

Interface Metaphors for Interactive Machine Learning

Robert J. Jasper^(✉) and Leslie M. Blaha

Pacific Northwest National Laboratory, Richland, WA, USA
{robert.jasper,leslie.blaha}@pnnl.gov

Abstract. To promote more interactive and dynamic machine learning, we revisit the notion of user-interface metaphors. User-interface metaphors provide intuitive constructs for supporting user needs through interface design elements. A user-interface metaphor provides a visual or action pattern that leverages a user’s knowledge of another domain. Metaphors suggest both the visual representations that should be used in a display as well as the interactions that should be afforded to the user. We argue that user-interface metaphors can also offer a method of extracting interaction-based user feedback for use in machine learning. Metaphors offer indirect, context-based information that can be used in addition to explicit user inputs, such as user-provided labels. Implicit information from user interactions with metaphors can augment explicit user input for active learning paradigms. Or it might be leveraged in systems where explicit user inputs are more challenging to obtain. Each interaction with the metaphor provides an opportunity to gather data and learn. We argue this approach is especially important in streaming applications, where we desire machine learning systems that can adapt to dynamic, changing data.

Keywords: Interactive machine learning · User interface · Interface metaphor

1 Introduction

Following the introduction of the desktop as a user interface for personal computers, together with the mouse as an extension of the user’s hand for interacting with the desktop, metaphors have dominated the design of user interfaces. Indeed, the use of metaphors is highlighted in early interface guidelines for the Macintosh: “You can take advantage of people’s knowledge of the world around them by using metaphors to convey concepts and features of your application” [2]. This increased the accessibility of computational systems for everyone by rendering the systems intuitive and familiar.

A user-interface metaphor provides a visual or action pattern that leverages a user’s knowledge of another domain. Metaphors provide the user a quick understanding of context and meaning of interface contents based on familiarity with another, typically physical, domain. For example, files, folders, tabs,

stick-on notes are common user-interface metaphors based on a user’s knowledge of office environments. Roots, trunks, branches, and leaves are metaphors for structural or hierarchical organization based on a user’s knowledge of trees. These metaphors not only provide the user an understanding of concepts and structural relationships, but also permissible actions and potential consequences of actions. In the office environment, files, folders and tabs have implications for intuitive means of organizing information in a nested fashion, while stick-on notes carry implications about methods for annotating information.

Metaphors derive their power from the user’s experiences with the real world prior to encountering them in the computational setting. This experience can be gained by direct experience in the world (e.g., working in an office) or by indirect observation of other people’s interactions (e.g., watching a master chef in a kitchen setting). People learning through observations of others’ interactions with the world gives inspiration to how machines could learn by observing user interactions through user-interface metaphors.

Despite the ubiquity of user-interface metaphors in practical applications, much of their use has been ad-hoc, based on intuition rather than formalism. We leverage an existing user-interface metaphor taxonomy [4,5] to help formalize the notion of metaphor and its role in user interfaces. We extend the use of this interface metaphor taxonomy for interactive machine learning. Such formalisms provide us with a more nuanced view of the role of these metaphors and how machines might learn from our interactions with them.

The crux of our argument is that, if it is “things” that make people smart about how to interact with the world [18], and metaphors about those things have made complex computational machines intuitively accessible to users [3,10], then those same metaphors can be leveraged by machine learning to render the computational systems smarter about the meanings of the interactions from the people who use them.

We believe that computers can learn by observing user interactions with user-interface metaphors. Our goal in the present paper, however, is not to simply reiterate the body of literature on how to select a good metaphor. Rather, our goal is to explore how machine learning might exploit good metaphors selected for (graphical) user interfaces to advance the capabilities of the machine learning-based system. We do not wish to be prescriptive about which metaphors should be used. We hope to identify synergies between user-interface metaphors, particularly those already familiar to users, and the goals and needs of machine learning to suggest fruitful next steps for using metaphors to support interactive machine learning.

