Beauty and the Code:

Does Compressive Information Affect the Perception of Beauty ?

Mostafa Mohsenvand (mmv@mit.edu)
MIT Media Lab, Social Machines Lab

Abstract

This project is an attempt towards a better understanding of computational cognitive science of beauty/interestingness perception (BIP). Here we will try to examine and improve a theory called Coherence Progress (or Compression Progress) by Jürgen Schmidhuber which puts forward an information theoretic framework for BIP and has been used in optimization of intelligent agents (Schmidhuber (2008, 2010); Schaul, Pape, Glasmachers, Graziano, and Schmidhuber (2011)). In order to come up with a computationally expressible model, we took mathematical formulas as the generative algorithms of 2-dimensional plots and examined changes in beauty/interestingness rating (BIR) of human subjects with respect to perceived complexity of the formulas. First we presented tens of formulas to more than a 100 human subject and modeled their formula complexity perception (FCP) using bayesian and evolutionary approaches. Then we used the obtained models to generate 2d plots of functions with a broad range of FCP. Next we showed the plots with two different kinds of formulas (simple and complex) to more than 200 human subject and gathered their BIR before and after seeing the formulas. By analyzing the results, we could confirm two hypotheses: (i) Rating of Interestingness of a plot increases after showing a mathematical formula that is used to generate the plot and (ii) Rating of Interestingness of a plot after seeing a simple formula is higher than its rating after seeing a complex formula. Both results are in agreement with Schmidhuber's theory. At the end, we modeled the relationship of change in BIR with FCP and speculated an improved version of Schmidhuber's work.

Keywords: Beauty Perception, Complexity Perception, Algorithmic Complexity, Bayesian Regression, Genetic Programming

Introduction

History of computational cognitive science has shown that some of complex cognitive and behavioral characteristics of us humans and animals can be understood and mimicked effectively through simple computational models.

Listening to an exquisite piece of music or seeing a beautiful face is rewarding for us humans but why ? Despite its huge marketing and industrial applications, there are not many accurate models for beauty/interestingness perception (BIP). One of the related areas of research is the study of facial attractiveness. It has been shown that perception of facial beauty has a direct relationship with symmetry, similarity (like-attracts-like) and perhaps more generally "averageness" (proximity to the average face, See Figure 1) of the face Langlois and Roggman (1990); Rubenstein, Langlois, and Roggman (2002); Perrett et al. (1999).

Compression Progress

In a 1991 paper (Jurgen Schmidhuber (1991)), Schmidhuber proposed an information theoretic approach for mathematical formulation of BIP. His theory suggests that we experience beauty or interestingness when we sense something *informative or novel* which is *highly compressible*. In other words,

Figure 1: Average face for different countries, source: faceresearch.org

our brain is interested in patterns that are meaningful (with respect to our previous knowledge) and have a small algorithmic complexity. For example, Schmidhuber reasons that the fact that we find jokes "funny" is because they express a meaningful social or human related concept in a very compact way; Or the reason we find symmetric, average and similar faces more beautiful is because they are easier to compress given our learned model of faces in our brain.

Motivation For the Experiments

How can we test these ideas? First of all, it is hard(if not impossible) to calculate or even clearly define the Kolmogorov complexity of sensible objects. Secondly, how do we make sure that subjects of our experiments really absorb the information from observing an object?

In order to overcome these important hurdles, we have come up with two simple assumptions that can help us design an experiment:

(i) We can almost certainly come up with the most simplified version of a conventional mathematical formula that does not include special functions; moreover, if we visualize the formula, the Kolmogorov complexity of the resulting plot would be equal to the length of the formula plus a constant (assuming that we use a declarative programming language such as Mathematica). For example Figure 2 shows a code and its output plot. Other than the part highlighted by pale green (which is the formula itself), the rest of the code stays unchanged for all such plots. Therefore we can claim that the Kolmogrov complexity of all the plots of this style is the length of the formula plus the constant baggage which determines the size, color and the resolution of the image and is the same for all the plots.

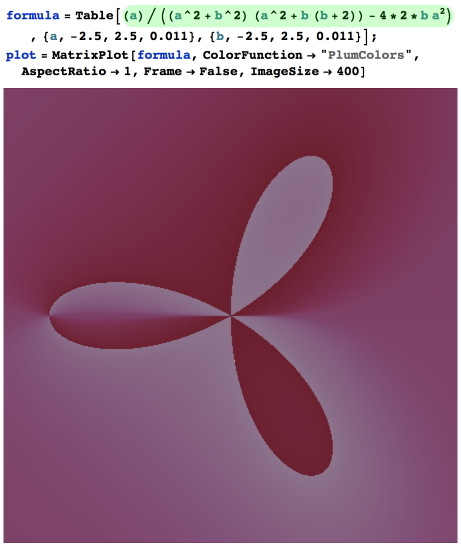


Figure 2: The code and the plot: only the highlighted part changes from plot to plot

(ii) Someone with a reasonable background in mathematics can understand and absorb a simple formula almost perfectly and can even mentally visualize very simple formulas. To such a person, showing the formula of a plot is almost equivalent to giving him/her a very short and understandable description of the image.

