hhline **Article type:** Opinion Article

WILEY Interdisciplinary Reviews
DATA MINING AND KNOWLEDGE DISCOVERY

Data Mining and Machine Learning in Computational Creativity

Hannu Toivonen

Department of Computer Science and HIIT, University of Helsinki, Finland

Oskar Gross

Department of Computer Science and HIIT, University of Helsinki, Finland

Abstract

Creative machines are an old idea, but only recently has computational creativity established itself as a research field with its own identity and research agenda. The goal of computational creativity research is to model, simulate or enhance creativity using computational methods. Data mining and machine learning can be used in a number of ways to help computers learn how to be creative, such as learning to generate new artefacts or to evaluate various qualities of newly generated arfefacts. In this review paper we give an overview of research in computational creativity with a focus on the roles that data mining and machine learning have had and could have in creative systems.

Introduction

While artificial intelligence has experienced remarkable advances in the last decades and has reached maturity as a research field, its sibling, *computational creativity*, is in earlier phases of its development. Computational creativity can be characterized in a manner parallel to artificial intelligence: Where artificial intelligence studies how to perform tasks which would be deemed intelligent if performed by a human, computational creativity studies performances which would be deemed creative if performed by a human.

As a research field, the goal of computational creativity is to model or simulate creativity, or to enhance human creativity using computational methods. Such tasks could be in musical creativity, verbal creativity, visual creativity, creative problem solving, or some other area requiring creative skills.

A major question in computational creativity is if and when computers can be credited with originality. Already Ada Lovelace, the "first programmer", is quoted to believe that machines can create but not really originate anything. Obviously, a purely pre-programmed generative system can be criticized for only doing what it was told to do and thus for having little if any creativity. Some form of adaptivity or self-determinism seems necessary to attribute any creative autonomy or originality to a creative system[1;2]. This is where methods from data mining and machine learning can help.

Data mining and machine learning are here broadly understood as methods that analyze data and make useful discoveries or inferences from them. Such methods can be used in creative systems, e.g., to learn how to recognize desirable qualities in produced artefacts, or even to produce artefacts, helping these systems produce novel and valueable results. There are obvious risks, however. For instance, automatic analysis of Western pop music[3] could reveal patterns that can be used to generate fairly good imitations of the given music. Imitation, however, is not really an original or creative act, and it is not the goal of computational creativity research. We will return to this in the next sections.

The goal of this paper is to introduce the research field of computational creativity to data miners and machine learners. We present computational creativity research from this viewpoint and structure the review by the different roles and opportunities that data mining and machine learning have had and could have in creative systems.

We start this review with a characterisation of the field of computational creativity.

Computational Creativity

Mechanical creativity is an old idea. For instance, methods for mechanistic composition of music can be traced back at least to Guido d'Arezzo in year 1026[4]. Later, Mozart introduced Musikalisches Würfelspiel, a dice game to compose waltzes. Artificial intelligence research has sporadically addressed the generation of creative artefacts such as poems, stories, paintings, and music during the last six decades. Computational creativity aims to go beyond mere mechanical creativity, and data mining and machine learning can play a major role here.

Despite this long history, computational creativity has only recently emerged as an independent, internationally recognized research field with its own identity and research agenda. The International Conference on Computational Creativity has been held annually since 2010[5;6;7;8;9;10], preceded by a series of workshops.

Reasons to study computational creativity are numerous. There are deep philosophical and theoretical questions about the models and mechanisms of creativity, a desire to better understand human creativity, as well as practical applications and needs. For reasons of brevity, we will here focus on computational models and methods for creative systems and applications.

Practical applications of computational creativity can be roughly categorized into two classes. The first class makes use of fully automatic creativity, e.g., in computer games to develop plots or to compose music [11] on the fly, or as a part of human-machine communication in dialogue systems and conversational agents [12]. Applications in the other class use creative techniques as components to support or enable human creativity, e.g., in music or advertising. Some of the most interesting applications will improve the usability and usefulness of various technological appliances and will support human creativity in different tasks, by creating "in new, unforeseen modalities that would be difficult or impossible for people" [2].

The different motivations and goals of computational creativity research are reflected in the definition by Colton & Wiggins [2], according to which computational creativity is *the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative*. Here, "responsibilities" emphasizes the creative intent of the system, and "unbiased observers" have no prejudice against the notion of creative machines.

The definition of computational creativity above essentially delegates the definition of what creativity is to external observers. Numerous definitions of creativity address this question. For instance, according to Boden [13], *creativity is the ability to come up with ideas or artefacts that are new, surprising, and valuable.* Unfortunately, novelty, surprise, and value are not only difficult to define objectively, they also overlap. For instance, an idea rarely is surprising or valuable if it is not novel.

Data miners will notice an interesting analogue to classic definitions of data mining as discovery of novel and useful information [14]. Does it imply that data mining algorithms are creative? According to Boden [13], possibly. On the other hand, according to Colton & Wiggins [2], a data mining algorithm would not be computationally creative unless external observers would think it is creative, which usually is not the case. This does not imply that data mining would have no role in creative systems, quite the contrary. Creative systems often consist of non-creative components, such as data mining, that together constitute a creative whole. This raises the question of how humans judge the creativity of any machine or method when they know exactly how it works? Instead of asking whether computers can be considered creative or not, it is more fruitful to identify different levels or ways of being creative.

Boden [13] identifies three increasingly complex types of creativity: (1) combinational creativity produces novel combinations of familiar ideas, (2) exploratory creativity is based on search in some space of concepts, and (3) transformational creativity involves a modification of the search space so new kinds of ideas can be generated. Data mining as a potentially creative activity usually is in the second category, looking for patterns, rules, or models of a fixed type in the given data, but creative systems can use data mining and machine learning for various tasks also in combinational and transformative creativity, as we will see in this review.

