

Best of LessWrong: February 2014

1. [The January 2013 CFAR workshop: one-year retrospective](#)
2. [A Fervent Defense of Frequentist Statistics](#)
3. [Bridge Collapse: Reductionism as Engineering Problem](#)
4. [A self-experiment in training "noticing confusion"](#)
5. [White Lies](#)
6. [Beware Trivial Fears](#)
7. [Book Review: Linear Algebra Done Right \(MIRI course list\)](#)

Best of LessWrong: February 2014

1. [The January 2013 CFAR workshop: one-year retrospective](#)
2. [A Fervent Defense of Frequentist Statistics](#)
3. [Bridge Collapse: Reductionism as Engineering Problem](#)
4. [A self-experiment in training "noticing confusion"](#)
5. [White Lies](#)
6. [Beware Trivial Fears](#)
7. [Book Review: Linear Algebra Done Right \(MIRI course list\)](#)

The January 2013 CFAR workshop: one-year retrospective

[About a year ago](#), I attended my first CFAR workshop and wrote a post about it here. I mentioned in that post that it was too soon for me to tell if the workshop would have a large positive impact on my life. In the comments to that post, [I was asked](#) to follow up on that post in a year to better evaluate that impact. So here we are!

Very short summary: overall I think the workshop had a large and persistent positive impact on my life.

Important caveat

However, anyone using this post to evaluate the value of going to a CFAR workshop themselves should be aware that I'm local to Berkeley and have had many opportunities to stay connected to CFAR and the rationalist community. More specifically, in addition to the January workshop, I also

- visited the March workshop (and possibly others),
- attended various social events held by members of the community,
- taught at the July workshop, and
- taught at [SPARC](#).

These experiences were all very helpful in helping me digest and reinforce the workshop material (which was also improving over time), and a typical workshop participant might not have these advantages.

Answering a question

[pewpewlasergun](#) wanted me to answer the following question:

I'd like to know how many techniques you were taught at the meetup you still use regularly. Also which has had the largest effect on your life.

The short answer is: in some sense very few, but a lot of the value I got out of attending the workshop didn't come from specific techniques.

In more detail: to be honest, many of the specific techniques are kind of a chore to use (at least as of January 2013). I experimented with a good number of them in the months after the workshop, and most of them haven't stuck (but that isn't so bad; the cost of trying a technique and finding that it doesn't work for you is low, while the benefit of trying a technique and finding that it does work for you can be quite high!). One that has is the idea of a **next action**, which I've found incredibly useful. Next actions are the things that to-do list items should be, say in the context of using [Remember The Milk](#). Many to-do list items you might be tempted to right down are difficult to actually do because they're either too vague or too big and hence trigger [ugh fields](#). For example, you might have an item like

- Do my taxes

that you don't get around to until right before you have to because you have an ough field around doing your taxes. This item is both too vague and too big: instead of writing this down, write down the **next physical action** you need to take to make progress on this item, which might be something more like

- Find tax forms and put them on desk

which is both concrete and small. Thinking in terms of next actions has been a huge upgrade to my [GTD system](#) (as was [Workflowy](#), which I also started using because of the workshop) and I do it constantly.

But as I mentioned, a lot of the value I got out of attending the workshop was not from specific techniques. Much of the value comes from spending time with the workshop instructors and participants, which had effects that I find hard to summarize, but I'll try to describe some of them below:

Emotional attitudes

The workshop readjusted my emotional attitudes towards several things for the better, and at several meta levels. For example, a short conversation with a workshop alum completely readjusted my emotional attitude towards both nutrition and exercise, and I started paying more attention to what I ate and going to the gym (albeit sporadically) for the first time in my life not long afterwards. I lost about 15 pounds this way (mostly from the eating part, not the gym part, I think).

At a higher meta level, I did a fair amount of experimenting with various lifestyle changes (cold showers, not shampooing) after the workshop and overall they had the effect of readjusting my emotional attitude towards change. I find it generally easier to change my behavior than I used to because I've had a lot of practice at it lately, and am more enthusiastic about the prospect of such changes.

(Incidentally, I think emotional attitude adjustment is an underrated component of causing people to change their behavior, at least here on LW.)