Let us suppose these assumptions are true. Then we can design an experiment through which we can measure relationship of algorithmic information and BIP. The experiment should have two parts:

Part I: First we should model how humans perceive the algorithmic information or in this case, *the formula complexity*. The straightforward approach is to show many formulas and ask subjects to rate their complexity. Then based on the features extracted from formulas, we might be able to learn a model (see next section: First Experiment: Rate My Complexity).

Part II: We will show plots (with and without formulas) to subjects and measure changes in their BIR with respect to complexity of the formulas. Note that each subject sees either a simple or complex formula not both. This allows us to measure the exact effect of formula complexity on the perception of beauty (See Second Experiment: Rate My Coolness).

First Experiment: Rate My Complexity

In the first experiment, we asked people to rate the complexity of shown formulas by an integer number between 1 and 7. Here are the details of the experiment:

Goals

The main goal of this experiment was to gather some data to model *formula complexity perception* (FCP) in people with at least a college-level background in mathematics.

A side goal was to run a pilot on assessing peoples' interest in participating in such experiments and also gaining some experience with online survey platforms. This experiment was my first field study ever, thus had a high pedagogical value and significantly helped with the design and execution of the second experiment.

Features Extraction for Formula Complexity Perception (FCP)

Before picking the formulas for the survey, we chose a set of ten intuitively relevant features. The list includes (1)number of characters[nc], (2) number of nodes in the arithmetic parse-tree or tree size [ts], (3) depth of the parse-tree[td], (4) number of "+" and "−" operators[ns], (5) number of multiplications[nm], (6) number of fractions[nd], (7) number of grouping parentheses pairs (excluding function calls)[ng], (8)number of exponents[ne], (9) number of function calls[nf] and (10) number of variables[nv].

Let us list the features for an example formula:

$$\frac{(a+b)(a-b)^3}{\sin(\frac{a^2}{b})-|a+b|}$$

The parse-tree for this formula is given in Figure 3.

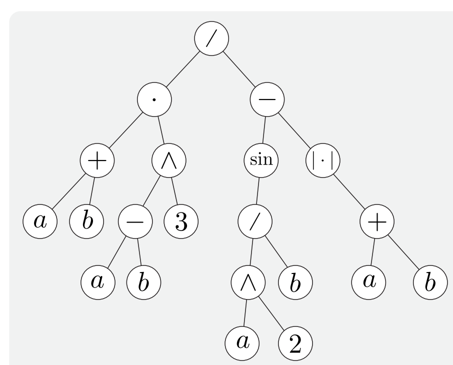


Figure 3: Parse tree of the example

Therefore the feature vector for this example formula is given by:

nc	ts	td	ns	nm	nd	ng	ne	nf	nv
26	21	5	4	1	2	2	2	2	2

Obviously we could have chosen a much more elaborate set of features. For example our set does not differentiate or add features like number of squares, number of trig functions, number of integer constants, symmetries of the equation, etc. But such detailed set of features will lead to very sparse experimental results unless we show hundreds of formulas to the subjects in order to get good variance for each feature. As we use an online survey platform, showing a high number of questions can significantly shrink the number of participants thus we limited ourselves to this set of features.

Experiment Design

Choosing the Formulas These points were considered in choosing the formulas:

- Considering the second experiment that uses plots of two variable functions, we only presented formulas with one and two variables to avoid the possible bias caused by the variable count.
- Although "College-Level" background was announced as the prerequisite for taking the test, formulas with pre-calculus level of complexity such as trigonometric functions and modulus operator were picked. This decision was made to make sure participants understand the formulas and functions.
- Formulas were picked to represent high variance of each feature discussed in the previous section.

Designing the Forms For both experiments we used an online survey platform called Typeform which provides a rich, modern and interactive interface that is more similar to a friendly single-page web application than a regular survey form. Moreover, the interface is completely mobile-responsive – it turned out more than 20 % of the participants used their smart phones (see Figures 5 and 17)–. In order to unlock logic-jump and randomization features of the form a short-term subscription was purchased.

The online form for the first experiment included a few short questions in the beginning (such as : do you like mathematics ? or does the length of a formula affect its apparent complexity ?) and then the subject was shown a set of formulas (large, high-resolution, LaTeX- type-set see Figure 4) and they could chose a rating between 1 and 7. For the variables of the formulas we used a and b consistently to avoid any bias towards real $(x, y, z, \ldots)$ or integer $(i, j, m, n, \ldots)$ numbers.

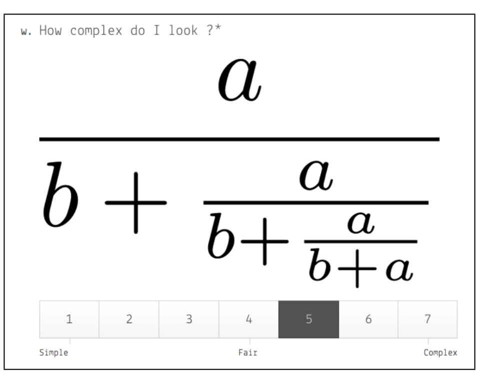


Figure 4: screenshot of an example question from experiment 1

Experiment Execution

The online form was posted on the author's Facebook wall and four student groups (Oxford Mathematics MSc , MIT MediaLab, Stanford EE and Berkeley EE). The cover story of the post clearly emphasized that the subjects should have a college-level background in mathematics. Figure 5 shows the statistics of the participants and the average time spent on the form by each subject (over a 6-day period).

