Evolutionary Computation and Games Tutorial

Julian Togelius, Sebastian Risi, Georgios Yannakakis

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3207860

Who are we?

Julian Togelius

Sebastian Risi

Georgios N. Yannakakis

Course Agenda

Playing Games

Neuroevolution in games

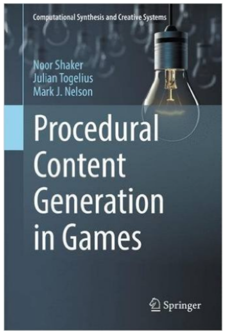
Search-based procedural content generation

Player Modelling

Objective of the Tutorial

To give you a taste of some of the many ways evolutionary algorithms (and related computational intelligence methods) can be used in games research

Want to know more?

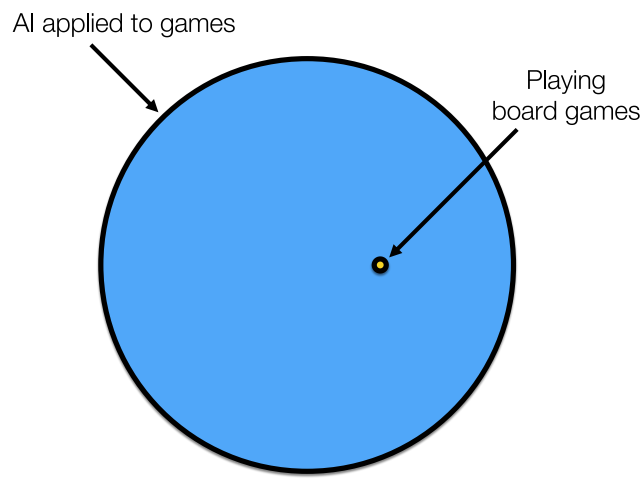


Yannakakis and Togelius: Artificial Intelligence and Games
www.gameaibook.org

Playing board games

1997: IBM vs Chess

2016: Google vs Go

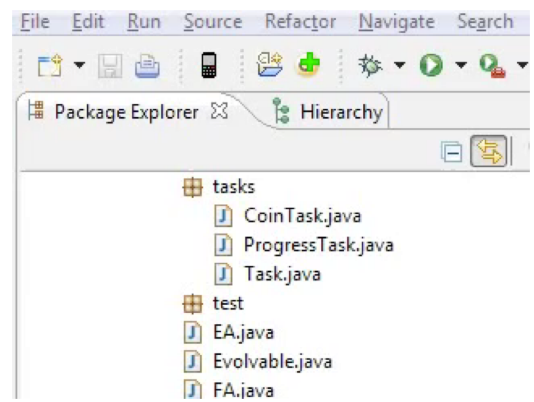


You already know about

- Tree search
- Basic idea of evolutionary computation
- Basic ideas of supervised learning and reinforcement learning, including neural nets

Time for a video

2009: ? vs Mario



Do you know A*?

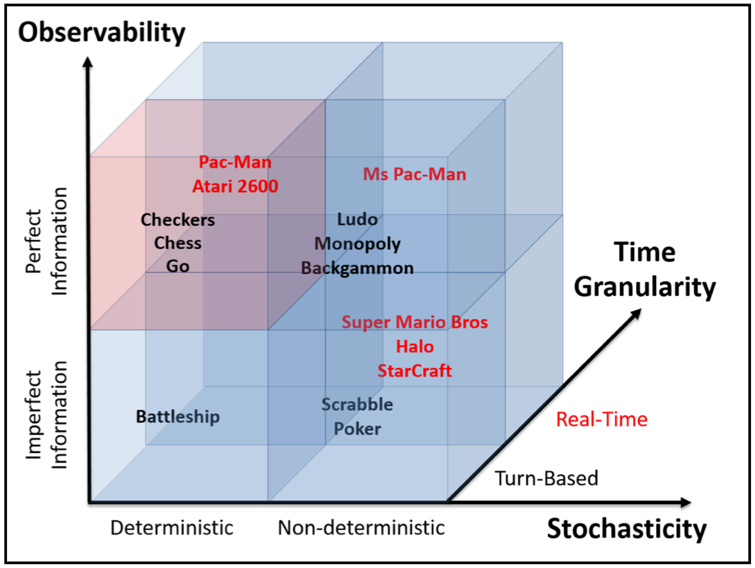
Why use AI to play games?

- Playing to win vs playing for experience
  - For experience: human-like, fun, predictable…?
- Playing in the player role vs playing in a non-player role

	Playing to win	Playing for experience
As a player	AI Benchmarking Really hard adversaries Goldfarming bots AI Playtesting (is the level beatable?)	Interesting adversaries in online games Tutorialization AI Playtesting (is the level hard/easy for a human?)
As a non-player	Really hard opponent NPCs? Team mates / allies	Most current "AI" in the game industry

Characteristics of games

- Number of players: 1, 1.5, 2, many…
  - Adversarial? Cooperative? Both?
- Stochasticity: does the same action in the same state lead to the same outcome?
- Observability: how much does the agent know?
- Action space and branching factor: how many actions?
- Time granularity: how many turns/ticks until end/reward?



Some board games

- Chess
  Two-player adversarial, deterministic, fully observable, branching factor ~35, ~70 turns
- Go
  Two-player adversarial, deterministic, fully observable, branching factor ~350, ~150 turns
- Backgammon
  Two-player adversarial, stochastic, fully observable, branching factor ~250, ~55 turns

Some video games

- Frogger (Atari 2600)
  1 player, deterministic, fully observable, bf 6, hundreds of ticks
- Montezuma's revenge (Atari 2600)
  1 player, deterministic, partially observable, bf 6, tens of thousands of ticks
- Halo
  1.5 player, deterministic, partially observable, bf ???, tens of thousands of ticks
- Starcraft
  2-4 players, stochastic, partially observable, bf > a million, tens of thousands of ticks

Applying AI to games

- How is the game state represented?
- Is there a (fast, accurate) forward model?
- Do you have time to train?
- How many games are you playing?

How to play games

- Different methods are suitable:
  - Depending on the characteristics of the game
  - Depending on how you apply AI to the game
  - Depending on why you want to make a game-playing
- There is no single best method (duh!)
- Often, hybrid architectures do best

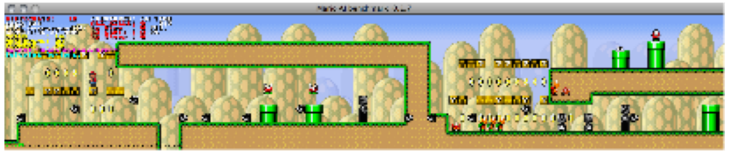
Surely, deep Q-learning is the best algorithm for game-playing!

- Planning (requires forward model)
  - Uninformed search (e.g. minimax, breadth-first)
  - Informed search (e.g. A*)
  - Evolutionary algorithms
- Reinforcement learning (requires training time)
  - TD-learning / approximate dynamic programming
  - Evolutionary algorithms
- Supervised learning (requires play traces to learn from)
  - Neural nets, k-nearest neighbors etc
- Random (requires nothing)

Informed search: A*

Not so fast!

Limitations of A*



New methods overcome limits

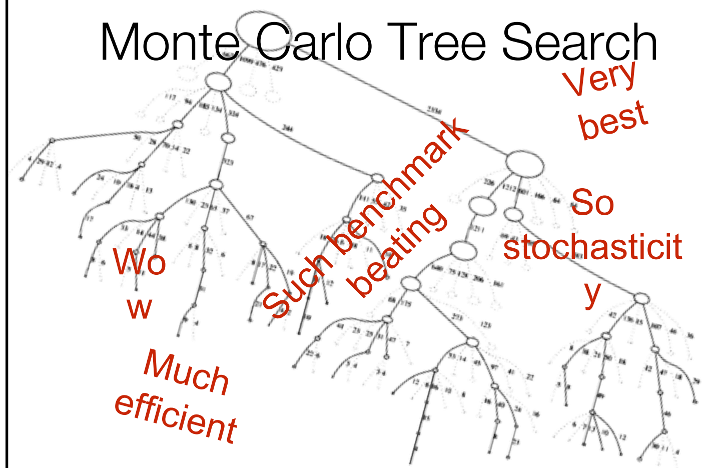
Slawomir Bojarski and Clare Bates Congdon: **REALM: A Rule-Based Evolutionary Computation Agent that Learns to Play Mario**.CIG 2010.