Using all of my strength

The workshop is the first place I really understood, on a gut level, that I could use my brain to think about something other than math. It sounds silly when I phrase it like that, but at some point in the past I had incorporated into my identity that I was good at math but absentminded and silly about real-world matters, and I used it as an excuse not to fully engage intellectually with anything that wasn't math, especially anything practical. One way or another the workshop helped me realize this, and I stopped thinking this way.

The result is that I constantly apply optimization power to situations I wouldn't have even tried to apply optimization power to before. For example, today I was trying to figure out why the water in my bathroom sink was draining so slowly. At first I thought it was because the strainer had become clogged with gunk, so I cleaned the strainer, but then I found out that even with the strainer removed the water was still draining slowly. In the past I might've given up here. Instead I looked around for something that would fit farther into the sink than my fingers and saw the handle of my plunger. I pumped the handle into the sink a few times and some extra gunk I hadn't known was there came out. The sink is fine now. (This might seem small to people who are more

domestically talented than me, but trust me when I say I wasn't doing stuff like this before last year.)

Reflection and repair

Thanks to the workshop, my GTD system is now robust enough to consistently enable me to reflect on and repair my life (including my GTD system). For example, I'm quicker to attempt to deal with minor medical problems I have than I used to be. I also think more often about what I'm doing and whether I could be doing something better. In this regard I pay a lot of attention in particular to what habits I'm forming, although I don't use the specific techniques in the relevant CFAR unit.

For example, at some point I had recorded in RTM that I was frustrated by the sensation of hours going by without remembering how I had spent them (usually because I was mindlessly browsing the internet). In response, I started keeping a record of what I was doing every half hour and categorizing each hour according to a combination of how productively and how intentionally I spent it (in the first iteration it was just how productively I spent it, but I found that this was making me feel too guilty about relaxing). For example:

- a half-hour intentionally spent reading a paper is marked green.
- a half-hour half-spent writing up solutions to a problem set and half-spent on Facebook is marked yellow.
- a half-hour intentionally spent playing a video game is marked with no color.
- a half-hour mindlessly browsing the internet when I had intended to do work is marked red.

The act of doing this every half hour itself helps make me more mindful about how I spend my time, but having a record of how I spend my time has also helped me notice interesting things, like how less of my time is under my direct control than I had thought (but instead is taken up by classes, commuting, eating, etc.). It's also easier for me to get into a [success spiral](#) when I see a lot of green.

Stimulation

Being around workshop instructors and participants is consistently intellectually stimulating. I don't have a tactful way of saying what I'm about to say next, but: two effects of this are that I think more interesting thoughts than I used to and also that I'm funnier than I used to be. (I realize that these are both hard to quantify.)

etc.

I worry that I haven't given a complete picture here, but hopefully anything I've left out will be brought up in the comments one way or another. (Edit: this totally happened! Please read [Anna Salamon's comment below](#).)

Takeaway for prospective workshop attendees

I'm not actually sure what you should take away from all this if your goal is to figure out whether you should [attend a workshop yourself](#). My thoughts are roughly this: I think attending a workshop is potentially high-value and therefore that even [talking to CFAR about any questions you might have](#) is potentially high-value, in addition to

being relatively low-cost. If you think there's even a small chance you could get a lot of value out of attending a workshop I recommend that you at least take that one step.

A Fervent Defense of Frequentist Statistics

[Highlights for the busy: de-bunking standard "Bayes is optimal" arguments; frequentist Solomonoff induction; and a description of the online learning framework. Note: cross-posted from my [blog](#).]

Short summary. This essay makes many points, each of which I think is worth reading, but if you are only going to understand one point I think it should be "Myth 5" below, which describes the online learning framework as a response to the claim that frequentist methods need to make strong modeling assumptions. Among other things, online learning allows me to perform the following remarkable feat: if I'm betting on horses, and I get to place bets after watching other people bet but before seeing which horse wins the race, then I can guarantee that after a relatively small number of races, I will do almost as well overall as the best other person, even if the number of other people is very large (say, 1 billion), and their performance is correlated in complicated ways.