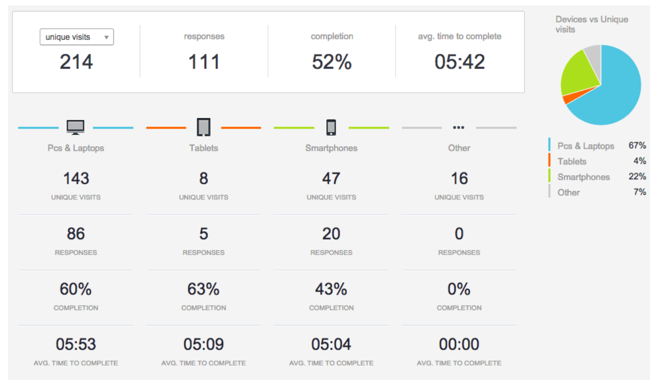


Figure 5: Experiment 1 participant statistics

Analysis and Modeling of the results of the first Experiment

Overall Statistics

From the 111 responses, 108 unique IPs were chosen for the analysis and modeling. From the 42 shown formulas, 15 formula only included a single operation or function and the rest were compound. Figure 6 shows the histograms of complexity ratings for different categories of formulas. The red-colored histograms are for the compound formulas which have a larger frequencies at higher ratings. However, as we see, single operations $(+, -, \times, /, \wedge)$ and single function calls have their highest mass at small ratings (1 and 2).

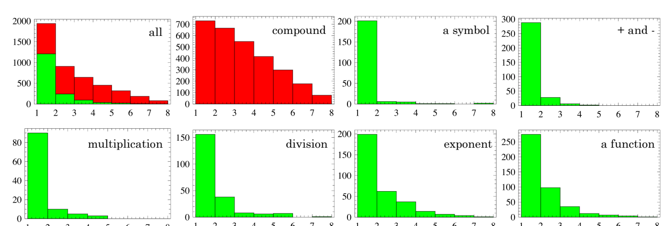


Figure 6: Histograms of ratings in Experiment 1

Bayesian Classification

In order to learn a model for FCP, let us treat the problem as a multi-label classification task where the goal is to assign each feature vector a label which is a whole number between 1 and 7.

To carry out the task we used Mathematica's built-in bayesian logistic regression algorithm. In the first try, the whole data-set was given as a training set which included 4536 data points. Mathematica's classifier (using `Classify[data,Method->"LogisticRegression"]`) performs cross-validation and model selection on the data and outputs a classifier function with minimum confusion across all the cross-validation sets.

The resulting classifier was L_1 and L_2 regularized and showed dependency on the first two variables only (number of characters and tree size). Figure 7 shows the marginal posterior probabilities for these two variables along with number of $+, -$ and an arbitrary independent variable.

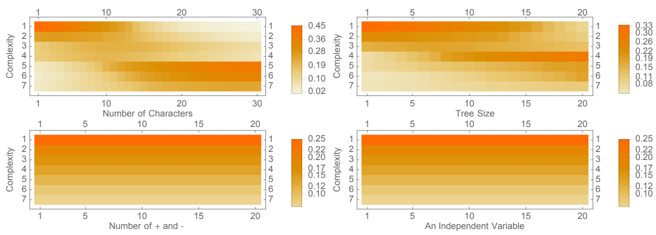


Figure 7: Marginal posteriors of the classifier trained on the whole data

As we see the classifier shows high confidence for very simple and very complex formulas.

Now let us try learning the median of complexity ratings. The new classifier takes almost all of the features into account. Figure 8 shows the marginal posteriors of the median classifier for 6 of the features.

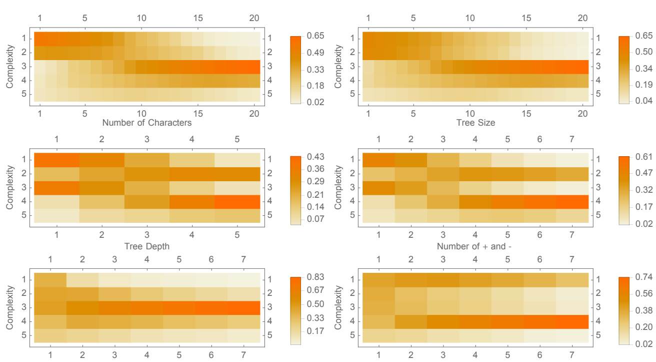


Figure 8: Marginal posteriors of the classifier trained on the medians

Bayesian Linear Regression

Linear Regression with a mixture of L_1 and L_2 regularizations ,gives a very sparse model only dependent on the tree size. Figure 9 shows the posterior distribution for different values of tree size.

