Different types of creativity

Three types of creativity (Boden 1992)

(Recap from an earlier lecture)

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

Q: How do their inputs differ? (How do the differences in input reflect what is done?)

A refined typology of creativity

– We propose the following, extended classification of different types of creativity (Xiao, Toivonen et al 2016, under review)

– The types differ in terms of the input they take, and thus in the processing they (can) do on it

A refined typology

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

Transformational/metacreativity

- Additionally, there is the transformational case: takes as input an explicit specification of any of the previous tasks and can manipulate the specification

- (Cf Wiggins' model of creativity and its metalevel, also Ventura's intent)

Transformational/metacreativity

(Recap from an earlier lecture)

Computational creativity is

– the philosophy, science and engineering

– of computational systems which,

METALEVEL/
INTENT

– ***by taking on particular responsibilities***,

– exhibit behaviours that unbiased observers would deem to be creative.

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

Creative Autonomy vs. Social Creativity

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

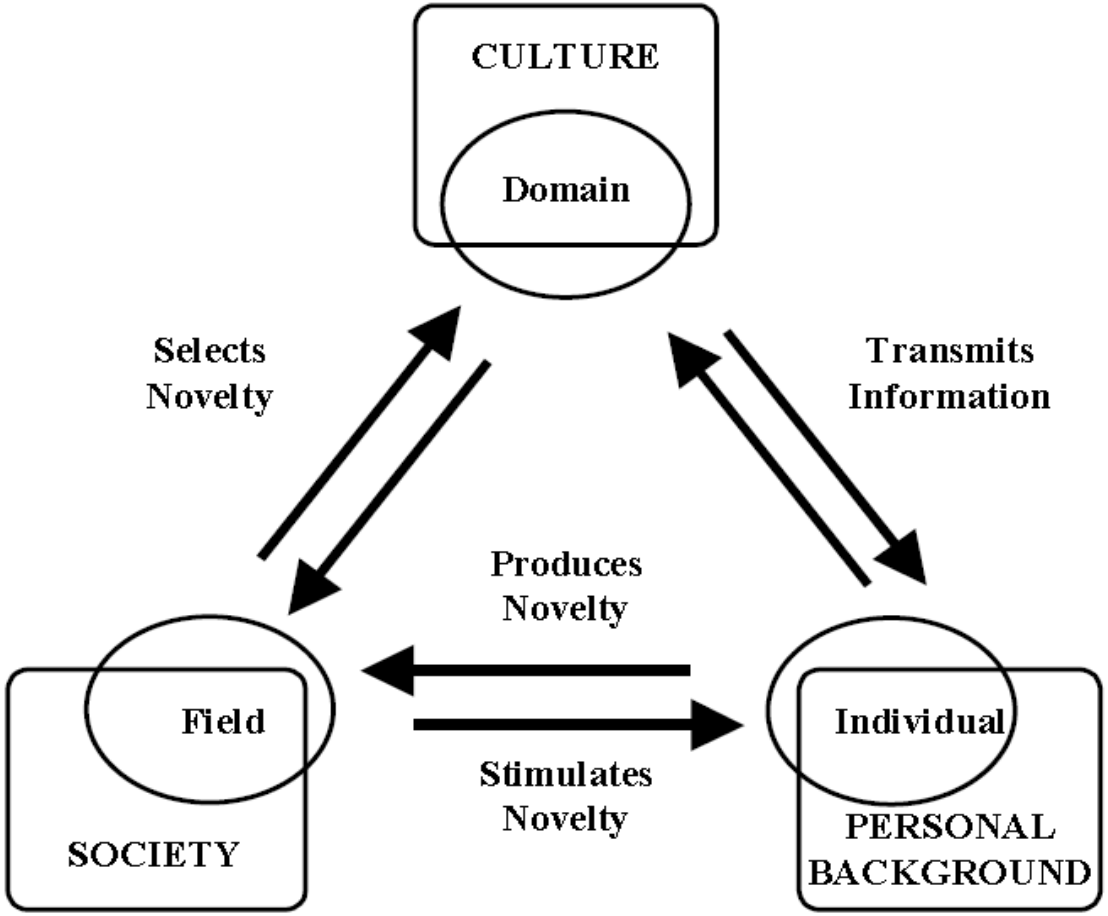
Autonomy Requires Sociality

- 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

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

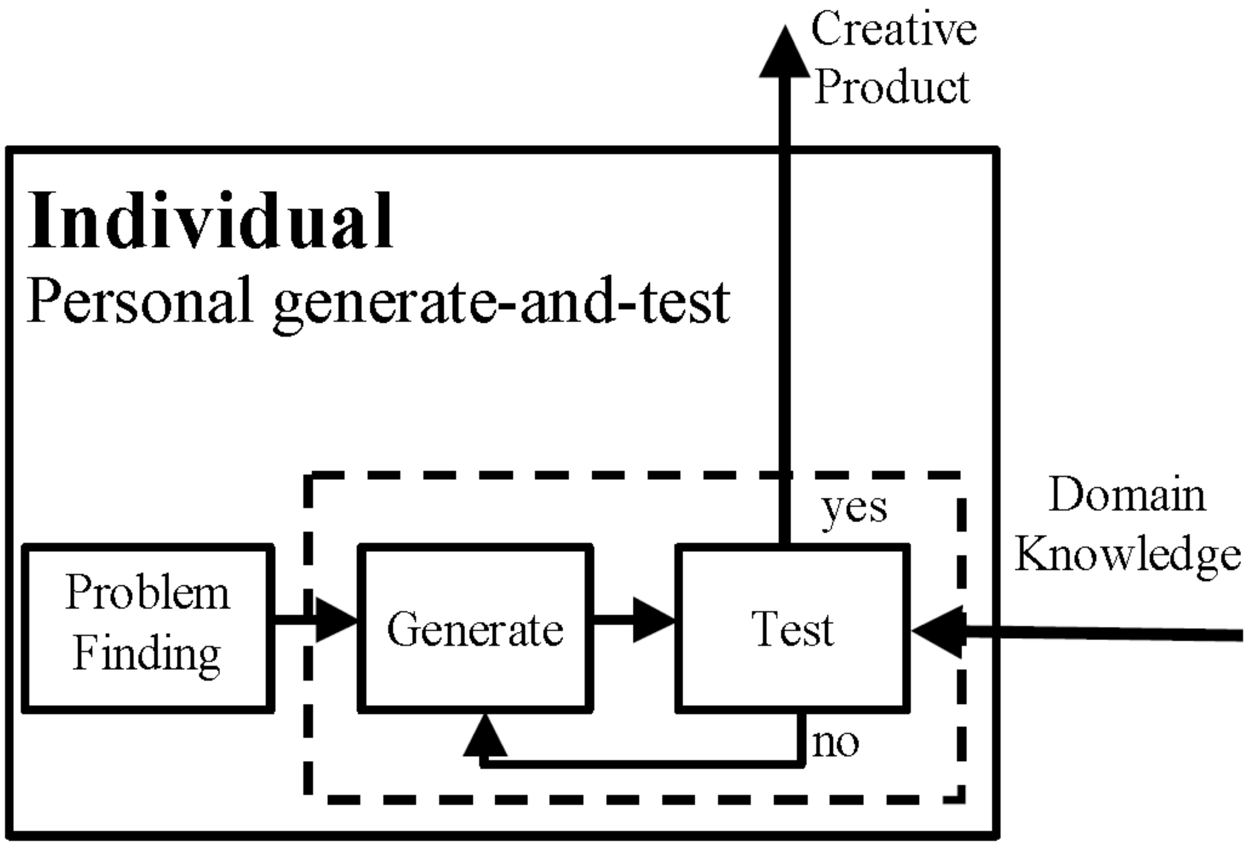
Saunders and Gero (2001)