If you're only going to understand two points, then also read about the frequentist version of Solomonoff induction, which is described in "Myth 6".

Main article. I've already written one essay on [Bayesian vs. frequentist statistics](#). In that essay, I argued for a balanced, pragmatic approach in which we think of the two families of methods as a collection of tools to be used as appropriate. Since I'm currently feeling contrarian, this essay will be far less balanced and will argue explicitly against Bayesian methods and in favor of frequentist methods. I hope this will be forgiven as so much other writing goes in the opposite direction of unabashedly defending Bayes. I should note that this essay is partially inspired by some of Cosma Shalizi's blog posts, such as [this](#) one.

This essay will start by listing a series of myths, then debunk them one-by-one. My main motivation for this is that Bayesian approaches seem to be highly popularized, to the point that one may get the impression that they are the uncontroversially superior method of doing statistics. I actually think the opposite is true: I think most statisticians would for the most part defend frequentist methods, although there are also many departments that are decidedly Bayesian (e.g. many places in England, as well as some U.S. universities like Columbia). I have a lot of respect for many of the people at these universities, such as Andrew Gelman and Philip Dawid, but I worry that many of the other proponents of Bayes (most of them non-statisticians) tend to oversell Bayesian methods or undersell alternative methodologies.

If you are like me from, say, two years ago, you are firmly convinced that Bayesian methods are superior and that you have knockdown arguments in favor of this. If this is the case, then I hope this essay will give you an experience that I myself found life-altering: the experience of having a way of thinking that seemed unquestionably true slowly dissolve into just one of many imperfect models of reality. This experience helped me gain more explicit appreciation for the skill of viewing the world from many different angles, and of distinguishing between a very successful paradigm and reality.

If you are not like me, then you may have had the experience of bringing up one of many reasonable objections to normative Bayesian epistemology, and having it shot down by one of many "standard" arguments that seem wrong but not for easy-to-

articulate reasons. I hope to lend some reprieve to those of you in this camp, by providing a collection of “standard” replies to these standard arguments.

I will start with the myths (and responses) that I think will require the least technical background and be most interesting to a general audience. Toward the end, I deal with some attacks on frequentist methods that I believe amount to technical claims that are demonstrably false; doing so involves more math. Also, I should note that for the sake of simplicity I’ve labeled everything that is non-Bayesian as a “frequentist” method, even though I think there’s actually a fair amount of variation among these methods, although also a fair amount of overlap (e.g. I’m throwing in statistical learning theory with minimax estimation, which certainly have a lot of overlap in ideas but were also in some sense developed by different communities).

The Myths:

- Bayesian methods are optimal.
- Bayesian methods are optimal except for computational considerations.
- We can deal with computational constraints simply by making approximations to Bayes.
- The prior isn’t a big deal because Bayesians can always share likelihood ratios.
- Frequentist methods need to assume their model is correct, or that the data are i.i.d.
- Frequentist methods can only deal with simple models, and make arbitrary cutoffs in model complexity (aka: “I’m Bayesian because I want to do Solomonoff induction”).
- Frequentist methods hide their assumptions while Bayesian methods make assumptions explicit.
- Frequentist methods are fragile, Bayesian methods are robust.
- Frequentist methods are responsible for bad science
- Frequentist methods are unprincipled/hacky.
- Frequentist methods have no promising approach to computationally bounded inference.

Myth 1: Bayesian methods are optimal. Presumably when most people say this they are thinking of either Dutch-bookings or the complete class theorem. Roughly what these say are the following:

Dutch-book argument: Every coherent set of beliefs can be modeled as a subjective probability distribution. (Roughly, coherent means “unable to be Dutch-booked”.)

Complete class theorem: Every non-Bayesian method is worse than some Bayesian method (in the sense of performing deterministically at least as poorly in every possible world).

