

Speculative Execution for Guided Visual Analytics

Fabian Sperrle*
University of Konstanz

Jürgen Bernard†
Technische Universität Darmstadt

Michael Sedlmair‡
University of Stuttgart

Daniel Keim§
University of Konstanz

Mennatallah El-Assady¶
University of Konstanz
University of Ontario Institute of Technology

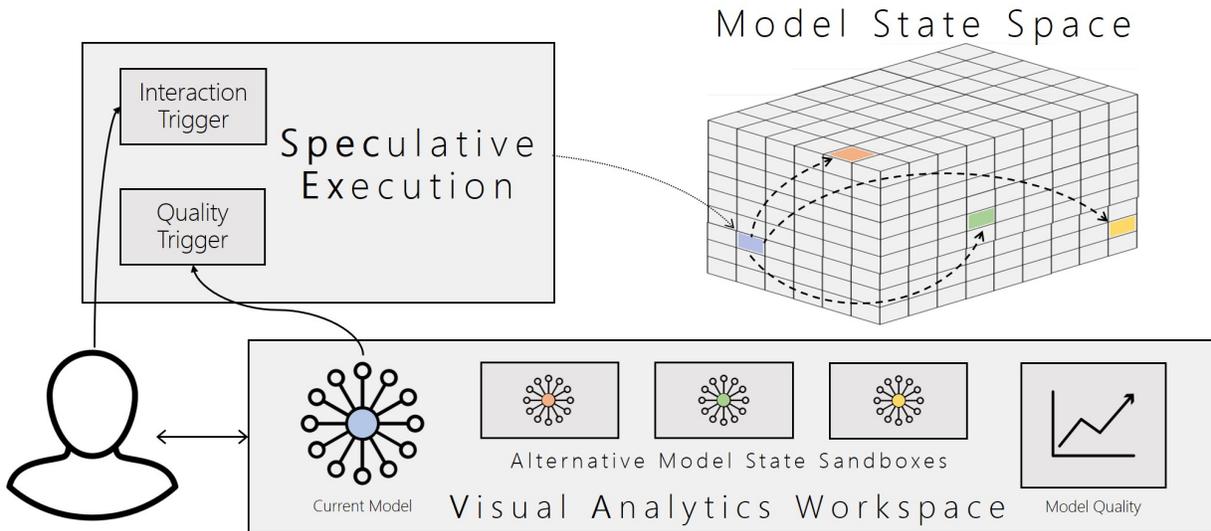


Figure 1: Speculative Execution for Visual Analytics. A model optimization process defines a path through the model state space (shown with only three dimensions here for simplicity). Speculative Execution proactively and automatically computes alternative model states in sandboxes and presents them using delta-visualizations, comparing them to the current model. Speculative Execution can be triggered either through user interaction or model quality metrics.

ABSTRACT

We propose the concept of Speculative Execution for Visual Analytics and discuss its effectiveness for model exploration and optimization. Speculative Execution enables the automatic generation of alternative, competing model configurations that do not alter the current model state unless explicitly confirmed by the user. These alternatives are computed based on either user interactions or model quality measures and can be explored using delta-visualizations. By automatically proposing modeling alternatives, systems employing Speculative Execution can shorten the gap between users and models, reduce the confirmation bias and speed up optimization processes. In this paper, we have assembled five application scenarios showcasing the potential of Speculative Execution, as well as a potential for further research.

Index Terms: Human-centered computing—Visual Analytics—Speculative Execution—SpecEx;

*e-mail: fabian.sperrle@uni-konstanz.de

†juergen.bernard@gris.tu-darmstadt.de

‡michael.sedlmair@visus.uni-stuttgart.de

§e-mail: daniel.keim@uni-konstanz.de

¶e-mail: mennatallah.el-assady@uni-konstanz.de

1 INTRODUCTION

In many disciplines, domain experts have to create and validate multiple possible solutions to a current situation. With the identification of a best possible outcome, a solution can be pursued. In politics, simulations and anticipations have been an integral part of successful campaigns for a long time: politicians and their advisers prepare for different, likely outcomes of an event or a meeting. This allows them to be prepared for different situations that might come up and ensures that they can react in a fast, yet precise way. Similarly, politicians and involved policy analysts create, analyze, and compare alternative solutions before a policy is to be implemented [25]. Although most of the potential solutions will not be needed in the end, their preparation is imperative to guarantee a systematic and timely decision process. In the world of computers, CPUs, for instance, pre-compute conditional code blocks before the outcome of the condition is known in order to avoid waiting and doing nothing. Similarly, multiple pages are fetched from disk or from memory, even when only one has been requested. All of these examples employ Speculative Execution, the principle of preparing or precomputing the result of a task at a time when it is not needed, but its calculation is easier or cheaper thanks to synergy effects or idling resources.

We propose to apply Speculative Execution as a concept for Visual Analytics, as well. There, it can simplify user interactions and propose parameter changes. It is inspired by the human-in-the-loop concept that has become increasingly popular in Visual Analytics over the last years. It integrates human decision-making into the analysis process to obtain results that are based on semantic understand-

ing and fit the user’s expected mental model. More recently, Ender et al. proposed a pattern named “the human is the loop” [14]. They call for new directions of Visual Analytics that focus on recognizing the user’s work process and “seamlessly fitting analytics into that existing interactive process” [14]. Speculative execution picks up this idea and proposes that systems learn the users’ goals from their interactions and provide optimizations that help to reach this goal faster. Additionally, Ender et al. also argue that “implicit steering is perhaps the ultimate form of in-context input”. As Speculative Execution is aimed at understanding the users’ intentions and executing them in the model, it can remove the need for unnecessary, explicit interactions, which are instead performed automatically by the system.