- 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

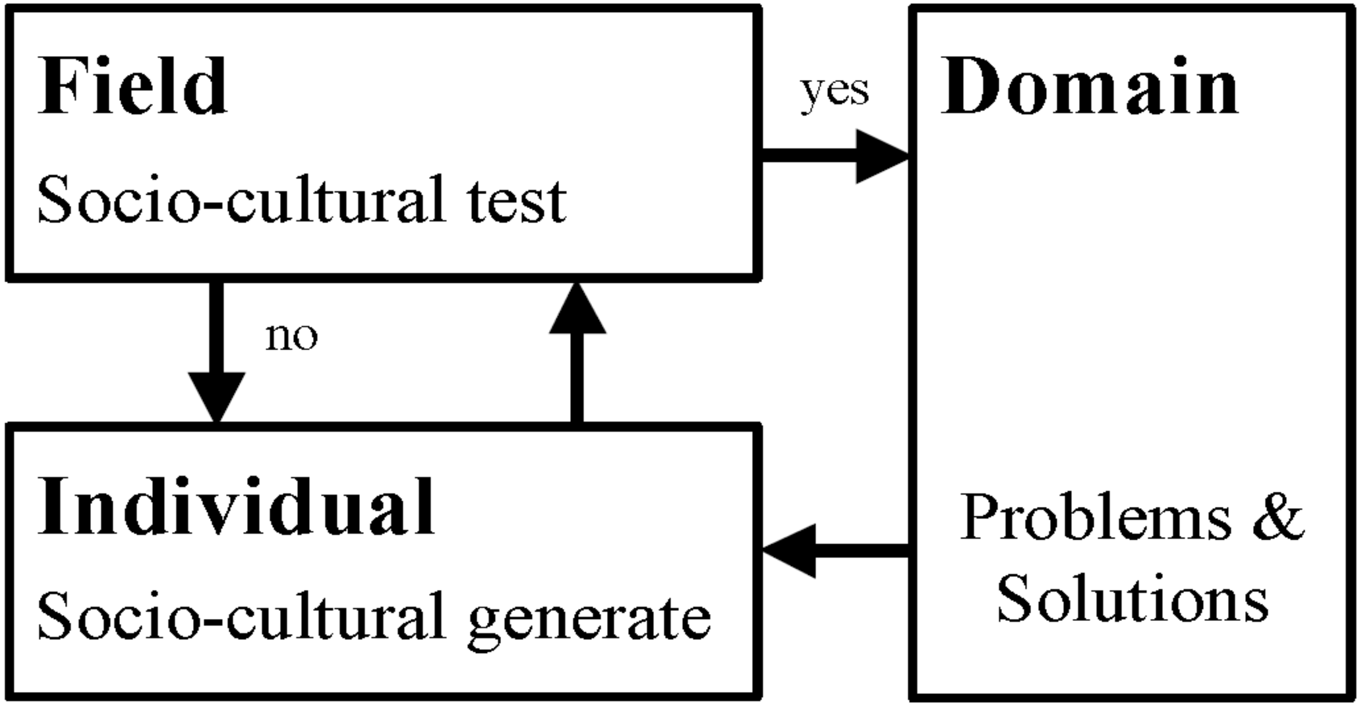
- The notions of whom and what are creative arise from multiple notions held by the individual agents
- Macro-level creativity from micro-level interactions