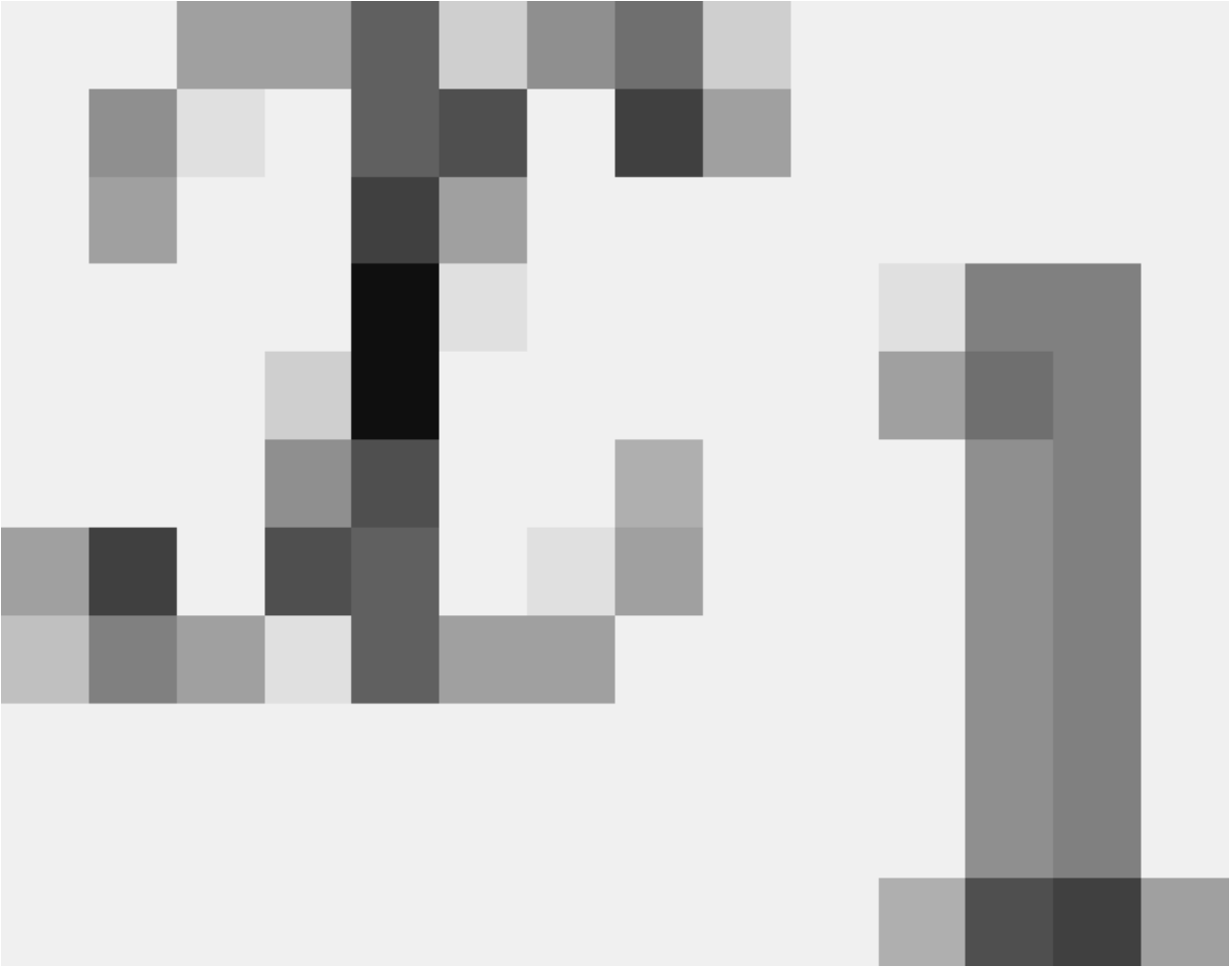
Let’s unpack both of these. My high-level argument regarding Dutch books is that I would much rather spend my time trying to correspond with reality than trying to be internally consistent. More concretely, the Dutch-book argument says that if for every bet you force me to take one side or the other, then unless I’m Bayesian there’s a collection of bets that will cause me to lose money for sure. I don’t find this very compelling. This seems analogous to the situation where there’s some quant at Jane Street, and they’re about to run code that will make thousands of dollars trading stocks, and someone comes up to them and says “Wait! You should add checks to your code to make sure that no subset of your trades will lose you money!” This just doesn’t seem worth the quant’s time, it will slow down the code substantially, and instead the

quant should be writing the next program to make thousands more dollars. This is basically what dutch-book arguments seem like to me.

