (a), showing the projected results of the similarity search algorithm for the query defined in the parameter designer (b). The tabular results view (c) shows the same results, linked to the projected nodes by color and hover events. Both the projection and table view switch to a speculative execution state on hovering a weight slider, indicating the changes occurring if the weight is adjusted accordingly.

ABSTRACT

The parameters of complex analytical models often have an unpredictable influence on the models' results, rendering parameter tuning a non-intuitive task. By concurrently visualizing both the model and its results, visual analytics tackles this issue, supporting the user in understanding the connection between abstract model parameters and model results. We present a visual analytics system enabling result understanding and model refinement on a ranking-based similarity search algorithm. Our system (1) visualizes the results in a projection view, mapping their pair-wise similarity to screen distance, (2) indicates the influence of model parameters on the results, and (3) implements speculative execution to enable real-time iterative refinement on the time-intensive offline similarity search algorithm.

1 INTRODUCTION

Similarity search in large database systems is a crucial feature in many applications and often requires a manual adjustment of parameters to suit various search scenarios [17]. Such parameters are hard to optimize by randomly probing the search space, but they significantly influence the retrieved results' quality [7]. In many cases, even experts with prior domain knowledge struggle to understand the inner workings of the used mining models and the influence of abstract model parameters, which prevents them from reaching the desired analysis goal. Systematic steering and exploration of different parameter settings can help to obtain the proper combination more effectively. Thus, domain experts need concurrent access to models, parameters, and results, enabling them to understand how parameters influence the results and how they can be refined to match the analysis goal.

Visual analytics enables users to explore and analyze data and models by providing integrated visual representations for data, models, and parameters. Such visual techniques enable interactive parameter adjustment during exploration and analysis [6]. Visual analytics bridges the gap between heuristics to find suitable parameters and domain experts with the knowledge to steer results in a human-centered direction. For instance, a visual interactive what-if analysis facilitates experts to understand black-box model decisions by enabling direct data and parameter manipulation [13]. The comprehensive understanding of the relationship between model parameter choices and outcome is a fundamental requirement for well-informed decision making [20]. By applying standard visual analytics techniques, such as aggregation, filtering, or speculative execution [18], the vast results- and parameter spaces can be interactively explored, despite the algorithms being time- and resource-consuming. Thus, visual analytics supports the comprehension of parameter choices in similarity search applications for users and domain experts. Visual analytics enables informed reasoning about a query's results, allows the understanding and diagnosis of parameters, and supports the user in refining those parameters to get the best possible results.

We propose a visual analytics workspace to support users in result understanding and model refinement on a ranking-based similarity search algorithm in the context of large data foundations. Our system consists of a user-centered visualization of

parameters and results to facilitate the users' exploration and understanding of the parameter choices. We enable users to interactively update model parameters based on their domain knowledge and findings during the analysis process. Our visual analytics system further facilitates real-time analysis using speculative execution on a time-intensive similarity search algorithm, enabling online exploration and execution of the offline algorithm.

Summarizing, we present a visual analytics system for similarity search, providing the following main contributions: (1) our system supports the *understanding* of results and parameters, emphasizing the most critical data characteristics by mapping similarity to spatial distance and highlighting communities of similar attribute combinations. (2) Our system enables the *diagnosis* of results and parameters by allowing the real-time interactive exploration of the parameter space to investigate the influence of parameter choices, enabled by the speculative execution. (3) Our system supports the *refinement* of the involved parameters, supporting the iterative guided optimization of the model to solve a given analysis task.

2 RELATED WORK

To cover the various involved research domains and applications, we structure our related work into sub-topics, summarizing the most relevant works regarding one aspect of our approach.

