From JPEGs to Quantum Field Theory:

Problem set for the Santa Fe Institute Renormalization MOOC

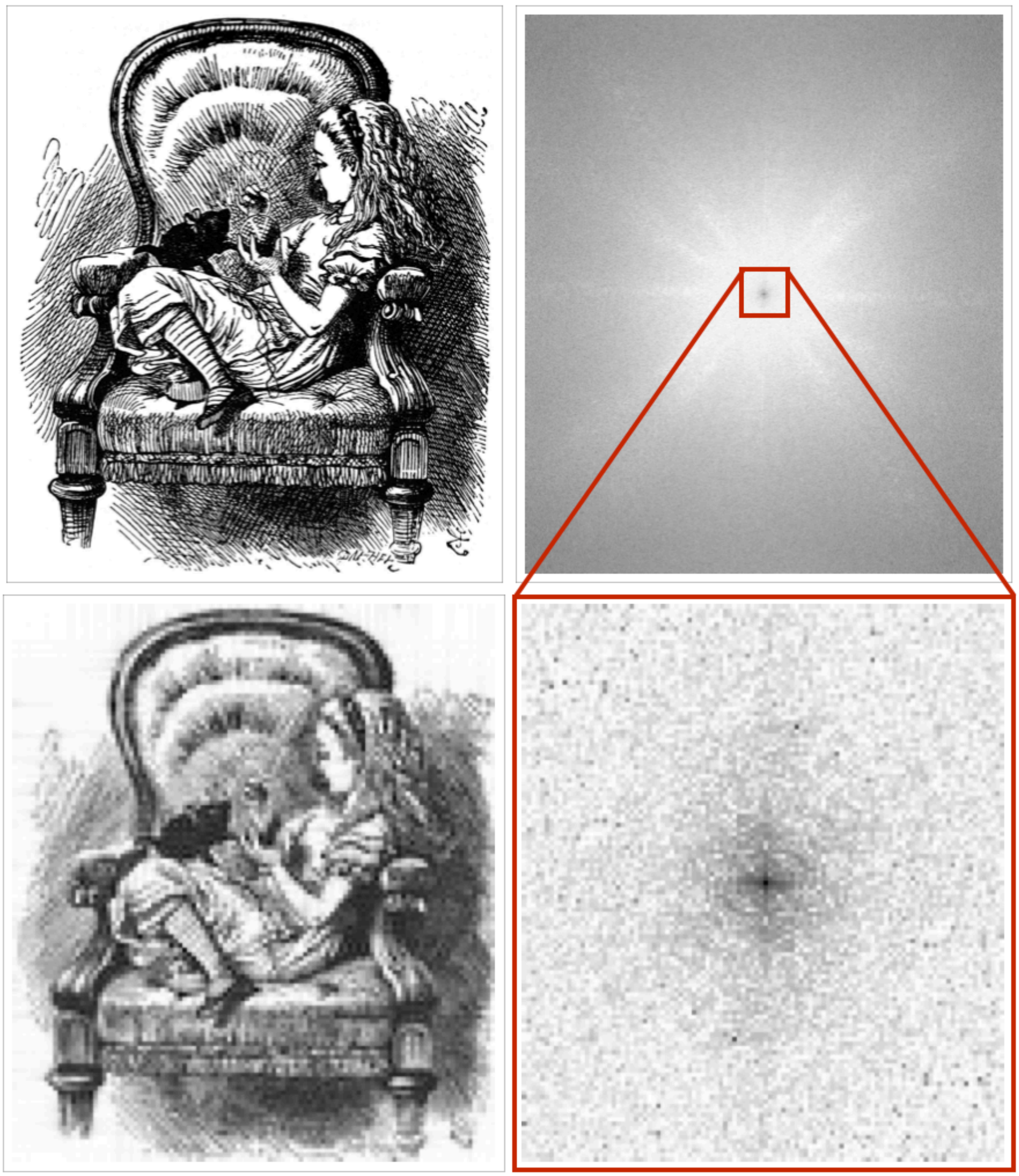
http://renorm.complexityexplorer.org

Solution set at http://santafe.edu/~simon/MOOC_solutions.pdf

Simon DeDeo

Assistant Professor, Carnegie Mellon University; External faculty, Santa Fe Institute.