Moving on, the complete class theorem says that for any decision rule, I can do better by replacing it with *some* Bayesian decision rule. But this injunction is not useful in practice, because it doesn't say anything about *which* decision rule I should replace it with. Of course, if you hand me a decision rule and give me infinite computational resources, then I can hand you back a Bayesian method that will perform better. But it still might not perform **well**. All the complete class theorem says is that every local optimum is Bayesian. To be a useful theory of epistemology, I need a prescription for how, in the first place, I am to arrive at a *good* decision rule, *not* just a locally optimal one. And this is something that frequentist methods do provide, to a far greater extent than Bayesian methods (for instance by using minimax decision rules such as the maximum-entropy example given later). Note also that many frequentist methods *do* correspond to a Bayesian method for some appropriately chosen prior. But the crucial point is that the frequentist *told* me how to pick a prior I would be happy with (also, many frequentist methods *don't* correspond to a Bayesian method for any choice of prior; they nevertheless often perform quite well).

Myth 2: Bayesian methods are optimal except for computational considerations. We already covered this in the previous point under the complete class theorem, but to re-iterate: **Bayesian methods are *locally* optimal, *not* global optimal. Identifying all the local optima is very different from knowing which of them is the global optimum.** I would much rather have someone hand me something that wasn't a local optimum but was close to the global optimum, than something that was a local optimum but was far from the global optimum.

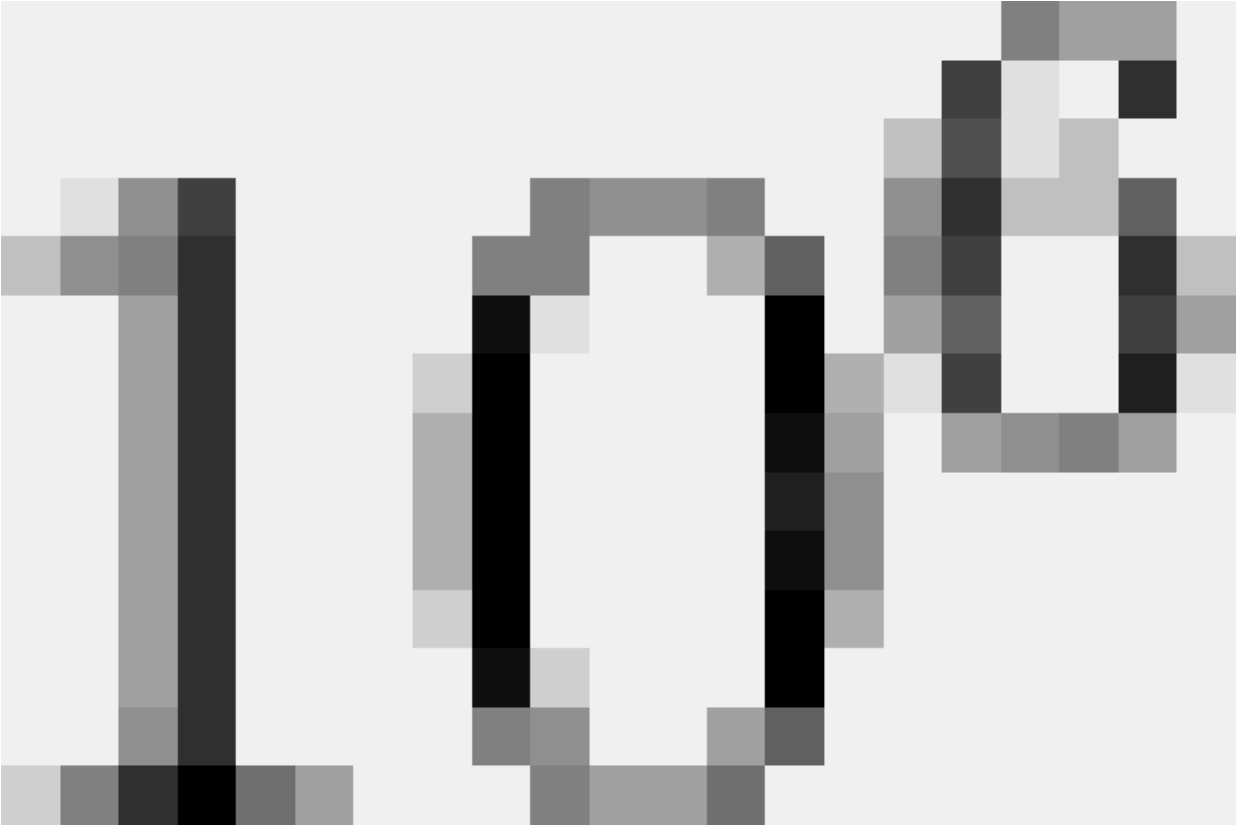
Myth 3: We can deal with computational constraints simply by making approximations to Bayes. I have rarely seen this born out in practice. Here's a challenge: suppose I give you data generated in the following way. There are a collection of vectors