Visual Analytics Foundations — Similarity searches in large database systems are often automatically executed using predefined similarity functions and distance measures. However, user-adaptable similarity search applications increase in importance, and user integration rises [17]. Visual analytics combines automated analysis techniques with interactive visualizations to enable users to understand and reason about large datasets [6]. Sacha et al. [14] have presented a knowledge generation model that describes how knowledge is generated during the analysis process, building upon prior methodologies in visual analytics [2, 12]. Besides the computer system that visualizes and models data, they describe the human as a core element whose creativity, interaction abilities, and perception help find and comprehend patterns hidden in the data.

Weight Space Exploration — As visual analytics is concerned with integrating human knowledge with automated machine learning, it is frequently used for model exploration and optimization. Sedlmair et al. [16] provide a conceptual framework of visual parameter space analysis, structuring the design space. Pajer et al. [10] present a tool for the visual analysis and exploration of weight spaces, tackling the problem of setting abstract weight parameters. Their tool supports the understanding of sensitivity and helps identify weight regions of interest for a desired output. Mühlbacher et al. [9] present TreePOD, a sensitivity-aware approach to selecting Pareto-optimal decision trees. In contrast to most existing work, we tackle the exploratory analysis of similarity queries and rely on the analyst's intuition rather than on quality metrics.

Parameter Optimization for Mining Models — Parameter optimization for data mining systems or hyper-parameter optimization in machine learning is an open problem that frequently occurs in scientific or industrial use-cases. Analytic optimization or exhaustive search for parameter optimization is often impossible in these models due to black-box methods or high-dimensional parameter spaces. Torsney et al. [22] apply a guided semi-automatic method to this problem by first sampling from the parameter space and then guiding the user by estimating the effects of parameter changes on the result. Schall et al. [15] propose a heat-map method to superimpose the prediction of a deep neural network over its input image. This allows the model engineer to identify problems in the prediction and tune the hyper-parameters accordingly. The resulting workflow is iterative and guided by the provided visualization. This method is applied to offline handwriting recognition, where spatial information is essential but not available in ground-truth data.

Speculative Execution and Guidance — Sperrle et al. [18] present an adaptation of *speculative execution* for visual analytics to support exploratory model analysis and -optimization in visual analytics. Inspired by speculative execution in CPUs, they define it as "the proactive, near-real-time computation of competing model alternatives" to support model state-space exploration. Our system uses speculative execution to execute queries automatically using adapted weights, serving two purposes: first, speculatively preparing those results while the system would otherwise idle enables a near-realtime analysis of related parameter configurations. Second, our system compares all obtained results and guides the user in their exploration by visually highlighting alternative feature weights that produce significantly different results. In recent years, such guidance has been identified as one of the main challenges in visual analytics [3, 4] characterized by user and machine teaching each other while mutually learning from each other [19]. Such guidance enables a more efficient human-machine collaboration and paves the way towards true mixed-initiative [1] systems.

Application Background — Related to our application, we focus on work for similarity search on heterogeneous data collections. Gionis et al. [5] tackle the curse of dimensionality for search in high-dimensional attribute spaces by hashing data entities and performing an approximate nearest-neighbor search on the hashes. Sun et al. [21] present a metapath-based search algorithm, deriving similarity from linkage paths in the network, addressing the advent of heterogeneous information networks. Patroumpas and Skoutas [11] frame the problem as search on enriched, geographical data, i.e., geospatial attributes with additional textual, numerical, or temporal information. Our approach builds upon their work, tackling the open challenge of user-centered model optimization.

3 THE SIMILARITY SEARCH SYSTEM

While search is an essential tool to locate entities of interest in large data foundations, it has significant limitations when the data distribution is unknown and, hence, explorative access to the data is required. Specifically, the exact attribute combination of the results might not be known beforehand, or multiple entities in a particular region might be of interest. The used similarity search (SimSearch) algorithm [11] fulfills these requirements by considering entities that feature attribute combinations close to the desired search parameters. By specifying the number k of ranked closest matches, the analyst can explore the region of interest and refine the search parameters according to the analysis goal. The high-dimensional search space poses particular challenges for the visual representation of the results: pair-wise distances between entities and the root search have to be considered, as well as the influence of each single search attribute.