Monte Carlo Tree Search

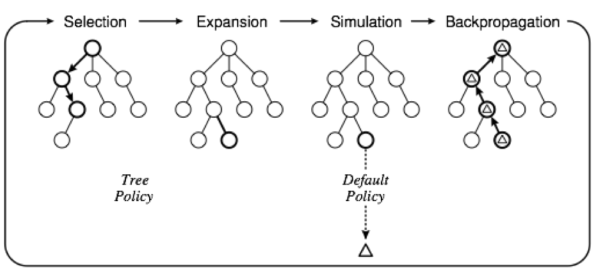
- The best new tree search algorithm you hopefully already know about
- When invented, revolutionized computer Go

Year	Program	Description	Elo
2006	INDIGO	Pattern database, Monte Carlo simulation	1400
2006	GNU GO	Pattern database, α-β search	1800
2006	MANY FACES	Pattern database, α-β search	1800
2006	NEUROGO	TDL, neural network	1850
2007	RLGO	TD search	2100
2007	MOGO	MCTS with RAVE	2500
2007	CRAZY STONE	MCTS with RAVE	2500
2008	FUEGO	MCTS with RAVE	2700
2010	MANY FACES	MCTS with RAVE	2700
2010	ZEN	MCTS with RAVE	2700

Monte Carlo Tree Search



Monte Carlo Tree Search



- Tree policy: choose which node to expand (not necessarily leaf of tree)
- Default (simulation) policy: random playout until end of game

UCB1 criterion

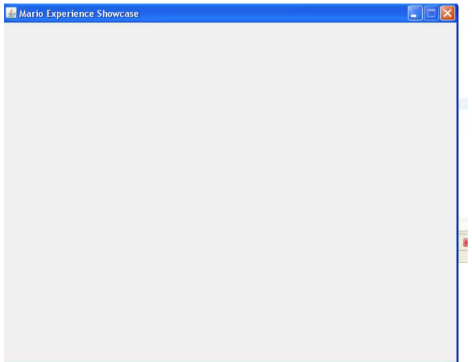
Choose which node to explore based so as to balance exploration and exploitation

Uses average reward for all children of a node, and number of visits

UCB1 (Auer et al (2002)). Choose node j so as to maximize:

$$\underbrace{\bar{X}_j}_{\text{Mean so far}} + \underbrace{\sqrt{\frac{2\log n}{T_j(n)}}}_{\text{Upper bound on variance}}$$

It's a me, Mario, again…



Noor Shaker, Julian Togelius, Georgios N. Yannakakis, Likith P. K. Satish, Vinay S. Ethiraj, Stefan J. Johansson, Robert Reynolds, Leonard Kinnaird-Heether, Tom Schumann and Marcus Gallagher (2013): **The Turing Test Track of the 2012 Mario AI Championship: Entries and Evaluation**. *IEEE Conference on Computational Intelligence and Games.*

Multi-action games

Sid Meier's Civilization

Heroes of Might and Magic

Advance Wars

Hero Academy

Acting in Hero Academy

- 5 action points each turn
- Actions: Movement, Healing, Attacking, Equipping, Swapping
- Branching factor:
  - One action: ~60
  - One turn: $60^5 = 7.78 \times 10^8 = 778{,}000{,}000$

Playing by search algorithm

- Random
- 1-ply search (Greedy on action-level)
- 5-ply (1 turn) depth-first search (Greedy on turn-level)
  - ~500,000 unique outcomes evaluated each turn (6 seconds)
  - Similar to MiniMax search depth-limited to 5 plies
- Monte Carlo Tree Search

Enormous branching factor beats MCTS

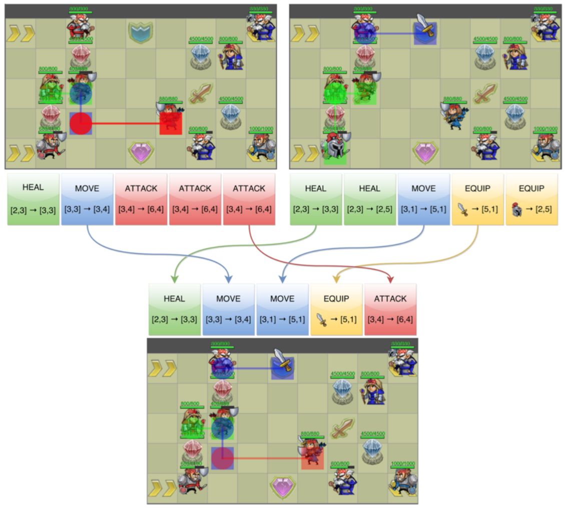
	Random	Greedy Action	Greedy Turn	MCTS
Greedy Action	100%			
Greedy Turn	100%	64.0%		
MCTS	100%	48.5%	22.0%	

Using evolution to plan?

- Some games have extremely high branching factor
  - Chess: 35
  - Go: 350
  - Civilization/StarCraft: say you have ten units, which can each take one of ten actions…
- Tree search cannot even get past the first ply
- One solution: treat the whole plan as a sequence of actions, the value of the final state as fitness…

Online Evolution

- Evolve the set of actions to take each turn
  - Chromosome is a sequence of five actions
- Simple evolutionary algorithm:
  - Population size of 100, 50% elitism, random selection of parents, uniform crossover, 10% mutation rate



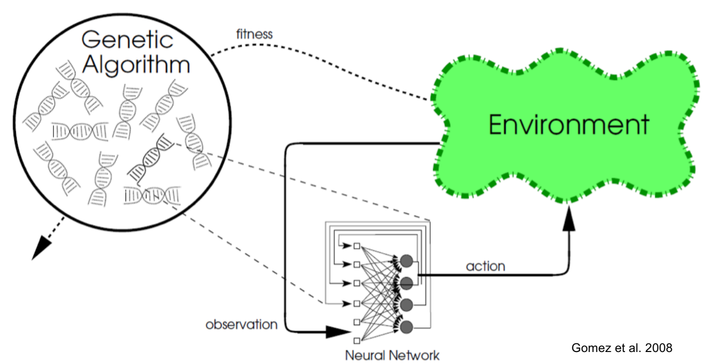
Results: wow

	Random	Greedy Action	Greedy Turn	MCTS
Online Evolution	100%	90.0%	80.5%	98%

- ~10,000 unique outcomes evaluated each turn (6 seconds)
- ~3,500 generations each turn on average

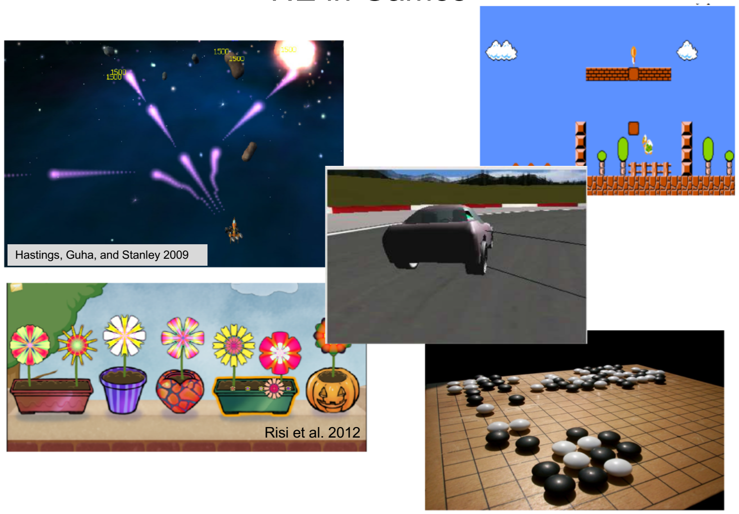
Neuroevolution in Games

Neuroevolution

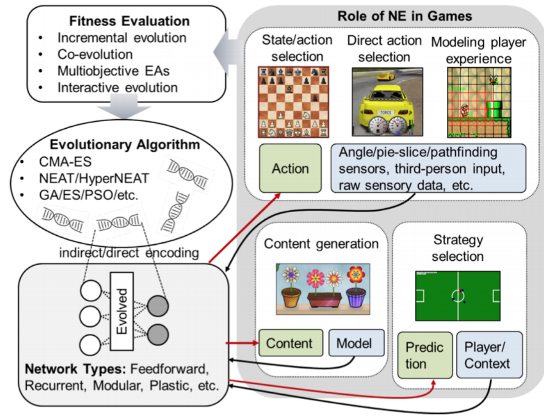


Gomez et al. 2008

NE in Games



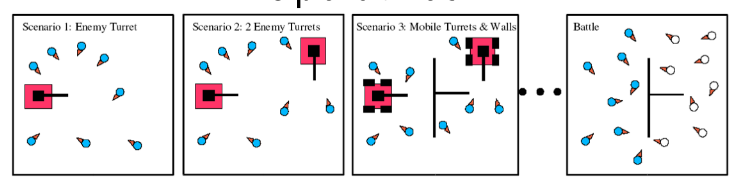
NE Role in Games



Why Neuroevolution

- Broad applicability
- Can be used for both supervised and RL problems
- Diversity
- Open-ended learning
- Enables new types of games

NERO: NeuroEvolving Robotic Operatives



- NPCs improve in real time as game is played
- Player can train AI for goal and style of play
- Each AI Unit Has Unique NN

EVOLVE
New game mechanics based on brain switching

https://www.youtube.com/watch?v=xFwjbCe5Zo8#t=22

Role of NE

- State/action evaluation
- Direct action selection
- Selection between strategies
- Modelling opponent strategy
- Content generation
- Modelling player experience

Evolving Neural Networks

- Direct encodings
  - NEAT (can evolve arbitrary topologies)
  - Evolutionary Strategies
- Indirect encodings
  - HyperNEAT
  - Compressed weight space

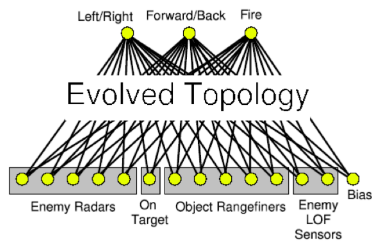
Fitness Evaluations in Games

- Incremental evolution
- Transfer learning
- Co-evolution
- Multiobjective evolution