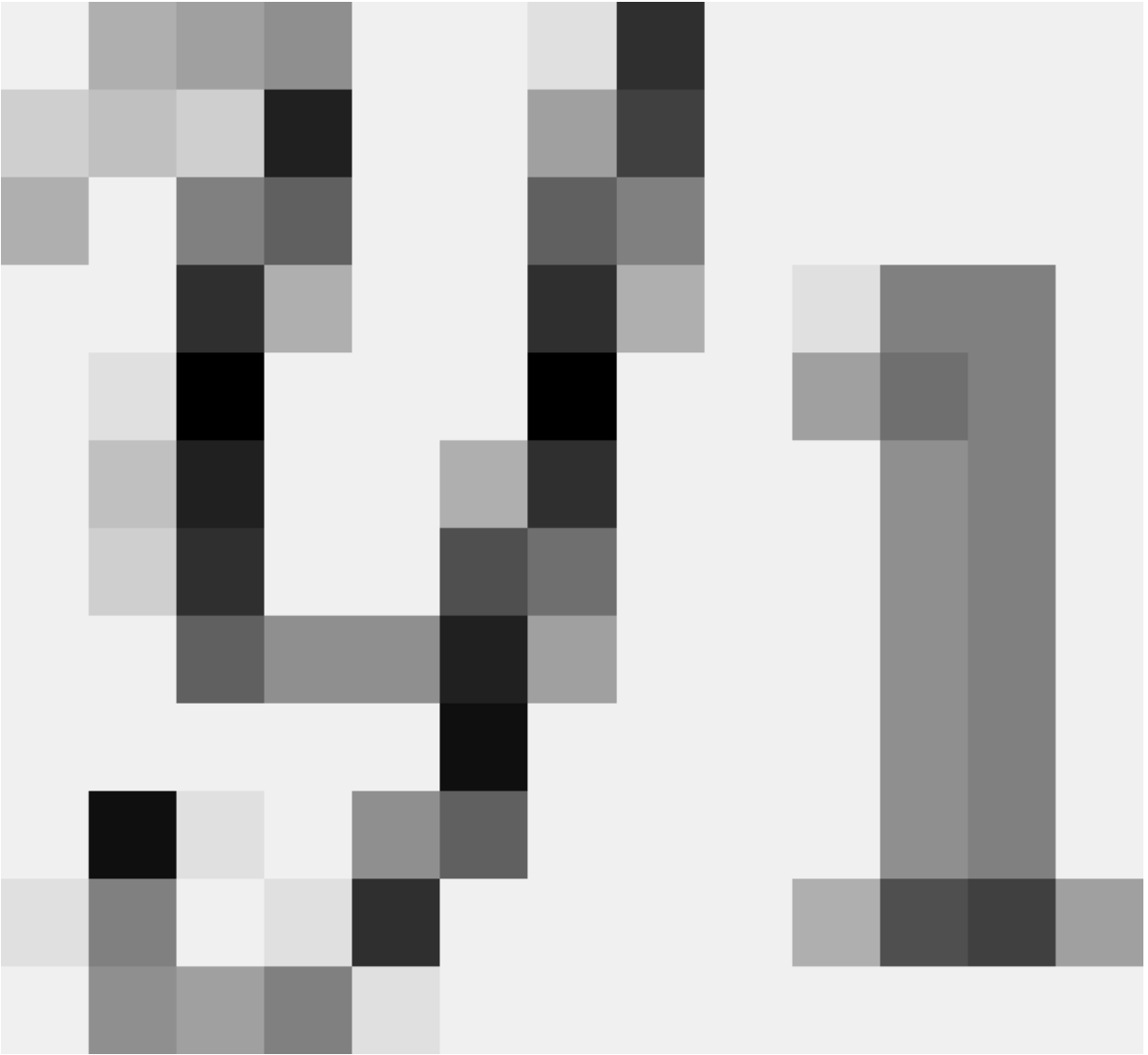
,