The variety of data types and -domains that might occur in the data attributes requires the concurrent use of different distance

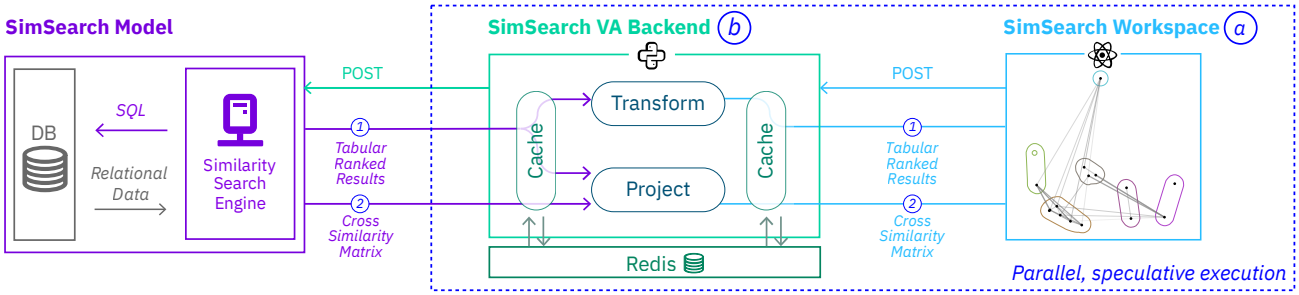

Figure 2: The SimSearch visual analytics system's architecture, split into frontend (a) and backend (b) applications. Search queries are issued to the SimSearch engine, which returns (1) a table of the top-k ranked results and (2) a $k \times k$ cross-similarity matrix, encoding the pair-wise similarities between entities. The results are cached, filtered, projected, and transformed by the SimSearch visual analytics backend before delivering them to the SimSearch workspace frontend.

functions, rendering an objective comparison between the obtained distances impossible. For example, a geospatial attribute might have a real-world geographical distance function associated, while a numeric attribute could exemplary have a logarithmic distance function defined. Figure 4a illustrates the non-comparability of those two measures in a two-axis plot. The similarity search algorithm allows specifying weight parameters in the interval $[0; 1]$ to balance the distance functions between different attributes, tackling this problem. Figure 4b illustrates, how applying weights can scale the search space accordingly.

However, no objective can be optimized to automatically determine the ideal set of weights for a query since it heavily depends on the data domain and the analysis task, rendering human feedback essential for parameter optimization.

We, therefore, identify three fundamental challenges: (1) the high-dimensional and interconnected results must be presented such that the analyst understands their meaning, mapping similarity to the spatial distances in the visualization, (2) the analyst must understand the influence of the parameters on the results, and (3) the interactive exploration of the parameter space must be possible to refine the parameters targeting the analysis goal.

Our proposed visual analytics system makes the similarity search model accessible in a comprehensive workspace, combining different views and panels to address the identified challenges.

3.1 The Similarity Search Backend

To avoid computationally-, time-, and storage-expensive operations in the frontend, our implementation splits the SimSearch system into frontend (2a) and backend (2b). The backend interfaces with the similarity search model, being exposed via REST API. The result of a request to the SimSearch API consists of (1) a ranked list of the top-k similar results together with (2) a $k \times k$ cross-similarity matrix, denoting the pair-wise similarities between every two entities. The raw results are cached by the backend application for later search queries with similar parameters. The results are then transformed from the n-dimensional attribute space down to the two-dimensional screen space and converted into a graph representation using a specified projection algorithm. We include different projection methods to achieve good results for varying search attributes and input parameters: for low-dimensional searches ($n \leq 4$), the system supports PCA and MDS, based directly on the attribute values or the cross-similarity matrix, respectively. For higher-dimensional searches, UMAP [8] can provide fast and stable projections highlighting connections in the data while preserving its global topology. The decisive criterion for choosing the provided projection methods was their ability to derive a stable transformation under a changing set of input vectors. The cross-similarity matrix is filtered for its top k values and converted into a list representation to reduce network load and computational complexity in the frontend. Both results, the projected graph, and the cross-similarity list are then cached and returned to the frontend application. Figure 2 shows the architectural details of the system, including data paths, caching, and the applied data transformations.