As a terminological remark, we mostly use the term *artefact* to refer to a product of a creative process, be it an idea, a concept, or a concrete result such as a poem or an image. In many applications of data mining and machine learning to computational creativity, existing artefacts are used as data, e.g., as training instances or examples.

In the review that follows, we focus on work that their authors characterize as computational creativity research. This implies that we exclude a large body of literature on computational generation of music, images, natural language, etc. that is not focused on creativity. Further, our focus is on the use of data mining and machine learning in creative methods and systems, and other aspects of these creative systems are largely ignored.

Learning to Be Creative

Conceptually, machine learning can be easily applied as the *test* component of a creative system that works in a generate-and-test manner. In such settings, one trains a machine learning system with examples of artefacts from the domain of interest. For instance, given a set of songs by The Beatles, a classifier could be trained to recognize if a given new composition is in the style of The Beatles or not. A generic music generator could then be used to produce novel songs, and those that pass the classifier as Beatles-like would be accepted.

In the following, we try to make the different roles of data mining and machine learning more clear by a simple formalization of the generate-and-test setting (even though many actual systems are not structured like that). Let $gen()$ denote the generative function of the system, and let a be an artefact generated by the function. Artefact a is then evaluated by another function $eval(a)$ and, if it passes the evaluation, is output by the system. The evaluation function can sometimes be broken down into components that measure different aspects of creativity, such as novelty, surprise, or value. For suggestions on how to evaluate the creativity of a computer program, we refer to Ritchie[15].

We will start our review by looking at how machine learning can be used to recognize or evaluate new artefacts, i.e., how an evaluation function $eval(a)$ can be learned. While this is not yet a creative act in itself, evaluation is a central part of creative systems and one of the possible creative "responsibilities" in the definition by Colton & Wiggins[2]. The rest of this section will then look at how machine learning and data mining can also be used for constructing the generation function $gen()$.

Learning to Evaluate using Classification or Regression

How do we produce images which look like a face? Or which express happiness? One approach is to work in clearly separate generate and test phases as already outlined above, where the generation phase is generic and the test phase checks the facelikeness or happiness. A popular choice then is to use genetic algorithms for generating candidate artefacts to be evaluated by classification or regression, as will be discussed below. In these cases, it is the fitness function $eval(a)$ that is learned, while a genetic algorithm aims to implement the generation function $gen()$ effectively by adapting to search for high quality artefacts.

An illustrative example of a framework where images are produced by an evolutionary algorithm is by Correia et al. [16]. The fitness of images is evaluated by a classifier trained to recognize if certain types of objects are present in the image. The evolutionary algorithm then tries to optimize the quality of produced images with respect to this evaluation (fitness) function.

Another example of learning to evaluate visual artefacts is the machine- learning based image generator DARCI [17;18;19]. The goal of DARCI is to generate images which can be described by a certain adjective (e.g. *DARCI, draw me a happy picture!*). Training of DARCI takes place via a website (http://darci.cs.byu.edu/) where visitors can associate images with adjectives, and also give feedback to DARCI regarding its success in producing an image that expresses the target adjective. In DARCI, the learning algorithm for the evaluation function $eval(a)$ is based on artificial neural networks.

The use of regression to evaluate creative artefacts is illustrated by applications in quite a different field, cooking recipes. PIERRE [20], a computational stew recipe generator, first extracts different possible ingredients of stews, soups and chilis from recipe web sites. PIERRE then learns multiple multi-layer perceptrons on different levels of abstraction to model the relation between different recipes (essentially weighted combinations of ingredients) and the ratings given to the recipes on web sites. Then a genetic algorithm is used to generate recipes, evaluated using the learned perceptrons. A more elaborate cooking recipe generation system has been developed at IBM, considering the cultural context, physiochemical properties of flavour compounds, and even the name of the dish as components that influence perception of flavours [21]. The system uses many different machine learning techniques to extract information and to evaluate the quality of the produced recipes.

Let us at this point make some general remarks about learning to evaluate creative artefacts. Despite the conceptual simplicity of the generate and test setting there are major bottlenecks in implementing this (or any other) model of creativity.

First, learning an evaluation (or fitness) function $eval(a)$ can be a very difficult problem. For instance for poetry we would need to take into account the semantics, metaphors, grammar (including the intentional twisting of rules in poems), rhythm *et cetera*. One way of making this problem easier is to only measure some specific aspect of the artefact, as was done in some of the systems above.

Second, generating complex artefacts (i.e., writing or learning the function $gen()$) is a difficult task as well. For a generate-and-test setting to be practical, the generation step must adapt to the task at hand, or it could go on forever generating artefacts that do not satisfy the evaluation function. The next subsection will describe learning methods for generation of artefacts.

Third, when learning what is good and what is bad from existing examples, there obviously is a great risk of just producing *pastiche* (i.e. imitation of the style of the training data) instead of being truly creative. In defence of this approach, it should be obvious that new and creative artefacts cannot be produced in a domain or genre without first having some understanding of the domain and, in a very concrete sense, without being able to produce artefacts that are recognized as members of the intended genre in the first place. How to break away from the limitations of conventions will be addressed in a later section of this review.

Learning to Generate Using Predictive Models

We now move on to review methods where machine learning and data mining are used more directly to generate new artefacts, i.e., to construct a generative function $gen()$.

Predictive methods such as classification and regression can be used to complete partial artefacts based on training examples of complete artefacts. Such methods are often used in music, e.g., to produce harmonies for melodies or *vice versa*. We will next review such methods.

Smith et al.[22] propose a k-nearest neighbours method for harmonization of music, i.e., for producing a suitable chord to match other properties of the music. (Their method actually generates music by using non-musical audio as the inspirational data to be harmonized.) The harmonization is produced using k-nearest neighbours in a collection of MIDI files containing different music styles. The features used for calculating the voice model are the extracted melody notes from non-musical audio. To produce a musical result, the learned voice model is added to the harmonization by using a *snap-to-grid approach*, i.e., by shifting notes in time and pitch so that the results sound musically more pleasing. In this case, the grids are decided at the system design time, not learned from data.