2 The User-Interface Metaphor Taxonomy

We adopt the metaphor taxonomy of Barr et al. [4] which presents an extension to the seminal work of Lakoff and Johnson [16]. This framework provides a taxonomy of metaphors and introduces several important concepts. Figure 1 depicts their taxonomy (the subset of the image with squared corners); we note

that the elements of this taxonomy are not mutually exclusive. We provide a short summary of these elements for completeness.

Orientalational metaphors organize a set of concepts in terms of a space. For example, GOOD IS UP, and BAD IS DOWN. They provide at least one dimension (e.g., goodness, time) along which we can relate concepts to one another. The fast-forward (right-pointing arrow) and rewind (left-pointing arrow) buttons on a media player are simple examples of orientational user-interface metaphors.

Ontological metaphors support understanding of system concept based on understanding of objects or entities in the physical world. A common ontological metaphor is TIME AS AN OBJECT with quantity (e.g., having enough time). In a computational system, DATA AS A FILE is a common ontological metaphor, where the file can be quantified and manipulated.

Structural metaphors characterize the structure of one concept through another concept. Where ontological metaphors state that X is an object, structural metaphors state the object that X is, which implies its structure. For example, the FOLDER AS AN OBJECT in the user interface is an ontological metaphor, and the FOLDER AS A CONTAINER for holding documents is a structural metaphor for the object [7]. Structural metaphors speak to how the user experiences the concept.

Process and **element** metaphors are types of structural metaphors introduced by Barr and colleagues [4], inspired by the work of Nielsen and Molich [17]. Process metaphors explain how something works, indicating functionality. For example, tools are process metaphors that use icons or words to indicate functionality within graphics software, such as SCISSORS to CUT CONTENT. Element metaphors are part of the user interface that indicate which process metaphors are applicable. Toolboxes containing collections of tools with common functionality are familiar element metaphors. Because of their common functionality for adding visual content, BRUSH, PENCIL, and PAINT BUCKET AS DRAWING IMPLEMENTS are collected into a PAINTING TOOLBOX.

Metonymy metaphors substitute the name or adjunct of an object for the object itself. For example, PEN FOR WORDS, as in “the pen is mightier than the sword.” Within computational systems, examples of metonymy include the MAGNIFYING GLASS FOR SEARCH and ZOOM, and a QUESTION MARK FOR HELP MENU.

Not included in the taxonomy diagram, but referred to in practice, there is also a concept of a conventional metaphor [6]. A conventional metaphor is one with which users are already familiar, and so it continues to be used. Common examples include BUSY PROCESSOR AS SPINNING ICON (replacing the cursor) and DATA AS DOCUMENT.

Metaphors connote meanings and potential affordances to the users through a concept of metaphoric entailment. A metaphoric entailment describes what one thing (the signifier) implies about another (the signified). This concept is both fundamental to Lakoff and Johnson’s work in language, and it is fundamental

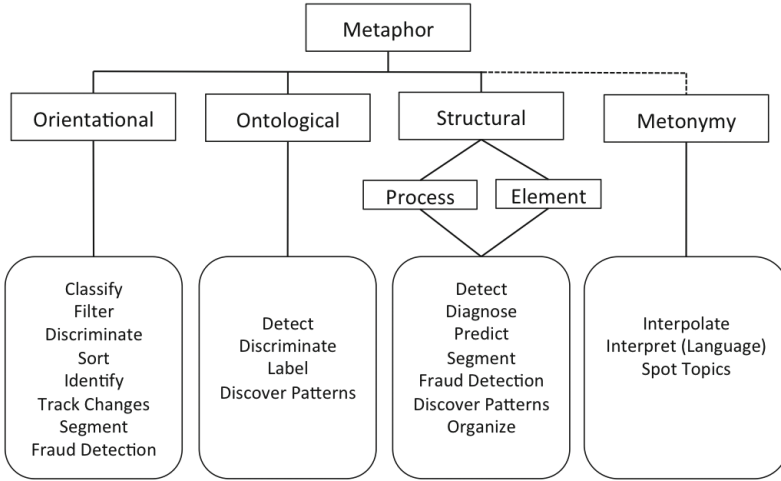


Fig. 1. User-interface metaphor taxonomy from [4] augmented with the types of machine learning activities that might benefit from leveraging each type of metaphor.

to user interface design. An example provided by [6] is USING THE DATA-STORAGE SYSTEM IS FILING. Entailments for this user-interface metaphor include:

- a. There are files in the data-storage system.
- b. There are folders in the data-storage system.
- c. Files can be placed (recursively) in folders in the data-storage system.