simon@santafe.edu; http://santafe.edu/~simon



Frontispiece to Renormalization: coarse-graining Alice in Wonderland, using a method roughly equivalent to the early JPEG algorithm. First, we re-represent the image in a new fashion (top-left panel to top-right panel): we go from "image space" to "Fourier space" (or, for a physicist: "configuration space" to "momentum space"). The new representation contains all of the information of the original, organized differently—each point refers to the strength of a ripple, at a particular frequency and in a particular direction, with the really fine-grained fluctuations on the edges. This re-representation allows us to describe the coarse-graining procedure in a particularly simple fashion: we simply cut out, and retain, only the middle portion (red box, top-right to bottom-right panel), dropping the fine-grained features. On transforming back into image space, we recover something that resembles the original image. In part because this compression matches some of the algorithms found in our own visual system, a dramatic reduction in image size is possible, while leaving the image largely still interpretable.

Contents

1 Unit One: An Introduction to Coarse-Graining **1**

2 Renormalizing Markov Chains **3**

3 Further Reading **4**

4 Renormalizing Cellular Automata **4**

5 Renormalizing the Ising Model **6**

6 Renormalizing the Creature (Krohn-Rhodes Theorem) **6**

7 Renormalizing the Thermal Plasma **7**

8 Rate-Distortion Theory: Keeping the Things that Matter **8**

1 Unit One: An Introduction to Coarse-Graining

Jane Austen's Pride and Prejudice as a bag of letters

Imaging sticking your hand into the novel *Pride and Prejudice*[1], pulling out a letter at random (with replacement, *i.e.*, when you pull out the letter and look at it, you then put it back in). If we image that a letter we see corresponds to a draw from a probability distribution, this defines a distribution over the letters of the alphabet. Ignoring case distinctions, and making the the distribution turns out to be:

[1]Full text at `https://www.gutenberg.org/files/1342/1342-h/1342-h.htm` if you want to do the text processing yourself. There's nothing special about this particular book from an information-theoretic or coarse-graining perspective. But it's always helpful to apply tools to a fragment of the real world, and why not pick a good one?

```
[["a", 0.0777093], ["b", 0.01694125], ["c", 0.02509587], ["d", 0.0415729],
["e", 0.1293273], ["f", 0.02237207], ["g", 0.01869932], ["h", 0.063514],
["i", 0.0705279], ["j", 0.00162383], ["k", 0.00598080], ["l", 0.04025481],
["m", 0.0275251], ["n", 0.0702613], ["o", 0.0746462], ["p", 0.01533419],
["q", 0.001168940], ["r", 0.0602125], ["s", 0.0617264], ["t", 0.0869565],
["u", 0.02793712], ["v", 0.01067520], ["w", 0.0229406], ["x", 0.001564180],
["y", 0.02368643], ["z", 0.001745021]]
```

e.g., the probability of pulling out a "z" is about 0.1%.

Q1: what is the entropy of the distribution?

Q2: Coarse-grain this distribution to "vowel" and "not-vowel" (*i.e.*, map each of the letters into one of two categories). What is the entropy of this coarse-grained distribution? For simplicity, assume the vowels are just the letters "a", "e", "i", "o" and "u". Neglect letters that can sometimes function as a vowel, like "y", or cases where a string of letters can indicate a vowel sound, as in "loose".

From bags of letters to bags of words

Imagine taking *Pride and Prejudice* and coarse-graining the words by keeping only their grammatical role: "noun", "verb", "adjective", "adverb", "determiner" (a or the), and so on.

For example,

> It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

becomes (in a rough and ready fashion)

```
PRONOUN VERB DETERMINER NOUN ADVERB VERB PREPOSITION DETERMINER ADJECTIVE
NOUN PREPOSITION VERB PREPOSITION DETERMINER ADJECTIVE VERB VERB PREPOSITION
VERB PREPOSITION DETERMINER NOUN
```

Q3: What are some of the rules that might apply to this coarse-grained time-series? If you want to go further, "Part of Speech Taggers" are widespread online—see, for example, the tagger in the Python NLTK package, `http://www.nltk.org/book/ch05.html`.