In addition to reducing the gap between users and machine learning models, Speculative Execution can also be helpful for model optimization and model understanding. As a consequence of the increasing complexity of (machine learning) models, interactions with model visualizations are important to foster model-understanding and trust-building [22]. However, these interactions are not always straight-forward. For example, Lee et al. have found that even “seemingly small changes can have unexpectedly large consequences” [21] on the output of the popular topic model LDA [4]. This often leads to users being cautious when interacting with models in fear of “breaking something” [9]. To avoid potentially worsening the current model, Speculative Execution provides isolated computation environments. In those, changes can be applied speculatively. In combination with a delta-visualization between two model states, this allows users to preview the model changes introduced by their interaction in a focused manner.

We deem Speculative Execution (SpecEx) particularly beneficial for the typical Visual Analytics (VA) scenarios. First, it provides the means for effective model optimization and refinement towards the users’ tasks and data. Second, it can help prevent confirmation bias by showing potential modeling alternatives. Finally, it is well-suited for mixed-initiative user-guidance. However, SpecEx has not yet been formally introduced to VA. We demonstrate the applicability of SpecEx for guided Visual Analytics in five usage scenarios in Sect. 4. These scenarios also highlight that SpecEx integrates well with existing VA concepts that will be introduced in Sect. 2.

The contributions of this paper are (1) the introduction of the concept of Speculative Execution to Visual Analytics; (2) an illustration of the value of Speculative Execution in five usage scenarios; (3) design considerations and an implementation model for Speculative Execution; (4) the identification of open research questions for efficient application of Speculative Execution.

2 BACKGROUND

SpecEx as a concept is broadly applicable in many VA scenarios. We discuss related VA techniques here, and present derived usage scenarios in Sect. 4.

Van den Elzen and van Wijk have introduced a visual exploration technique called “Small Multiples, Large Singles” [33] that is similar to Speculative Execution. Starting from a “large single” visualization a small multiples visualization shows alternative models, model parameters, visual mappings, or visualization techniques to the user. Following an alternating sequence of large singles and small multiples, users can either use the tool to explore the data space or to adjust both the model and visualization for their use case. While the approach focuses on the navigation-support provided with the small multiples (e.g., by trial-and-error), the SpecEx concept also incorporates the suggestion of next step that may be meaningful to succeed in some task.

SpecEx integrates well with the concept of provenance tracking. Systems like AVOCADO [32], VisTrails [6] or SenseMap [23] focus on visualizing the provenance of results during or after an analysis session. Some systems also give users the possibility to revert to a previously seen model configuration. SpecEx enables a more

straightforward comparison of two model states, as they can both be instantiated in isolated environments at the same time. Combined with a delta-visualization this provides a powerful tool for understanding how specific interactions have influenced the model building process.

If Speculative Execution is based on quality metrics instead of user interactions, it becomes an alternative to Visual Parameter Space Analysis, e.g., conducted by Sedlmair et al. [28]. In this case, sandboxes would be created with different parameter configurations following the speculation dimensions presented in Sect. 5.2. However, as SpecEx is an interactive, mixed-initiative approach, the exploration of the search space and computation of alternative models can be more restricted to those in which users are interested. Also, this interactivity allows Speculative Execution to modify the parameter sampling methods, focusing on parameter ranges that seem to be relevant to the users in a given analysis session.

The concept of SpecEx pairs well with progressive computation. As SpecEx observes user interactions with the system (see Sect. 5.1), systems can observe which sandboxes users are most interested in, and progressively refine their models. This allows initial sandbox models to be computed with a lower degree of detail, enabling the computation of more sandboxes with the same resources. The necessary details can then be computed once they are necessary.

3 SPECULATIVE EXECUTION AS A VA CONCEPT

We first define *Speculative Execution* and *Computational Sandboxes*, as well as the *Model Search Space*. We then combine the introduced concepts into an *implementation model* that can be applied to existing Visual Analytics systems.

3.1 Definitions

All three of the important terms have previously been used in different areas of computer science. We present related definitions that are tailored towards the use of the terms in Visual Analytics.

Speculative Execution Typically, *Speculative Execution* describes a set of CPU optimization techniques like branch prediction [39] or data prefetching [16] that can improve their performance. In the context of Visual Analytics, we describe SpecEx as the proactive computation of alternative, competing model states that are isolated from the current model state and do not influence it. On the one hand, such a proactive computation can be triggered by user interaction. In this case, the computation completes the interaction or performs similar operations. More detailed distinctions will be made in Sect. 5.1. On the other hand, the computations can have the goal of model optimization. They are then typically triggered by model quality metrics and can, for example, be used to explore different parameter settings. Combining these two aspects, we define SpecEx as follows:

Speculative Execution describes the proactive, near real-time computation of competing model alternatives that do not influence the current model state, explores the model state search space, and is triggered by either interaction or quality measures.