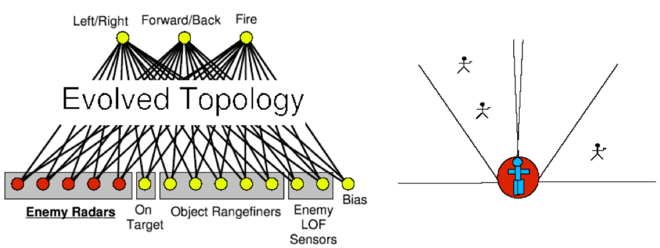
Input Representation

- Straight line sensors and pie slice sensors
- Angle sensors and relative position sensors
- Pathfinding sensors
- Third-person input
- Learning from raw sensory data

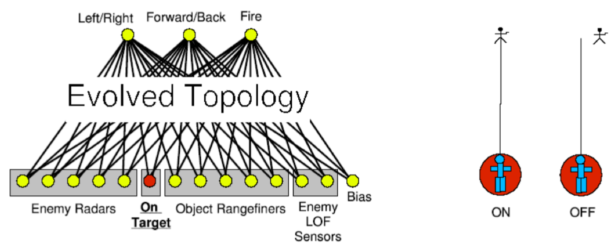
NERO Inputs and Outputs



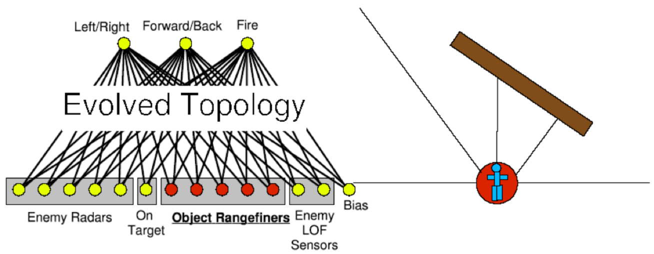
Enemy/Friend Radars



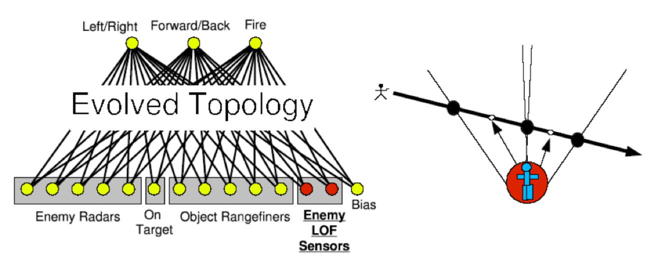
Enemy On-Target Sensor



Object Rangefinder Sensors



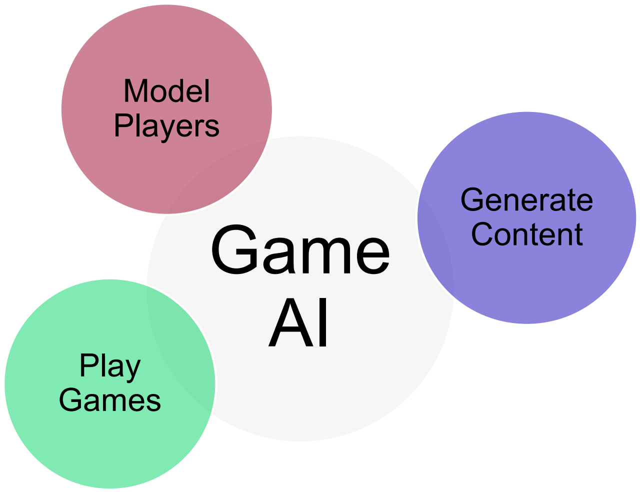
Enemy Line-of-Fire Sensors



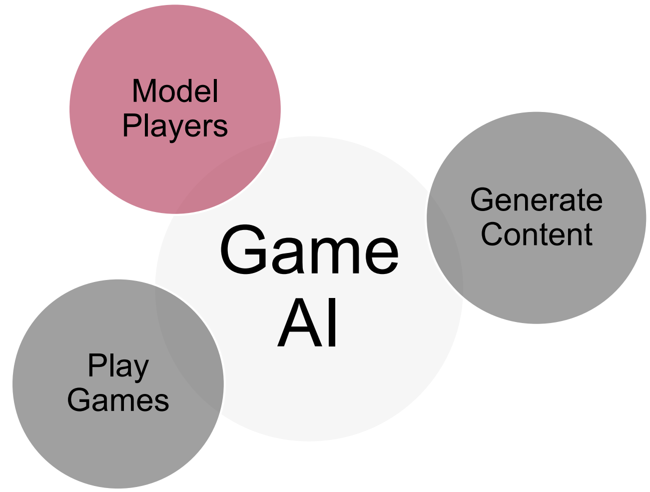
Open NE in Games Challenges

- ~~Reaching Record-beating Performance~~
- Combining evolution with other learning methods
- ~~Learning from high-dimensional/raw data~~
- General video game playing
- Combining NE with life-long learning
- Competitive and cooperative coevolution
- Fast and reliable methods for commercial games

Procedural Content Generation



G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer, 2018



G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer, 2018.

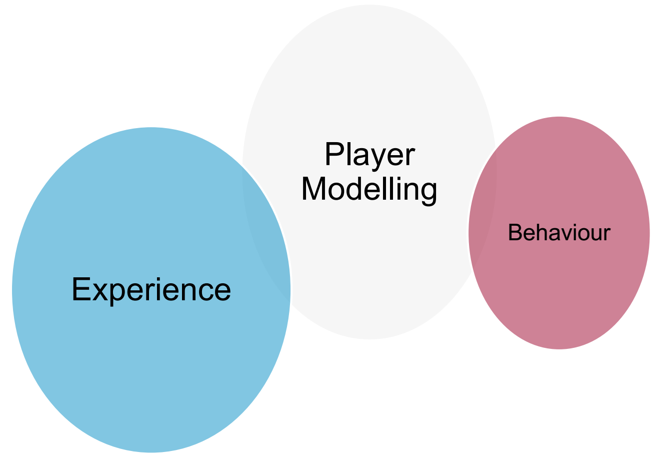
Modelling Players… **Why?**

Why model players?

- Why not?
- Machines (and some people) understand numbers
- Player Experience is the holy grail for design and development
- But most importantly because…

Why model players?

- The perfect game is tailored to you!
- We are different (and many more than before)
- If you learn to play…. it is only fair that the game learns you

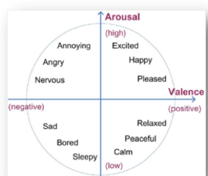


G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer, 2018

Player Experience vs Player Behavior

Experience: how you **feel during play**
- A set (a synthesis) of affective, cognitive and behavioral states
- Or else *user states*
- Emotions: Appraisal theory, …
- Cognition/Behavior: several models (e.g. BDI,…)

Behavior: what you **do during play**



Core Player Modeling Tasks for EC/ML

Supervised/Reinforcement Learning
Imitation
Prediction

Unsupervised Learning
Clustering
Association mining

Player Modelling: In a nutshell

Yannakakis et al., **Player Modeling**, in *Dagstuhl Seminar on AI/CI n in Games*, 2013

Theory (model-based)	Data (model-free)

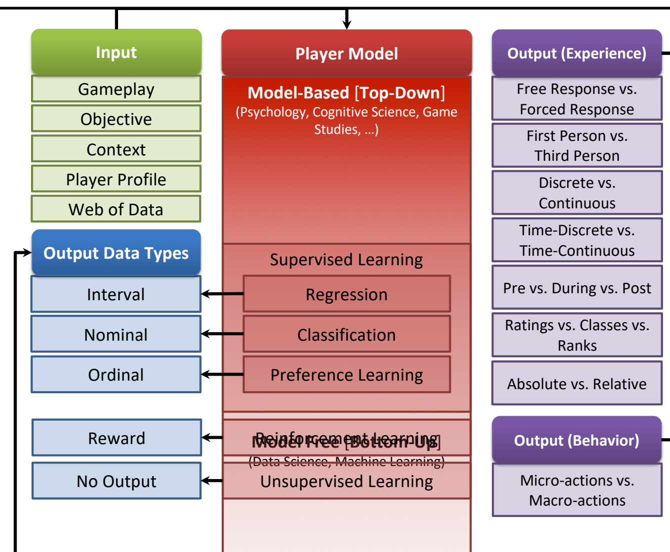
Player Modelling: In a nutshell

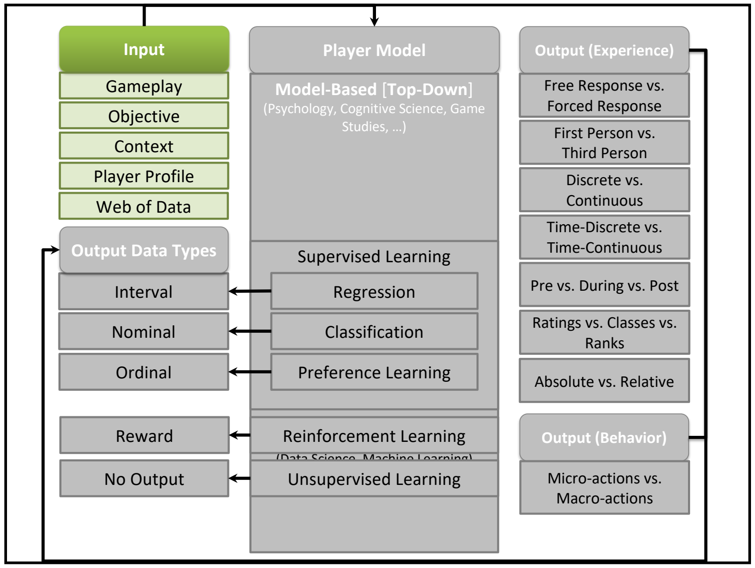
Yannakakis et al., **Player Modeling**, in *Dagstuhl Seminar on AI/CI n in Games*, 2013

Theory (model-based)	Data (model-free)

How – In a Nutshell

G. N. Yannakakis, P. Spronck, D. Loiacono and E. Andre, "**Player Modeling**," in Togelius et al., (Eds.) *Dagstuhl Seminar on Artificial and Computational Intelligence in Games*, 2013.