Open problem: Coarse-graining real world data

If you have some programming skills, working with real data is the quickest way to build intuitions about coarse-graining.

If you are a chess player, for example, try building an algorithm that takes an PGN file and coarse-grains the state of the board. For example, by adding up the "point values" of the pieces each player possesses: each pawn counts for one point, each knight for three, and so on.[2] If you are more ambitious, you might think of dividing up the board into quadrants and summarizing who controls each quadrant.

[2]See `https://en.wikipedia.org/wiki/Chess_piece_relative_value`

Once you've built a coarse-graining, you can ask questions about the information content (the system entropy) and how much is lost upon coarse-graining compared to the base. For a system like chess, the entropy of the microscopic system can be very hard to measure because there are an enormous number of valid potential configurations and you need enough samples to estimate the probability of each).

Finally, to anticipate the arc of this tutorial, you can then ask about the laws and rules that might apply to the coarse-grained variables. Can you infer, for example, probabilistic laws for how the relative piece-scores of two players evolve during a game?

Further Reading

- A crash-course in information theory is *Information Theory for Intelligent People* `http://tuvalu.santafe.edu/~simon/it.pdf`.
- Measuring information-theoretic quantities on data; plus, a crash course in coarse-graining and its relationship to information-theoretic quantities is *Bootstrap Methods for the Empirical Study of Decision-Making and Information Flows in Social Systems*; `http://www.mdpi.com/1099-4300/15/6/2246/htm`
- A big-picture overview of coarse-graining can be found in *Major Transitions in Political Order*; `https://arxiv.org/abs/1512.03419`

2 Renormalizing Markov Chains

Q1: Prove the existence of two-state slippy counters that do not have a corresponding two-state microtheory (*i.e.*, that *can't* be derived as a coarse-graining of a two-state Markov chain of any form).

Q2: Experiment with the three-state case. Take a Markov Chain over three states, represent it as a matrix, and use a system like Mathematica or Python to find the fixed points by repeated multiplication of the matrix on itself. Confirm the result using theory: find the eigenvector corresponding to the eigenvalue equal to unity, and show that the columns of the fixed point Markov chain are equal to the components of the first eigenvector.

Q3: Consider two rock-paper-scissors playing robots. Each robot decides what to do next based on both its move, and the move its opponent made, the step before.

A. Write down two decision rules for each robot. You can make them up yourself; for example, a robot might play "best response" to its opponent's prior move; another robot might play noisy best-response (*i.e.*, with some small probability to play something other than best response). Then, for example, if the system is in state {R,P}, and robot one plays best response, its next move will be S (since Scissors would have beaten its opponent).

In general, this will be a 9x3 matrix: for each of the nine possible states at the previous step {RR, RP, RS, PR, PP, PS, SR, SP, SS}, the robot has some probability to emit R, P, or S. If you write the matrix as A_{ia}, then $A_{1,2}$, for example is the probability that if the previous state of play was RR, then robot A will play paper. You might find it easier to use a dictionary representation if you're working in Python, rather than keeping track of matrices.

B. Combine these two decision rules together to get the evolving Markov Chain for the pair. Technically is a bit tricky, because you have to convert the pair of output symbols (R, and R) to a single symbol in the larger list of 9 possible combinations.

In general, the probability of going from state i to state j is equal to the product of A_{ia} ("robot A sees i and plays move a") and B_{ib} ("robot B sees i and plays move b"), and where j is the combination of the a and b moves—equal to $3 \times (a-1) + b$. For example, if j is equal to PS, then $a = 2$, and $b = 3$). Be careful to define the player B's rules so that, for example PS means that he (player B) played S. Again, you might find it easier to use a dictionary representation if you work in Python.

C. Find the fixed points of this system. (In general, to make sure that you flow to the 2-dimensional subspace, you can add a small jitter probability epsilon to make sure that you don't get stuck in "limit cycles".) Which robot wins in the long run?