Individual's generate-and-test model



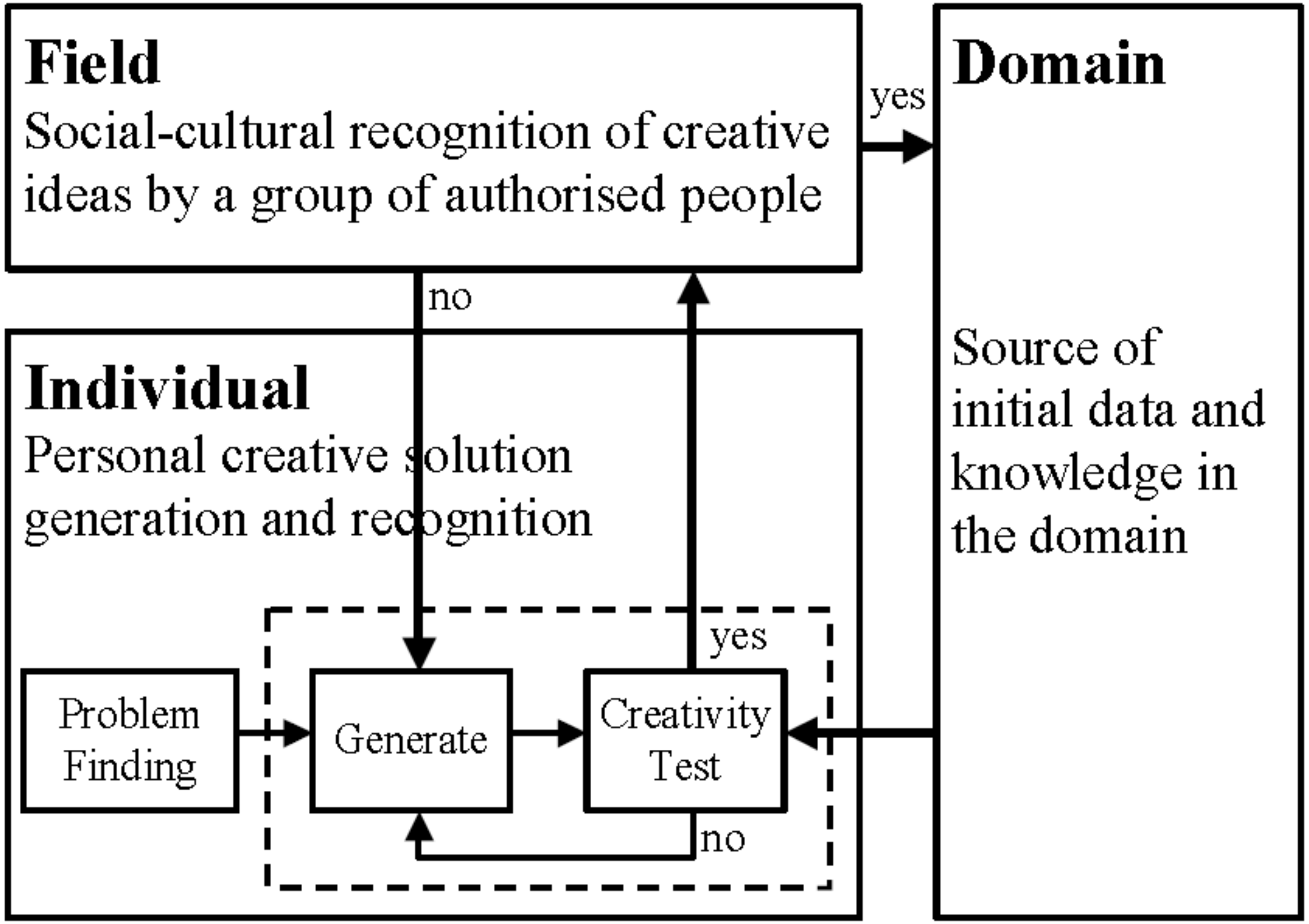
Cf. personal creativity (p-creativity)

Socio-cultural generate-and-test model



Cf. historical creativity (h-creativity)

A dual generate-and-test model



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

Machine Learning and Data Mining for Computational Creativity

Toivonen and Gross (2015)