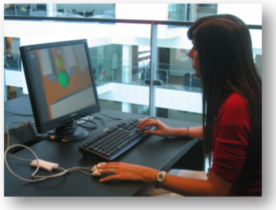
Gameplay

Gameplay Input

- Player game preferences, behavioral patterns
- **Micro** vs **macro** actions
- Examples: tactics, strategy, play patterns, clickthroughs, deaths, weapon selection, character selection, etc…
- **Pros:** real-time efficiency
- **Challenge:** we can't tell much beyond player behavior…

Objective

Objective Input

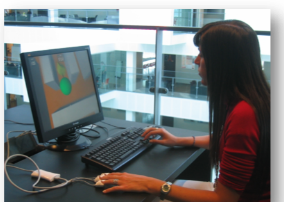


- Bodily and physiological manifestations of gameplay
- Captured via a multitude of sensors (e.g. EEG, BVP, ECG, EMG, eyetracking,…)
- **Pros:** reliable measures of user experience
- **Challenges:** many; let's see them in more detail

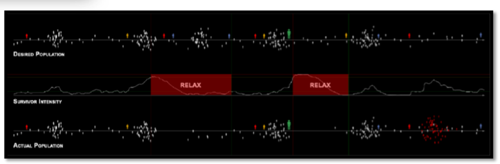
Visual Cues

- **Pros:** every laptop has a camera, off-the-shelf cheap solution, natural interaction
- **Challenges:** do we really express emotions (facially) while playing? Head-pose might be more relevant?

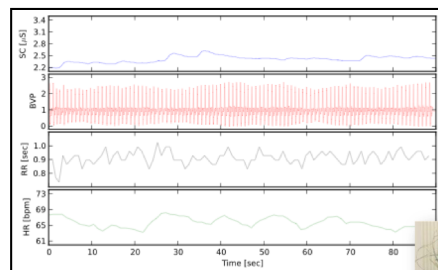
Physiology



- **Pros:** directly linked to arousal – immediate response
- **Challenges:** signal denoising/normalization; control for subjectivity of physiological responses



Measuring physiology can be obtrusive…



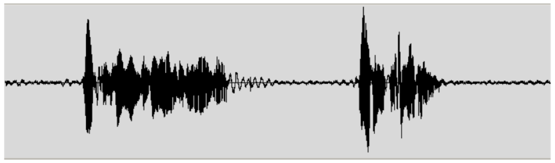
Eye-tracking

- **Pros:** you know where your player looks at/focuses on/pays attention to
- **Challenges:** practicality, lab conditions (illumination), pupilometry doesn't really work

J. Munoz, G. N. Yannakakis, F. Mulvey, D. Witzner, G. Gutierrez and A. Sanchis, "**Towards Gaze-Controlled Platform Games**,"in *Proceedings of 2011 IEEE Conference on Computational Intelligence and Games*, 2011.

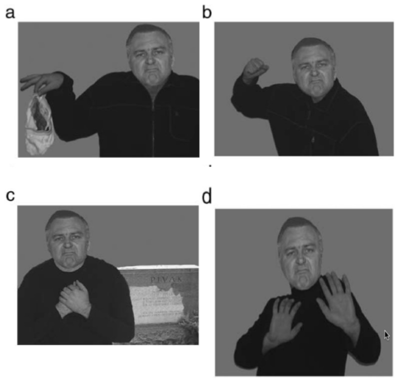
Speech

- **Pros:** speech (pitch, loudness, quality) is linked to emotions (arousal/valence); useful in game-child interaction studies
- **Challenges:** verbal cues are rare; environment noise; multi-player games



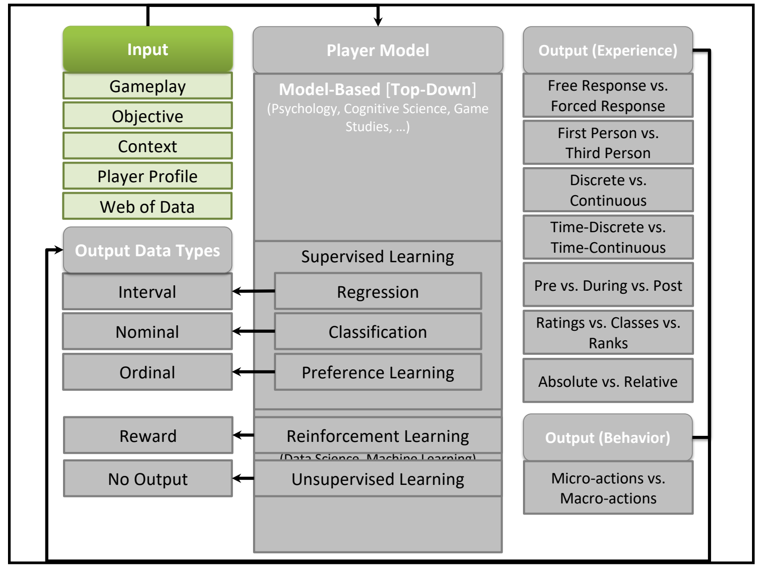
(Game) Context

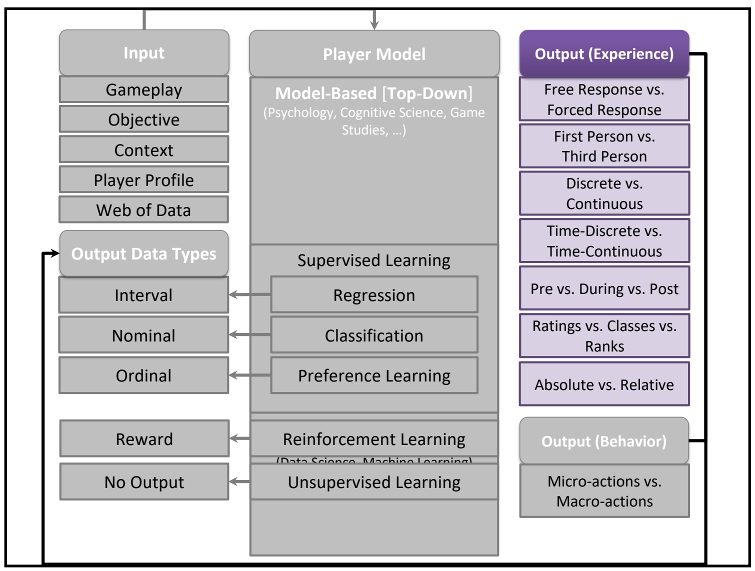
Context matters!



Player Profile

- Player profile
  - Information about ones' personality, demographics, culture, age, sex, experience with games etc…
  - In general information that does not change due (or not altered via) games – at least not that rapidly… :)
- A player profile can form additional input(s) to a player model
- Player Model vs. Player Profile : what are the differences?
  - A **profile** is built on *static* data and not influenced by game
  - A **model** is built on *dynamic* data from the gaming interaction and is (temporally) influenced by the game





Output = Annotation

- Annotation is the labelling of experience (states/values/ranks etc.)
- This is ultimately the *ground truth* of experience
- This is the training signal for your computational models

Key questions

- Who annotates?
- When?
- How often?
- How?

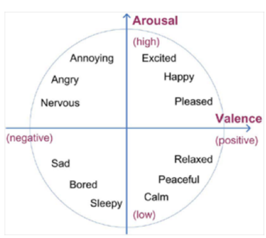
Who annotates?

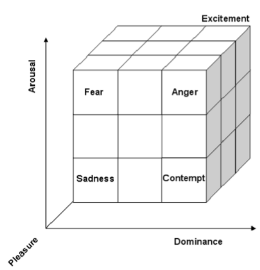
- Third Person
  - Usually a domain expert (game designer) or a psychologist
- First Person
  - The person actually experiencing the emotion/affect

	Third person	First person
+	• Expert knowledge	• Reported true experience
-	• Assumptions about the true emotion • Reporting effects	• Self-deception • Reporting effects • No expert knowledge

How is Experience Best Represented?

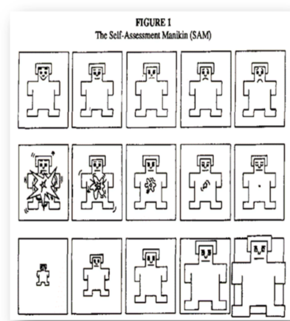
- Discrete states (e.g. fun, engagement, frustration)
- Continuous dimensions (e.g. arousal and valence)

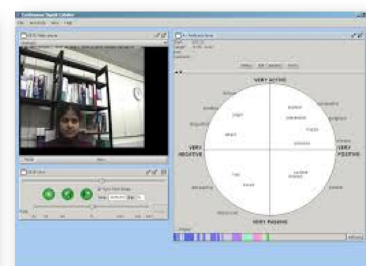




How Often to Annotate?