Similarly, the system MATT[23] uses a combination of case-based reasoning and wave table synthesis to simulate the creative interpretation of traditional Irish tunes on the wooden flute. The MATT system uses a learning method which creates the cases from corpora. The cases are a set of vectors containing features (e.g., notes, duration, key, tune part et cetera) for each tune.

To formalize the use of predictive methods to complete partial artefacts, we need to extend our simple formalization a bit. Let $\hat{a}$ be a partial artefact (e.g., a melody) and let a_i be a feature missing from the partial artefact. If the missing feature has a simple structure, such as a note in the bassline at a given point in time, then a predictive model can potentially be trained to generate it, i.e., to learn a function $gen_i(\hat{a}) \mapsto a_i$.

For the generation of more complex structures, such as an accompaniment, it is sometimes possible to reduce the generative task to selection and to train a classifier to perform the selection. For instance, the harmonization method[22] described above can be seen as a classifier that chooses an appropriate chord among a limited number of alternatives. Once the chord is selected, the concrete accompaniment can be generated using predetermined or mined patterns.

In some cases, it is possible to generate ("predict") complex structures directly with simple methods. An easy way to do this is to (partially) copy structures and content from existing examples, i.e., to use instance-based approaches to learning. Instance-based methods such as k-nearest neighbours and case-based reasoning (cf. work mentioned above) avoid using models, decision structures, or patterns whose specification and application to producing complex creative artefacts could be overwhelming and restrictive.

An example of instance-based methods in computational creativity is by Toivanen et al.[24;25;26], whose poetry generation system obtains the syntactical structure for the poem to be generated by copying the structure from a random poem in a large corpus. This avoids specification of generative grammars and limitations of manually constructed templates. The actual content of the poem is then generated with the help of pattern mining methods (see below). There are other approaches which have similar mechanisms for generating poetry[27;28;29].

Learning to Generate Using Generative Models

An interesting method for generating artefacts is to first learn a generative model from data and then use this model to generate new artefacts, i.e., as function $gen()$. Generation using a learned model directly aims to produce artefacts that are good in some respects, taking away some of the need for a separate evaluation function $eval()$. Often, however, these generative models are used together with separate evaluation functions to check those components of creativity that are not covered by the generative functions so well.

Markov models, especially Markov chains, are a popular approach for generative modelling of sequential artefacts such as music and text. For instance, Markov models have been used to learn the task of four-part harmonization[30]. system and it produces a harmonization for it[30]. There are many other methods which generate music based on Markov models[31;32;33].

Some researchers use Markov models for more specific tasks. For instance, Monteith et al.[34] describe a learning, creative system which generates music to induce a given target emotion. In this case, the training corpora consisted of movie soundtrack MIDI files. The MIDI files are first grouped into different emotion bins (i.e. love, joy, surprise, anger, sadness and fear) by ratings from six subjects. Then for each emotion, the respective melodies are transposed into the same key and an n-gram model (i.e., a Markov chain of order $n-1$) is extracted from the resulting dataset. For generating original melodies targeting a certain emotion, the authors do not use the resulting Markov model directly. Instead, the n-grams are analysed and eventually hidden Markov models and entropy-based models are used for the generation task.

Chuan et al.[35] propose a method for automatic accompaniment for amateur music writers. The authors use three songs by Radiohead for training their model (and used one song for evaluation). The chord progressions are first modeled as a tree, constructed by music-theoretical neo-Riemannian transforms between checkpoints. Chord progressions for user-given melodies are then generated from the tree with a Markov chain.

Thaler has proposed a method for creativity by using neural networks[36]. The idea is that first the neural network is trained by using know examples. Then the input is held constant and the weights between the neurons are slightly modified. The effect is, that the neural network starts to give new and unseen, yet plausible outputs.

Togelius and Schmidhuber propose a neural-network based agent architecture method for automatic game design, which defines general rules for a board game and uses artificial players to measure the complexity of the game[37].

As our last example of generative models in music, Bickerman et al.[38] propose a method for Jazz improvisation based on deep belief nets, a form of probabilistic neural network based on restricted Boltzmann machines. 4-bar jazz *licks* are used for training the belief nets and then the trained nets are used for generating novel improvisations.

Mining Patterns for Creative Tasks

We now move on to consider uses of data mining and machine learning in tasks related to the generation of new artefacts but without such distinctive roles as in the previous subsections. Here, data mining is typically used to discover patterns in a given domain such as text or music, and these patterns are then utilized by a generation function $gen()$.

Veale and others have extensively defined and studied linguistic patterns for creative inference. For instance, similes ("strong as a bull") can be extracted as patterns of the form *T is as P as a|an V* from large masses of text. If *P* ("strong") and *V* ("bull") co-occur often then *P* likely is a typical property of *V* ("bulls are strong"). These and other patterns mined from text documents can then be used in creative tasks such as metaphor generation ("he is a bull")[39;40;41;42] or conceptual blending[43]. Such creative functions can be used, for instance, in the generation of poetry[44;45].

Gastronaut[46] is a (cooking) recipe generator which uses a combinational approach. It learns relations between different options (different cooking methods, different ingredients, and different types of dish) from documents in the Web by analysing their frequencies. The relations are then further exploited to generate new combinations where all the ingredients match pairwise but the overall combination of the ingredients cannot be found on the Web. The trained model is used for validating the possible usefulness of generated recipes.

Hong et al.[47] proposed T-PEG, Template Based Pun Extractor and Generator, to analyze puns and generate new ones. T-PEG takes a pun as an input and analyses the word relations in the pun. The extracted template specifies the place for variables and also how the variables have to be related to each other (e.g., X and Y have to be conceptually related, or X is used for Y). Then, given a keyword the system generates puns by looking for words which satisfy the word relations defined in the template.