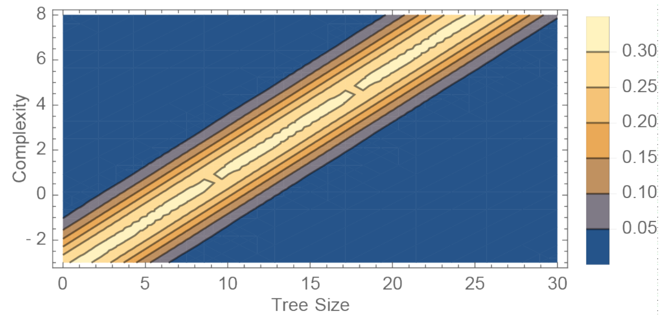


Figure 9: Linear Regression on the whole data: Posterior Marginal of complexity conditioned on Tree Size

Training with the medians gives more interesting results. Here one of the most significant relationships is the number of function calls which did not show up as boldly with previous models . See Figure 10 for the marginal posteriors.

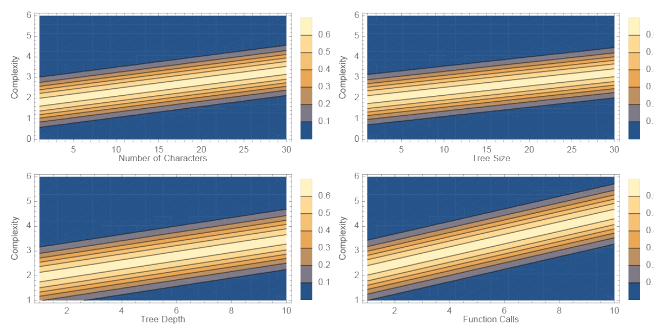


Figure 10: Linear Regression on Medians: Posterior Marginals of complexity

Evolutionary Symbolic Regression

Besides bayesian models we used genetic programming for symbolic regression.

For this purpose we used Eureqa(a commercial symbolic regression tool). Eureqa utilizes genetic programming to combine and evolve mathematical expressions in order to fit the data.

To perform a search in Eureqa, one must choose a set of mathematical operators that are guessed to be in the final model. Intuitively, all the ten features we picked positively effect on the complexity. Therefore we avoided division and subtraction in choosing the operators to shrink the hypothesis space.

Eureqa's fitness function is not only dependent on the regression error but also on the complexity of the resulting formulas. In other words, Eureqa looks for a simple model to describe the data; Therefore it allows users to change the complexity of each individual operator. Setting low complexity for an operator gives it a higher priority to show up in the final formulas. Thus selecting a set of operators and functions and setting their complexity for symbolic regression can be thought of setting priors.

Figure 11 shows a screenshot of search-space setting in Eureqa.

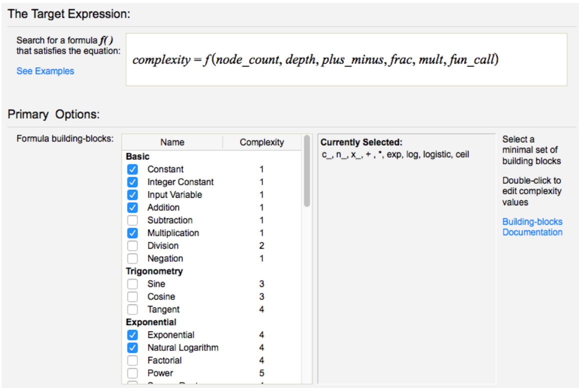
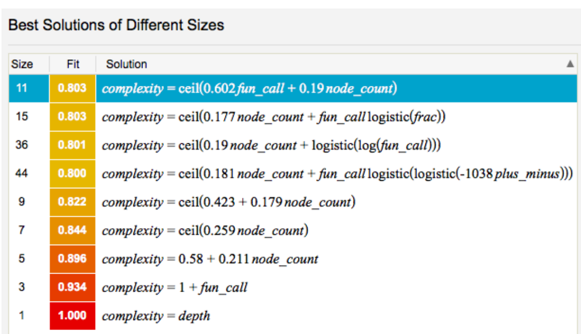


Figure 11: Defining the search space

Running Eureqa on the whole data-set for about an hour gives a few simple formulas that are very easy to interpret. Figure 12 shows the results.



Best Solutions of Different Sizes

Size	Fit	Solution
11	0.803	$complexity = \mathrm{ceil}(0.602\,fun\_call + 0.19\,node\_count)$
15	0.803	$complexity = \mathrm{ceil}(0.177\,node\_count + fun\_call\,\mathrm{logistic}(frac))$
36	0.801	$complexity = \mathrm{ceil}(0.19\,node\_count + \mathrm{logistic}(\log(fun\_call)))$
44	0.800	$complexity = \mathrm{ceil}(0.181\,node\_count + fun\_call\,\mathrm{logistic}(\mathrm{logistic}(-1038\,plus\_minus)))$
9	0.822	$complexity = \mathrm{ceil}(0.423 + 0.179\,node\_count)$
7	0.844	$complexity = \mathrm{ceil}(0.259\,node\_count)$
5	0.896	$complexity = 0.58 + 0.211\,node\_count$
3	0.934	$complexity = 1 + fun\_call$
1	1.000	$complexity = depth$

Figure 12: search results on the whole data-set

The two simplest models coming out of this search describe complexity with just the depth of the tree and number of function calls. These simple models are very intuitive in the sense that depth of the tree is simply a measure of nesting in the formula– it is easy to understand what $\sin(x)$ does but very difficult to have any idea about $\sin(\cos(\tan(x)))$. Number of function calls is also very understandable because we usually learn to understand behavior of a single function but not the combination of many of them.

