A short tutorial on Computational Creativity

Hannu Toivonen
University of Helsinki
www.cs.helsinki.fi/hannu.toivonen

I am domed like an universe
hither and thither.
The salon about me
thrusts itself into voice of reedy whores,
station and lso,
domes shredded by the cleverness.

Computational creativity

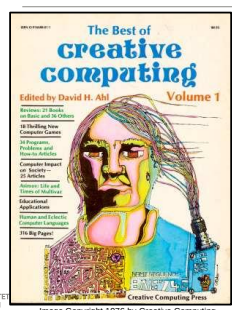


Image Copyright 1976 by Creative Computing

- Creative computers, machine creativity
- Computers supporting human creativity
- Studies of creative computational processes

From Ping Xiao and Simo Linkola: Vismantic: Meaning-making with Images, ICCC 2015

- What do you call a murderer with fibre?

- A cereal killer.

Tutorial overview

- The tutorial is an introduction to
- Concepts and theories of computational creativity
  - different types of creativity, formalization of creativity as search, social creativity, …
- Computational creativity in some fields
  - associations, poetry, music, …
- Philosophy of computational creativity
  - what is creativity, what is creative autonomy, how to assess creativity, …

Outline of the tutorial

Module I:
- General introduction to computational creativity
- Examples of creative methods based on associations

Module II:
- Formalization of creativity as search, higher levels of creativity
- Data analysis for computational creativity
- Computational creativity for data analysis

Module III:
- Social aspects and creative autonomy,
- Evaluation of computational creativity
- References and resources

Bias

- Only selected topics are covered
  - The diversity of topics is huge
- I try to cover the most central concepts
- Example applications come from our own work
- Some bias towards use of data mining and machine learning with computational creativity

Computational Creativity I

General introduction to computational creativity,
Examples of creative methods based on associations

Can Computers Be Creative?

Example: A computer takes a psychometric test of creativity (Gross et al 2012)

Creativity and associations

A common characterization of creativity:

Creativity: ability to associate concepts, ideas and problems

Association-based computational creativity?

Text corpus

✓

- 2-grams
- co-occurrence analysis

Word associations

?

- RAT, a psychometric test
- computational poetry

Creativity

Remote Associates Test (RAT)

- A psychometrical test of creativity
- Correlates with IQ and originality in brain storming
- How can computers perform?
- Disclaimer: The test is controversial. Does it *really* measure creativity?

Remote Associates Test (RAT)

- Which single word relates to each of these?
  - *coin* – ***silver** coin*
  - *quick* – *quick **silver***
  - *spoon* – ***silver** spoon*
- Measures the ability to discover relationships between remotely associated concepts
- That's psychometrics of creativity. Now let's move to *computational* creativity.

Modeling RATs computationally

- A single RAT question is a quadruple
  $r = (c_1, c_2, c_3, a)$
- A probabilistic approach: find a that maximizes
  $P(a|c_1, c_2, c_3) \propto P(a, c_1, c_2, c_3)$
- Maximize
  $$P(a)\prod_{i=1}^{3} P(c_i|a)$$
  (cf. naïve Bayes)

Modeling RATs computationally

- Learn word frequencies from a large corpus
  - Use Google 2-grams to estimate probabilities $P(a)$ and $P(c_i|a)$
  - (Google n-grams: a large, publically available collection of word sequences and their probabilities)
- A lot more could be done, but we want to keep things as simple as possible
  - Creative behavior *without* an explicit semantic resource (such as WordNet)

Experiments

- Data: published psychometric RATs with 212 questions in total
- No preprocessing at all, alternatively just simple stop word removal
- Numbers of correct answers:

Humans [1]	Computer: 2-grams	Computer: 2-grams, stopwords removed	Computer: 2-grams, plurals removed	…
50%	54%	66%	?	?

Clearly better than humans, with extreme simplicity

Comments

- An empirical proof that a computer can perform a task that is (or was) considered to require creativity
- Some creative tasks are actually easier to computers than humans

What is (computational) creativity?

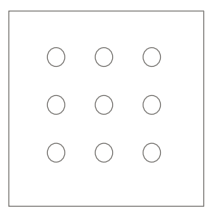
Defining creativity

– Many definitions. A representative one:

"Creativity is the ability to come up with ideas or artefacts that are new, surprising, and valuable."
- Boden 1992

– Note: Human creativity is typically defined by the output
– Tests like Torrance (below) are used in practical settings

– Connect the nine dots with four straight lines, without lifting the pen



Measuring creativity in humans

E.g., Torrance test of creative thinking:

- *Fluency:* ability to produce of *many* ideas
- *Flexibility:* ability to produce *different* ideas
- *Originality:* ability to produce *unusual* ideas
- *Elaboration:* ability to *explain* ideas

Note: in this tutorial, "idea" ≈ "artefact" ≈ "concept" = the product of creation

Three types of creativity (Boden 1992)

1. *Combinational:* new combinations of familiar ideas
2. *Exploratory:* generation of new ideas by exploration of a space of concepts
3. *Transformational:* involves a transformation of the search space so new kinds of ideas can be generated.

An alternative typology

– We propose the following, extended classification of different types of creativity (Xiao, Toivonen et al 2015)
– The types differ in terms of the input they take
– Additionally, there is the transformational case

An alternative typology

- Concept Extraction: extraction and transformation *from an existing but different representation*
- Concept Induction: learning *from examples*
  - Concept Learning: supervised, *labeled examples*
  - Concept Discovery: unsupervised, *unlabeled examples*
- Concept Recycling: creative reuse of *existing concepts*, e.g.
  - Concept Mutation: modify *one* existing concept, e.g., by generalization, specialization, or mutation
  - Concept Combination: combine *many* existing concepts
- Concept Space Exploration: takes as input *a search space* of possible new concepts

P-creativity vs. H-creativity (Boden 1992)

A different distinction between creations:

- *P-creativity* or psychological (or personal) creativity: novel just to the agent that produces it
- *H-creativity* or historical creativity: creativity that is recognized as novel by society

- In machine creativity research, emphasis is on p-creativity, i.e., the system be able to produce something novel to itself.
- H-creativity can then, in principle, be achieved with a database of existing artefacts

Credits for pictures above:

1. Kristiina Isola: Forest People (2007)
2. Maria Primatšenko: Rat (1963)

What is computational creativity?