Comparing the differences between two competing model states, users can decide whether any SpecEx was useful or not. If yes, they can accept the proposed sandbox, or else reject it. Using these two simple operations, users can steer how the model state space is searched. Over time, SpecEx can learn from this user interaction and rank proposed sandboxes higher if they conform to similar schemas as sandboxes that have previously been accepted by the user. This improves the quality of suggestions and facilitates the intended tasks of model optimization and exploration further. Imagining model optimization as a “walk” through the model state space, SpecEx allows users to “look to the left and right of the path”. This differentiates

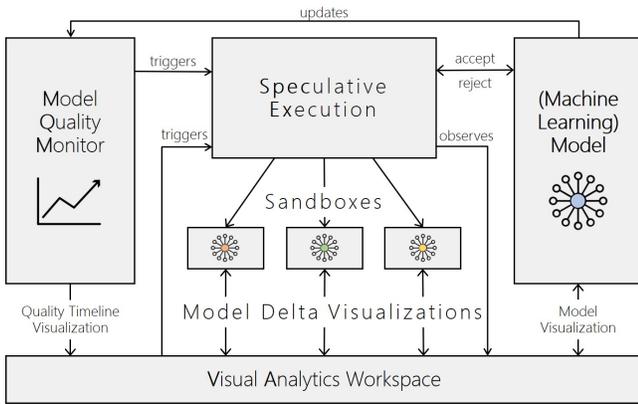


Figure 2: Implementation architecture for Speculative Execution in Visual Analytics. The *Speculative Execution* and *Model Quality Monitor* components can be retrofitted to existing VA systems, together with one or multiple *Model Delta Visualizations*.

SpecEx from “normal” human-in-the-loop Visual Analytics that incorporates the user without necessarily focusing on exploration or proposing modeling alternatives.

Sandboxes—isolated containers holding models—and the model state search space—the theoretical, high-dimensional space of possible model configurations—will be introduced and defined in the following paragraphs.

Sandboxes In general, sandboxes are environments with limited connections to their respective outside world, enabling a separated and encapsulated computation. In computer science, the term sandbox is often used to describe shielded environments or local working copies that are used to execute some task. Application areas include virtual machines, software installation facilities, or security-related environments. Sandboxes in visual analytics have often been used in collaborative or multi-device environments to show all users different and tailored views of the data [19]. “Sandbox” is also the name of a visual sensemaking system introduced by Wright et al. [36]. Additionally, “sandboxes” is a term often used in relation with computer- and system-security: browsers use JavaScript sandboxes [7, 18] and systems move suspect executables to sandboxes [37].

Building upon these characterizations and examples, we adopt the notion of sandboxes and describe them as *isolated computation environments* that can be initialized with any model state. The result of any sandbox computation should always be a valid model state, such that any model can be replaced with a sandbox derived from it easily. These sandboxes can help to explore different, alternative hypotheses, which analysts often do to avoid confirmation bias [17].

Search Space The search space for Speculative Execution is defined by all possible model states, given by the input data and all parameters. SpecEx defines any number of these inputs as *speculation dimensions* that can be modified before starting a proactive computation. As a result, the Speculative Execution Space is significantly smaller than the entire model state space, as it does typically not make sense to change all variables before launching a computation. An elaboration on the different potential sizes of the search space can be found in Sect. 5.3.

3.2 Implementation Model for VA with SpecEx

Having defined the concepts of Speculative Execution, we now describe how it can be integrated into an existing VA workspace. The result of such an integration is shown in Fig. 2. The central component is the VA workspace itself with visualizations of both the

model and, if available, a set of quality metrics computed by the *Model Quality Monitor*. The *Speculative Execution Component* can create model sandboxes and instantiate them with copies of the current model. These sandboxes can either be visualized individually or in a delta-view comparing them to a different model state; typically the “original” model from which the sandbox was started. The *Speculative Execution Component* also constantly monitors the VA workspace and tracks the users interactions, searching for patterns and trying to determine the users intentions. The possible outcomes of this process will be described in Sect. 5.1. Thus, any sandbox can either be triggered by observed user interactions or, more simply, by the *Model Quality Monitor*. This component combines multiple metrics indicating the quality of the model, and employs different strategies for combining these metric values into a “trigger-decision”. This generic implementation model can be taken into account when designing new Visual Analytics systems, or be retrofitted to existing VA systems thanks to its modularity. It follows the architecture for “human centered machine learning” presented by Sacha et al. [26]. Speculative interaction ties into the *Validation & Interaction* stage of their proposed framework.

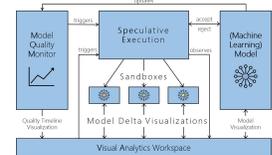
SpecEx can reuse existing techniques to determine the users intentions [5, 8], and employ them in sandboxes. The ranking and proposition of sandboxes to the user can be incrementally improved throughout the analysis session. For example, a higher weight can be given to those sandboxes that have been created according to a schema that has often been accepted by the user in the past. However, such adaptations to the weighting scheme have to be carefully considered to avoid creating and confirming biases in the user’s mental model.