- Time-Discrete (e.g. self-assessment manikin)
- Time-Continuous (e.g. FeelTrace, *AffectRank*)





How often to annotate?

- Depends on
  - Application (speed of interaction: e.g. games vs. movies vs. e-learning apps)
  - Signal (e.g. physiology is slower than body movement and speech)
- No gold standard

A note about time and self-report!

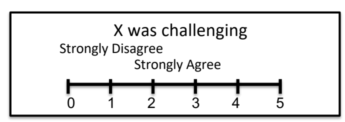
- Self-reports are time-dependent
- Real experience vs. Post-experience
  - Few seconds → Real experience
  - Few minutes/hours → *Episodic* memory (context retrieval)
  - More → *Semantic* Memory (beliefs)

- NB. The gap between our **memory of experience** and our **experience** is more prominent when we report unpleasant emotions such as anger, sadness and tension. Also: The experience felt near the end of a session (e.g. a game level or a game) affects our report – aka ***peak-end rule***.

Which Annotation (Data) Type?

- Scalar (a value of arousal, valence, SAM, Geneva wheel, Likert scale) – **Rating**
- Binary value or a class – **Class**
- Preference between two or more options – **Rank**

Rating



Examples: Geneva Wheel, SAM, Likert Scales, PAD values

Class

Examples:
- This facial expression is **happy**! (Eckman)
- Arousal values higher than 0.6 belong to class **aroused**
- This skin conductance peak denotes **stress**

Rank

X is **more/less** than **Y**

challenging
frustrating
arousing
boring
fearful
…

- Requires at least two instances!
- N-Alternative Forced Choice (4-AFC is popular)

☐ X is **more/less** frustrating than Y
☐ Both are **equally** frustrating
☐ **Neither** is frustrating

Use ratings (e.g. Likert items, SAM, etc.)?

Yannakakis and Hallam, **Rating vs. Preference: A comparative study of self-reporting**, ACII, 2011
Yannakakis and Martinez, **Ratings are Overrated!** *Frontiers in Human-Media Interaction*, 2015

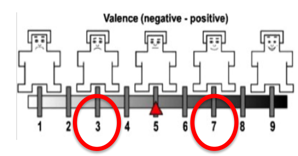
X is frustrating
Strongly Disagree
Strongly Agree
0 1 2 3 4 5

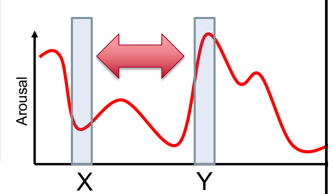
Ordinal data (ratings) is **not** interval…

☐ X is **more/less** frustrating than Y
☐ Both are **equally** frustrating
☐ **Neither** is frustrating

The **ordinal** (relative) approach

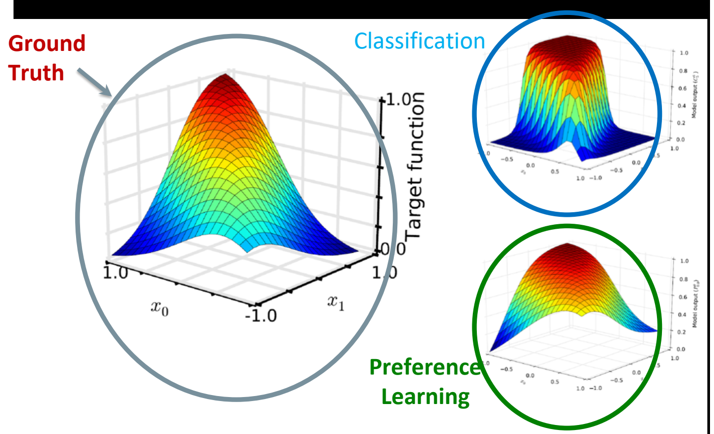
Yannakakis, Cowie, Busso, **The Ordinal Nature of Emotions**, ACII, 2017 **[Best Paper Award]**





Classified Ratings vs Ranked Ratings

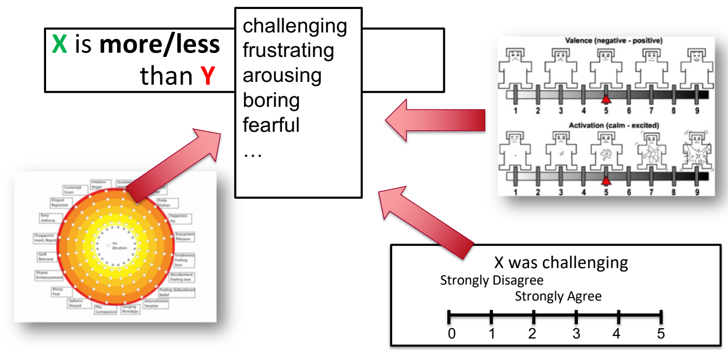
Martinez, Yannakakis and Hallam, **Don't classify ratings of affect; Rank them!** *IEEE Trans. on Affective Computing*, 2014



Ratings (and Classes) vs. Ranks

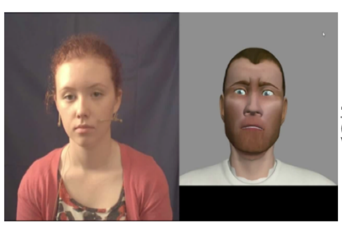
Martinez, Yannakakis and Hallam, **Don't classify ratings of affect; Rank them!** *IEEE Transactions on Affective Computing*, 2014

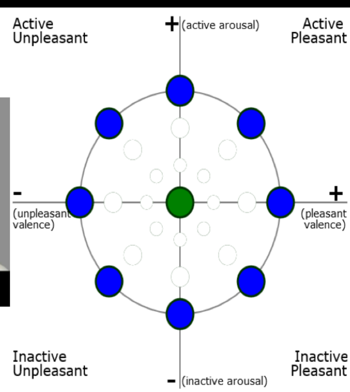
Treat ratings as **ordinal data** : **rank them**!



AffectRank: Ordinal emotion annotation

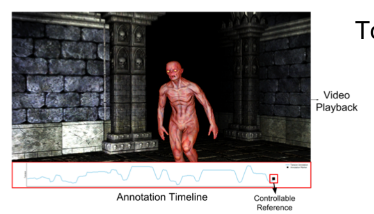
Yannakakis and Martinez, **Grounding Truth via Ordinal Annotation,** *Affective Computing and Intelligent Interaction*, 2015.





RankTrace: Relative Unbounded Annotation

Lopes et al., **RankTrace: Relative and Unbounded Affect Annotation** *Affective Computing and Intelligent Interaction*, 2017.



Tools @ emotion-research.net

Annotation – Take away messages

- 1st vs. 3rd person: depends on the application
- Try to get reports as close to the *true experience* as possible (time-wise)
- No report is ideal (they suffer from biases)
- Annotate experience as **ranks** whenever possible
- If ratings are available
  - Regression of ratings is **fundamentally wrong**
  - Do not convert them to classes – it will cost you on model performance
  - **Convert them to ranks** (treat them as ordinal scales)!

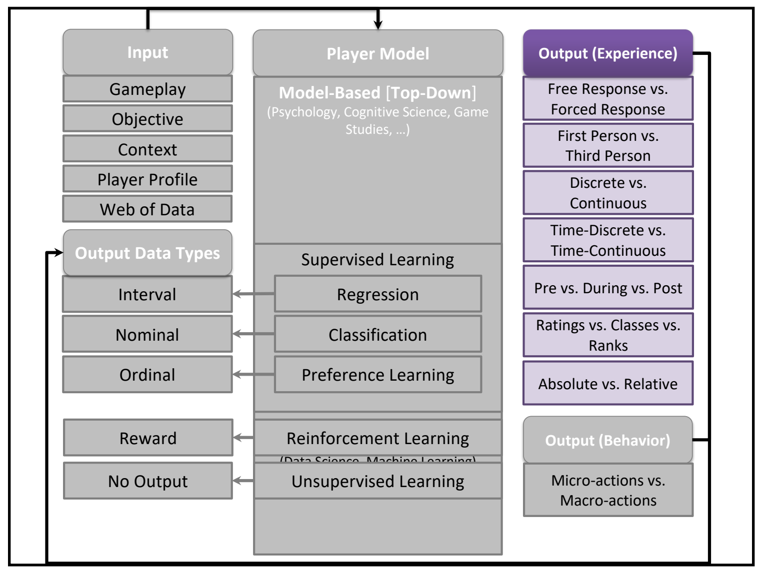
Subjective Notions Summary

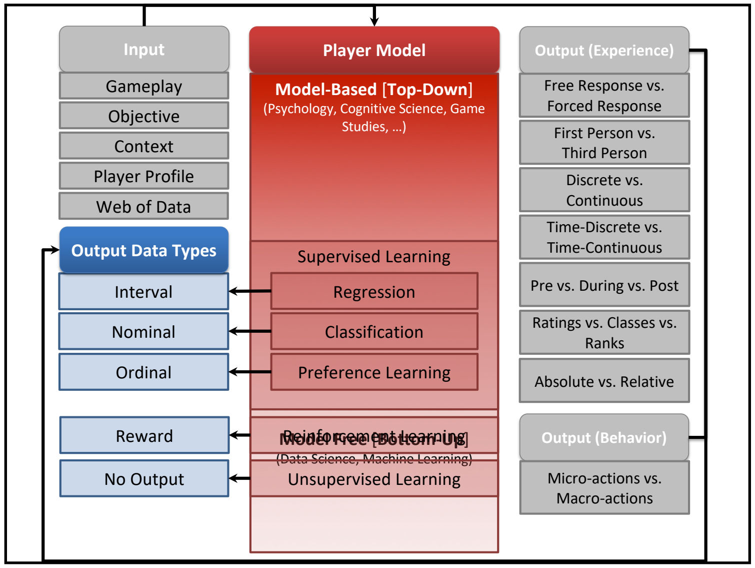
- Don't try **this**
- Wasteful Info due to
  - Scale-bias
  - Personal-bias
  - Labels are **NOT** numbers
  - High inconsistency (randomness)
  - ...