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.

- Colton and Wiggins 2012

Computational creativity – why on earth?

- An ultimate AI challenge
- A test bed for AI methods
- Applications
  - Games
  - User interfaces, usability
  - Applications where human creativity is not feasible, e.g., instant creativity
  - Support of human creativity
- An intellectual challenge

Image from the cover of "Creative Computer Graphics" by Jankel and Morton, copyright Cambridge University Press 1984

Creating Open-ended Associations

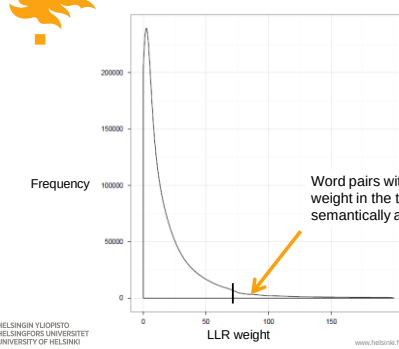
Example: Generative creativity using semantic relations

(Gross et al 2012)

Data mining for semantic (but unlabeled) associations

- Analysis of word co-occurrence *within sentences*
- Data: English Wikipedia, 2 million articles
- Preprocessing:
  - Extraction of nouns and named entities
  - Lower-casing
  - Lemmatization
- Co-occurrence analysis: log-likelihood ratio (LLR)
- Top 5% word pairs are used

Weight distribution



Word association network (a small extract related to winter)

Creative associations

- Generalize the RAT idea to semantic associations
- Find associations that are more creative than in RAT
- Practical motivation: supporting creativity

- Example: What could *riding* be associated with?
  - *horseback*
  - *accident*
  - *election (riding ≈ electoral district)*

Creative associations

- An example model: the reverse of RAT
  - Given a word a find a set of words that
    - are related to a
    - are not mutually related
- Simple greedy method:
  - Consider words in a decreasing order of association to a
  - Output a word if it is not associated to any other word already output

Example results

Seed word	Associated words		
imperialism	colonialism	lenin	american
missile	warhead	defense	flight
packaging	product	paper	artwork
slope	steep	ski	western
medley	relay	yankovic	beatles
far	north	greater	moon
kpmg	firm	young	report
concert	band	hall	benefit

Evaluation using Torrance test of creative thinking

- *Fluency:* ability to produce of many ideas
  - **production of many relevant words**
- *Flexibility:* ability to produce different ideas
  - **discovery of complementary words**
- *Originality:* ability to produce unusual ideas
  - **use of rare words in the output**
- *Elaboration:* ability to explain ideas
  - illustrating word relations and contexts

The myth of creativity

- Can computers be creative?
  - In the examples given above with word associations, was the computer creative? Why? Why not?
  - Can an algorithm be creative?
- The myth of creativity:
  - Since we can't explain creativity, we can't see how computers could be creative (Minsky 1983)

Computational poetry

A case in creative utilization of word associations (Toivanen et al 2012)

Poetry

- Poetry is a marriage between **content** and **form**
- Used e.g. to express something, to evoke emotions or opinions, and to give aesthetic pleasure
- Why would one want to automate poetry generation
  - Gaining insight into poetry
  - Gaining insight into verbal creativity
  - Applications of instant poetry
    - Turn a news story into a poem to provoke new views to it

Challenges for computer poetry

- **Content**
  - How to control the topic of a poem?
  - How to obtain semantic coherence?
  - …
- **Form**
  - How to generate (roughly) grammatical poems?
  - How to control rhyme and other phonetic features?
  - …
- How to balance
  - Control vs. surprise (or creativity)
  - Manually coded expert knowledge vs. mining/learning

A data mining inspired approach

- Goal: minimize explicitly coded expertise
  - Instead, use given corpora as resources
- Generation of grammatical structure:
  - Take an existing example (by random) and copy the syntax from it
- Generation of content:
  - Use words that are statistically related to given topic

Word association network construction

- Simple word association analysis carried out in the background corpus using the multinomial log likelihood ratio test
- Background corpus in our experiments: Finnish Wikipedia + Finnish poetry
- Result: a network of hundreds of thousands of words

Morphological analysis and synthesis

- Omorfi (Linden et al, 2009) used for Finnish morphological analysis and generation
  - Finnish is an agglutinative language with rich morphology, with several morphemes in single words
  - English: "*I wonder if I would run around*"
  - Finnish: "*juoksentelisinkohan*", derived from "juosta" ("run")

Example of operation of the method (translation from Finnish)

- An original piece of text is randomly selected:
  - *how she once played in a big, green park […]*
- Topic for new poem: (children's) play
  - *how she then played in a daring, daring whispering*
    under the pale trees.
    She had heard for fun
    how her whispering drifted as jingle to the wind.

More examples (translated from Finnish)

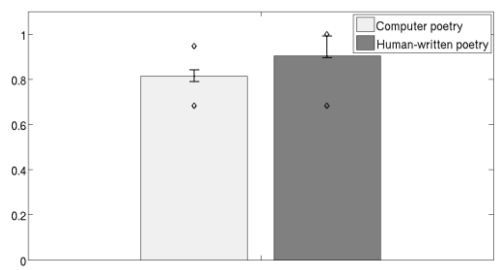
In a pale fist
in a well-balanced fist, the buds are so pale
in your image lies a dear child god.

Lives got the frolic ways,
snow the home of time,
softly chimed abandoned homes,
softly got frolics beloved –
ripening crop got the snows' joys.

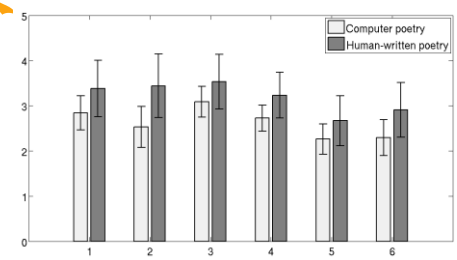
I am domed like an universe
hither and thither.
The salon about me
thrusts itself into voice of reedy whores,
station and lso,
domes shredded by the cleverness.