Searching to fit median of the complexity gives much smaller values of error and produces models very similar to the previous case. Figure 13 shows the search results on the median data. Note that one of the very simple models ($complexity = 0.133 * nodeCount + 2.044 * logistic(funCall^2)$) shows a high correlation $R \approx 0.9$ with the median data. We will use this model along with our Bayesian Logistic Regression and Linear Regression models in analyzing the results of the second experiment.

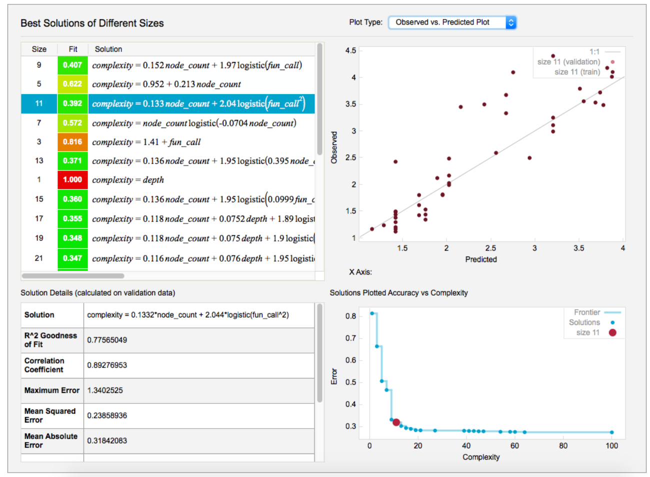


Figure 13: search results on the median data-set

Second Experiment: Rate My Coolness

In the second experiment, we asked participants to rate the beauty or coolness or interestingness of two dimensional plots before and after seeing the formulas used to create them.

Goals

The main goal of the experiment was to assess if people find images more interesting if they are shown the formula used to create those images. We also wanted to know if there is a correlation between the complexity of given formulas and changes in subjects' opinion.

Formulas and Plots

For the purpose of the experiment, 10 formulas which have a quite surprising or at least none-obvious visualization were picked and a more complicated version of them were constructed. Then each formula was plotted as a density plot with a consistent color space, size and resolution amongst all the plots. A guide for understanding how plots were made was shown in the beginning of the survey (See Figure 14).

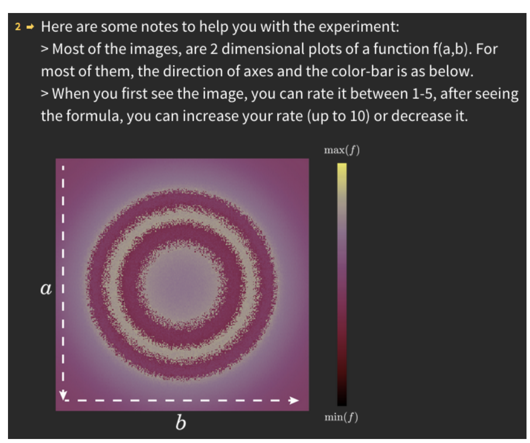


Figure 14: The guide shown to the subjects

The reason that we showed a guide was to provide a way for participants to connect the formulas to the images. We believe that this connection is necessary in perception of beauty if Schmidhuber's theory about compression progress is right.

Figure 15 shows some of the plots used for the experiment.

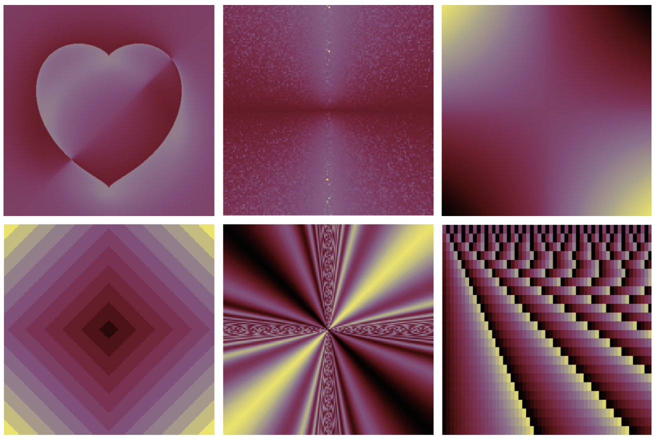


Figure 15: Some of the plots used in the experiment 2

Experiment Design and Execution

Similar to the first experiment, we used Typeform as our questionnaire platform. In order to monitor the effect of seeing formulas on BIP, first we showed a plot without a formula and asked subject to rate its interestingness. Then a formula (simple or complex by random) was shown with the same plot and the subjects were asked to update their vote.