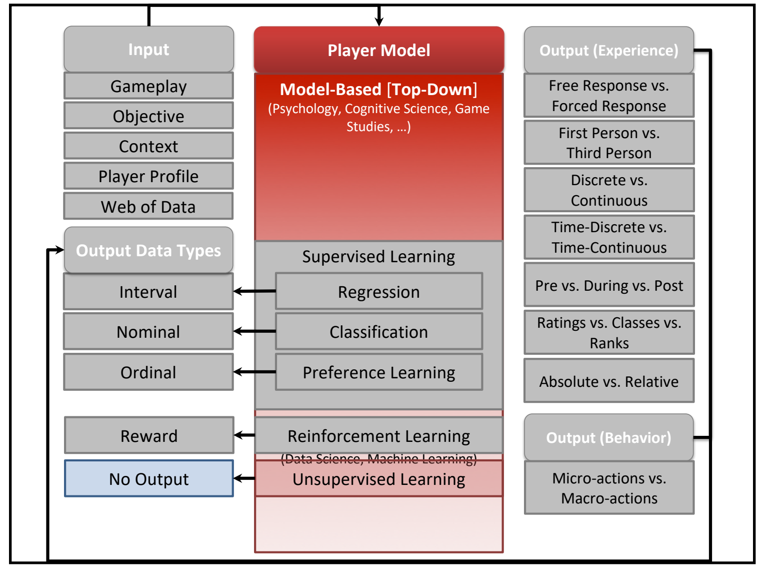
Subjective Notions Summary

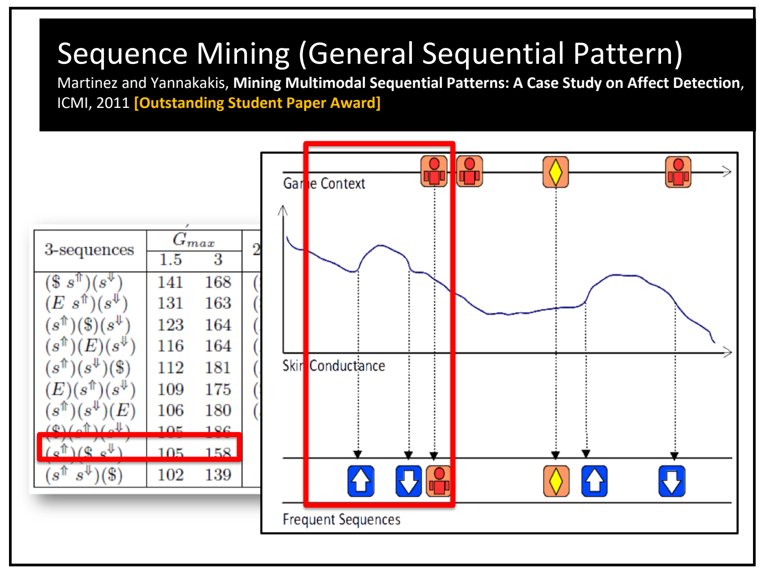
Try out something like this instead:

- ☐ I like **Julian's** class more/less than **Georgios'** class
- ☐ I like them both equally
- ☐ I like neither

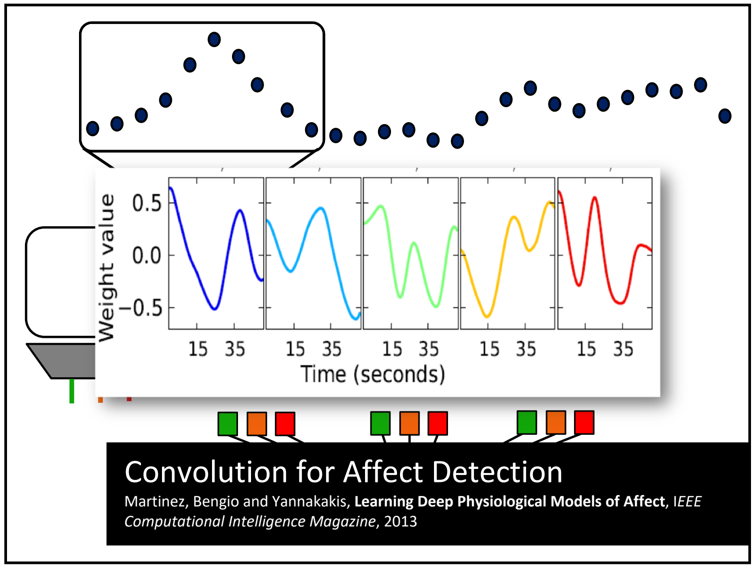




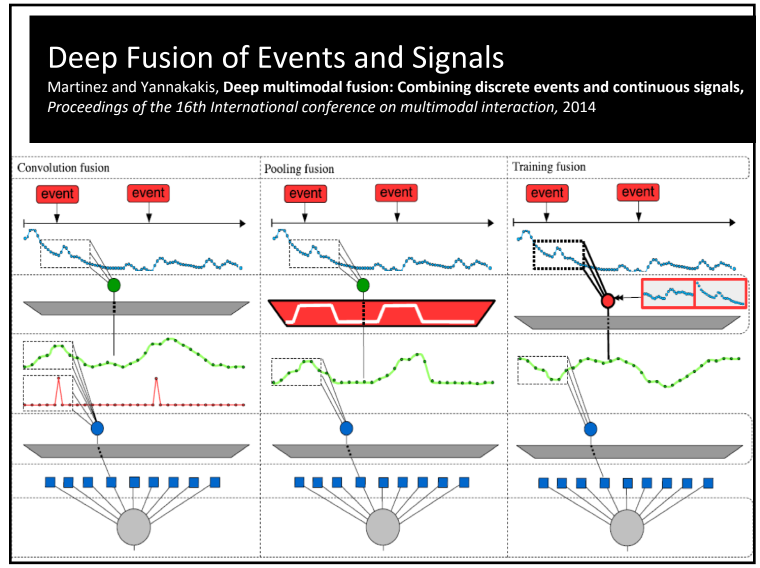




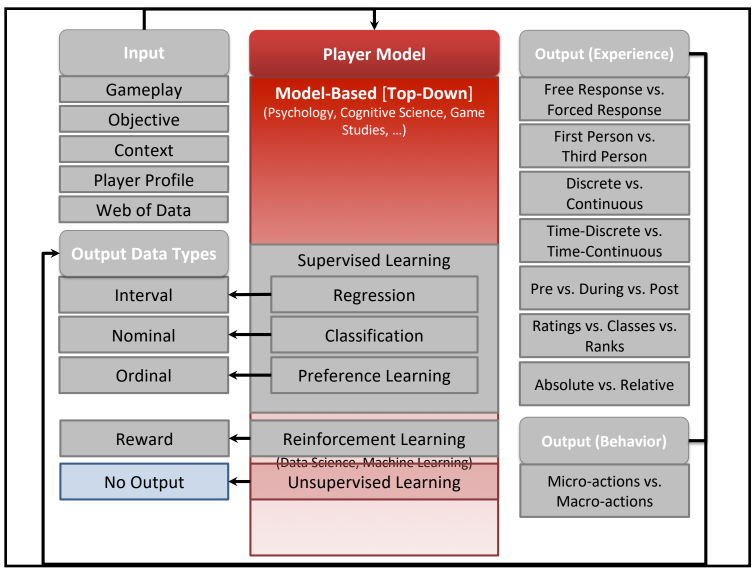
3-sequences	G_{max} 1.5	G_{max} 3
$(\$\ s^{\Uparrow})(s^{\Downarrow})$	141	168
$(E\ s^{\Uparrow})(s^{\Downarrow})$	131	163
$(s^{\Uparrow})(\$)(s^{\Downarrow})$	123	164
$(s^{\Uparrow})(E)(s^{\Downarrow})$	116	164
$(s^{\Uparrow})(s^{\Downarrow})(\$)$	112	181
$(E)(s^{\Uparrow})(s^{\Downarrow})$	109	175
$(s^{\Uparrow})(s^{\Downarrow})(E)$	106	180
$(\$)(s^{\Uparrow})(s^{\Downarrow})$	105	186
$(s^{\Uparrow})(\$\ s^{\Downarrow})$	105	158
$(s^{\Uparrow}\ s^{\Downarrow})(\$)$	102	139