How dull and good and melodramatic
and sonic it was!
On miraculous locs it was always good
I thrust to re-create on an entrance
and look at it from the rca.

Evaluation: Is this text a poem?



Evaluation



- Questions: (1) typicality as a poem, (2) understandability, (3) quality of language, (4) mental images, (5) emotions, and (6) liking
- Scale from 1 (very poor) to 5 (very good)

Data mining (DM) and Artificial Intelligence (AI) vs. Computational Creativity

Data Mining vs. Computational Creativity

"Creativity is the ability to come up with ideas or artefacts that are new, surprising, and valuable."
- Boden 1992

"KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data."
- Fayyad et al. 1995

So is computational creativity ≈ data mining?

Data Mining vs. Computational Creativity

- Data mining (like most other subfields of AI) looks at
  - well-specified problems ("find all frequent patterns")
  - with obvious and objective success criteria
  - that can be measured with relative ease
- Computational creativity looks at
  - ill-defined, open-ended problems ("write a poem")
  - with subjective and non-explicit criteria
  - that cannot be computed easily

Artificial Intelligence vs. Computational Creativity

Artificial Intelligence	Computational Creativity
Split into several subfields (robotics, natural language, inference, learning, planning…)	No obvious structure beyond applications (verbal, musical, …)
Well-formulated problems	Open tasks
Obvious measures of success (quality of the solution)	No good measures of success

Escape from the blocks world

- A generative system can be programmed to perform well in limited settings
  - E.g., poetry: use hand-crafted generative grammars, knowledge bases, and lexicon to obtain better control
  - Leads to the same issues as the "blocks world" in AI:
    - Nice demos but no scalability beyond toy examples
- Data mining can make an opposite approach feasible
  - Assume minimal knowledge as input
  - Use data and data mining instead
- Trade-off: control vs. wide applicability

Computational Creativity II

Formalization of creativity as search, higher levels of creativity
Data analysis for computational creativity
Computational creativity for data analysis

Formalization of creativity as search

Wiggins (2006)

Creativity as search

Wiggins: "A preliminary framework for description, analysis and comparison of creative systems" (2006)

- A conceptual framework for talking about creative systems and their properties
- Views creativity as search (cf. search in AI)
- Looks like an architecture but is not intended to be used as one
- In this tutorial, a simplified version will be presented

Components of the framework

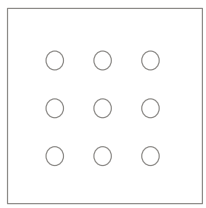
- Universe U contains all possible concepts
- Rules R define the acceptable conceptual space
- Evaluation function E assigns a value to a concept
- Method $T_{R,E}$ for searching U with respect to R and E

Components of the framework

- Universe U contains all possible concepts
  - E.g., all possible sequences of words
- Rules R define the acceptable conceptual space
  - E.g., those sequences that match a given meter
- Evaluation function E assigns a value to a concept
  - E.g., does the text express the desired emotion
- Method $T_{R,E}$ for searching U w.r.t. R and E
  - E.g., produce poems using a generative grammar and expressions reflecting the desired emotion

U, R, T, E are system-specific

Recall this problem with four lines connecting the dots.



What was your

- Universe U?
- Rules/acceptable search space R?
- Evaluation function E?
- Traversal (search) method T?

Data mining tasks in computational creativity

- Mine, learn, or model:
  - the universe U and/or rules R for acceptable cases from existing examples
  - the evaluation function E from recognized examples or from the user,
  - methods T that leverage existing examples and their properties
  - changes to any of the above from experience and from interaction with others (cf. transformational creativity and social creativity)

Creativity as Search vs. Boden's Three Types of Creativity

- Recall Boden's three types of creativity
  - Combinatorial (combining old ideas to new ones)
  - Exploratory (generating new ideas within rules)
  - Transformational (also changing the rules)
- Wiggins' model looks like exploratory search
  - A space defined by U, R and E explored by T
- However, Wiggins' model is generic and allows U, R, E and T to be defined in various ways
  - E.g., T can be based on recombinations of existing ideas (leading to combinatorial creativity)

Higher Levels of Creativity – Transformational Creativity

Wiggins (2006)

Creativity as search: metalevel

Wiggins introduces the following additional notation:

- A language L, in which R, E, T are expressed
  - R ∈ L, E ∈ L, T ∈ L
- An interpreter ⟦ ⟧ for rules R
  - ⟦R⟧(c) evaluates c ∈ U using R
- An interpreter ⟪ ⟫ for search method T
  - ⟪R,T,E⟫(c_{in}) produces c_{out}, concepts to traverse next
- This allows rules R and search method T (and evaluation function E) to be modified during runtime
  → Boden's *transformational creativity*

Transformational Creativity as Metasearch

- Consider the transformational case where rules R are modified in the creative process
- Formulate Wiggins' model to search for artefacts *and rules*
  - E.g. in poetry: select a set of poetic features (meter, number of syllables and lines, alliteration, rhyme pattern, …) and generate a matching text
- Metauniverse
  U_L = {(R, c) | R is a possible rule set, c ∈ U}

Transformational Creativity as Metasearch

- R_L: metarules about valid (R, c) pairs
- E_L: evaluation of (R, c) pairs
- T_L: search method for (R, c) pairs
- **Exploratory search w.r.t. U_L, R_L, E_L, and T_L is transformational creativity**
- In more general, allow modification of E and T, too, and search for tuples (R, E, T, c)

Formulation of Metasearch

- "Normal" search is defined by tuple
  ⟨U, L, ⟦ ⟧, ⟪ ⟫, R, T, E⟩
- Metasearch:
  - The universe consists of all possible R, T, E, i.e., of expressions in L, i.e., U_L = L
  - A metalanguage L_L is needed to talk about L
- Metasearch is thus defined by tuple
  ⟨L, L_L, ⟦ ⟧, ⟪ ⟫, R_L, T_L, E_L⟩