In order to reduce confusion, the original vote of the subject was also shown with the second question so that they can easily decide weather they are increasing, decreasing or leaving their vote unchanged (See Figure 16).

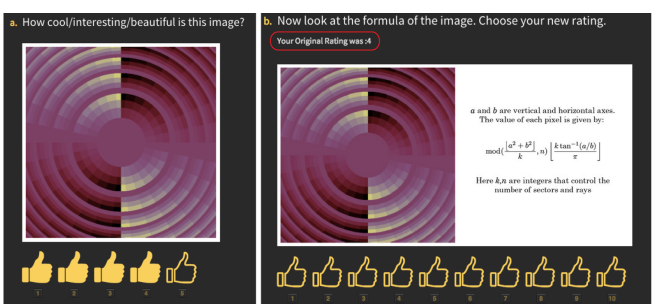


Figure 16: an example of a question-pair in experiment 2

As the questions in this experiment are supposed to come in pairs, it was impossible to randomize the form with the provided tools in Typeform. Therefor, we had to come up with a way to randomize the experiment through other functionalities of the form. Considering that the day component of subjects' date of birth ($d \in 1, \ldots, 31$) is distributed almost uniformly, we generated 6 different random sequences of the images for every 5 day of the month. Therefore by using the logic-jump function of the form, say if a subject's birthday is on the 3rd day of the month ($1 \leq 3 \leq 5$), (s)he will see the first sequence.

On the other hand, we made sure that subjects do not see the formulas before rating the image without the formula.

The form was posted in the same student groups mentioned in experiment 1, a few student groups associated with Max Planck Institute (Tubingen) and a few LinkedIn groups and other relevant Facebook pages. The statistics of participants is given in Figure 17.

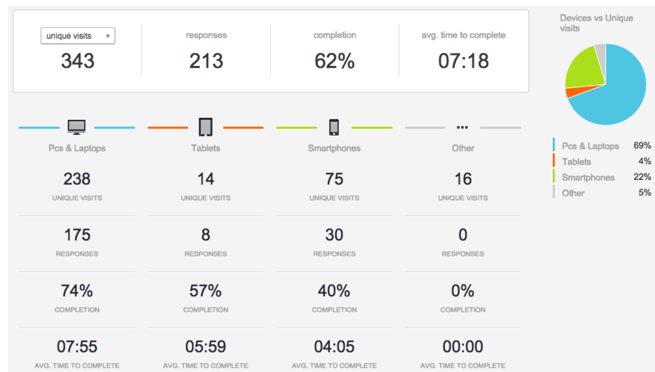


Figure 17: participation statistics of experiment 2

Analysis and Modeling of the results of the Interestingness Experiment

Overall Statistics

from 213 participants 195 unique IPs were sifted for analysis. For each plot, we gathered 3 data-sets: (N) no formula shown, (C) complex formula shown, (S) simple formula shown.Figure 18 shows the results for the whole dataset.

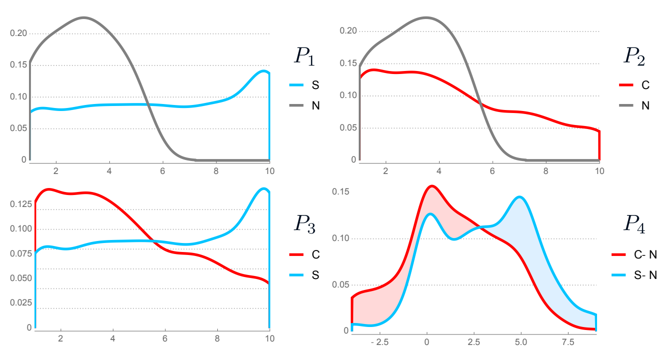


Figure 18: Experiment 2 overall results. (P_1) distribution of S vs. its corresponding N, (P_2) distribution of C vs. its corresponding N, (P_3) S vs. C and (P_4)comparison of distributions of vote-change (C-N vs S-N)

Figure 19 shows the results for three individual examples.

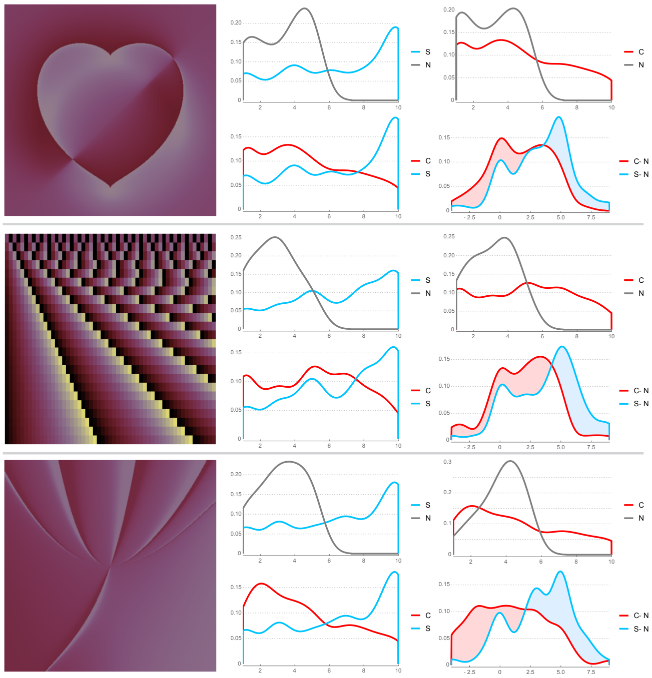


Figure 19: Experiment 2 results for 3 example plots.