4 USAGE SCENARIOS

To illustrate the idea of Speculative Execution in Visual Analytics, we now discuss five usage scenarios in which SpecEx was either already applied successfully, or that would build a valuable basis for an extension towards SpecEx. The first scenario reports the results of the initial implementation of SpecEx for topic modeling. The second and third scenario highlight the use of SpecEx for implicit steering and user guidance, respectively. Scenario four exemplifies the value of SpecEx for cooperative VA, and scenario five shows that SpecEx can also be applied to visualization and is not limited to VA. Except for scenario one, these usage scenarios are hypothetical and have not been implemented yet. The small figures under each usage scenario repeat the implementation architecture model from Fig. 2. The individual components that are “active” in each of the scenarios have been coloured in blue, highlighting the versatility of SpecEx. Different amounts of sandboxes have been coloured throughout the examples and represent whether a small, medium or large number of sandboxes are expected to be computed.

4.1 Model Optimization

Topic modeling is a popular technique to segment a text corpus into thematically related clusters. Consequently, refining the results of topic models and adapting them to a particular set of users, data, and tasks, is an active area of research [10]. In our recent work, we examined the optimization of the Incremental Hierarchical Topic Model (IHTM) using SpecEx [11]. This model represents topics as a tree—with leaf nodes being documents, and inner nodes representing hierarchically ordered topics. This data structure can be explicitly manipulated through various optimization strategies that, for example, split or merge topics, remove outliers, or compact the topic tree. In combination with typical topic model quality metrics like the number and size of topics, their coherence or pointwise mutual information, a system can speculatively start the available optimization strategies in



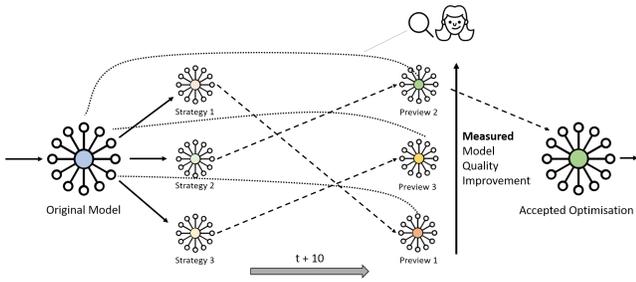


Figure 3: Example of two-dimensional SpecEx over time and different optimization strategies. Three optimization strategies for an incremental topic model are triggered in sandboxes and forecast over the next ten document inserts. The resulting models are sorted according to their quality, and presented to users as a diff with the current model. Applying their domain knowledge, users select the best model to continue the process with.

individual sandboxes whenever any (combination) of the metrics declines. Once the sandbox computations are complete, the results can be ranked according to their measured quality and be presented to the user. We have implemented and successfully tested such a system in our previous work [11]. The topic trees of two topic models are merged into a single tree, and added, moved and removed topics and documents are highlighted to help users see differences between the two models and select a sandbox to continue the computation with. This workflow is outlined in Fig. 3.

As the user observes the model building process of the IHTM on the “20news” dataset, a document from the “atheism” group is added to the “christian” topic, making it too broad and leading to Speculative Execution. The user inspects the proposed optimization leading to the highest measured quality improvement: merging the two topics “mideast” and “baseball”. While this merge optimizes the quality metrics, the user quickly rejects it, employing their semantic understanding. Instead, they select a different sandbox in which the erroneously combined “car” and “gun” topics have been split.

Using SpecEx the analysis system was able to integrate the user into the model steering process seamlessly. Instead of defining must-link and cannot-link constraints to optimize topics, the user could select from a set of prepared optimizations. Especially in topic models, where often subtle semantic differences decide over the quality or even correctness of a topic attribution, SpecEx can help to achieve good results by exploring a wider area of the search space instead of only a single model.

4.2 Implicit Steering

Our second example addresses the task of labeling datasets which is increasingly supported with VA techniques having the human in the loop in interactive machine learning settings. In their current state, these VA systems support users in the labeling process with visual-interactive interfaces showing data characteristics as well as information about the current model state [3]. Example labeling interfaces include scatterplots [2] in combination with dimensionality reduction [27], radvis-like visualizations [29], or list-based interfaces [24], e.g., in combination with active learning models [30]. Interaction techniques that enable the assignment of labels are based on simple item selection or drag-and-drop facility. Users of the systems start by creating a small set of training data with some labeling interactions, triggering the machine learning algorithm to be re-trained in an iterative way. Intermediate results of the mod-

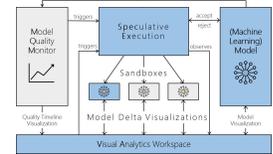
els are visualized in the labeling interfaces which closes the human-centered feedback loop [3].

Observing these interactions such a system could understand and learn the pattern in the users’ interaction. It could then find potentially misclassified data instances by using quality metrics or assessing the spatial relationships of data instances in the high-dimensional data space or the visual space, or both. In a series of parallel sandboxes, the system could “auto-complete” different interaction patterns the users have started, and present a list of changes together with the re-trained alternative models. The user can then inspect the list of instances which switched their prediction of the machine learning model and accept or reject the sandbox.

If the system can detect and understand these *semantic interactions* [14] as implicit steering commands, it can learn about the user’s intuition (intents [38]). It can then apply this intuition to the remaining data, removing the need for users to explicitly complete their interaction pattern, reassigning even more instances.