Caching Strategy and Requirements — The cache has a crucial impact on the system's responsiveness, requiring the caching strategy to obtain the best possible balance between data topicality and system performance. Since this choice is strongly dependent on the frequency with which data evolution events occur in the data foundation, we tackle this challenge by occasionally querying the similarity search engine despite the results already being present in the cache. Since this strategy triggers a request of multiple similar parameter combinations, the results in the local search space are updated, maximizing the probability of future cache hits with the most recent data entities. The cache's required storage space is neglectable since, in a typical scenario, $k = 50$ can be taken as a reasonable upper-bound for the top-k results of interest. The storage consumption for a query grows linearly with k, except for the $k \times k$ cross similarity matrix, which grows quadratically. Taking the upper bound of $k = 50$, we can estimate its storage consumption as $50 \cdot 50 \cdot 64\,\text{bit} = 160\,000\,\text{bit} \approx 20\,\text{kB}$.

3.2 The Similarity Search Workspace

To allow the interactive analysis of the SimSearch algorithms' results and enable informed decision making during the parameter tuning process, our proposed similarity search workspace combines multiple components in a comprehensive user interface, shown in Figure 1.

Central Projection View (1a) — After defining a search query and receiving the similarity search engine results, the analyst must understand (1) the connection between results and root query and (2) the pair-wise relationship between the results. The SimSearch workspace is built around a central projection view, mapping the n-dimensional data points to the two-dimensional screen space while preserving the distances between entities as well as possible. The search attributes are projected as an additional, virtual entity to set the result entities into relation with the specified search parameters.

Besides the spatial position of entities in the result space, the pair-wise relation between entities is essential to interpret connections and reveal proximities in the data that the projection could not preserve. Therefore, we indicate these relations by extracting the top k values from the cross-similarity matrix and displaying them as links between the respective entities. The edges' line width is proportional to the similarity between two entities, visually highlighting the most important connections.

Important information for each entity is attached directly to the projected node: the similarity rank is annotated persistently on each node, while the exact attribute combinations and similarity scores for each attribute can be displayed by hovering an entity either in the projection view or in the tabular results view. By coloring the results according to their spatial position in the projection using a two-dimensional colormap, the entities are visually clustered and linked to the tabular results view.

Besides displaying the inter-linkage, we also apply k-means clustering to the projected points, reducing visual clutter by forming local groups and highlighting results with spatial proximity in the projection space instead of the attribute space. While the cross-similarity would ideally correspond with the k-means clusters in the n-dimensional space, this is not valid for the projected entities since not all information can be preserved in the projection. Therefore, the clustered entities can share similar attributes, which, at the same time, might diverge from the most similar entities denoted by the cross-similarity matrix. I.e., entities might be close in only a subset of their attributes, causing them to be assigned to the same cluster, while the total similarity across all attributes might be vanishing, preventing their cross-similarity link from being strong enough to be displayed.

Tabular Result View (1b) — Complementing the projection view, we include the tabular results view in the SimSearch workspace, showing the ranked entities together with their attribute set and the corresponding similarity scores. The table's rows are linked to the nodes in the projection view, simultaneously highlighting a specific node in both views on mouse hover. By clicking the table header for one attribute column, the column can be re-ordered according to its contained values, enabling the direct comparison between the individual similarity scores for each attribute.