3 Further Reading

- This unit described how to coarse-grain Markov Chains that are already known; but what if you want to infer a chain from a time-series? In *Conflict and Computation on Wikipedia: A Finite-State Machine Analysis of Editor Interactions* `http://www.mdpi.com/1999-5903/8/3/31/htm` we present a worked example, and introduce the `SFIHMM` code for machine-learning of time-series on data, available at `http://bit.ly/sfihmm`.
- If you successfully completed **Q3**, you've built what we call a "joint machine"—a representation of the emergent behavior when decision-makers come together in a game. *Group Minds and the Case of Wikipedia* `https://arxiv.org/abs/1407.2210` describes the properties of these representations in greater detail.

4 Renormalizing Cellular Automata

Israeli and Goldenfeld's work is primarily computational in nature. Once the problem is set up, and the space of projections and evolution operators defined, they're then forced to boot up a computer and search. In some cases, they can *post hoc* interpret what they've found (for example, the Garden of Eden states). But in most cases, the overall structure of the network is mysterious, "baked in" to the mathematics, and not derivable by any simpler method.

The best way to understand their system is to write your own code to search the space. There's a lot you can learn just by sticking to the one-step coarse-graining, where the super-cells are size two, and the clock jumps two units into the future at the coarse-grained level.

Q1: demonstrate the renormalization of Rule 105 to Rule 150. What other projections work?
You'll almost certainly want to code this up, but there's no reason you can't do it by hand with lots of paper. If you do try to code, here's a guide for Python.

1. First decide on a representation of the fine-grained rule-set (the lookup table f). In Python language, a simple choice would be an eight element list, where the first element is the output for the {0,0,0} state, the second for the {0,0,1} state, and so forth. Note that Wolfram's definition of rules is a bit funny; Rule 0, for example, takes 000 (all black states) to 1 (white). The easiest way to get this is by converting the number to binary, and then reversing the bits—so, *e.g.*, for Rule 105, the binary expansion is `01101001`. If you reverse this, you get

`10010110`. An input cell set "black, white, black" is then `010`, or two; the third bit in the string (or, second counting from zero) is 0, so "black, white, black" goes to "black".

To access the lookup table, you'll generically need to convert a list of 1s and 0s to the corresponding integer. In Python, you can use the "int" function; for example, the list position corresponding to the state {0,1,1} is 3: `int('010', 2)` gives `2`. So in Python, you can figure out the evolution of Rule 105 given "black, white, black" initial conditions as `'10010110'[int('010', 2)]`.

2. Then build a function that maps a lookup table f to the two-step evolution on the fine-grained scale. If the states at the micro-level are denoted {a, b, c, d, e, f}, then the left and right elements of the super-cell at time t+1 are given by $A = f(f(a, b, c), f(b, c, d), f(c, d, e))$ and $B = f(f(b, c, d), f(c, d, e), f(d, e, f))$.

3. Decide on a representation for the projection operator. Again, in python language, a simple choice is a four element list, where the first element is the projection of {0,0}, the second of {0,1}, and so forth.

4. With these in place, you can take an f (the fine grained rule), a p (the projection lookup table), and a g (another rule, with the same 8-element lookup table), and determine whether the diagram commutes. We need only determine whether $p(A, B)$ is equal to $g(P(a, b), P(c, d), P(e, f))$ for all 2^6 values of {a, b, c, d, e, f}.

   To cycle through the different values, it's simplest to take a for-loop and convert the decimal values back to a string representation. In python, you can use the string format (definitely not optimized for speed): `[[j for j in '{0:b}'.format(i).rjust(6, '0')] for i in range(2**6)]`

   Using the f and g corresponding to rule 105 and 150, along with the "edge detector p", you should be able to confirm the result of the lecture.

5. The final step in their analysis is "simple": for each f, cycle through the 2^4 possibilities for p, and the 2^8 possibilities for g, to find choices of g and p that enable the diagram to commute. For p, don't forget that you'll want to avoid the "trivial" projections—[0,0,0,0] and [1,1,1,1]—which are easy to avoid by setting the range from 1 to $2^4 - 1$.