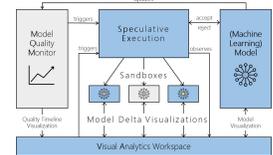
4.3 User Guidance

Recent work in Visual Analytics has often focused on measuring and avoiding confirmation bias [1, 34]. Speculative Execution can help to prevent confirmation bias by providing alternative sandboxes, as well. A different approach has been taken by Wall et al. with their system PODIUM [35]. It aims at making the users mental model visible by letting them rank data according to their intuition and preference. In their example with college football teams, users drag and drop some teams that they have an opinion about to a new position in the list. Additionally, users can indicate whether the model should put more or less weight on individual features. The system then learns feature weights from that ranking and reorders the list according to the learned model. As a result, the relative differences between teams ranked by the user can change. Here, Speculative Execution could explore alternative feature weightings that lead to a ranking that is closer to what the user originally expressed. The system could then guide users and highlight that changing certain feature weights would lead to the model more closely representing their originally expressed order. Users can then verify the changes and accept them, if they agree. Alternatively, this could lead to users realizing that the proposed changes do not fit their mental model. As a result, they might start questioning the said model, exploring the data further, and overcoming their biases. This scenario shows that SpecEx can not only be used to speed up model optimization or auto-complete user interactions. Instead, it can be used to explain model changes by highlighting features with a high impact on the current modeling situation.



4.4 Cooperative Visual Analytics

Linguists are interested in classifying questions on whether they are information-seeking (ISQ) or non-information-seeking (NISQ), i.e., rhetorical [31]. Their data consists of transcripts of conversations or written text. The task of an expert analyst is to train a classifier using a VA system. Such a system might, for example, provide the context before and after the question, as well as information on the respective speakers. This task is interesting for cooperative analysis because discussions with linguists have shown that there is frequent disagreement on whether a question is an ISQ or not, even between experts. In current classifier training systems, this would lead to prolonged decisions on how to classify training data. With Speculative Execution, disagreeing experts can create two sandboxes, and each classify some training data according to their understand-



ing. If this task is executed on a large, collaborative touchscreen, both experts can even train their model in parallel. After retraining the respective models, the experts can compare which model captured the particularities of the current dataset better. In a provenance-tracking view, they can compare which training decisions were most important for making one of the classifiers more adapt to the data. After a short discussion, the experts can now agree on a common understanding of ISQ/NISQ that is suitable for the dataset.

4.5 Speculative Execution beyond Machine Learning

Speculative Execution can also be extended beyond machine learning scenarios. One possible extension is towards visualization systems that focus on presenting data without an underlying model. In this scenario, we use the example of a system visualizing geographical movement data of cars on a map. The “envirocar” data set¹ contains GPS tracks of driving cars and is annotated with speed, the fuel consumption, rpm, et cetera. As it contains about 1.7 million rows, it cannot be visualized on a map without some preprocessing and aggregation. One such aggregation step is the bundling of trajectories to avoid overplotting. Additionally, a system may show a summary of the presented data in a detail panel. Such information is, for example, useful to families searching for a home in a quiet area without too much traffic and with low emissions. To make the interaction with the system more seamless, it uses eye-tracking to determine regions that users are interested in. Once such a region is determined, it zooms in and provides more detail. In combination with SpecEx this zooming process is smoother. While the system is still determining whether a user is interested in a region or was only glimpsing at it, it can already precompute the aggregations on the new level and prepare the detail panel. If the zoom-in action is performed later, the data is already preprocessed and can be displayed.

5 ASPECTS OF SPECULATIVE EXECUTION

Having shown the applicability of SpecEx in various VA scenarios, we provide a list of theoretical considerations. They stem from both our experience with implementing SpecEx for topic modeling and the hypothetical usage scenarios. We conclude by providing design principles for SpecEx.

5.1 Semantic Complexity of Interactions

User interaction can build the basis for relevant input for SpecEx. Depending on the level of *semantic complexity* of such an interaction, the goals of the Speculative Execution are different. We define semantic complexity as the amount by which the mental model of the user changes by performing this interaction. The different levels will be described in the following.

L1: Start of Interaction Once the user has started an interaction, the system can assess what the user is trying to achieve. It can then try to predict how this interaction can be finished using extrapolation techniques. The goal of Speculative Execution on this level is to prepare the context of the interaction target. During a drag and drop operation, for example, such computations could include the search for relevant “drop targets”.

L2: Completion of Interaction Every interaction that was completed can build a basis for similar interactions that may be performed in future. Whenever an interaction has been finished, the system can try to predict the next interaction on a similar level of complexity that the user might want to perform. The goal of such speculations is to guide the user in exploring the potential impact of interactions. Also, Speculative Execution at this level could make

users aware of data points that might have been missed, for example when removing outlier nodes.

L3: Repetition of Interactions In many cases, users combine different low-level interactions to solve a higher-level intent [38]. As such, combinations of interactions are at the highest of the semantical levels of speculation that we want to address. Tracking and contextualizing of low-level interactions to make sense of user intents builds the basis for the support of SpecEx at this level. Once a user intent has been identified, the system can begin searching for similar intents that users might have, and that could be solved with a similar set of repeated low-level interactions.

To some extent, the goal on all levels of semantic complexity is “User Intention Guessing”: the system needs to determine the *implicit interaction* [14] that the user is trying to achieve by performing the explicit interactions. Once the user’s intention has been identified, appropriate optimizations, parameter changes, or additional computations can be searched in a much more focused manner.