Parameter Designer (1c) — The parameter designer is the primary interface for specifying and refining search queries, projection settings, and weight parameters. Search attributes can be added from a list of all available attributes in the dataset, allowing to set a target value for each selected parameter. A slider attached to each attribute enables the analyst to set the attribute's relative importance concerning all other defined attributes, giving full control over the balance between attributes and their corresponding distance function.

To diagnose the weight parameters' influence on the result set, hovering a weight slider triggers the projection and tabular result view to switch to the speculative execution state. In the speculative execution state, the views indicate the change in the result set under a speculative de- and increase of the respective attribute weight. In the projection view, this is done by inserting the possible new positions of the entities under the changing projection, marking the results under a positive weight adjustment with a red outline and the results under a negative weight adjustment with a green outline. Complementing the projection, the tabular results view is extended by two additional columns, indicating the change in each result entity's rank and marking

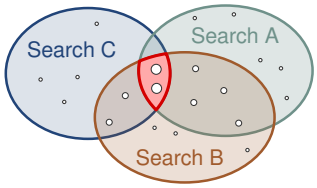

Figure 3: Cached subsets of the search space covered by three consecutive searches with slightly changed parameters. The stability of the central entities is maximal, while the stability for the border-cases vanishes.

entities that are descending from the top k results, causing them to lose their place in the table.

Time-consuming search operations are executed speculatively before an actual user interaction is performed, enabling the iterative refinement of search parameters. When the user performs an action, and the resulting parameter combination causes a cache hit, the results can be delivered and visualized in real-time. Besides the increase in responsiveness, more and more discrete samples of the local search space are present in the cache with the ongoing analysis process. By setting the frequency of an entity in the latest result sets into relation with the total number of results, we derive a measure for an entity's stability over the changing search parameters, as shown in Figure 3. The stability is then mapped to the node size in the projection view, with larger nodes indicating entities that appear more frequently in the recent result sets.

4 USE-CASES

We show the applicability and advantages of the proposed SimSearch workspace based on two exemplary use-cases. The first use-case (subsection 4.1) is hands-on and describes in detail how our proposed system can be used to reach the analysis goal, while the second use-case (subsection 4.2) demonstrates how our system can be applied to varying tasks and domains.

4.1 Assessing the Local Business Landscape

This use-case is based on a real-world, large-scale ($\approx$ 120 GB) dataset containing information about companies in Italy.

In the use-case, a small company with $\approx$ 50 employees plans to expand, for which several potential new locations are considered. Since the company is dependant on the local infrastructure and other supplying companies, geographical proximity to those companies is an essential requirement. Simultaneously, the company wants to avoid direct local competition through other companies working in the same sector and having a similar corporate structure. Our proposed SimSearch workspace supports the search and interactive exploration of the potential company locations to fulfill the company's requirements.

By specifying the attribute combinations in the parameter designer according to the desired or declined company profiles together with the considered company location, the local search space can be explored. The projection view reveals the most similar companies and indicates their pair-wise relationships, revealing communities and enabling the analyst to assess the most influential search attributes. In doing so, it becomes clear that the geolocation only has marginal influence on the search results, and the shown companies are too far for a business relationship.

Since the numerical search attributes, such as the number of employees, can not be objectively compared to the geospatial

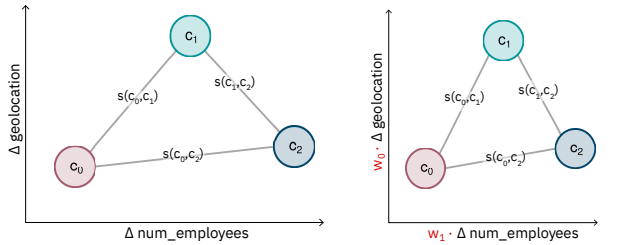

Figure 4: Similarity search results $\{c_0, c_1, c_2\}$ and cross-similarities $s(c_a, c_b)$ with $a, b \in \{0, 1, 2\}$. Search attributes originating from different data domains render an objective comparison of the similarity scores impossible (a). By applying weightings $\{w_1, w_2\}$, the analyst can adjust the distance functions according to his domain knowledge (b).