Entailments provide a way of transferring the user’s knowledge about the signifier onto the signified. These can be used to construct deductive arguments, hence the use of the verb entails. For example:

A FOLDER IS A FILE.
 FILES CAN BE DUPLICATED.

 Therefore: A FOLDER CAN BE DUPLICATED.

2.1 Why Are User-Interface Metaphors Important for Humans?

Interface metaphors allow users to more quickly learn and adapt to new user interfaces through reasoning by analogy. Analogical reasoning is central to cognition [12–14]. Analogies operate as a mapping between domains, providing context for finding patterns and relationships between patterns. Analogies provide a way of recreating complex patterns from personal feeling and experience. They form the foundation of mental models to support mental simulation and prediction for novel situations. In the context of data analytics, analogical reasoning

is a foundation for contextualizing cues and using them for appropriate recall and inferences over the course of the analytics process [19]. By tapping into analogical reasoning, user-interface metaphors take advantage of the extensive cognitive capabilities supported by analogy. When user-interface metaphors are successful, they seem invisible to the users, providing an intuitive and seamless user experience [10].

Interfaces that do not rely on user-interface metaphors (or worse yet, break them) make learning a new interface or system more difficult. Many user-interface metaphors have become ubiquitous because interfaces that break with them frustrate users, which limits adoption. One of the most familiar cases is the aforementioned personal computer OPERATING SYSTEM AS DESKTOP interface metaphor. Most non-specialist users (non-computer-scientist or engineer) are comfortable navigating the desktop environment, but they are not comfortable working within the command line interface. The ubiquity of the desktop and the familiarity of the metaphor have established user expectations for system interactions. The additional time and effort needed to work from the command line to accomplish the same goals are costs that many users do not want to pay. In fact, one can argue that the use of the metaphor itself becomes a kind of metaphor for other applications. Because of the desktop work environment, nearly every operating system using a graphic interface employs the TRASH CAN AS CONTAINER metaphor for unwanted files. Emptying the trash can becomes the metaphor for permanently removing files. Not using this metaphor or adopting a new metaphor would have to provide significant benefits over the existing metaphor for people to want to make the change.

2.2 Why Are User-Interface Metaphors Important for Machines?

In addition to supporting intuitive user interface design, we can conceive of metaphors as an additional rich source of data for machine learning systems. Many traditional machine learning approaches rely on pre-labeled training data with minimal direct user input into the training process. The resulting trained algorithms are non-adaptive during application; even if the user's understanding of the data has changed, the machine algorithm remains fixed. Active learning approaches have resulted in more adaptive machine learning systems that are responsive to explicit user feedback [1]. Such systems require humans to engage in the training process by providing explicit inputs, such as labels for images, to create a training data set for semi-supervised machine learning. But human-interface interactions can provide additional implicit data to the system. For the image labeling situation, for example, the speed at which labels are input, the similarity in labels between images, and the number of other activities with which the user engages concurrent to the labeling task could all provide information to the machine.

Metaphors have the potential to provide critical contextual information and constraints on user interactions that can be leveraged to guide machine learning. Paralleling the human use of interface metaphors, they might provide a mechanism for a process akin to analogical reasoning for the machine learner. Past

interaction with data via metaphors, for example, offers a template for how to handle new data [20]. Consider a case where a user places all email from advertisers into the trash can, using the TRASH CAN AS CONTAINER metaphor. Explicitly, the user has input to the system a specific set of data that need to go into a cluster with all other objects in that container. Implicitly, the user is indicating a set of data that s/he wishes to eliminate or not even receive in the inbox. Knowledge of this intention can be derived from the DELETE FILES AS EMPTY TRASH CAN metaphor. If the machine learner could access this metaphor, it could derive implicitly the user's intention from the explicit actions. From this, perhaps the system could learn to predictively place new messages from advertisers into the trash can as well. Thus, the user input provides direct feedback to the machine learner, and the metaphor provides the context critical for interpreting and generalizing the user's actions. This approach has obvious benefits (off-loading the effort from the user) as well as risks (the machine places something important in the trash). We anticipate that transparency into how the system is handling new data will be important for avoiding overgeneralizations from user interaction with metaphors. User feedback or guidance to the system to correct machine errors can aid in avoiding the overgeneralization as well.