5.2 Dimensions of Speculation

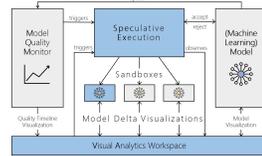
Speculative Execution and sandboxes provide the theoretical framework for an efficient, mixed-initiative guidance approach to visual analytics, extending the Visual Analytics pipeline by Keim et al. [20]. Whenever SpecEx is not proposing optimizations to complete user interactions, it can target improvement in the measured model quality and prepare alternative models. Their usefulness depends on the model-specific *speculation dimensions*. These speculation dimensions are defined by parameters or properties of the underlying model. In any computed sandbox, some of the possible speculation dimensions may be altered with respect to the “original” model. The following section highlights guidelines for the selection of speculation dimensions. Although SpecEx is not limited to those, we present five distinct categories of dimensions here.

Temporal Dimensions Temporal dimensions include the actual time as well as an iteration number, depending on the model. They are especially interesting for incremental algorithms. Incremental streaming models can build a buffer of events or data and “forecast” the development of the model in any given sandbox. For iterative algorithms, a sandbox can show the model development over the next iterations. *Progressive* sandboxes can be continuously refined as new data becomes available. This can allow users to pursue multiple model alternatives in parallel, before deciding for one.

Optimization Strategies For various models, direct optimizations, manipulating the models’ underlying data structures, can be conceived. Operations could include merging or splitting tree nodes, changing values of matrices and vectors, or introducing a threshold. These optimization strategies can be tailored towards known potential model issues and provide bespoke solutions for these problems, without having to explore and change the model’s parameters. Such strategies have been implemented in the system presented in 4.1 [11] and provide sandboxes avoiding topic chaining or combining small, overly specific topics into more easily understood generalized ones.

Utilizing bespoke optimization strategies incurs an additional implementation cost for identifying potential model issues and developing possible solutions. However, it can directly address problems that would be unintuitive, difficult, or even impossible to change through model parameter changes.

Model Parameters If no bespoke optimization strategies are available, the sandboxes of SpecEx can perform a Visual Parameter Space Analysis [28], creating different sandboxes for different parameter configurations. However, instead of sampling and precomputing the entire parameter space, the analysis can be focused on regions of the search space that are similar to the users current model. As soon as users start exploring new regions of the model state space, new sandboxes can be created and prepared in the background.



¹ <https://old.datahub.io/dataset/envirocar>, last accessed 7/15/2018

Input Transformations If the preprocessing pipeline is integrated into the Visual Analytics system, SpecEx sandboxes can utilize these preprocessing algorithms to transform the underlying data. Examples of such transformations include a stricter outlier-removal, filtering out stopwords from a collection of text documents, or introducing minimum and maximum-thresholds for time-series data.

Algorithm Modifications Ultimately, SpecEx sandboxes can also explore modifications of the original model or any of its parts. They can replace similarity functions, feature weighting schemes or merging strategies, to name just a few. As this dimension is specific to the underlying model, it can be very effective and powerful. As with any *Input Transformations*, the system needs to explicitly inform the users of any changes made to these dimensions. Modifications here might have an impact on the user’s mental model and how well it fits.

5.3 Towards Design Principles

In the following, we provide design considerations for effective Speculative Execution in five areas.

Speculation Dimensions When performing speculative parameter space analysis or employing bespoke optimization strategies, there is a trade-off between the number of sandboxes that are created and their usefulness. As one goal of Speculative Execution is a more guided exploration of the model state space, more sandboxes are beneficial: they lead to more model configurations being calculated and presented to the user. However, in addition to the increased need for computational resources, users cannot and will not inspect and compare a large number of speculative sandboxes. In our first implementation of Speculative Execution for the optimization of topic models, we offered users the results of seven speculative optimization strategies [11]. In the evaluation study, users often focused on the top-two or top-three optimizations according to our provided ranking. One possible reason is that model comparison is a difficult task. It might be alleviated by effective delta-visualizations, but users will remain unable to compare all sandboxes that can be computed.

Runtime Depending on the level of semantic complexity of an interaction triggering speculative execution different runtime requirements apply. With increasing complexity of the performed interactions, increasingly complex Speculative Executions are necessary to support and guide the user. However, the runtime requirements for these more “complex” sandboxes triggered by L2 or even L3 interactions are not as strict as for those triggered by L1 interactions that need to be executed while the user is performing an interaction like dragging and dropping an object. Here, the aim should be on focused and short computations of less than 500ms. As soon as the user ends the interaction, the speculation becomes meaningless. However, the resulting sandboxes for L2 and especially L3 interactions are likely still useful after a couple of seconds. It is important that such longer-running speculations do not block the user interface as to not interrupt the analysis workflow.

Search Space Size As we have previously implemented SpecEx for IHTM [11], an incremental topic model building a tree structure, we elaborate on the size of the individual search- and model-spaces using a concrete example. We consider a corpus containing 280 documents (k) and a two-dimensional speculative execution with $n = 7$ optimization strategies and $b = 10$ additional documents being inserted into the model from the buffer during a speculation. Running the model without optimization will produce exactly one topic-tree as its output. With SpecEx, we compute at most $(k/b) \cdot n = 196$ sandboxes. This allows us to involve the user in the algorithmic decision-making process but is significantly more scalable than considering all possible optimization paths (even considering the buffer), which would result in $n^{(k/b)} \approx 10^{23}$ options. Even though this would be an exponentially large search space, it is still

multiple orders of magnitude smaller than all possible trees an incremental algorithm would consider ($k! \approx 10^{565}$), or all possible trees with k nodes ($k^{(k-2)} \approx 10^{680}$). This example shows that the Speculative Execution reduces the factorial search space to a linear one. This is due to the use of only two dimensions with a very limited set of possible values. Here, careful considerations weighing search space exploration against computation time are necessary. One interesting area for future research is the formalization of a cost-benefit model for Speculative Execution that can be used as guidance when selecting speculation dimensions.