company location, the weights in the parameter designer have to be iteratively refined to match the analyst's understanding of each attribute's desired influence on the results. Figure 4 shows how the weight adjustment helps to balance the different distance functions. By indicating the changes in the result set for a possible weight adjustment, the analyst can exploit the systems speculative execution feature to observe changes in real-time and assess the most purposeful operation before the actual, time-consuming execution. Using the tabular results view, the analyst can verify the possible changes in detail by observing how each attribute's ranking would change under the operation or if the company would be excluded from the result set. By iteratively refining the search parameters, the analyst can explore the search space ideally for each potential location, leading to well-informed decision making for the new location.

4.2 Mail Forwarding

This use-case is based on internal mail forwarding within a large company. Incoming postal mail is automatically opened and digitized, using an OCR system, on arrival at the company headquarters. The digitized mail item is then used as a search query on a structured database of the company's customers, contracts, products, or projects to electronically forward the scanned document to the staff responsible for working this task needing the document. This use-case requires both a robust search engine for retrieving database entries (e.g., contracts) containing keywords, names, or numerical values similar to the query document and semantic understanding of the content to weigh these attributes.

The structured data in the database consists of categorical attributes, person or item names, as well as spatial, temporal, numerical, or general ontological values. These may occur within the scanned document with different individual similarities as well as in many different combinations. Thus the need arises to weigh these database attributes against each other to model the overall semantic similarity. This configuration of the search query likely is done by a human engineer with expert domain knowledge. One approach here is to use a set of example documents for evaluation and repeatedly querying for them and modifying the attribute weights until relevant database entries are found with high overall similarity to the query document, with less relevant entries being significantly dissimilar.

We propose to use SimSearch in this process to both see the overall similarity of the different database entries using the current configuration and identify clusters in the embedding space. The embedding method will be chosen to reflect the expert's domain knowledge of semantically different and similar documents. Cross-similarities will show potential miss-classifications. This allows adjusting the weights of the similarity search to increase the similarity to semantically relevant documents and separate them from semantically distinct ones.

5 DISCUSSION AND FUTURE WORK

While the presented similarity search workspace implements a variety of features and techniques to make the data search space and the model parameter space accessible by the analyst, possible extensions could further strengthen the system's usefulness. Such extensions could include improvements to the search functionality and the explanation of results or the implementation of advanced guiding techniques. Furthermore, the presented approach could be generalized to other domains and tasks with a similar problem setting, i.e., where high-dimensional result entities of complex mining models have to be visualized, and the model must be refined to match a particular analysis task.

Extending the Search Functionality — Additional views could augment the existing visualizations with an abstract overview of possible actions and the resulting changes, enabling the analyst to identify possible changes at first glance before descending into detailed views. For example, an additional view visualizing all possible weight combinations probed by the speculative execution component and their likely outcomes could provide first hints where the region of interest might be located. Additional interestingness measures could augment the parameter designer's weight sliders with information on the intervals corresponding with the most significant changes in the result set. Extending the interestingness feature, decision boundaries could be estimated by probing the search space in regions with a high gradient, providing a sensitivity analysis for each parameter.

Extending Guidance — The system currently provides orienting guidance to users alerting them to similar weight configurations that produce significantly different search results. In addition to highlighting different possible weight settings, the system could actively propose user actions like moving weight sliders or switching to different projection methods. By analyzing and learning from user interactions, the system could identify the users' preferences and provide suggestions adapted to their understanding of the domain and analysis task. By giving the system more initiative in the exploration process, the system should become both more effective and efficient to use.