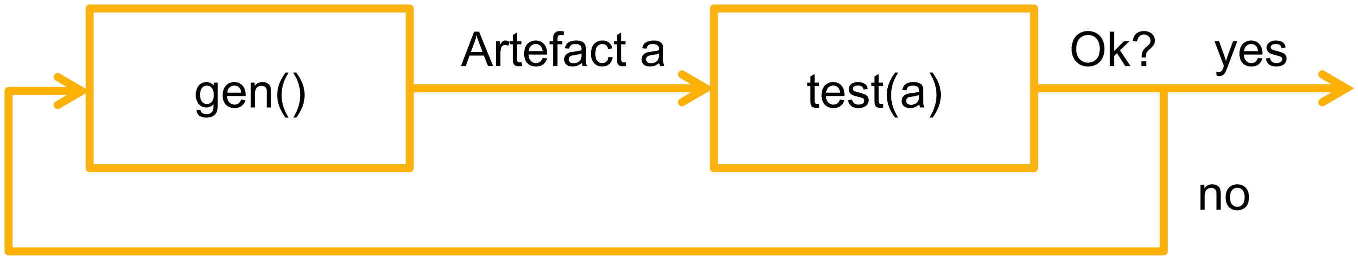
Self-determinism and creativity

– 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

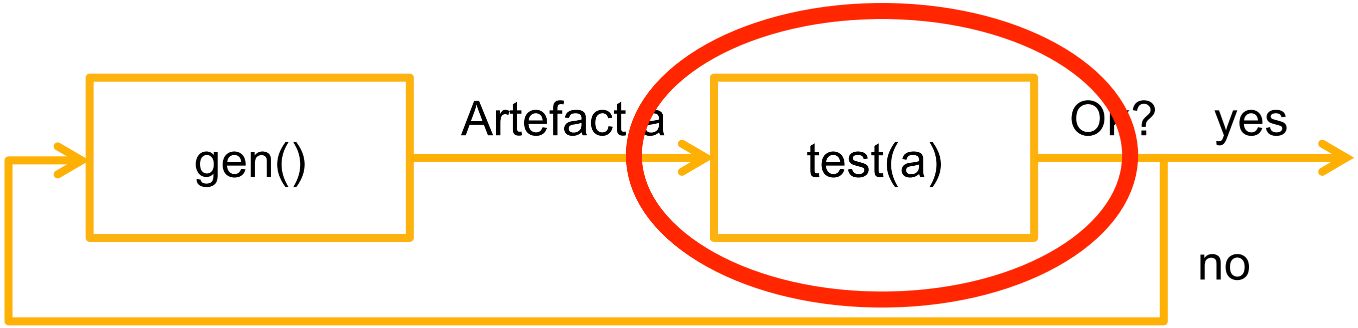
ML and DM in CC

– 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 on 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!”

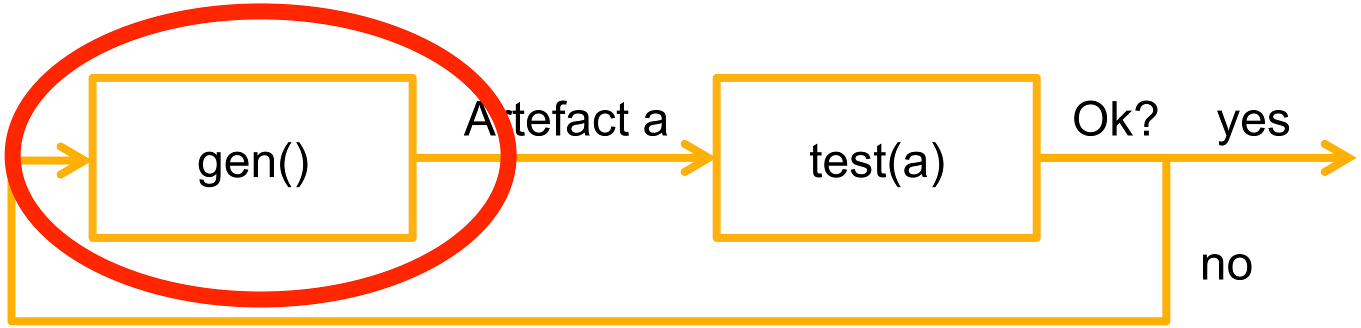
A happy image by DARCI, http://darci.cs.byu.edu/

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)

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 problems	Computational Creativity problems
Well-specified (e.g., ”induce a classifier”, ”find all frequent patterns”)	*Ill-defined, open-ended* (e.g. ”write a poem”)
Have obvious and objective success criteria (e.g. classification accuracy)	*Have subjective and non-explicit criteria* (e.g. when is a poem good?)
Success can be measured with relative ease (e.g. evaluate on test set)	*Evaluation cannot be computed easily* (e.g. ask subjects to evaluate)

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