Quality Metrics Whenever Speculative Execution is not triggered by user interaction it needs quality metrics to trigger and assess sandboxes. The user typically performs a multi-objective optimization of these metrics when trying to improve a model. As a result, good consensus strategies between the used metrics are necessary to sort the computed sandboxes before presenting them to the user for exploration. Note that it is typically not possible to fully automate this optimization process, as most quality metrics do not capture (all) semantic details.

Delta-Visualization For an efficient SpecEx, an effective delta-visualization is necessary to highlight the differences between the current model and a selected sandbox. This visualization needs to be tailored to the underlying model, the visualization from which the Speculative Execution was triggered, and the number of changes introduced in the speculation. Gleicher et al. [15] introduced various patterns for visual comparison. While different patterns are useful in different situations, we argue that explicit encoding should be present in comparative sandbox visualizations to help users to quickly focus on the introduced changes that decide over accepting or rejecting the proposed sandbox. For some tasks like labeling, a list of elements with changed labels might be more useful than a complex visualization trying to highlight the differences between two dimensionality-reduction results.

5.4 Research Opportunities

With Speculative Execution being a novel concept in Visual Analytics, many interesting questions remain.

Mapping Interaction to Optimization Understanding the intent of user interactions is paramount for effective Speculative Execution. While systems using implicit steering exist today, it is an open field of research how interactions on the different levels of semantic complexity can be understood and mapped to concrete goals for a speculative sandbox. Endert et al. have already identified the capturing of user interaction intentions as relevant future work [12]. In addition, they have elaborated on design considerations and confidence levels of captured interactions [13]. Further research should investigate how such captured interactions can be generalized and re-applied to complete the user’s semantic interactions.

Cost-Benefit-Model As was alluded to in the previous section, choosing the wrong (number of) speculation dimensions is detrimental to the usefulness of SpecEx. An information-theoretic cost-model for Speculative Execution could define the size of the search space, the number of visited states, the computation time and the cognitive load on users. Such a cost-model would then allow informed choices on the sandboxes SpecEx computes.

Interaction Design Speculative Execution constantly computes alternative models that users might want to explore and needs to present them for inspection. However, systems should not constantly interrupt the analyst’s workflow for model-comparison. Further research should be conducted to determine when and how to show sandbox results, and when to refrain from interrupting the user. Furthermore, research should investigate how different presentation styles of SpecEx results impact the creation and confirmation of biases.

6 CONCLUSION

We have introduced Speculative Execution as a new methodology for Visual Analytics. Based on a definition and formalization of SpecEx for VA, we presented five possible usage scenarios that demonstrate the applicability of SpecEx in VA. Finally, we characterized different aspects of SpecEx in VA, showing that SpecEx is a multifaceted concept that can support VA in different ways. The primary benefits of successfully adopting SpecEx into the VA processes are multiple ways in which user guidance can be provided algorithmically, conflated with visual-interactive interfaces. Next steps include the implementation of Speculative Execution in several VA applications to further demonstrate the applicability of SpecEx for guided VA. Likewise, the elaboration of formerly discussed research opportunities will lead to new insights.

7 ACKNOWLEDGMENTS

This work has been funded in part by the Deutsche Forschungsgemeinschaft (DFG) within the project “Visual Analytics and Linguistics for Interpreting Deliberative Argumentation (VALIDA)”, Grant Number 376714276, as part of the Priority Program “Robust Argumentation Machines (RATIO)” (SPP-1999).