Further Reading

- This entire unit is based on a series of papers by Navot Israeli and Nigel Goldenfeld—in particular, *Computational Irreducibility and the Predictability of Complex Physical Systems* `https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.92.074105`, and *Coarse-graining of cellular automata, emergence, and the predictability of complex systems* `https://journals.aps.org/pre/abstract/10.1103/PhysRevE.73.026203`

- We'll return to how renormalization can induce long-range couplings and interactions in the next unit, on the Ising Model. If a theory extends in time as well as space, we can think about long-range couplings as a new form of memory: if I'm directly coupled to distant times, I have a new form of memory. For an unusual perspective on the relationship between renormalization and memory, see *Origin Gaps and the Eternal Sunshine of the Second-Order Pendulum* `http://fqxi.org/community/forum/topic/2918`.

5 Renormalizing the Ising Model

A short "Ising Model Practical" written for the 2011 Santa Fe Institute Summer School is at `http://tuvalu.santafe.edu/~simon/practical.pdf`. This walks through the Ising model in much greater, and slower, detail, along with suggestions for how to understand the Ising model as an analogy in biological and social systems.

If you work through this handout you'll come out the other side with a hands-on understanding of the Ising model, including the simulation techniques that (*e.g.*) allow Douglas Ashton to demonstrate the renormalization group flow in his Ising model simulations.

Further Reading

- The classic reference for the particular renormalization transformation we do in this unit is Leo Kadanoff's book, *Statistical Physics: Statics, Dynamics, and Renormalization* `https://www.amazon.com/STATISTICAL-PHYSICS-STATICS-DYNAMICS-RENORMALIZATION/dp/9810237588`— Chapter 14 covers the clever decimation technique where the new lattice is a 45 degree rotation of the old. It's a lovely introduction to the ways in which smart people flailed around at a problem they didn't quite yet know how to solve.
- *Ising Model Practical* `http://tuvalu.santafe.edu/~simon/practical.pdf`
- The Ising model simulation is by Bernd Nottelmann (with some nice interface design by A. Peter Young) `http://physics.ucsc.edu/~peter/java/ising/keep/ising.html`
- Douglas Ashton's YouTube of life at the critical point is at `https://www.youtube.com/watch?v=MxRddFrEnPc`
- We dealt with the Ising model on the 2D lattice, but what about the Ising model on arbitrary graphs? You can also take a look at a story about renormalization in "self-similar" (fractal) network structures in *Dynamics and processing in finite self-similar networks* `http://rsif.royalsocietypublishing.org/content/9/74/2131?sa=X&ved=0CDcQ9QEwEGoVChMIqsL9tpT1xgIVwu0` (and many references therein) – or, if you really want to dig down into the different ways of modeling the Ising model on arbitrary graphs, consider the Linked-Cluster Expansion, `http://tuvalu.santafe.edu/~simon/wortis-1974.pdf`. A great deal of work on the Ising model on arbitrary graphs is done by simulation (see the first link, the Ising Model practical, for how this works).

6 Renormalizing the Creature (Krohn-Rhodes Theorem)

You got a secret crash-course in group theory. The "creature" we drew on the board is also known as the "permutation group on three elements". It's one of the smallest "groups" (a technical, mathematical use of the word group) with interesting properties.

Consider now the "cyclic" group Z_6. In our language, Z_6 is a creature with six internal states, and a single operator, `C`. If we number the states, then `C` takes $1 \to 2$, $2 \to 3$, $3 \to 4$, $4 \to 5$, $5 \to 6$, and $6 \to 1$. (So you can think of it as a counter that only goes up to six before jumping back to one.)

Q1: Find a coarse-graining of Z_6 with two states. Hint: you might find it helpful to draw the transitions in a 2×3 grid.

Q2: Find a coarse-graining of Z_{36} with 3 states.

Q3: Explain why Z_{13} doesn't have a (deterministic) coarse-graining.