Simple word associations extracted from co-occurrences in large corpora have been used to choose content words in computer-generated poetry by Toivanen et al.[24]. In contrast to more controlled methods of creating text, statistical associations between words can be used to choose less predictable sets of words that are still relatively likely to produce coherent content. Another possible application for word associations is proposed in the conjunction of a method for generating pictorial metaphors for advertisement[48]. In the first step of the method, concepts with high imageability in various word networks are looked for. While the search is tailored for this purpose, the word networks could be the result of mining for work associations.

Gross et al.[49] show empirically that statistical word associations from bigrams in a large corpus can be used to solve a psychometric test of creativity better than humans do on average. In the Remote Associates Test, three cue words are given at a time (e.g., *coin*, *quick*, and *spoon*) and the subject tries to identify a fourth word that is related to each of the three cue words (*silver*).

It is debatable whether the Remote Associates Test is a good test of creativity, but the results of Gross et al.[49] do suggest two observations. First, some purely conceptual tasks once considered to require human creativity can now be solved efficiently by computers. Second, automatically identifying/discovering/creating non-obvious associations between concepts, a difficult task for humans, could be a powerful way of supporting human creativity in creative tasks such as problem solving. Let us next briefly mention work that goes in the latter direction.

Discovery of non-obvious links

A non-obvious association or connection, especially one that links items from domains that usually are not related, is potentially a *bisociation*. Koestler[50] coined the term, and proposed that discovery or production of bisociations is a central form of human creativity. Graph mining and analysis are attractive choices for computational implementation of bisociation in the sense of discovering non-obvious connections, such as domain-crossing links[51].

For instance, graph mining can be applied to a network of biological concepts to identify potentially relevant but non-obvious links[52], in order to help biologists in their creative problem solving. A classic example of non-obvious linkage is between "magnesium" and "migraine". It was discovered that many articles wrote how migraine can be treated with calcium blockers, while another set of seeming unrelated articles which described how magnesium works as a calcium blocker, yet the potential of magnesium in the treatment of migraine had not been realized[53].

Another creativity supporting approach using link discovery is proposed by Juršič et al.[54]. They use literature mining to discover terms that bridge two otherwise separate scientific (sub)domains, aiming to trigger scientific creativity in experts working in either of the domains.

Transformational creativity

The methods reviewed in the earlier sections show how data mining and machine learning can be useful tools for creating and evaluating novel artefacts without being explicitly told how to do so. Automatically constructing a generation function $gen()$ or an evaluation function $eval()$ (or some significant part of them) is a major conceptual step towards systems that have creative responsibilities when compared to directly programming $gen()$ and $eval()$ to work in a certain way. At the same time, even the automatically constructed methods could be criticized for not having much creative responsibility but rather just producing and evaluating the way they have been trained.

In this section we will review models and methods that aim to go beyond this criticism by being transformationally creative, i.e., by showing greater autonomy in modifying their generation function $gen()$ or evaluation function $eval()$. We also indicate little explored areas for data mining and machine learning in computational creativity.

Different Levels of Creativity

What does it mean for a system to be autonomously creative, not just as a servant of the user or the programmer? Jennings [1] proposes three criteria for attributing creative autonomy to a system.

(1) *Autonomous Evaluation:* The system is able to evaluate new creations autonomously. It thus has its own opinion on which creations are better than others, giving it some very elementary "self-awareness" needed, e.g., to guide its productions towards better artefacts. Any system that has an evaluation function $eval()$, whether given or learned, fulfills this criterion.

(2) *Autonomous Change:* The system is able to change its evaluation function $eval()$ without explicit directions. Systems that use machine learning to adapt their evaluation function are clearly moving in this direction. However, autonomous change means more than just learning to adapt to external authorities or to new training data, so that the system can in some loose sense decide for itself what it wants to create. Autonomous change typically involves exposure to external artefacts or evaluations as triggers of change, i.e., some interaction with other agents [1;55]. Data mining or machine learning is a likely component in implementing the change [1].

(3) *Non-Randomness (Aleatoricism):* Random behavior is not creative, so the evaluation and change referred to above should not be totally random. However, there can be random elements in them.

So, somewhat paradoxically, creative autonomy appears to be strongly related to social interactions: triggers to change the evaluation criteria naturally come from outside the system, such as from feedback to its own creations, or from its observations of creations from other creative systems. Such social creativity is an area of research within computational creativity [1;55].

A formal view to creativity that helps analyse different levels of creativity and also identify further opportunities for data mining and machine learning, is given by Wiggins [56]. Consider a creative system producing artefacts in some domain, say poetry. In the model of Wiggins, in a simplified form, the system is characterized by the quadruple $(\mathcal{U}, \mathcal{R}, \mathcal{E}, \mathcal{T})$ as follows. The universe of all possible artefacts is denoted by $\mathcal{U}$, including artefacts outside the domain, e.g., all possible strings of text. (The motivation for the existence of $\mathcal{U}$ is to allow the system to operate outside its foreseen domain, as will be explained in the next subsection.) The acceptable conceptual space is defined by a set $\mathcal{R}$ of rules, e.g., telling if a string is a valid poem of the desired form. An evaluation function $\mathcal{E}$ assigns a value for a given artefact, e.g., the quality of the poem. Roughly speaking, Wiggins thus divides the evaluation function $eval()$ of the generate and test model into two separate components: one that defines the search space for artefacts ($\mathcal{R}$), and one that measures their quality ($\mathcal{E}$). However, Wiggins' model is not limited to the generate-and-test setting.