Generalization as Visual Analytics Technique — There are several other problems in automated data mining pipelines with the same or a similar structure as the similarity search application addressed in the presented system, such as clustering, classification, or graph merging. Specifically, our approach can be generalized to understand, diagnose, and refine models where (1) the result is a number of n-dimensional entities with arbitrary distance functions associated, and (2) the outcome depends on a set of parameters whose influence on individual results is opaque.

Scalability — The system's scalability is directly dependent on the underlying similarity search algorithm. Despite implementing various techniques (caching, speculative execution) to enable interactive visual analytics on the offline search algorithm, the similarity search model's response time is the limiting factor for the approach. While response times of $1 - 30$ s can be bridged by applying the implemented techniques, longer response times

render an online analysis increasingly difficult since (1) non-ideal sampling points might have been chosen for speculative execution or (2) the analyst might change the search space context more rapidly than results can be preemptively queried and cached. The response times of the similarity search algorithm could be reduced by parallelizing the main stages of the algorithm, namely (1) generating a ranked list of results for each queried attribute and (2) compiling the ranked lists into a list of top-k results [11].

Limitations and Future Work — Currently, views of higher abstraction giving the analyst reference points on promising analysis directions are missing. We will tackle this issue by adding a third view to the similarity search workspace, showing all possible weight combinations in a matrix view and indicating the regions of the highest expected result change. Currently, the analyst has to evaluate the speculative changes in results manually by observing the predicted outcomes and comparing them across the different parameter combinations. In future versions, we will automatically highlight regions of interest using the number of changes in the result set for each combination as an interestingness measure. This functionality will be strengthened by implementing interactive, adaptive guidance. If one operation has significantly higher interestingness than others, it will be actively proposed as a possibly rewarding action. Furthermore, by tracking recent interactions of the user with the system, we will estimate the likelihood of future interactions based on the history, adapting the guidance to user preferences. Despite the presented use-cases proving our approach's applicability in different real-world scenarios and data domains, a future user study will further validate the system's usefulness and provide insights on both benefits and open challenges. Besides measuring quantitative criteria, such as task completion time and comparing the analysis results to ground-truth data, an additional qualitative evaluation will expose additional user requirements and future points for improvements of the system.

6 CONCLUSION

Applying complex data mining models to large data foundations introduces particular challenges to the analysis process. Both the parameter space and the search space might be opaque, requiring manual probing to approach the regions of interest and, hence, rendering an interactive exploration impossible. Applying visual analytics, models, parameters, and results can be made accessible through interconnected visualizations, revealing hidden connections between components and providing advanced mechanisms, such as speculative execution, to enable the real-time exploration of otherwise time-consuming data processing pipelines.

The presented system implements views and techniques to make the parameters and results of a novel similarity search algorithm accessible to the analyst. Specifically, we provide a projected view of the search results, highlighting the similarity to the root query, the pair-wise similarity between the result entities, the stability of the results, as well as communities of close entities. The projected view is complemented with and linked to a tabular view of the results, indicating their rank and providing sorting functions on distinct attributes or their corresponding similarity. Supporting parameter refinement and search space exploration, the system implements speculative execution on the time-consuming similarity search operation, presenting the user with possible outcomes of parameter changes on-demand before actually performing an action. The projection and tabular views are coupled with the parameter refinement functionality, integrating the speculative results into their visual representation.

We show our proposed similarity search workspace's applicability and usefulness based on two use-cases, both anchored in real-world application examples and datasets.

ACKNOWLEDGEMENTS

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825041.