As described above, user-interface metaphors provide both context for interpretation as well as constraints on the possible user interactions afforded by the metaphor. Continuing our trash can metaphor, one would naturally fill and empty the trash, but one does not usually organize the trash, as one might with a file system metaphor. Machines capable of learning metaphors could leverage the affordance constraints to make predictions about future interactions, with the interpretation grounded in the metaphor-provided context. Metaphor entailments would further augment the machine's ability to interpret and generalize user behaviors. The user-interface taxonomy in Fig. 1 offers a way to consider the classes of possible metaphors in a way that describes the context and constraints within each class, as we have defined above. Certain contexts and their associated interactions have variable amounts of usefulness for different machine learning tasks. We augment the user-interface taxonomy with sets of machine learning tasks for which each class should have a high degree of usefulness. Our hypothesized associations are shown in the rounded-corner boxes in the lower tier of Fig. 1. Specific choices of user-interface metaphors, however, should be made in the context of the desired system. We do recommend leveraging common user-interface metaphors as much as possible from existing systems. To aid a system designer in thinking through metaphor selection for interactive machine learning systems, we next discuss a set of questions that define ways in which metaphors could shape system development.

3 Metaphor Considerations

As in user interface design, choice of metaphor is critical to promoting effective user engagement and understanding of an IML system. It is also a design decision made at the discretion of the designer, though often shaped and honed by user

evaluations. Erickson [10] posed five critical questions that should be asked in the process of user-interface metaphor generation to evaluate candidate choices. These questions are also applicable to selection of metaphors for use in machine learning:

1. *How much structure does the metaphor provide?* This speaks to the usefulness of the metaphor for aiding the user in analogical reasoning and scaffolding knowledge.
2. *How much of the metaphor is actually relevant to the problem [for which the interface is designed to solve]?* The inclusion of too much irrelevant detail could be misleading or result in misuse or overgeneralization of the metaphor.
3. *Is the interface metaphor easy to represent?* Simplicity is key to adoption because the analogical reasoning will not place heavy cognitive demands on the user.
4. *Will your audience understand the metaphor?* Metaphors cannot help users if the users cannot understand the metaphor.
5. *What else do the proposed metaphors buy you?* Metaphors can be selected that provide useful structure that can be built upon later with additional metaphors or additional functionality.

IML from metaphors is one way that good choice of metaphors can be built upon for additional system functionality, going more deeply than just user interaction functionality implied by the last question. But this suggests that we should start the process of developing IML systems on the metaphors that have already proven successful in interface design, such as the desktop and toolboxes. Moving toward IML with metaphors wherein the user interactions and the content of the metaphor become additional inputs to the machine learner (in addition to the data of interest), we should pose some additional questions to inform good choices of metaphors.

6. *How consistent is the structure of the user interactions with the metaphor?* Metaphors that encourage consistent interaction patterns within and across users will provide consistent structure to the interaction-based inputs to the machine learner. This includes both the volume (number of types of interactions, frequency of interaction) and the variability of the interactions. Of particular concern in the process of learning is how many interpretations could be attributed to the same user input. For example, double clicking a mouse can mean open a file, launch an application, highlight a word, or zoom in/out, depending on the context. Context learned from the metaphor and metonymy (icons) become critical for disambiguating user behavior for machine interpretability.
7. *How many machine learning tasks can be supported with a selected metaphor or method of representation?* This speaks to an IML-specific dimension of usefulness for the metaphor. If a metaphor supports a basic machine learning task (cluster, rank) in a manner that is not strictly tied to a data type or domain, then that metaphor may be re-usable across systems that all need the same basic task. VOTING AS AGREEMENT is an example of a simple