Two immediate hypotheses can be deduced from Figures 18 and 19:

- *Rating of Interestingness of a plot increases after showing a mathematical formula that is used to generate the plot.* We will call this hypothesis H_A
- *Rating of Interestingness of a plot after seeing a simple formula is higher than its rating after seeing a complex formula.* We will call this hypothesis H_B

Now let us perform bootstrap hypothesis testing for H_A on the means. If we show the mean rating of plots with formulas with $\bar{F}$ and mean rating of plots without formulas with $\bar{N}$, then observed difference of the means will be $t^*_{\text{obs}} = \bar{F} - \bar{N}$. Figure 20 shows the distribution of $t^*_{\text{bootstrap}}$ for $B = 100000$ bootstrap samples versus t^*_{obs}. For this sample size we get $p = 0$. Therefore we reject the Null hypothesis (H_{0_A}) with very significant p-value of at most $p = 10^{-6}$.

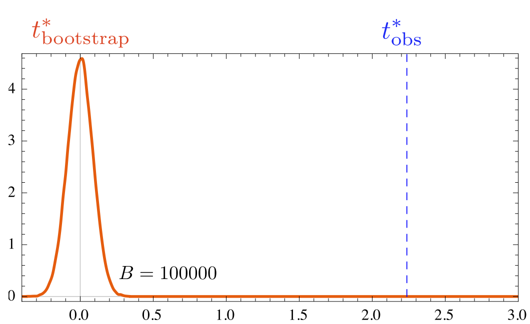


Figure 20: bootstrap hypothesis test for H_A

Similarly for H_B If we show the change in the mean rating of plots with simple formulas with $\bar{SN}$ and mean rating of plots with complex formulas with $\bar{CN}$, then observed difference of the means will be $t^*_{\text{obs}} = \bar{SN} - \bar{CN}$. Figure 21 shows the distribution of $t^*_{\text{bootstrap}}$ for $B = 100000$ bootstrap samples versus t^*_{obs}. Although in this case the difference of the means is smaller and the distribution of $t^*_{\text{bootstrap}}$ is wider, still we get $p = 0$. Therefore we reject the Null hypothesis (H_{0_B}) with very significant p-value of at most $p = 10^{-6}$.

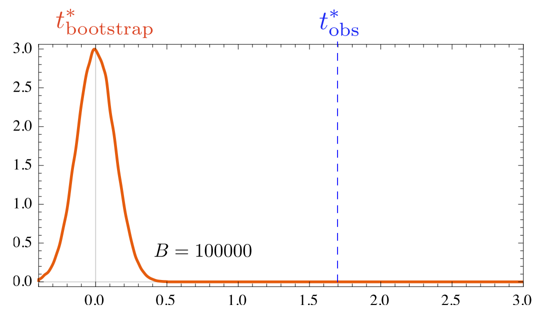


Figure 21: bootstrap hypothesis test for H_B

The bottom-line is both hypotheses agree with Schmidhuber's theory. Now we will look at the actual values of complexity and seek a relationship between perception of complexity and changes in beauty rating.

Beauty vs. Complexity

In this section we compare our beauty rating data with the formula complexity models that we obtained in the first experiment. We are seeking a relationship between the perception of complexity and the change in perception of beauty.

For each image, we take the average of change in beauty rating for two cases of simple and complex formula. Therefore, for each image we have two data-points. We also calculate the complexity of used formulas using our classifiers, predictors and the models obtained from genetic programming.

Bayesian Logistic Regression Figure 22 shows the results of comparison of change in perception of beauty with our bayesian logistic regression model of perceived complexity. As we see both classifiers find a negative relationship between complexity and change in beauty perception. The classifier trained on the whole data gives a higher correlation of $R = -0.46$ which is in agreement with Schmidhuber's theory. In other words, lower complexity gives rise to a larger change in perception of beauty.

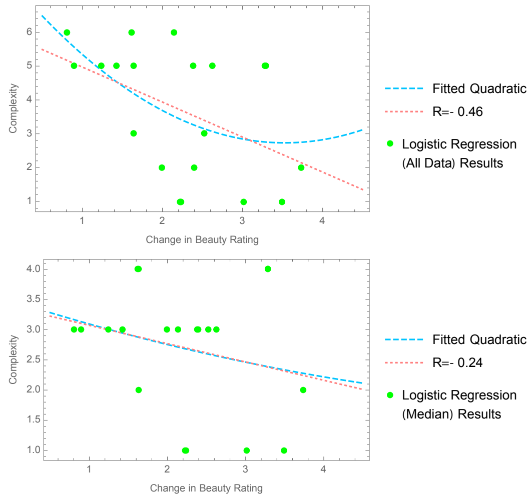


Figure 22: complexity (modeled by logistic regression) vs. change in beauty perception

Note that in both cases a quadratic fit produces a smaller fitting error than a linear fitting. Apparently, very small values of complexity do not lead to the highest changes in beauty rating. This effect can be seen in the results of our other models too. In the Discussion section we will try to interpret this effect.