REFERENCES

- [1] D. Arnott. Cognitive biases and decision support systems development: a design science approach. *Information Systems Journal*, 16(1):55–78, 2006.
- [2] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, and M. Sedlmair. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):298–308, Jan 2018.
- [3] J. Bernard, M. Zeppelzauer, M. Sedlmair, and W. Aigner. VIAL: a unified process for visual interactive labeling. *The Visual Computer*, pages 1–19, Mar 2018.
- [4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.
- [5] E. T. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding waldo: Learning about users from their interactions. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):1663–1672, Dec 2014.
- [6] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Vistrails: visualization meets data management. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 745–747. ACM, 2006.
- [7] A. Dewald, T. Holz, and F. C. Freiling. Adsandbox: Sandboxing javascript to fight malicious websites. In *Proc. ACM Symp. on Applied Computing*, pages 1859–1864. ACM, 2010.
- [8] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning processes from user interactions. *IEEE Computer Graphics and Applications*, 29(3):52–61, May 2009.
- [9] M. El-Assady, R. Sevastjanova, D. Keim, and C. Collins. Thread-Reconstructor: Modeling Reply-Chains to Untangle Conversational Text through Visual Analytics. *Computer Graphics Forum*, 37(3):351–365, 2018.
- [10] M. El-Assady, R. Sevastjanova, F. Sperrle, D. A. Keim, and C. Collins. Progressive Learning of Topic Modeling Parameters: A Visual Analytics Framework. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):382–391, Jan. 2018.
- [11] M. El-Assady, F. Sperrle, O. Deussen, D. Keim, and C. Collins. Visual Analytics for Topic Model Optimization based on User-Steerable Speculative Execution. *to appear, IEEE Trans. on Visualization and Computer Graphics*, 2018.
- [12] A. Endert, R. Chang, C. North, and M. Zhou. Semantic interaction: Coupling cognition and computation through usable interactive analytics. *IEEE Computer Graphics and Applications*, 35(4):94–99, July 2015.
- [13] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 473–482. ACM, 2012.
- [14] A. Endert, M. S. Hossain, N. Ramakrishnan, C. North, P. Fiaux, and C. Andrews. The human is the loop: new directions for visual analytics. *Journal of intelligent information systems*, 43(3):411–435, 2014.
- [15] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.
- [16] J. González and A. González. Speculative execution via address prediction and data prefetching. In *Proc. Int. Conf. on Supercomputing*, pages 196–203. ACM, 1997.
- [17] R. J. Heuer. *Psychology of intelligence analysis*. 1999.
- [18] L. Ingram and M. Walfish. Treehouse: Javascript sandboxes to help web developers help themselves. In *USENIX Annual Technical Conference*, pages 153–164, 2012.
- [19] W. Jentner, M. El-Assady, D. Sacha, D. Jäckle, and F. Stoffel. Dynamo: Dynamic Monitoring Interface for Task Ensembles. In *IEEE Conf. on Visual Analytics Science and Technology (VAST Challenge MCI)*, 2016.
- [20] D. Keim, G. Andrienko, J. D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual analytics: Definition, process, and challenges. In *Lecture Notes in Computer Science*, pages 154–175, 2008.
- [21] T. Y. Lee, A. Smith, K. Seppi, N. Elmqvist, J. Boyd-Graber, and L. Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *Int. J. of Human-Computer Studies*, 105:28–42, 9 2017.
- [22] S. Liu, X. Wang, M. Liu, and J. Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, 1(1):48–56, 3 2017.
- [23] P. H. Nguyen, K. Xu, A. Bardill, B. Salman, K. Herd, and B. W. Wong. Sensemap: Supporting browser-based online sensemaking through analytic provenance. In *IEEE Conf. on Visual Analytics Science and Technology*, pages 91–100. IEEE, 2016.
- [24] C. Ritter, C. Altenhofen, M. Zeppelzauer, A. Kuijper, T. Schreck, and J. Bernard. Personalized Visual-Interactive Music Classification. In *EuroVis Workshop on Visual Analytics*, 2018.
- [25] T. Ruppert, J. Bernard, and J. Kohlhammer. Bridging knowledge gaps in policy analysis with information visualization. In *Conf. on Electronic Government*, volume 221 of *LNI*, pages 92–103, 2013.
- [26] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164 – 175, 2017. Advances in artificial neural networks, machine learning and computational intelligence.
- [27] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):241–250, 2017.
- [28] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringner, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.
- [29] C. Seifert and M. Granitzer. User-based active learning. In *IEEE Int. Conf. on Data Mining Workshop*, pages 418–425, 2010.
- [30] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.
- [31] R. Sevastjanova, M. El-Assady, A. Hautli-Janisz, A.-L. Kalouli, R. Kehlbeck, O. Deussen, D. Keim, and M. Butt. Mixed-initiative active learning for generating linguistic insights in question classification. In *Workshop on Data Systems for Interactive Analysis (DSIA) at IEEE VIS*, 2018.
- [32] H. Stütz, S. Luger, M. Streit, and N. Gehlenborg. Avocado: visualization of workflow-derived data provenance for reproducible biomedical research. In *Computer Graphics Forum*, volume 35, pages 481–490. Wiley Online Library, 2016.
- [33] S. van den Elzen and J. J. van Wijk. Small multiples, large singles: A new approach for visual data exploration. In *Computer Graphics Forum*, volume 32, pages 191–200. Wiley Online Library, 2013.
- [34] E. Wall, L. M. Blaha, L. Franklin, and A. Endert. Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics. In *IEEE Conf. on Visual Analytics Science and Technology*, 2017.
- [35] E. Wall, S. Das, R. Chawla, B. Kalidindi, E. T. Brown, and A. Endert.

- Podium: Ranking data using mixed-initiative visual analytics. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):288–297, 2018.
- [36] W. Wright, D. Schroh, P. Proulx, A. Skaburskis, and B. Cort. The sandbox for analysis: concepts and methods. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*, pages 801–810. ACM, 2006.
- [37] B. Yee, D. Sehr, G. Dardyk, J. B. Chen, R. Muth, T. Ormandy, S. Okasaka, N. Narula, and N. Fullagar. Native client: A sandbox for portable, untrusted x86 native code. In *IEEE Symposium on Security and Privacy*, pages 79–93, May 2009.
- [38] J. S. Yi, Y. a. Kang, and J. Stasko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1224–1231, Nov 2007.
- [39] C. Zilles and G. Sohi. Execution-based prediction using speculative slices. *ACM SIGARCH Computer Architecture News*, 29(2):2–13, 2001.