metaphor (up/down orientational metaphor) that can be represented with straightforward interactions (select up/down button, which may be depicted as arrows, ✓ and × marks, or thumbs up and thumbs down). Depending on the context, voting can signify agreement between users or between user and machine, individual user preferences, rankings, correctness, or popularity. As we will discuss later, metaphors may be scaffolded upon each other to develop more complex systems, so the development of metaphor-based IML approaches may benefit from starting with combinations erring on the side of simplicity over complexity.

8. *Does the metaphor seamlessly integrate the machine learning into the user's natural activities on the system?* Even if a selected metaphor is intuitive and easy for the user to grasp and use, if it does not integrate naturally into their activities or workflow then it will may be deemed a burden or distraction by the user, who will avoid it altogether. Active learning suffers a pitfall of placing a cognitive burden on the user of providing explicit labels or supervision to the machine learner, which can detract from a user's primary goals or needs for using the computational system and machine learner in the first place. A promise of IML systems is the ability to extract supervision for the machine learner from natural interactions by the user within their normal workflow. Choice of metaphor may be crucial to this smooth integration.
9. *What can the machine learner learn from the metaphor?* We address this at length in the next section.

4 Learning from Metaphors

Metaphors make complex computational machines intuitively accessible to users. By analogy, those same metaphors can be leveraged by machine learning to render the computational systems smarter about the meanings of the interactions from the people who use them. We argue that metaphors provide a natural focal point for learning from the user. This learning may occur at two levels. When the metaphor is fixed and well known (e.g., trash can), we can learn about the user's goals, preferences, and needs through their interaction with the metaphor. When the metaphor is not obvious (e.g., organizing objects on a canvas), machines can learn the metaphor by looking at how the user interacts with user interface elements. For example, if user is organizing objects horizontally based on chronology, the machine might infer the user is applying a TIME AS LINE metaphor. Machines can learn these metaphors much in the same way we learn these metaphors, by observing the actions of others.

Each interaction with the metaphor provides an opportunity to gather data and learn. Placing a file in the trash suggests the file has little future utility for the user. Organization of elements on a canvas may suggest how those objects relate to one another. These insights come from how the user interacts with the particular metaphor. The trash can provides a structural metaphor; spacial grouping on a canvas suggests an orientational metaphor. We focus herein on learning from interactions, specifically interaction with metaphors. Interactions

provide the clearest insight into user intentions, preferences, goals, and needs. Metaphors provide structure that may be absent from non-metaphorical user interface elements.

User-interface metaphors are typically more constrained due to inherent limitations of the physical objects and processes they represent. This provides a level of consistency and regularity that makes learning from them easier than from a metaphor-free user interfaces, which is unconstrained. Furthermore, the metaphorical entailments by definition follow a particular form and can be reasoned about.

Organization of objects into folders and subfolders or grouping objects on a desktop provides clues to the relationship between those objects. If like items are clustered, the machine can learn to leverage the SIMILARITY AS PROXIMITY metaphor. Changing the sort order of a list or reordering individual list items provides clues to our preferences. The machine can learn the VALUE AS POSITION metaphor. Each user interaction with a metaphor is a potential clue and opportunity for machine learning to better support the user. When the metaphor is unknown, we want to learn these metaphors from user interaction. This approach provides the opportunity to learn new metaphors that maybe unknown to the designer of the system.

Fails and Olsen [11] presented an approach to constructing a perceptual user interface (PUI) using an IML model. IML departs from the standard machine learning (SML) model in which models are built offline then used interactively. IML creates a loop in which the user supports training of the classifier, which is built incrementally and interactively. Done properly, user interactions in IML provide both benefit to the user and feedback to the underlying machine learning system. Crayons is a system described in [11] that uses IML to create image classifiers. Crayons leverages the ITEM TAGGING AS PAINTING and USER FEEDBACK AS TUTOR metaphors. Crayons users can refine the image classifiers by iteratively adding more tags until satisfied with the machine learning performance.