Bayesian Linear Regression Similarly, we can obtain the same plots for our linear regression model. Figure 23 shows the results. As we see our linear regression model of complexity trained on medians of data gives a higher correlation value $R = -0.61$. Again both results agree with Schmidhuber's theory. Note that the linear regression models have produced higher variance of results for complexity in comparison to the logistic regression models. Note that here again the quadratic model is a better fit than a linear fit.

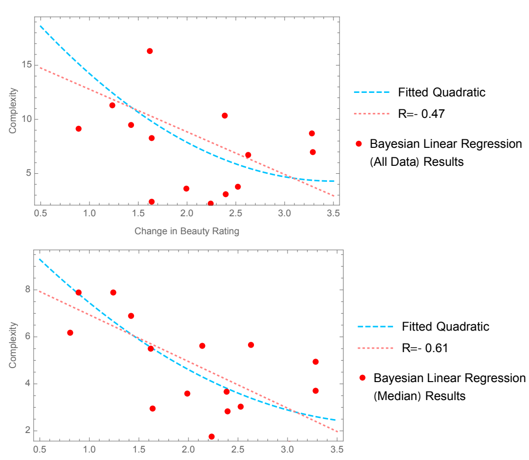


Figure 23: complexity (modeled by linear regression) vs. change in beauty perception

Evolutionary Symbolic Regression Similarly, we can obtain the same plots for our genetic programming models. Figure 24 shows the results. As we see our genetic programming model of complexity trained on all of data gives a higher correlation value $R = -0.62$ than all of the previous models. Not to mention that this model is also the simplest model we obtained (See Figure 12).

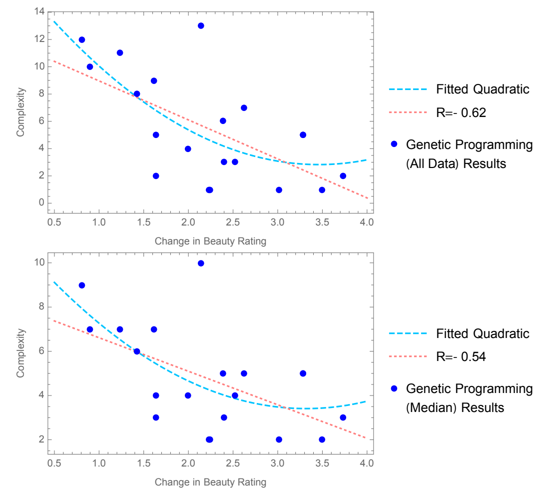


Figure 24: complexity (modeled by Genetic Programming) vs. change in beauty perception

Both results agree with Schmidhuber's theory again the quadratic model is a better fit than a linear one.

Summary and Discussion

Our two experiments led to promising results in confirming the compression progress. In summary, the results show that giving subjects a compressive information or a short way to describe an image makes them find the images more compelling. Apart from that, simpler formulas lead to higher excitement and make subjects increase their vote.

Also, we observed that very simple formulas has slightly smaller effect on the perception of beauty (recall the parabolic shape of the complexity-beauty curve See Figure 22,23,24). Despite possible biases of the experiment, we might be able to find a bit more sophisticated explanations for this phenomenon. One possible explanation which directly follows Schmidhuber's theory might be that subjects already knew the visualization for that simple formula and showing the formula did not give them much information.

Another explanation might be that in formulas that are a bit more complicated than the very simple ones a 2-way learning process happens which causes a higher BIP. By 2-way learning we mean that the plot teaches something about the formula and the formula teaches something about the plot to the subject.

References

Jurgen Schmidhuber, J. (1991). A possibility for lmplementing curiosity and boredom in model-building neural controllers.

Langlois, J. H., & Roggman, L. A. (1990). Attractive faces are only average. *Psychological science*, *1*(2), 115–121.

Perrett, D. I., Burt, D. M., Penton-Voak, I. S., Lee, K. J., Rowland, D. A., & Edwards, R. (1999). Symmetry and human facial attractiveness. *Evolution and human behavior*, *20*(5), 295–307.

Rubenstein, A. J., Langlois, J. H., & Roggman, L. A. (2002). What makes a face attractive and why: The role of averageness in defining facial beauty.

Schaul, T., Pape, L., Glasmachers, T., Graziano, V., & Schmidhuber, J. (2011). Coherence progress: a measure of interestingness based on fixed compressors. In *Artificial general intelligence* (pp. 21–30). Springer.

Schmidhuber, J. (2008). Driven by compression progress. In *Knowledge-based intelligent information and engineering systems* (pp. 11–11).

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *Autonomous Mental Development, IEEE Transactions on*, *2*(3), 230–247.