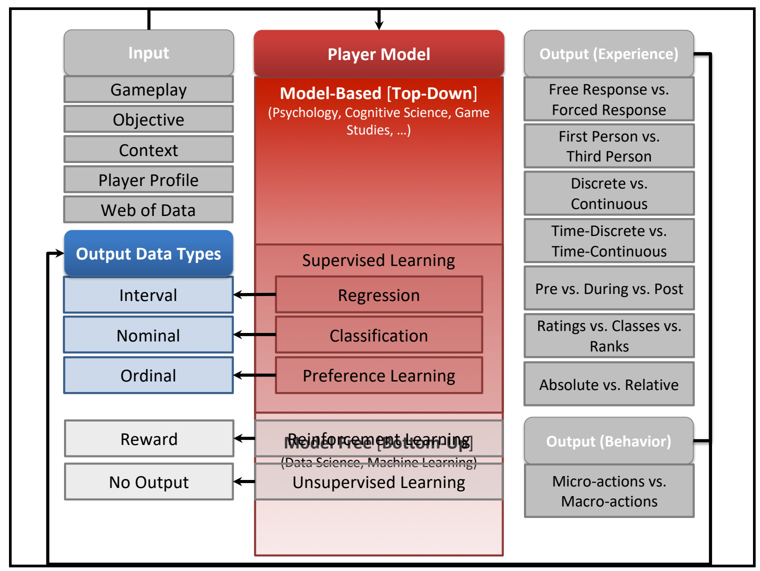
Weight value
0.5
0.0
−0.5
15 35
Time (seconds)
Convolution for Affect Detection
Martinez, Bengio and Yannakakis, Learning Deep Physiological Models of Affect, IEEE Computational Intelligence Magazine, 2013



Deep Fusion of Events and Signals
Martinez and Yannakakis, Deep multimodal fusion: Combining discrete events and continuous signals, Proceedings of the 16th International conference on multimodal interaction, 2014
Convolution fusion
Pooling fusion
Training fusion
event

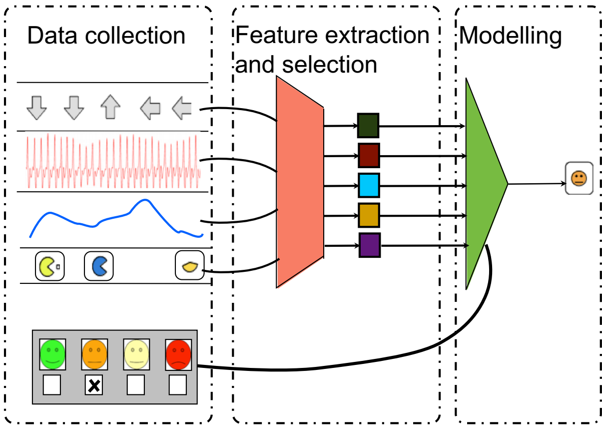


Input
Gameplay
Objective
Context
Player Profile
Web of Data
Output Data Types
Interval
Nominal
Ordinal
Reward
No Output
Player Model
Model-Based [Top-Down]
(Psychology, Cognitive Science, Game Studies, …)
Supervised Learning
Regression
Classification
Preference Learning
Reinforcement Learning
(Data Science, Machine Learning)
Unsupervised Learning
Output (Experience)
Free Response vs. Forced Response
First Person vs. Third Person
Discrete vs. Continuous
Time-Discrete vs. Time-Continuous
Pre vs. During vs. Post
Ratings vs. Classes vs. Ranks
Absolute vs. Relative
Output (Behavior)
Micro-actions vs. Macro-actions



Input
Gameplay
Objective
Context
Player Profile
Web of Data
Output Data Types
Interval
Nominal
Ordinal
Reward
No Output
Player Model
Model-Based [Top-Down]
(Psychology, Cognitive Science, Game Studies, …)
Supervised Learning
Regression
Classification
Preference Learning
Reinforcement Learning
Model-Free [Bottom-Up]
(Data Science, Machine Learning)
Unsupervised Learning
Output (Experience)
Free Response vs. Forced Response
First Person vs. Third Person
Discrete vs. Continuous
Time-Discrete vs. Time-Continuous
Pre vs. During vs. Post
Ratings vs. Classes vs. Ranks
Absolute vs. Relative
Output (Behavior)
Micro-actions vs. Macro-actions

Modelling Steps



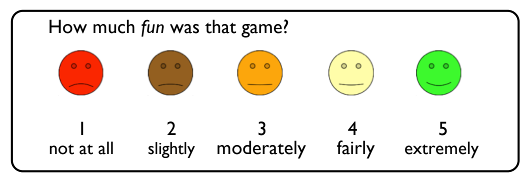
Supervised learning for modelling experience

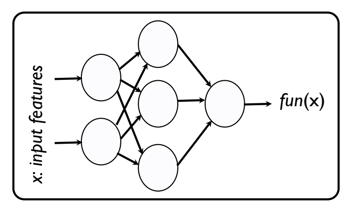
- The output of the model is the *estimated experience*
- The **ground truth** is given by annotated experience given as
  - Nominal values (e.g. sample A is frustrated)
  - Numerical values (e.g. sample A is 0.86 frustrated)
  - Ordinal values
    - Ranks (e.g. sample A is more frustrating than sample B)
    - Ratings (e.g. sample A is 'extremely frustrating' and sample B is 'fairly frustrating'

Which Training Method?

Preference learning   Classification   Regression

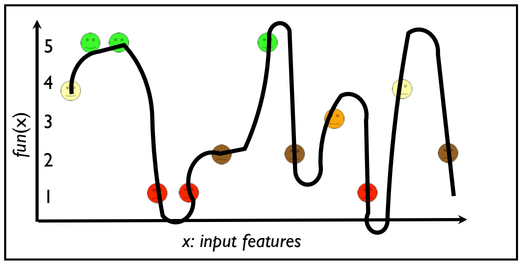
Example: modelling *fun* ratings



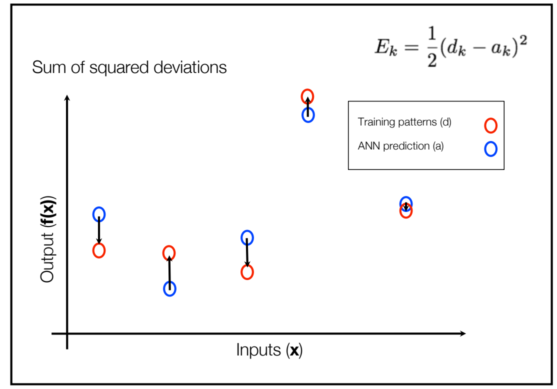


The bad: Regression

- Remember: ratings are **NOT** numbers!
  - Not everyone uses the scales in the same way
  - Items in the scale are not equidistant

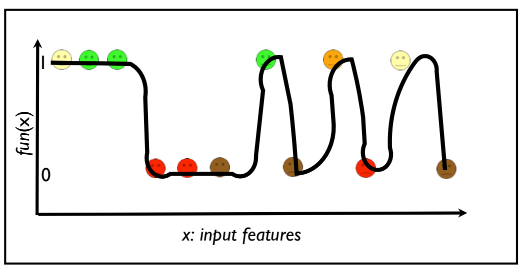


Regression with backpropagation



The ugly: Classification

- Converting ratings into classes eliminates a lot of information and it can introduce biases



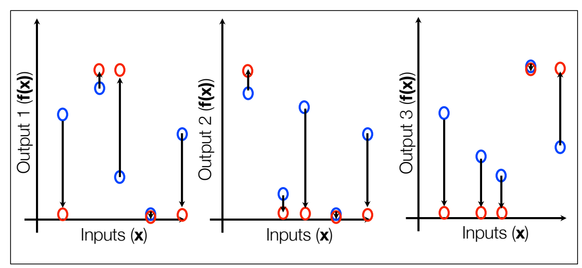
H. P. Martinez, G. N. Yannakakis and J. Hallam, "**Don't Classify Ratings of Affect; Rank them!**," *IEEE Transactions on Affective Computing,* 2014

Classification with backpropagation

- Same as regression but with one output per class

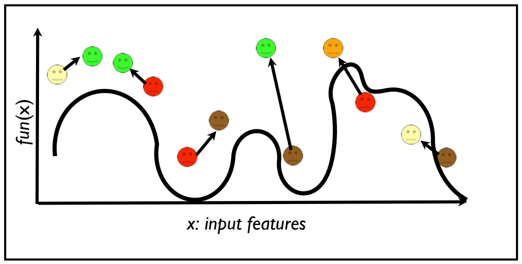
Sum of squared deviations $E_k = \frac{1}{2}(d_k - a_k)^2$

Training patterns (d)
MLP prediction (a)



The **good**: Preference Learning

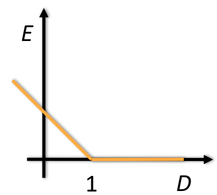
- Learn only the ordinal relations
- Valid whenever the annotator is consistent on her use of the scale

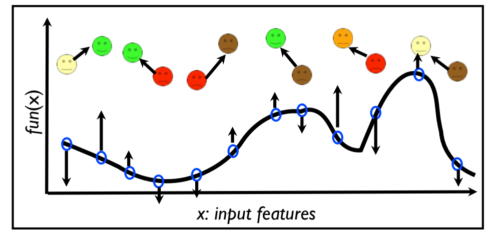


(Deep) Preference Learning with BP

- Error function maximizes the distance between the output for the preferred sample (*d*[A]) and the output for the non preferred sample (*d*[B])

$$E = max(0, 1-(d^A - d^B)) \qquad \frac{\partial E}{\partial w_{ij}} = \begin{cases} -\frac{\partial d^A}{\partial w_{ij}} + \frac{\partial d^B}{\partial w_{ij}} & \text{, if } d^A - d^B < 1 \\ 0 & \text{, otherwise} \end{cases}$$