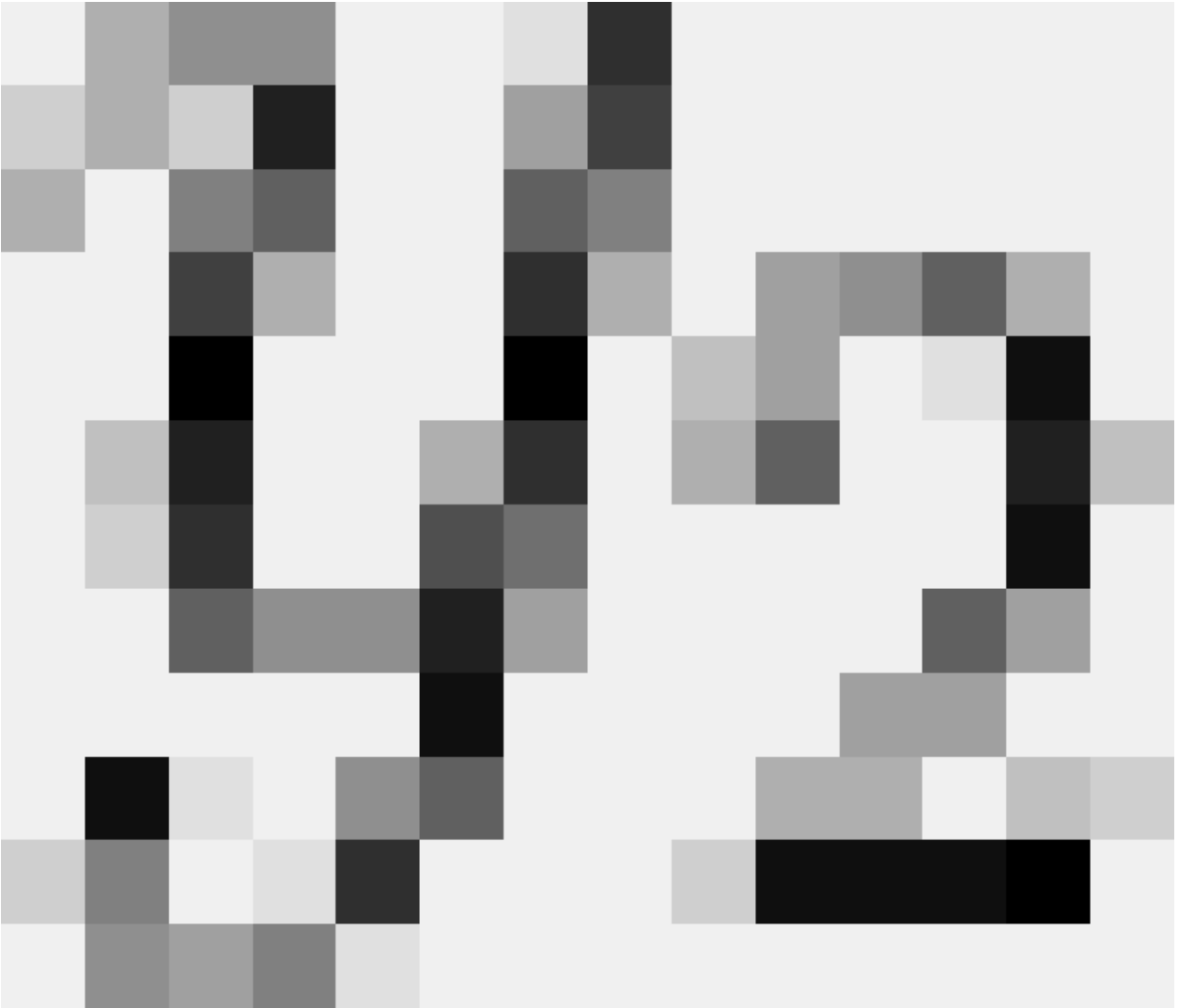
, each with