Notations by Wiggins

Misc. notation/a reading guide

- $F^n(X) = F(...(F(X)))$
  i.e., F applied recursively n times
- $F^\diamond(X)$ = union of all recursive applications, i.e., all that can be obtained from X by F
- ⟪R,T,E⟫$^\diamond$({T}) = everything that $T_{R,E}$ can reach in universe U
- ⟦E⟧(⟪R,T,E⟫$^\diamond$({T})) = everything of value that $T_{R,E}$ can reach

Possible Properties of Creative Agents

- "Generative uninspiration": $T_{R,E}$ does not reach anything valuable
  - ⟦E⟧(⟪R,T,E⟫$^\diamond$({T})) = ∅
- A milder form: a lot of (highly) valued concepts cannot be reached by TR,E
  - ⟦E⟧(⟦R⟧(U)) \ ⟪R,T,E⟫$^\diamond$({T}) is significant
- Transformation of T is required
- Help from outside is needed, e.g., valued concepts
  - Learning, social aspects!

Possible Properties of Creative Agents

- "Aberration": $T_{R,E}$ reaches concepts outside R
  - A = B = ⟪R,T,E⟫◇({T}) \ ⟦R⟧(U) ≠ ∅
  - (Wiggins refers to this set first as B, later as A)
- "Pointless aberration": the extra concepts are not valued
  - V = ⟦E⟧(B) = ∅
  - Need to transform T to avoid the useless search

Possible Properties of Creative agents

- "Productive aberration": $T_{R,E}$ reaches some valued concepts outside R
  - V = ⟦E⟧(B) ≠ ∅
  - Transform R to include the valued concepts?
  - (Possibly transform T to exclude unvalued ones)

Machine Learning and Data Mining for Computational Creativity

Toivonen and Gross (2015)

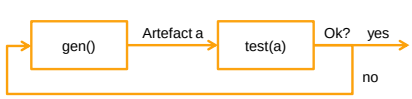
Self-determinism and creativity

- A purely preprogrammed generative system
  - only does what it was told to do
  - has little creativity
- Adaptivity or self-determinism
  - Is necessary to attribute any creative autonomy or originality to a creative system
- Transformative or meta-level creativity (cf. Boden, Wiggins) can be attributed with higher creativity
  - …but how to build a system to deal with unanticipated cases?

→ Opportunities for ML and DM

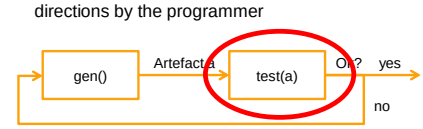
ML and DM in CC

- Let's use a simple generate-and-test model to illustrate uses of machine learning (ML) and data mining (DM) in CC



Learning to evaluate

- Use ML to learn an evaluation function eval(a) from training examples
  - E.g. a classifier that tells if the result is good
- Assuming a generator gen() exists, its outputs are filtered by the trained classifier without explicit directions by the programmer



Learning to evaluate

An example system, DARCI (Ventura et al)

- Creates images that express an emotion
- Emotion detection is based artificial neural networks trained by users of the system
- A genetic algorithm is used as generator gen()
  - Adapts to the evaluation/fitness function eval()
- http://darci.cs.byu.edu/
- ”DARCI, draw me a happy picture!”

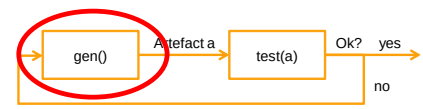
A happy image by DARCI, http://darci.cs.byu.edu/

Learning to evaluate

Bottlenecks in learning the eval() function

- Learning an evaluation (or fitness) function eval(a) can be very difficult
  - How does one evaluate the quality of a poem?
- Generating complex artefacts, i.e., writing (or learning) the function gen(), can be very hard
  - In practice, the generation step must be adaptive in order to be effective
- Pastiche generation, i.e., mere imitation of training examples rather than creativity

Learning to Generate



- Predictive models
- Generative models

Learning to Generate Using Predictive Models

1. Completion of partial artefacts
   - Given some part of the artefact, predict the values of the remaining parts
   - Based on training on complete artefacts

   E.g. harmonization of music:
   - Given a melody (possibly created by the system itself), choose suitable chords to accomppany the melody

Learning to Generate Using Predictive Models

2. Reduce the task of generating complex structures to selection.

   E.g. generation of accompaniment by running a classifier to pick a suitable chord, and then using (possibly automatically extracted) patterns to generate the exact accompaniment

Learning to Generate Using Predictive Models

3. Generate complex structures using instance-based techniques
   - E.g. k-nearest neighbours and case-based reasoning
   - avoids using models, decision structures, or patterns
     - can be difficult to specify or learn
     - could be restrictive.

Example: Corpus-based poetry by Toivanen et al.

- No explicit grammar, instances simply copied from a corpus

Learning to Generate Using Generative Models

Generative models (from ML and statistics) can be used more directly to generate artefacts

- E.g. Markov models for sequencies such as text and music
- Artificial neural networks, with slight modification of weights (and keeping the input constant)

Mining patterns for creative tasks

1. Use data mining to discover patterns in, say, text
2. Utilize these patterns in a generation function gen()

Examples:

- Association-based creativity (Gross et al)
- Corpus-based poetry (Toivanen et al)

Mining patterns for creative tasks

Example: metaphor generation (Veale et al)

1. Extract similes (“strong as a bull”) from a corpus
   - Look for patterns of the form “T is as P as a V”
2. P (“strong”) is a typical property of V (“bull”) if the pattern “T is as strong as a bull” occurs often
3. To express ”he is strong” in a metaphorical way, find a noun V for which ”strong” is a typical property
   - Bull is found as a suitable V
4. Output “he is a V”, i.e., “he is a bull”

http://ngrams.ucd.ie/metaphor-eye/

Metaphor-Eye

Why are scientists like artists?