Machine learning can be used to learn metaphors and leverage those metaphors to better support users. Orientational and structural metaphors provide the greatest opportunity to leverage machine learning. The following two sections provide more detail into how we can learn from interaction with these classes of metaphors.

4.1 Learning from Orientational Metaphors

Orientational metaphors provide meaning to objects in terms of a space. Examples include, GOOD IS UP, BAD IS DOWN, PAST TIME IS LEFT, FUTURE TIME IS RIGHT, HOT IS ABOVE COLD. Orientational metaphors extend to and are embedded in everyday objects, symbols and speech. Examples from speech include “she was at the top of her class” or “left of boom” referring to time prior a horrible event. Most typically, organizational metaphors provide meaning to a collection of objects and therefore describe how they relate to one

another along an important dimension. For example, using the common orientational metaphor *TIME AS LINE*, users can depict the temporal ordering of events by organizing them along a horizontal line from left (earliest) to right (latest). We can depict that “*A* occurred before *B*” and “*B* occurred before *C*” by placing these symbols horizontally organized from left to right. This metaphor implies a number of entailments such as: *A*, *B*, and *C* are different events; *A*, *B*, and *C* didn’t occur at the same time; and *A* occurred before *C*.

When the metaphor is unknown, we would like the machine to learn from user interaction with orientational interface metaphors while avoiding the hard-coding or pre-programming of specifics into a system (e.g., good := up, bad := down).

As previously discussed, learning can occur at two levels. First, we discuss how we could learn the metaphor itself from user interaction.

While we could pre-program a particular orientational metaphor into a system, such a system would always be brittle. Suppose the user is given a 2-d canvas in which to organize objects needed to perform a task, and s/he is employing an organizational metaphor. Given sufficient access to the underlying structure and attributes of objects, a system could learn the metaphor being employed by the user. For example, one user might be organizing hotter objects on the right and colder objects on the left. A second user could similarly be organizing hotter objects on the top and the colder objects on the bottom of the canvas. The machine should learn the metaphor *TEMPERATURE AS LINE* regardless of the orientation on the screen.

Given sufficient examples, the system could review the attributes of each object and determine the attribute that provides an ordering consistent with the user’s layout. This could be employed in both dimensions across all the attributes. Of course, it is possible that there are multiple (or no attributes) that result in a consistent ordering. Multiple attributes providing a consistent ordering suggests some level of ambiguity on the part of the learner. Finding no consistent order may suggest that the user is not using an orientational metaphor, or they are organizing by an attribute not available to the learner.

Regardless, there exists opportunities for the machine learning to make plausible inferences regarding the use of orientational metaphors by the user. Similar techniques could be used to derive metaphorical entailments the user has made based on the organization of objects.

The second level entails learning the user’s preferences, goals, and concepts. Given the system understands the metaphor, the system could learn the user’s preferences based on interactions with those metaphors. User interaction with objects organized in a space provides clues into how the objects are related based on the metaphor. For example, the user places important items above less important items in a list. Second level learning would have to determine what makes items important, which could be obtained by examining the items. Having learned which items are important and not important, the system could recommend where to place incoming items based on their importance.

Such an intelligent system could warn the user when they are using the metaphor inconsistently. Widely used metaphors across users could be suggested to new users of the system. Entailments that have found to be useful could be leveraged with new users.

4.2 Learning from Structural Metaphors

Structural metaphors reveal the structure of one object (signified) through reference of another object (signifier). They are more powerful than orientational or ontological metaphors as they often leverage more of our personal experiences. ARGUMENT IS WAR is a classic example, where ARGUMENT is the signified and WAR is the signifier. People “attack” and “defend” themselves in argument. There are “winners” and “losers” or someone might “come to my defense”. Such metaphors are powerful in that they can aid the user in more quickly discovering how a system works through the analogy.

iTunes uses many orientational and structural metaphors, including the album/song metaphor and stop, reverse, forward, and play button metaphors. In fact, these metaphors arise from multiple sources. The album/song metaphor comes from vinyl records. The control button metaphors come from the cassette recorder. These metaphors instantly clue the user into what operations are valid and what consequences the associate actions have on the iTunes system. They bring forth a number of entailments.