Given a universe $\mathcal{U}$, rules $\mathcal{R}$, and an evaluation function $\mathcal{E}$, the task of the creative system is to search the universe $\mathcal{U}$ for artefacts that satisfy rules $\mathcal{R}$ and score high on the quality measure $\mathcal{E}$. To carry out the search, the system has some method $\mathcal{T}$ for traversing the search space and generating new artefacts. The traversal function $\mathcal{T}$ roughly corresponds to the generation function $gen()$. In Wiggins' formulation, the traversal function $\mathcal{T}$ is informed of $\mathcal{R}$ and $\mathcal{E}$ so it can adapt to be more effective with respect to them.

In the review above, we have already seen how data mining and machine learning have been used to automatically construct an evaluation function $\mathcal{E}$ (e.g., measuring the quality of a stew recipe) or rules $\mathcal{R}$ (e.g., checking if an image represents a face or not). Markov models can be seen as constructing a traversal method $\mathcal{T}$ from data, but strongly intertwined with the rules and evaluation at the same time. This is typical of creative systems: the conceptually separate components $\mathcal{R}$, $\mathcal{E}$, and $\mathcal{T}$ of Wiggins are often not implemented in a modular fashion. Wiggins' framework is not intended to be used as an architecture for building creative systems, but rather as a conceptual tool for describing and comparing them.

Transformational Creativity Using Data Mining and Machine Learning

Higher levels of creativity, or meta-creativity, is obtained when one of the three central components in Wiggins' model[56] ($\mathcal{R}$, $\mathcal{E}$, or $\mathcal{T}$) is modified by the system itself during its runtime, in the spirit of Jennings' creative autonomy[1] and Boden's transformational creativity[13]. This happens when any of the components is changed conceptually so that the system can be creative in unforeseen ways, especially by reaching other areas of the universe $\mathcal{U}$ than originally intended. Early work that already went in this direction include AM[57] and especially Eurisko[58]. Wiggins[56] identifies the following opportunities for mining and learning on the meta-level.

A creative system faces "generative uninspiration" if it is not able to reach valuable areas (as defined by $\mathcal{R}, \mathcal{E}$). In this case the traversal method $\mathcal{T}$ should be transformed. One way of doing this could be based on using observations of valued artefacts from other creative agents as positive training examples for adapting the traversal method. Instance-based learning methods can directly take advantage of new positive examples, so do those genetic algorithms that operate directly on representations of the artefacts. It is a matter of debate if it is the traversal function $\mathcal{T}$ that is changed or just its parameters that affect where the search moves. For genetic algorithms in computational creativity it is more common to operate on programs that produce artefacts rather than on the artefacts themselves. These are conceptually deeper changes, and can be argued to more truly manipulate $\mathcal{T}$ as the artefact generation method. However, the cases of aberration below are then more difficult to take advantage of.

In the case of "aberration", the traversal function $\mathcal{T}$ reaches artefacts outside the acceptable space defined by the rules $\mathcal{R}$. Then, the search could be adapted on the fly to exclude those areas by using artefacts found outside $\mathcal{R}$ as negative training examples.

An interesting special case is "productive aberration" where the extra artefacts outside $\mathcal{R}$ are actually valued by the evaluation function $\mathcal{E}$: are these perhaps exceptionally creative artefacts as they have value (according to $\mathcal{E}$) but are not within the conventional limits (as defined by $\mathcal{R}$)? The system has two options for adapting its operations: to modify $\mathcal{R}$ to include these new positive examples and other similar ones, or to modify $\mathcal{T}$ as above to exclude them as negative examples. How to have the system make an informed choice is a big question.

While Wiggins does not consider modifications to $\mathcal{E}$, from the above discussion on creative autonomy and transformational creativity it is obvious that adapting one's likings, e.g., by receiving external feedback on one's own work or observing the works of others, is a central creativity capability or "responsibility". The importance of meta-level manipulation of $\mathcal{E}$ and $\mathcal{R}$ is well summarized in a quote attributed to Getzels and Csikszentmihalyi: *the difference between greater and lesser creativity lies not in how you solve problems, but rather in what problems you choose to solve.*

In this paper, our discussion on evaluation of creative artefacts has been on the generic level of some function ($eval()$) or pair of functions ($\mathcal{E}$, $\mathcal{R}$). However, creativity is related to a number of concepts such as novelty, surprise, value, and interest. In the past, computational creativity research has paid little attention to these areas, but this has now been changing[59;60;61;62;63;64;65]. These are central topics to which data mining and machine learning can have much to offer.

Conclusion

Data mining and machine learning have been used in a number of different roles for building creative systems. Most of the current roles fall into one of four main categories. (1) Learning an evaluation function from existing artefacts. (2) Using existing artefacts in an instance-based manner to produce new ones (cf. combinational creativity by Boden[13]). (3) Learning models from existing artefacts and then generating new artefacts that match the models (cf. exploratory creativity by Boden[13]). (4) Mining patterns in artefacts and using them in generative functions. — In contrast, transformational creativity offers exciting, little explored opportunities for data mining and machine learning to help build creative systems that go beyond mere generation and pastiche. This relates to the question of how much data mining and machine learning methods can extrapolate, i.e., to produce novel solutions (artefacts) different from the training examples. Another question is how data mining and machine learning could contribute more to the creation of structurally complex artefacts. Here, the recent interest in machine learning methods for "structured prediction"[66;67] could have interesting contributions to computational creativity.

Finally, one can also ask if and how computational creativity could be used for data mining or machine learning? There are interesting opportunities for creative mining and learning systems, but let us just briefly highlight recent attempts to use computational creativity to perceptulize data. Tulilaulu et al.[68] propose a method for subjectively experiencing one's own data instead of analysing it objectively. Their application is in sleep analysis, where sleep measurements are used to automatically compose a novel, short piece of music, allowing the user to "listen" to her sleep in the morning. In the same general direction, Johnson and Ventura[69] discover musical motifs in non-musical data, and then their system composes music inspired by the data. These examples go beyond sonification by including creative components specifically tuned to produce musical results. They also demonstrate how computational creativity and data mining can be combined in creative ways.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments that helped us improve the manuscript.