- Scientists
  - …develop ideas like artist
  - …explore ideas like artist
  - …acquire skills like artist
  - …spread ideas like artist
  - …nurture ideas like artist
  - …develop techniques like artist

Transformational Creativity Using Data Mining and Machine Learning

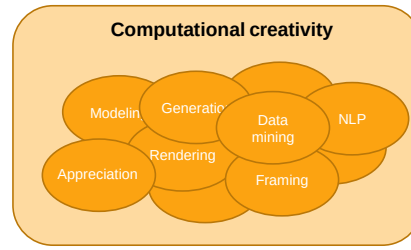
Wiggins suggests uses of ML/DM:

- Automatic adaptation of R or T
  - To remedy aberration: use aberrant concepts as positive or negative examples, depending on their value
  - To remedy generative uninspiration: use positive (and negative) examples received from outside
- Automatic adaptation of E
  - Use feedback and evaluations received from outside (not covered by Wiggins)

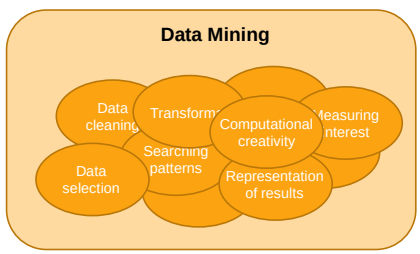
Computational Creativity for Data Analysis

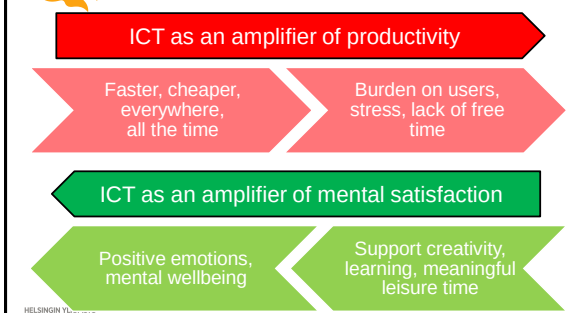
Tulilaulu et al (2012 + manuscript)

Data mining for creativity



Creativity for data mining





Can data be turned into a subjective, esthetic experience?

– Given **your** data, could **you** enjoy listening to it as music?
– Could you even feel **joy of creativity**?

Data Musicalization

– Using given data as input,
– automatically compose a novel piece of music

New goals for data analysis

Traditional data analysis and representation (statistics, visualization, sonification, etc.):

- Transfer of information
- Objectivity
- (Cognitive emphasis)

Data musicalization:

- Experiences involving feelings and emotions
- Subjective
- (Affective emphasis)

Motivation for data musicalization

- Esthetic pleasure
- Joy of creativity
- Transfer of (some) information in a novel way
  - Possibly in the background, unconsciously
  - Like sonification, but musical
- Building an emotional attachment to an application

Sleep musicalization

- An application of data musicalization
- A piece of music is composed from a night's sleep
- *Goal: help and motivate users track their sleep and eventually improve their sleep*
- Two phases
  1. Data analysis:
     from sensor data to sleep measurements
  2. Composition algorithm:
     from sleep measurements to music

Musicalization vs. sonification

Sonification:

- Maps data to sounds
- Not necessarily musical or creative
- Focus on conveying information objectively

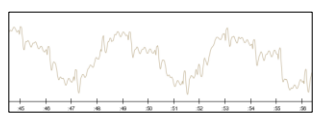
- Novelty here: musicality, subjective experience

Phase 1, data analysis

From sensor data to sleep measurements

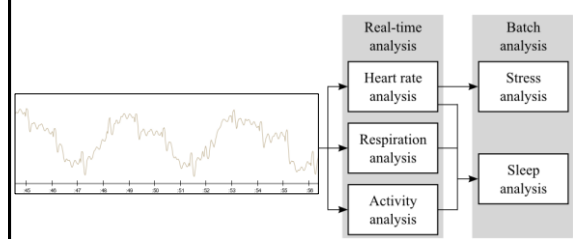
Sensor data

A thin piezo-electric force sensor placed under the mattrass topper (from Beddit Ltd)

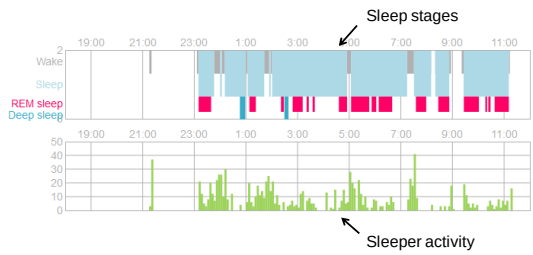


Force signal with sample rate 140 Hz (here a 12-second signal excerpt)

Analysis of sensor data



Sleep analysis results



Phase 2, composition

From sleep measurements to music

Composition of music from sleep

Design principles/goals:

- Compose a novel piece of music
- Use sleep measurements to guide the composition
- Produce music, not sonification
- Make the music reflect properties of sleep
- Compress 8 hours of sleep to couple of minutes
- Use simple methods, build a proof of concept

Composition in 6 steps

1. Generation of harmonic progression (chord seq.)
2. Melody generation
3. Generation of rhythm
4. Generation of accompaniment
5. Adjusting volume
6. Regulation of tempo

Correspondence between sleep and music

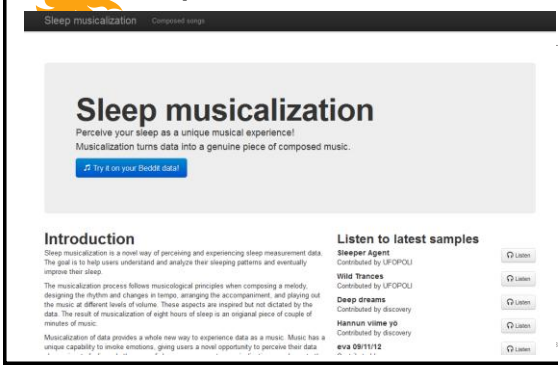
In a nutshell:

- Deeper sleep corresponds to calmer music
- Lighter sleep corresponds to livelier music
- Each sleep stage has a different accompaniment (and theme)
- Tempo is regulated by the heart rate
- More activity corresponds to louder music