Further Reading

- *Effective Theories for Circuits and Automata* and references therein; `http://aip.scitation.org/doi/abs/10.1063/1.3640747` See, in particular, Atilla Egri-Nagy's papers with Nehaniv, as well as Oded Maler's "On the Krohn-Rhodes Cascaded Decomposition Theorem". For a wild ride, take a look at John Rhodes' "Applications of Automata Theory and Algebra" (which used to be passed around in mimeographed form as "the wild book")—`https://www.amazon.com/Applications-Automata-Theory-Algebra-Mathematical/dp/9812836977/`

- Atilla Egry-Nagy's `SgpDec` is a package for the GAP system that implements the Holonomy Decomposition; see `http://gap-packages.github.io/sgpdec/` and `https://arxiv.org/abs/1501.03217` The clearest description of the decomposition I've been able to find (other than Atilla's papers) is Holcombe's book *Algebraic Automata Theory* from Cambridge University Press, `https://www.amazon.com/Algebraic-Automata-Cambridge-Advanced-Mathematics/dp/0521604923`

- *Visual Group Theory*, `https://www.amazon.com/Visual-Classroom-Resource-Materials-Problem/dp/088385757X`. A lovely way to understand groups through network diagrams—the full creature bestiary. Fundamental theorems in group theory are translated into statements about a subclass of networks.

7 Renormalizing the Thermal Plasma

Q1: Plot the effective electric charge at some fixed distance r, as a function of temperature, and plasma density. How do these effects compete?

Q2: If you have a background in physics, work through the full derivation of the Debye length using Richard Fitzpatrick's material (see recommended reading). My blackboard work here is a partial account of that derivation.

Further Reading

- A slow, and clear version of the plasma story, with all the mathematical steps laid out, can be found at Richard Fitzpatrick's site at the University of Texas, `http://farside.ph.utexas.edu/teaching/plasma/Plasmahtml/node7.html`

- "The renormalization group and effective field theories", by two philosophers of science, tells the story of renormalization in a high-level fashion `http://link.springer.com/article/10.1007%2FBF01063904` ; you can also take a look at Michael E. Fisher's article "Renormalization group theory: its basis and formulation in statistical physics", which appears in Tian Yu Cao's edited volume, *Conceptual Foundations of Quantum Field Theory*, `https://www.amazon.com/Conceptual-Foundations-Quantum-Field-Theory/dp/0521602726`

- A conceptual account of the relationship between renormalization in physics, and in other fields, can be found in the review *Major Transitions in Political Order*, `https://arxiv.org/abs/1512.03419`; see also the references at the beginning of section 1.

8 Rate-Distortion Theory: Keeping the Things that Matter

Recall that a distortion function, $d(x, y)$, tells you the cost of "thinking" the system is in state y, when it's actually in state x. You can think of y as a coarse-graining of x.

Consider the coarse-grained creatures from the Krohn-Rhodes section, and (in particular) your solution for the Z_6 creature. You should be able to find a second coarse-graining, with only two states.

Q1: Write down a distortion matrix that leads you to prefer the two-state model. Write down a distortion matrix that leads you to prefer the original three-state coarse-graining. In this case, don't worry about the mutual information term; just find a distortion matrix that makes one preferable to the other as a representation of the system.

Further Reading

- *Major Transitions in Political Order* discusses the coarse-graining procedure in social and cognitive environments where we have to consider not just how to construct a model, but also which of many possible coarse-grainings we might want to select. `https://arxiv.org/abs/1512.03419`.

- The article *Optimal high-level descriptions of dynamical systems* makes explicit the coarse-graining and renormalization story, showing how desires for accuracy and for efficient prediction and modeling can trade off each other. `https://arxiv.org/abs/1409.7403`

- Both of these articles can be found in the book *From Matter to Life: Information and Causality*, from Cambridge University Press, and edited by Sara Imari Walker, Paul C. W. Davies and George F. R. Ellis. `https://www.amazon.com/Matter-Life-Information-Causality/dp/1107150531/`

- In *The evolution of lossy compression*, Sarah Marzen and I dig into the rate-distortion story; see that article and references therein for an introduction to rate-distortion – and an account of how different utility functions (distortions) lead to different dynamics for the evolution of coarse-graining systems. `https://arxiv.org/abs/1506.06138` A lovely account of the utility function aspect of this theory, and how it connects to "live" human behavior can be found in an article by Chris R. Sims in the journal Cognition: —it Ratedistortion theory and human perception `http://www.sciencedirect.com/science/article/pii/S0010027716300750`