This work has been supported by the Academy of Finland (decisions 276897, CLiC, and 118653, Algodan) and by the European Commission (FET grant 611733, ConCreTe).

References

[1] Kyle E. Jennings. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines*, 20:489–501, 2010.

[2] Simon Colton and Geraint A Wiggins. Computational creativity: The final frontier? In Luc De Raedt, Christian Bessiere, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter Lucas, ed. *20th European Conference on Artificial Intelligence, ECAI*, pages 21–26, Montpellier, France, 2012.

[3] Yizhao Ni, Raul Santos-Rodriguez, Matt McVicar, and Tijl De Bie. Hit song science once again a science? In *4th International Workshop on Machine Learning and Music*, 2011.

[4] H. Kirchmeyer. On the historical constitution of a rationalistic music. *Die Reihe*, 8:11–24, 1968.

[5] Dan Ventura, Alison Pease, Rafael Pérez, Graeme Ritchie, and Tony Veale, ed. *Proceedings of the International Conference on Computational Creativity*. Department of Informatics Engineering, University of Coimbra, Lisbon, Portugal, January 2010.

[6] Dan Ventura, Pablo Gervás, D. Fox Harrell, Mary L. Maher, Alison Pease, and Geraint Wiggins, ed. *Proceedings of the Second International Conference on Computational Creativity*. México City, México, April 2011.

[7] Mary L. Maher, Kristian Hammond, Alison Pease, Rafael Pérez, Dan Ventura, and Geraint Wiggins, ed. *Proceedings of the Third International Conference on Computational Creativity*. Dublin, Ireland, May 2012.

[8] Mary L. Maher, Tony Veale, Rob Saunders, and Oliver Bown, ed. *Proceedings of the Fourth International Conference on Computational Creativity*. Sydney, Australia, June 2013.

[9] Simon Colton, Dan Ventura, and Nada Lavrac, ed. *Proceedings of the 5th International Conference on Computational Creativity*. Ljubljana, Slovenia, June 2014.

[10] Hannu Toivonen, Simon Colton, Michael Cook, and Dan Ventura, ed. *Proceedings of the 6th International Conference on Computational Creativity*. Park City, UT, USA, June–July 2015.

[11] Kenneth McAlpine, Eduardo Miranda, and Stuart Hoggar. Making music with algorithms: A case-study system. *Computer Music Journal*, 23(2):19–30, 1999.

[12] Goh Ong Sing, Kok Wai Wong, Chun Che Fung, and Arnold Depickere. Towards a more natural and intelligent interface with embodied conversation agent. In *Proceedings of the 2006 International Conference on Game Research and Development*, CyberGames '06, pages 177–183, Murdoch University, Australia, Australia, 2006. Murdoch University.

[13] Margaret A. Boden. *The Creative Mind*. Abacus, London, 1992.

[14] William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. Knowledge discovery in databases: An overview. In Gregory Piatetsky-Shapiro and William J. Frawley, ed. *Knowledge Discovery in Databases*, pages 1–27. AAAI Press, Menlo Park, CA, 1991.

[15] Graeme Ritchie. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, 17:76–99, 2007.

[16] João Correia, Penousal Machado, Juan Romero, and Adrian Carballal. Evolving figurative images using expression–based evolutionary art. In Mary L. Maher, Tony Veale, Rob Saunders, and Oliver Bown, ed. *Proceedings of the Fourth International Conference on Computational Creativity*, pages 24–31, Sydney, Australia, June 2013.

[17] David Norton, Derrall Heath, and Dan Ventura. Establishing appreciation in a creative system. In Dan Ventura, Alison Pease, Rafael Pérez, Graeme Ritchie, and Tony Veale, ed. *Proceedings of the International Conference on Computational Creativity*, pages 26–35, Lisbon, Portugal, January 2010. Department of Informatics Engineering, University of Coimbra.

[18] David Norton, Derrall Heath, and Dan Ventura. Autonomously creating quality images. In Dan Ventura, Pablo Gervás, D. Fox Harrell, Mary L. Maher, Alison Pease, and Geraint Wiggins, ed. *Proceedings of the Second International Conference on Computational Creativity*, pages 10–15, México City, México, April 2011.

[19] David Norton, Derrall Heath, and Dan Ventura. Autonomously managing competing objectives to improve the creation and curation of artifacts. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia, June 2014.

[20] Richard Morris, Scott Burton, Paul Bodily, and Dan Ventura. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In Mary L. Maher, Kristian Hammond, Alison Pease, Rafael Pérez, Dan Ventura, and Geraint Wiggins, ed. *Proceedings of the Third International Conference on Computational Creativity*, pages 119–125, Dublin, Ireland, May 2012.

[21] Lav R Varshney, Florian Pinel, Kush R Varshney, Debarun Bhattacharjya, Angela Schoergendorfer, and Yi-Min Chee. A big data approach to computational creativity. *arXiv preprint arXiv:1311.1213*, 2013.

[22] Robert Smith, Aaron Dennis, and Dan Ventura. Automatic composition from non-musical inspiration sources. In Mary L. Maher, Kristian Hammond, Alison Pease, Rafael Pérez, Dan Ventura, and Geraint Wiggins, ed. *Proceedings of the Third International Conference on Computational Creativity*, pages 160–164, Dublin, Ireland, May 2012.

[23] Bryan Duggan, Zheng Cui, and Pádraig Cunningham. MATT - a system for modelling creativity in traditional Irish flute playing. In Simon Colton and Alison Pease, ed. *The Third Joint Workshop on Computational Creativity*, 17th European Conference on Artificial Intelligence, ECAI, Riva del Garda (Trentino), Italy, August 2006. Universita di Trento.