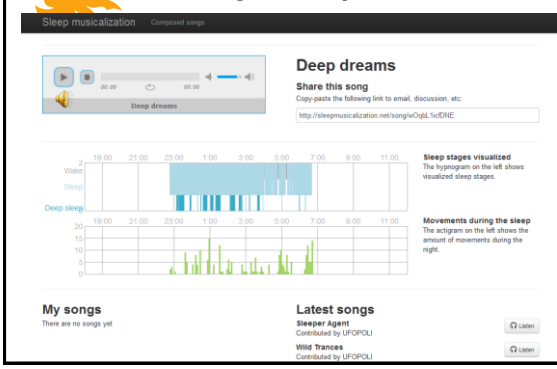
sleepmusicalization.net

- Sleep musicalization is available as a public web app at http://sleepmusicalization.net
- You can have your sleep measurements composed into music (if you have a Beddit sleep sensor), or
- Listen to songs published by other users

sleepmusicalization.net



Listening to sleep music



A different example

Independent reviews

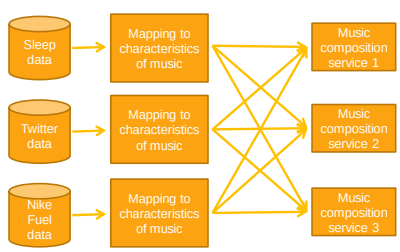
"[The songs] do have a uniquely personal and (dare I say it?) dreamlike feel"

- Leslie Katz, CNET

"[…] these songs left me with a sense of incompletion, as if the piano-dominate music would be an inadequate representation of my dreams."

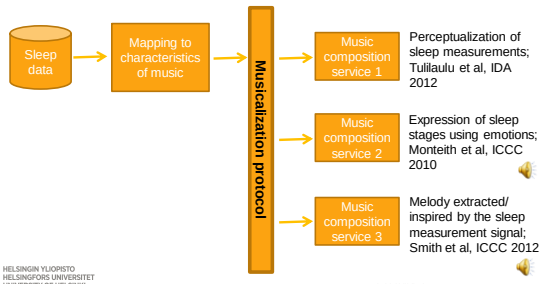
- Nic Halverson, Discovery News

Modular musicalization?



From Juslin, 2001

Alternative sleep musicalizations

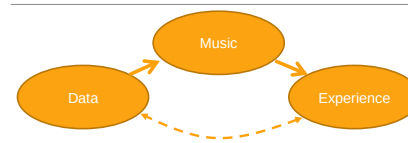


Perceptualization of sleep measurements; Tulilaulu et al, IDA 2012

Expression of sleep stages using emotions; Monteith et al, ICCC 2010

Melody extracted/ inspired by the sleep measurement signal; Smith et al, ICCC 2012

Future work in data musicalization



- Understand how subjective experiences can be affected by automatically composed music
- Develop principles for mapping properties of data to intended experiences
- Musicalization of patterns and other DM results

Applicability of data musicalization?

Where could data musicalization potentially be used?

The following characteristics may help:

- Personal data
  - Affectivity + creativity factors
- Data with a temporal dimension
  - Natural mapping to the temporal dimension of music

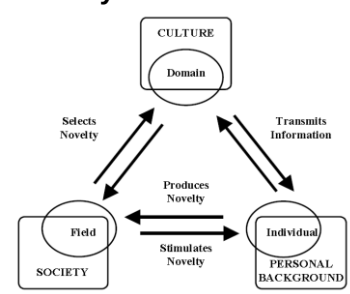
Computational Creativity III

Social aspects and creative autonomy,
Evaluation of computational creativity,
References and resources

Social Aspects of Creativity

Saunders and Gero (2001)

Creativity is a socio-cultural activity



Source of pictures in this lecture: Saunders and Gero (2001)

Socio-cultural aspects

- The context and background of creativity
- Interaction, development
- The audience of results
- What and where is the impact?
  - Historical creativity (h-creativity) is a social aspect
- …
- What could be a minimal computational model of socio-cultural creativity?

A model of social artificial creativity

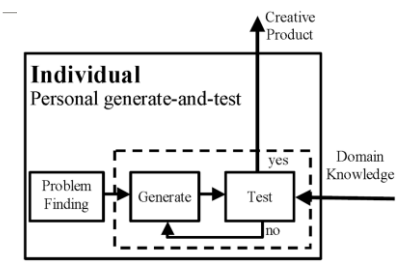
Saunders and Gero (2001)

- A society of agents in a cultural environment
- No agent can direct the behaviour of others
- No rules dictate global behaviour
- Agents interact with other agents to exchange artefacts and evaluations
- Agents interact with the environment to access cultural symbols
- Agents evaluate the creativity of artefacts and other agents

Social aspects in creativity

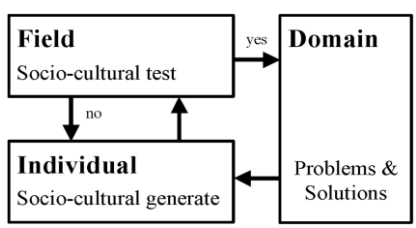
- The notions of whom and what are creative arise from multiple notions held by the individual agents
- Macro-level creativity from micro-level interactions

Individual's generate-and-test model



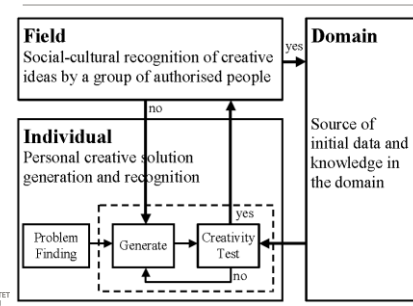
Cf. personal creativity (p-creativity)

Socio-cultural generate-and-test model



Cf. historical creativity (h-creativity)

A dual generate-and-test model



Creative Autonomy

Jennings (2010)

"The difference between greater and lesser creativity lies not it how you solve problems, but rather in what problems you choose to solve."

- Getzels and Csikszentmihalyi

- What is the programmer's influence on what a creative program creates?