- You can organize songs into albums.
- You can play, reverse, or forward a song.
- You can play an album.
- Playing an album starts with the first song.

They also indicate which actions you cannot perform.

- You can not put in album in an album.
- You can play an album, but not reverse an album.

Because the metaphors are, by design, abstracted away from the unchangeable properties of the physical objects, the computational system can combine multiple metaphors to introduce new functionality. For example, iTunes songs are not hardcoded on media in a fixed order. The system can take advantage of the LIST AS DECK OF CARDS metaphor, providing a new shuffle entailment, which randomly reorders the song list.

Similarly, learning from structural metaphors can occur at two levels. When the metaphor is fixed and well known, we’re interested in learning the user’s needs through their use of the metaphor. When the metaphor is not known, the system must first learn the structure being implied by the metaphor. A learning system could learn the types of relationship and hierarchies that are possible based on user interaction. Again, we would like to avoid hard coding learning systems.

An IML system, having learned the iTunes AS ALBUM PLAYER metaphor, could further leverage the metaphor and related user interactions to support a system for DATA STREAM AS MEDIA PLAYER metaphor. Samples of data could be treated as songs. Activation icons can be re-used for the interface. The IML system takes advantage of the metaphor entailments:

- User can organize samples into data stream albums (related groupings).
- You can play, reverse, or forward a data stream sample.
- You can play an album of data streams.
- Playing a data stream album starts with the first sample.

The efforts of the analyst, then, can be re-focused on more challenging problems of stream fusion or out-of-order samples. Further, because of the ability of the IML approach to learn metaphors, ongoing interactions by the user on the streaming player system could evolve additional metaphors. The IML system can also learn which metaphor elements are not useful in the new setting (e.g., track shuffle would render the stream out of temporal order and may not be useful for stream interpretation). The adoption of existing metaphors serves to facilitate the learning and system development process.

5 Limitations and Implications

We believe the most promising applications for learning from metaphors will be through interactive machine learning (IML). General purpose learning-based agent support faces a number of challenges identified by Horvitz [15]. Such a learning system may make poor guesses about the user's goals and intents, or the costs and benefits of taking action to support the user. These limitations stem from a number of underlying root causes.

- Data is limited.
- The number of user interactions may not be sufficient for the system to generalize about the user's goals and intent.
- The user's goals may not be static but may change over time.
- The underlying data may also be shifting over time.
- The underlying object may not reveal enough information for a learning system.
- The user may be making decisions based on background knowledge or insights unavailable to the machine learner.

While Horvitz [15] proposed mixed-initiative systems to address these limitations, our goals are more modest. IML systems focus on solving a more limited set of problems that center-around machine learning. These systems, by definition, focus on learning from user interaction on a continuous basis. This seems like the natural place to leverage metaphors for the purpose of learning.

We share a vision for machine learning, packaging algorithms into small, discrete components. Designers and developers will then build systems using pre-built learning components. This is a departure from traditional systems which

rely on a centralized learning component. Ideally, we would like to support specific tasks (e.g., filter, sort, organize) through a collection of suitable component-level interface metaphors. Each metaphor would have its own learning algorithm, learning from interactions with that component. Machine learning will need to understand context (e.g., user, time, environment) to be effective.

Learning could occur at multiple levels in a hierarchical fashion. General purpose learning could be used to identify orientational or structural metaphors. Higher level learning could be used to determine orientational axis or structure of a metaphor. Other learning algorithms could focus on individual preferences, goals, and priorities of the user through interaction. Such an approach would be much more flexible than hard-coded single learner systems.