REFERENCES

[1] James F. Allen, Curry I. Guinn, and Eric Horvitz. 1999. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications* 14, 5, 14–23.
[2] Matthew Brehmer and Tamara Munzner. 2013. A multi-level typology of abstract visualization tasks. *IEEE Trans. on Vis. and Comput. Graphics* 19, 12, 2376–2385.
[3] Davide Ceneda, Theresia Gschwandtner, and Silvia Miksch. 2019. A Review of Guidance Approaches in Visual Data Analysis: A Multifocal Perspective. *Comput. Graphics Forum* 38, 3, 861–879.
[4] C. Collins, N. Andrienko, T. Schreck, J. Yang, J. Choo, U. Engelke, A. Jena, and T. Dwyer. 2018. Guidance in the human–machine analytics process. *Visual Informatics* 2, 166–180.
[5] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 1999. Similarity Search in High Dimensions via Hashing. In *Proc. of the 25th Intl. Conference on Very Large Data Bases (VLDB '99)*. San Francisco, CA, USA, 518–529.
[6] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. 2010. *Mastering The Information Age – Solving Problems with Visual Analytics.* Eurographics Association.
[7] Sean D MacArthur, Carla E Brodley, Avinash C Kak, and Lynn S Broderick. 2002. Interactive content-based image retrieval using relevance feedback. *Comput. Vision and Image Understanding* 88, 2, 55–75.
[8] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:stat.ML/1802.03426
[9] T. Mühlbacher, L. Linhardt, T. Möller, and H. Piringer. 2018. TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees. *IEEE Trans. on Vis. and Comput. Graphics* 24, 1, 174–183.
[10] S. Pajer, M. Streit, T. Torsney-Weir, F. Spechtenhauser, T. Möller, and H. Piringer. 2017. WeightLifter: Visual Weight Space Exploration for Multi-Criteria Decision Making. *IEEE Trans. on Vis. and Comput. Graphics* 23, 1, 611–620.
[11] Kostas Patroumpas and Dimitrios Skoutas. 2020. Similarity search over enriched geospatial data. In *Proc. of the Sixth Intl. ACM SIGMOD Workshop on Managing and Mining Enriched Geo-Spatial Data.* ACM.
[12] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. of Intl. Conference on Intelligence Analysis*, Vol. 5. McLean, VA, USA, 2–4.
[13] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. 2017. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing* 268, 164–175.
[14] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim. 2014. Knowledge Generation Model for Visual Analytics. *IEEE Trans. on Vis. and Comput. Graphics* 20, 12, 1604–1613.
[15] Martin Schall, Dominik Sacha, Manuel Stein, Matthias O Franz, and Daniel A Keim. 2018. Visualization-assisted development of deep learning models in offline handwriting recognition. In *Symp. on Vis. in Data Science at IEEE VIS.*
[16] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. 2014. Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Trans. on Vis. and Comput. Graphics* 20, 12, 2161–2170.
[17] Thomas Seidl and Hans-Peter Kriegel. 1997. Efficient user-adaptable similarity search in large multimedia databases. In *VLDB*, Vol. 97. 506–515.
[18] Fabian Sperrle, Jürgen Bernard, Michael Sedlmair, Daniel Keim, and Mennatallah El-Assady. 2019. Speculative Execution for Guided Visual Analytics. arXiv:cs.HC/1908.02627v1
[19] Fabian Sperrle, Astrik Jeitler, Jürgen Bernard, Daniel A. Keim, and Mennatallah El-Assady. 2020. Learning and Teaching in Co-Adaptive Guidance for Mixed-Initiative Visual Analytics. In *EuroVis Workshop on Visual Analytics (EuroVA)*, K. Vrotsou and C. Turkay (Eds.). The Eurographics Association.
[20] Thilo Spinner, Udo Schlegel, Hanna Schäfer, and Mennatallah El-Assady. 2019. explAIner: A visual analytics framework for interactive and explainable machine learning. *IEEE Trans. on Vis. and Comput. Graphics* 26, 1, 1064–1074.
[21] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proc. of the VLDB Endowment* 4, 11, 992–1003.
[22] Thomas Torsney-Weir, Ahmed Saad, Torsten Moller, Hans-Christian Hege, Britta Weber, Jean-Marc Verbavatz, and Steven Bergner. 2011. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Trans. on Vis. and Comput. Graphics* 17, 12, 1892–1901.