[24] Jukka M. Toivanen, Hannu Toivonen, Alessandro Valitutti, and Oskar Gross. Corpus-based generation of content and form in poetry. In *Proceedings of the Third International Conference on Computational Creativity (ICCC)*, pages 175–179, Dublin, Ireland, May 2012.

[25] Jukka M. Toivanen, Matti Järvisalo, and Hannu Toivonen. Harnessing constraint programming for poetry composition. In *Proceedings of the Fourth International Conference on Computational Creativity (ICCC)*, pages 160–167, Sydney, Australia, June 2013.

[26] Jukka M. Toivanen, Oskar Gross, and Hannu Toivonen. The officer is taller than you, who race yourself! Using document specific word associations in poetry generation. In *5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia, June 2014.

[27] Hugo Gonçalo Oliveira. PoeTryMe: a versatile platform for poetry generation. In Kai-Uwe Kühnberger, Peter König, and Sven Walter, ed. *Proceedings of the Workshop Computational Creativity, Concept Invention, and General Intelligence*, volume 1, pages 21–26, Montpellier, France, August 2012.

[28] Hugo Gonçalo Oliveira, Raquel Hervás, Alberto Díaz, and Pablo Gervás. Adapting a generic platform for poetry generation to produce Spanish poems. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia, June 2014.

[29] Fam Rashel and Ruli Manurung. Pemuisi: a constraint satisfaction-based generator of topical Indonesian poetry. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia, June 2014.

[30] Raymond Whorley, Geraint Wiggins, and Marcus T. Pearce. Systematic evaluation and improvement of statistical models of harmony. In Amílcar Cardoso and Geraint Wiggins, ed. *Proceedings of the 4th International Joint Workshop on Computational Creativity*, pages 81–90, London, UK, September 2007. Goldsmiths, University of London.

[31] Arne Eigenfeldt and Philippe Pasquier. Considering vertical and horizontal context in corpus–based generative electronic dance music. In Mary L. Maher, Tony Veale, Rob Saunders, and Oliver Bown, ed. *Proceedings of the Fourth International Conference on Computational Creativity*, pages 72–78, Sydney, Australia, June 2013.

[32] Tom Collins, Robin Laney, Alistair Willis, and Paul Garthwaite. Using discovered, polyphonic patterns to filter computer-generated music. In Dan Ventura, Alison Pease, Rafael Pérez, Graeme Ritchie, and Tony Veale, ed. *Proceedings of the International Conference on Computational Creativity*, pages 1–10, Lisbon, Portugal, January 2010. Department of Informatics Engineering, University of Coimbra.

[33] François Pachet and Pierre Roy. Non-conformant harmonization: The real book in the style of take 6. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia, June 2014.

[34] Kristine Monteith, Tony Martinez, and Dan Ventura. Automatic generation of music for inducing emotive response. In Dan Ventura, Alison Pease, Rafael Pérez, Graeme Ritchie, and Tony Veale, ed. *Proceedings of the International Conference on Computational Creativity*, pages 140–149, Lisbon, Portugal, January 2010. Department of Informatics Engineering, University of Coimbra.

[35] Ching-Hua Chuan and Elaine Chew. A hybrid system for automatic generation of style-specific accompaniment. In Amílcar Cardoso and Geraint Wiggins, ed. *Proceedings of the 4th International Joint Workshop on Computational Creativity*, pages 57–64, London, UK, September 2007. Goldsmiths, University of London.

[36] Stephen Thaler. Neural nets that create and discover. *PC AI Intelligent Solutions for Today's Computers*, 10(3):4, 1996.

[37] J. Togelius and J. Schmidhuber. An experiment in automatic game design. In *Computational Intelligence and Games, 2008. CIG '08. IEEE Symposium On*, pages 111–118, Perth, Australia, Dec 2008.

[38] Greg Bickerman, Sam Bosley, Peter Swire, and Robert M. Keller. Learning to create jazz melodies using deep belief nets. In Dan Ventura, Alison Pease, Rafael Pérez, Graeme Ritchie, and Tony Veale, ed. *Proceedings of the International Conference on Computational Creativity*, pages 228–237, Lisbon, Portugal, January 2010. Department of Informatics Engineering, University of Coimbra.

[39] Tony Veale and Yanfen Hao. Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1471–1476. AAAI Press, 2007.

[40] Tony Veale and YF Hao. Learning to understand figurative language: From similes to metaphors to irony. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society (CogSci 2007)*, pages 683–688, Nashville,USA, August 2007.

[41] Tony Veale and Guofu Li. Creative introspection and knowledge acquisition: Learning about the world through introspective questions and exploratory metaphors. In W. Burgard and D. Roth, ed. *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 1243–1248, San Francisco, USA, August 2011.

[42] Tony Veale. Once more, with feeling! Using creative affective metaphors to express information needs. In Mary L. Maher, Tony Veale, Rob Saunders, and Oliver Bown, ed. *Proceedings of the Fourth International Conference on Computational Creativity*, pages 16–23, Sydney, Australia, June 2013.

[43] Tony Veale. From conceptual "mash-ups" to "bad-ass" blends: A robust computational model of conceptual blending. In Mary L. Maher, Kristian Hammond, Alison Pease, Rafael Pérez, Dan Ventura, and Geraint Wiggins, ed. *Proceedings of the Third International Conference on Computational Creativity*, pages 1–8, Dublin, Ireland, May 2012.

[44] Tony Veale. Less rhyme, more reason: Knowledge–based poetry generation with feeling, insight and wit. In Mary L. Maher, Tony Veale, Rob Saunders, and Oliver Bown, ed. *Proceedings of the Fourth International Conference on Computational Creativity*, pages 152–159, Sydney, Australia, June 2013.

[45] Simon Colton, Jacob Goodwin, and Tony Veale. Full-FACE poetry generation. In Mary L. Maher, Kristian Hammond, Alison Pease, Rafael Pérez, Dan Ventura, and Geraint Wiggins, ed. *Proceedings of the Third International Conference on Computational Creativity*, pages 95–102, Dublin, Ireland, May 2012.