Interactive machine learning from user-interface metaphors is especially appealing in streaming data environments. Relative to static or batch analytics environments, streaming data is characterized by increased velocity and volatility. That is, data captured from an inherently dynamic and streaming world can result in a user environment that is shifting, with changing context and constraints. Leveraging user interfaces for analytics that learn through metaphors supports adaptation of the machine learner to the changing context and constraints without the need for explicit user input. This enables an analyst to be continually supported by the machine analytics and focus mental efforts on the data interpretations, rather than supervising the machine learning. Recent work in visual analytics has demonstrated the utility of leveraging interface interactions to learn functions of the data and make visualization recommendations [8,9,20]. User-interface metaphors smoothly integrated into interactive machine learning could be the key to extending such learning to streaming analytics environments.

Acknowledgments. The research described in this document was sponsored the U.S. Department of Energy (DOE) through the Analysis in Motion Initiative at Pacific Northwest National Laboratory. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

1. Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T.: Power to the people: the role of humans in interactive machine learning. *AI Mag.* **35**(4), 105–120 (2014)
2. Apple Computer Inc: Macintosh Human Interface Guidelines. Addison-Wesley, Boston (1992)
3. Barr, P.: User-interface metaphors in theory and practice. Master's thesis, University of Wellington, Victoria (2003)
4. Barr, P., Biddle, R., Noble, J.: A taxonomy of user-interface metaphors. In: Proceedings of the SIGCHI-NZ Symposium on Computer-Human Interaction, pp. 25–30. ACM (2002)
5. Barr, P., Biddle, R., Noble, J.: A semiotic model of user-interface metaphor. In: Liu, K. (ed.) *Virtual, Distributed and Flexible Organisations*, pp. 189–215. Springer, Dordrecht (2004)

6. Barr, P., Khaled, R., Noble, J., Biddle, R.: A taxonomic analysis of user-interface metaphors in the microsoft office project gallery. In: Proceedings of the Sixth Australasian Conference on User interface, vol. 40, pp. 109–117. Australian Computer Society, Inc. (2005)
7. Brockerhoff, R.: User interface metaphors. In: MacHack Conference proceeding (2000)
8. Brown, E.T.: Learning from users' interactions with visual analytics systems. Ph.D. thesis, Tufts University (2015)
9. Brown, E.T., Ottley, A., Zhao, H., Lin, Q., Souvenir, R., Endert, A., Chang, R.: Finding waldo: learning about users from their interactions. *IEEE Trans. Vis. Comput. Graph.* **20**(12), 1663–1672 (2014)
10. Erickson, T.D.: Working with interface metaphors. In: Baecker, R.M. (ed.) *Readings in Human-Computer Interaction: Toward the Year 2000*, vol. 11, pp. 147–151 (1995)
11. Fails, J.A., Olsen, D.R.: Interactive machine learning. In: Proceedings of the 8th International Conference on Intelligent User Interfaces, pp. 39–45. ACM (2003)
12. Gentner, D.: Structure-mapping: a theoretical framework for analogy. *Cogn. Sci.* **7**(2), 155–170 (1983)
13. Gentner, D., Holyoak, K.J.: Reasoning and learning by analogy: introduction. *Am. Psychol.* **52**(1), 32–34 (1997)
14. Gentner, D., Holyoak, K.J., Kokinov, B.N.: *The Analogical Mind: Perspectives from Cognitive Science*. MIT Press, Cambridge (2001)
15. Horvitz, E.: Principles of mixed-initiative user interfaces. In: CHI (1999)
16. Lakoff, G., Johnson, M.: *Metaphors We Live By*. University of Chicago Press, Chicago (1980)
17. Nielsen, J., Molich, R.: Heuristic evaluation of user interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (1990)
18. Norman, D.A.: *Things that Make us Smart: Defending Human Attributes in the Age of the Machine*. Basic Books, New York (1993)
19. Patterson, R.E., Blaha, L.M., Grinstein, G.G., Liggett, K.K., Kaveney, D.E., Sheldon, K.C., Havig, P.R., Moore, J.A.: A human cognition framework for information visualization. *Comput. Graph.* **42**, 42–58 (2014)
20. Saket, B., Kim, H., Brown, E.T., Endert, A.: Visualization by demonstration: an interaction paradigm for visual data exploration. *IEEE Trans. Visual Comput. Graph.* **23**(1), 331–340 (2017)