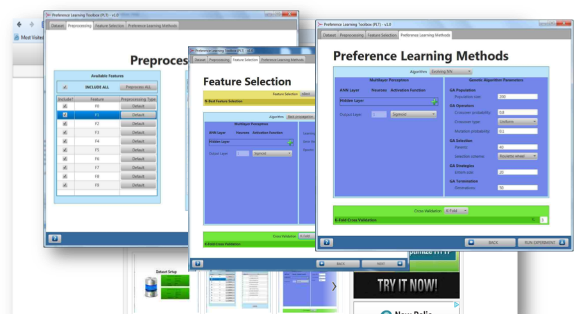


(Deep) Preference Learning beyond BP

- The concept of **learning from pairs of preferences** can be implemented in most supervised learning methods by adapting the error/fitness function
  - NeuroEvolution
  - SVMs (RankSVM)
  - Decision Trees
  - …

An open-source Preference Learning Toolbox

Farrugia, Martinez and Yannakakis, **The Preference Learning Toolbox**, arXiv preprint, 2015

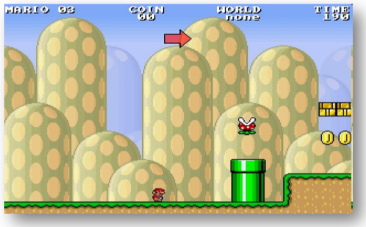


https://sourceforge.net/projects/pl-toolbox/

Examples

An Example: Player Experience Modeling in Super Mario

- 327 subjects (1308 games)
- Input: Playing Behavior and Content Features
- Output: Engagement, Frustration, Challenge self-reported ranks (pairwise) of short games
- ANN trained via **neuroevolutionary preference learning**
- Player experience model accuracy: **73-92**%

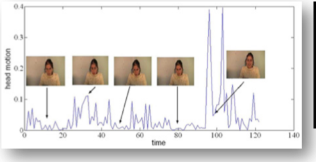


The Super Mario Example: Player Experience Modelling (Visual + Behavioral)

Shaker, Asteriadis, Yannakakis and Karpouzis, **Fusing Visual and Behavioral Cues for Modelling User Experience in Games,** *IEEE Trans. on Systems, Man and Cybernetics* (B), 2013

- 58 subjects (28 Male) – Played: 167 game pairs
- Player Experience model (ANN) accuracy: **88-92**%
- Input: Visual features and behavioral features
- States (Output) : Engagement, Frustration, Challenge

The Super Mario Example: Head Expressivity Features (ANN Input)



Category	Feature	Description
		Head Movement Features throughout whole sessions
Mean Head Movement	Avg	Absolute first order derivative of Head Pose Vector
	OA	Overall Activation
	SE	Spatial Extent
	TE	Temporal Expressivity parameter
	PO	Energy Expressivity parameter
	FL	Fluidity
	$M_{horizontal}$	Median value for horizontal head rotation
	$M_{vertical}$	Median value for vertical head rotation
		Head Movement Features during gameplay events
Visual Reaction Features	Avg_a	Absolute first order derivative of Head Pose Vector when the gameplay event, a occur
	OA_a	Overall Activation when the gameplay event, a occur
	SE_a	Spatial Extent when the gameplay event, a occur
	TE_a	Temporal Expressivity parameter when the gameplay event, a occur
	PO_a	Energy Expressivity parameter when the gameplay event, a occur
	FL_a	Fluidity when the gameplay event, a occur
	M_a	Median value for head rotation norm when the gameplay event, a occur

Category	Feature	Description
		Content (Level) Features
Content (Level) Features	G	Number of gaps
	$\bar{G}_w$	Average width of gaps
	E	Number of enemies
	E_p	Placement of enemies
	N_w	Number of powerups
	B	Number of boxes
		GamePlay Features
Time	t_{comp}	Completion time
	$t_{lastLift}$	Playing duration of last life over total time spent on the level
	t_{duck}	Time spent ducking (%)
	t_{jump}	Time spent jumping (%)
	t_{left}	Time spent moving left (%)
	t_{right}	Time spent moving right (%)
	t_{run}	Time spent running (%)
	t_{small}	Time spent in Small Mario mode (%)
	t_{big}	Time spent in Big Mario mode (%)
Interaction with items	n_{coins}	Free coins collected (%)
	$n_{coinBlocks}$	Coin blocks pressed or coin rocks destroyed (%)
	$n_{powerups}$	Powerups pressed (%)
	n_{boxes}	Sum of all blocks and rocks pressed or destroyed (%)
Interaction with enemies	$k_{cannonFlower}$	Times the player kills a cannonball or a flower (%)
	$k_{goombaKoopa}$	Times the player kills a goomba or a koopa (%)
	k_{stomp}	Opponents died from stomping (%)
	$k_{unleash}$	Opponents died from unleashing a turtle shell (%)
Death	d_{total}	Total number of deaths
	d_{cause}	Cause of the last death
Miscellaneous	n_{mode}	Number of times the player shifted the mode between: Small, Big, and Fire
	n_{jump}	Number of times the jump button was pressed
	n_{gJump}	Difference between the # of gaps and the # of jumps
	n_{duck}	Number of times the duck button was pressed
	n_{state}	Number of times the player changed the state between: standing still, run, jump, moving left, and moving right

The Super Mario Example: Gameplay/Content Features (ANN Input)

The Super Mario Example: The Annotated Experience (ANN output)

Shaker, Asteriadis, Yannakakis and Karpouzis, **Fusing Visual and Behavioral Cues for Modelling User Experience in Games,** *IEEE Trans. on Systems, Man and Cybernetics* (B), 2013

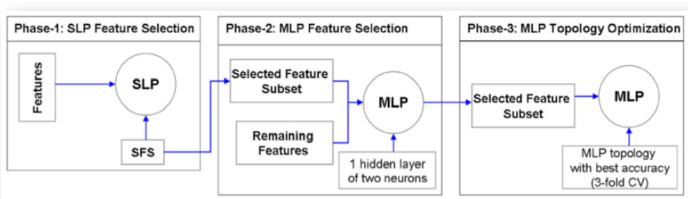
- Three Player experience states modelled:
  - *Engagement*, *Frustration*, *Challenge*
- Player Experience is self-reported (post-experience) via a 4-alternative forced choice questionnaire:

☐ **Game A** is more/less **engaging** than **B**
☐ Both are equally **engaging**
☐ Neither is **engaging**

The Super Mario Example: The Modelling Approach

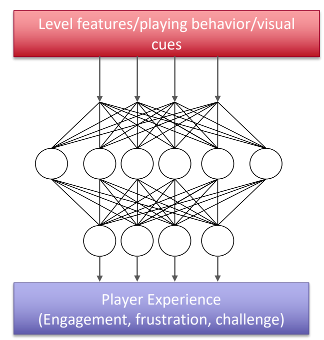
Shaker, Asteriadis, Yannakakis and Karpouzis, **Fusing Visual and Behavioral Cues for Modelling User Experience in Games,** *IEEE Trans. on Systems, Man and Cybernetics* (B), 2013

- NeuroEvolutionary Preference Learning: SLPs and MLPs
- Feature Selection: Sequential Forward Selection (SFS)



The Super Mario Example: The Modelling Approach

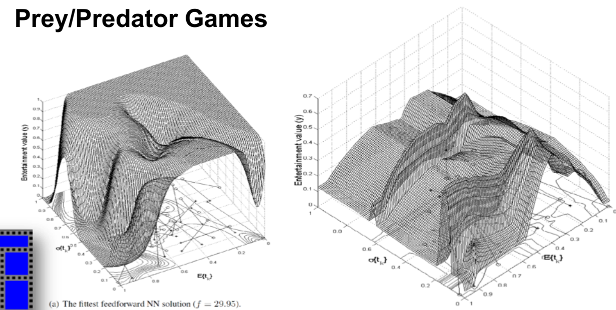
Shaker, Asteriadis, Yannakakis and Karpouzis, **Fusing Visual and Behavioral Cues for Modelling User Experience in Games,** *IEEE Trans. on Systems, Man and Cybernetics* (B), 2013

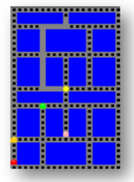


More Preference Learning Examples

Entertainment Modelling

Prey/Predator Games





(a) The fittest feedforward NN solution ($f = 29.95$).

G. N. Yannakakis, and J. Hallam, "**Modeling and Augmenting Game Entertainment through Challenge and Curiosity**," *International Journal on Artificial Intelligence Tools,* vol. 16, issue 6, pp. 981-999, December 2007.

Entertainment Modelling

Playware Playground

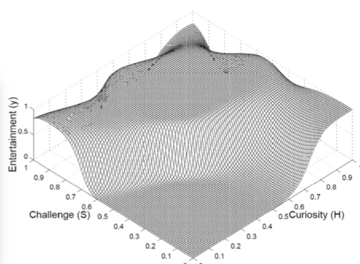


Fig. 4. Fittest feedforward NN ($f = 22.82$).

G. N. Yannakakis, and J. Hallam, "**Modeling and Augmenting Game Entertainment through Challenge and Curiosity**," *International Journal on Artificial Intelligence Tools,* vol. 16, issue 6, pp. 981-999, December 2007.

Takeaway and Future

- We encode information in **relative** terms
- Machine learning/EC should probably do so too!
- **Preference learning** is a way!
- Do regression and classification become irrelevant?