[46] Cristina Butnariu and Tony Veale. Lexical combinatorial creativity with "Gastronaut". In Simon Colton and Alison Pease, ed. *The Third Joint Workshop on Computational Creativity*, 17th European Conference on Artificial Intelligence, ECAI, Riva del Garda (Trentino), Italy, August 2006. Universita di Trento.

[47] Bryan Anthony Hong and Ethel Ong. Automatically extracting word relationships as templates for pun generation. In Anna Feldman and Birte Lönneker-Rodman, ed. *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 24–31, Boulder, USA, June 2009. Association for Computational Linguistics.

[48] Ping Xiao and Josep Blat. Generating apt metaphor ideas for pictorial advertisements. In Mary L. Maher, Tony Veale, Rob Saunders, and Oliver Bown, ed. *Proceedings of the Fourth International Conference on Computational Creativity*, pages 8–15, Sydney, Australia, June 2013.

[49] Oskar Gross, Hannu Toivonen, Jukka M. Toivanen, and Alessandro Valitutti. Lexical creativity from word associations. In Vincent C. S. Lee and Kok-Leong Ong, ed. *Knowledge, Information and Creativity Support Systems - KICSS*, pages 35–42. IEEE, November 2012.

[50] A. Koestler. *The act of creation*. Hutchinson, London, 1964.

[51] Werner Dubitzky, Tobias Kötter, Oliver Schmidt, and Michael R. Berthold. Towards creative information exploration based on Koestler's concept of bisociation. In Michael R. Berthold, ed. *Bisociative Knowledge Discovery*, number 7250 in Lecture Notes in Artificial Intelligence (LNAI), pages 11–32. Springer Verlag, 2012.

[52] Igor Mozetič, Nada Lavrač, Vid Podpečan, Petra K. Helena Motaln, Marko Petek, Kristina Gruden, Hannu Toivonen, and Kimmo Kulovesi. Bisociative knowledge discovery for microarray data analysis. In Dan Ventura, Alison Pease, Rafael Pérez, Graeme Ritchie, and Tony Veale, ed. *Proceedings of the International Conference on Computational Creativity*, pages 190–199, Lisbon, Portugal, January 2010. Department of Informatics Engineering, University of Coimbra.

[53] Don R Swanson. Undiscovered public knowledge. *The Library Quarterly*, pages 103–118, 1986.

[54] Matjaż Juršič, Bojan Cestnik, Tanja Urbančič, and Nada Lavrač. Cross-domain literature mining: Finding bridging concepts with CrossBee. In Mary L. Maher, Kristian Hammond, Alison Pease, Rafael Pérez, Dan Ventura, and Geraint Wiggins, ed. *Proceedings of the Third International Conference on Computational Creativity*, pages 33–40, Dublin, Ireland, May 2012.

[55] Rob Saunders. Artificial creative systems and the evolution of language. In Dan Ventura, Pablo Gervás, D. Fox Harrell, Mary L. Maher, Alison Pease, and Geraint Wiggins, ed. *Proceedings of the Second International Conference on Computational Creativity*, pages 36–41, México City, México, April 2011.

[56] Geraint A Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, 19(7):449–458, 2006.

[57] Douglas Bruce Lenat. *Am: An Artificial Intelligence Approach to Discovery in Mathematics As Heuristic Search.* PhD thesis, Stanford University, Stanford, CA, USA, 1976. AAI7707115.

[58] Douglas B. Lenat. Eurisko: A program that learns new heuristics and domain concepts. *Artificial Intelligence*, 21(1–2):61–98, March 1983.

[59] Rob Saunders and John S Gero. Designing for interest and novelty. In *Computer Aided Architectural Design Futures 2001*, pages 725–738. Springer, 2001.

[60] R Saunders and John S Gero. A curious design agent: A computational model of novelty-seeking behaviour in design. In *Proceedings of the Sixth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA 2001*, volume 1, pages 345–350. The University of Sydney, 2001.

[61] Mary Lou Maher, Katherine Brady, and Douglas H Fisher. Computational models of surprise in evaluating creative design. In Mary L. Maher, Tony Veale, Rob Saunders, and Oliver Bown, ed. *Proceedings of the Fourth International Conference on Computational Creativity*, volume 147, Sydney, Australia, June 2013.

[62] Jürgen Schmidhuber. Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In *Anticipatory Behavior in Adaptive Learning Systems*, pages 48–76. Springer, 2009.

[63] Jürgen Schmidhuber. Simple algorithmic theory of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *Journal of SICE*, 48(1):21–32, 2009.

[64] Alison Pease, Daniel Winterstein, and Simon Colton. Evaluating machine creativity. In *Workshop on Creative Systems, 4th International Conference on Case Based Reasoning*, pages 129–137, Vancouver, Canada, 2001.

[65] Lus Macedo and Amlcar Cardoso. Creativity and surprise. In *Proceedings of the AISB01 Symposium on Creativity in Arts and Science*, pages 84–92. Brighton: SSAISB, 2001.

[66] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd international conference on Machine learning*, pages 896–903, Bonn, Germany, 2005. ACM.

[67] Hal Daumé Iii, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine learning*, 75(3):297–325, 2009.

[68] Aurora Tulilaulu, Joonas Paalasmaa, Mikko Waris, and Hannu Toivonen. Sleep musicalization: Automatic music composition from sleep measurements. In Jaakko Hollmén, Frank Klawonn, and Allan Tucker, ed. *Eleventh International Symposium on Intelligent Data Analysis (IDA)*, volume 7619 of *LNCS*, pages 392–403, Helsinki, Finland, November 2012.

[69] Daniel Johnson and Dan Ventura. Musical motif discovery in non-musical media. In *Proceedings of the 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia, June 2014.