Criteria for Creative Autonomy (1/3), Jennings (2010)

1. **Autonomous Evaluation**:
   The system can evaluate its liking of a creation without seeking opinions from an outside source.
   - Any opinion is formed by the system itself
   - However, it may consult others at other times
   - Examples: preprogrammed evaluation, evaluation function learned from the user

Criteria for Creative Autonomy (2/3)

2. **Autonomous Change:**
   The system initiates and guides changes to its standards without being explicitly directed when and how to do so.
   - External event and evaluations may prompt and guide changes
   - The system decides when and how to change them
   - The system decides if new standards are acceptable
   - Fixed or learned evaluation functions can be used to bootstrap the process

Criteria for Creative Autonomy (3/3)

3. **Non-Randomness:**
   The system's evaluations and standard changes are not purely random.
   - The two first criteria could be easily met by random decisions
   - Not all randomness is excluded, however

Autonomy Requires Sociality

- What influences can a creative system experience to modify its standards?
- Introspection?
  - Cf. "uninspiration" and "aberration" in the search model of Wiggins
- Social interaction!
  - New influences, ideas, feedback
  - An apparent paradox: a system can only be autonomous if it is social
  - Think of the opposite: a system that is not influenced by external information can be argued to only express the programmer's creativity

Human-Computer Co-Creativity

Human-computer co-creation

- Shared creative responsibility between a human and a computer
- Joint ”ownership” of the result

- A major opportunity for computational creativity:
  - Enhancement of human creativity
  - Giving joy of creativity to everyone
  - Educational applications

Co-creation: Case Poetry Engine

Co-creation: Case Musicreatures

- App Store: Musicreatures

Evaluation in Computational Creativity

Evaluation is important – and difficult

- Evaluation of creativity allows us to compare methods and control progress
- However, evaluation of creativity is very difficult
  - No precise definition of creativity
  - Various goals (novelty, value, originality, ...)
  - Context-dependence
  - Cost of evaluation
  - ...

What to evaluate?

- Machine creativity:
  Creative performance of creative programs
- Computer-supported creativity:
  Increase in creativity of humans using CC tools
- Creativity studies: Increase in knowledge about creative processes

- Focus here: evaluation of machine creativity

Evaluation of Machine Creativity

Two possible targets in evalution of machine creativity (Colton 2008):

- Artefact-based evaluation: are the results creative?
  - e.g: novelty and value of results
- Process-based evaluation: is the process creative?
  - e.g: combinatorial/ exploratory/ transformational creativity; creative acts of the FACE model

Ritchie's Framework for Artefact Based Evaluation

Ritchie (2007)

Essential properties

Consider a set R of artefacts produced by a system.

Primitive properties that can be considered:

- **Typicality**: Is the artefact a typical/ recognizable example of the target genre?
- **Novelty**: How (dis)similar is the artefact to existing examples of its genre?
- **Quality** [= Value]

Formal definitions

- typ(a) = amount of typicality associated to artefact a
- val(a) = amount of quality associated to a
- $T_{\alpha,\beta}(X) = \{a \in X \mid \alpha \leq typ(a) \leq \beta\}$
  - Set of artefacts a with typicality between α and β
- $V_{\alpha,\beta}(X) = \{a \in X \mid \alpha \leq val(a) \leq \beta\}$
  - Set of artefacts a with value between α and β
- size(X) = number of elements of X
- ratio(X,Y) = size(X) / size(Y)

Some criteria

Criterion 2 $ratio(T_{\alpha,1}(R), R) > \theta$

- at least fraction θ of results R have high typicality (>α)

Criterion 4 $ratio(V_{\gamma,1}(R), R) > \theta$

- at least fraction θ of results R have high value (>γ)

Criterion 5 $ratio(V_{\gamma,1}(R) \cap T_{\alpha,1}(R), T_{\alpha,1}(R)) > \theta$

- at least fraction θ of results R have both high value (>γ) and high typicality (>α)

Inspiring set

– Any creative system is based on some existing examples, in one way or another. These can – and should – be taken into account.
– The *inspiring set* consists of all the relevant artefacts known to the program designer, or items which the program is designed to replicate, or a knowledge base of known examples which drives the computation within the program
– Inspiring set ≈ training set in ML/DM

Some more criteria

Criterion 9 *ratio*($I \cap R$, I) > θ

- Results R reproduce at least fraction θ of the inspiring set I
- Is the system able to reproduce its training examples?

Criterion 10 *ratio*(R, $I \cap R$) > θ

- Results R contain at least θ-1 times as many items outside the inspiring set I as inside it
- Can the system extrapolate outside the training examples?

Novelty vs. typicality?

Novelty and typicality are subtly different:

- Not recognizable as a member of the genre → low typicality
- Very different from the inspiring set (but possibly very clearly within the genre) → high novelty

Comments

Note: Ritchie does not prescribe a set of criteria. Instead, the criteria must be designed and chosen according to the goals and needs of each work; Richie gives examples of some of the possible criteria that one may want to use .

FACE Model for Process-Based Evaluation

Pease and Colton (2011)

F, A, C, E

– Focus on *creative processes*, not their results
– In the FACE model, systems can be characterized by their creative acts
– The four aspects of the model:
  – F – framing
  – A – aesthetics
  – C – concept
  – E – expression
– Here we present a simplified version

FACE aspects

- C: the *concept* or the idea of the artefact
  - E.g. use of excessive rhyming in poetry
- E: a concrete *expression* of the concept
  - E.g. a poem that uses excessive rhyming
- A: a measure of *aesthetics* of the work of art
  - E.g. grammaticality etc. of a poem
- F: all background information about the piece (*framing*)
  - E.g. a description of why excessive rhyming could be interesting, and what the poem expresses

Framing

- Framing is especially important for computational creativity
- It is difficult to appreciate the output (expression) without knowing anything about the process, its goals, etc.
  - E.g., is the resulting image pretty just by change? Or did the system produce it based on some specific criteria and goals? Was the process complicated? Is there some intention, e.g., a message that is being conveyed?

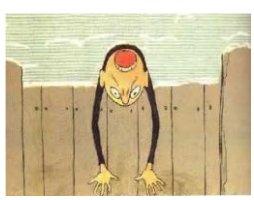
Ground level of FACE

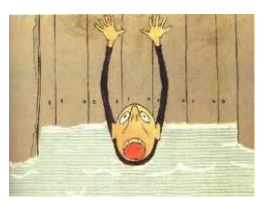
- Ground-level generative acts and their products
  - $F^g \rightarrow$ an item of framing information
  - $A^g \rightarrow$ an aesthetic measure
  - $C^g \rightarrow$ a concept
  - $E^g \rightarrow$ an expression of a concept
- Any system can now be described in terms of who generates these, and how
  - A simple generative system only performs E^g
  - A system that learns to evaluate also performs A^g
  - (The programmer and other humans probably perform the other acts)

Meta-level of FACE

- FACE also has a meta-level: processes that produce ground-level generators
- Process-level acts and their outputs:
  - $F^p \rightarrow$ a method for generating framing information
  - $A^p \rightarrow$ a method for generating aesthetic measures
  - $C^p \rightarrow$ a method for generating concepts
  - $E^p \rightarrow$ a method for generating expressions of a concept
- E.g., E^p generates new methods for generating expressions

Example from Pease et al, 2011
The Upside Downs by Verbeek





FACE	Upsidedowns
F^p:	Methods for generating the contextual history of this genre of art
F^g:	The contextual history of this genre of art, motivation, justification, etc.
A^p:	Methods for generating the idea of art having multiple meanings when viewing from multiple perspectives
A^g:	The idea of art having multiple meanings when viewing from multiple perspectives
C^p:	Methods for generating new perspectives from which the art might make sense
C^g:	The constraint that a picture must make sense when upside down
E^p:	Methods for generating expressions of art which have a different meaning when viewed upsidedown
E^g:	Expressions of art which have a different meaning when viewed upsidedown (see figure 1)

Conclusions

Summary

- Computational creativity
  - A fascinating area
  - A huge variety of topics and viewpoints
  - A small but active community (ICCC, ACC)
  - Lots of potential uses for data analysis techniques
    - In avoiding limitations of block worlds
    - In providing higher autonomy and creativity

Acknowledgements

- Oskar Gross (word associations, ML+DM for CC)
- Daniel Johnson (music creation)
- Anna Kantosalo (co-creativity)
- Petri Myllys (Musicreatures)
- Joonas Paalasmaa (sleep measurements)
- Jukka Toivanen (poetry generation)
- Aurora Tulilaulu (music composition)
- Alessandro Valitutti (humor)
- Dan Ventura (music creation)
- Mikko Waris (sleep musicalization)
- Ping Xiao (concept representation)

Resources and Pointers

Computational creativity resources

Web sites:

- Association for Computational Creativity
  http://computationalcreativity.net/
- European coordination action for promoting computational creativity
  http://prosecco-network.eu/

Books, journal special issues, bibliographies, article repositories, videos: see the Resources sections of the above web sites

Computational creativity resources

Conference:

- International Conference on Computational Creativity, ICCC, annual since 2010
  http://computationalcreativity.net/iccc2015

Other scientific publications:

- Conferences and journals of general AI and of specific application fields

On-line forum (under construction):

- https://groups.google.com/forum/#!forum/computational-creativity-forum

Cartoon tutorials: http://robotcomix.com

References

General:

– Margaret A. Boden: Creative Mind: Myths and Mechanisms (2nd ed.). Routledge, Oxon, UK, 2004 (1st edition:1992)

– Simon Colton and Geraint A. Wiggins: Computational Creativity: The Final Frontier? ECAI 2012 - 20th European Conference on Artificial Intelligence, 21-26, Montpellier, France, August 2012.

Theory:

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

References

Reviews:

– Ping Xiao, Hannu Toivonen, et al: Review of Conceptual Representations for Concept Creation. Manuscript.

– Hannu Toivonen and Oskar Gross: Data Mining and Machine Learning in Computational Creativity. Manuscript.

– (both can be obtained by contacting Hannu)

References

Social aspects:

– R. Saunders and J.S. Gero. Artificial creativity: A synthetic approach to the study of creative behaviour. In JS Gero and ML Maher (eds), *Computational and Cognitive Models of Creative Design V*, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, 113-139, 2001.

Creative autonomy:

– K.E. Jennings: Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4): 489-501, 2011.

References

Word associations and creativity:

– Oskar Gross, Hannu Toivonen, Jukka M Toivanen, Alessandro Valitutti: Lexical Creativity from Word Associations. International Conference on Knowledge, Information and Creativity Support Systems (KICSS), Melbourne, Australia, November 2012.

Corpus-based poetry:

– Jukka Toivanen, Hannu Toivonen, Alessandro Valitutti, Oskar Gross: Corpus-Based Generation of Content and Form in Poetry. International Conference on Computational Creativity (ICCC), 175-179, Dublin, Ireland, May 2012.

References

Metaphor:

– T. Veale. Creative Language Retrieval: A Robust Hybrid of Information Retrieval and Linguistic Creativity. In Proc. of the ACL'2011, the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, June 19-24, 2011.

Data musicalization:

– Aurora Tulilaulu, Joonas Paalasmaa, Mikko Waris, Hannu Toivonen: Sleep Musicalization: Automatic Music Composition from Sleep Measurements. Eleventh International Symposium on Intelligent Data Analysis (IDA), LNCS 7619, 392-403, Helsinki, Finland, October 2012.

– Aurora Tulilaulu, Matti Nelimarkka, Joonas Paalasmaa, Daniel Johnson, Dan Ventura, Petri Myllys, Hannu Toivonen: Data Musicalization. Manuscript.

References

Evaluation:

– Graeme Ritchie. Some Empirical Criteria for Attributing Creativity to a Computer Program. Minds and Machines, 17(1):76-99, Springer, 2007.

– Anna Jordanous. Evaluating Evaluation: Assessing Progress in Computational Creativity Research. Proceedings of the Second International Conference on Computational Creativity, pp. 108-110, Mexico City, 2011

– Colton, Simon, A. Pease, and J. Charnley. "Computational creativity theory: The FACE and IDEA descriptive models." Proceedings of the Second International Conference on Computational Creativity. 2011.

– Pease, Alison, and Simon Colton. "Computational creativity theory: Inspirations behind the FACE and the IDEA models." Proceedings of the Second International Conference on Computational Creativity. 2011.