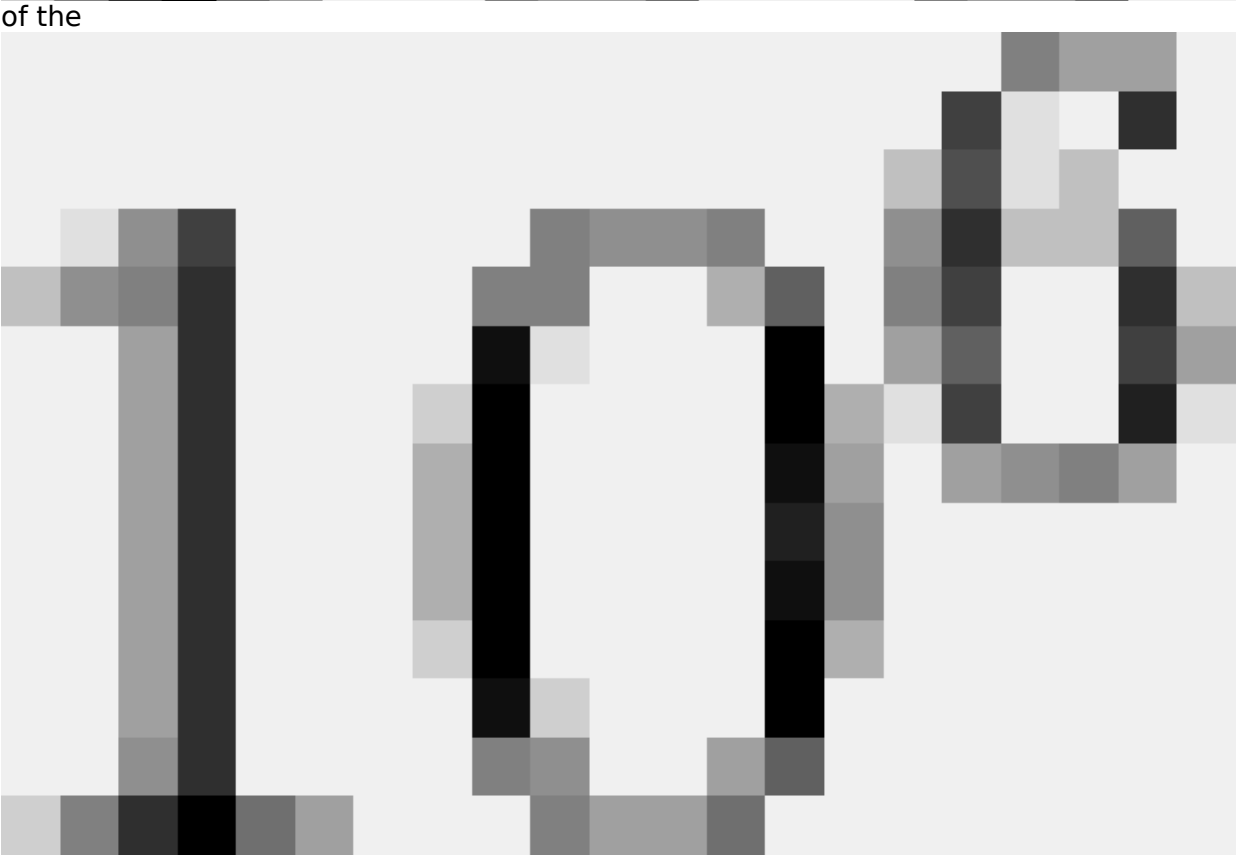
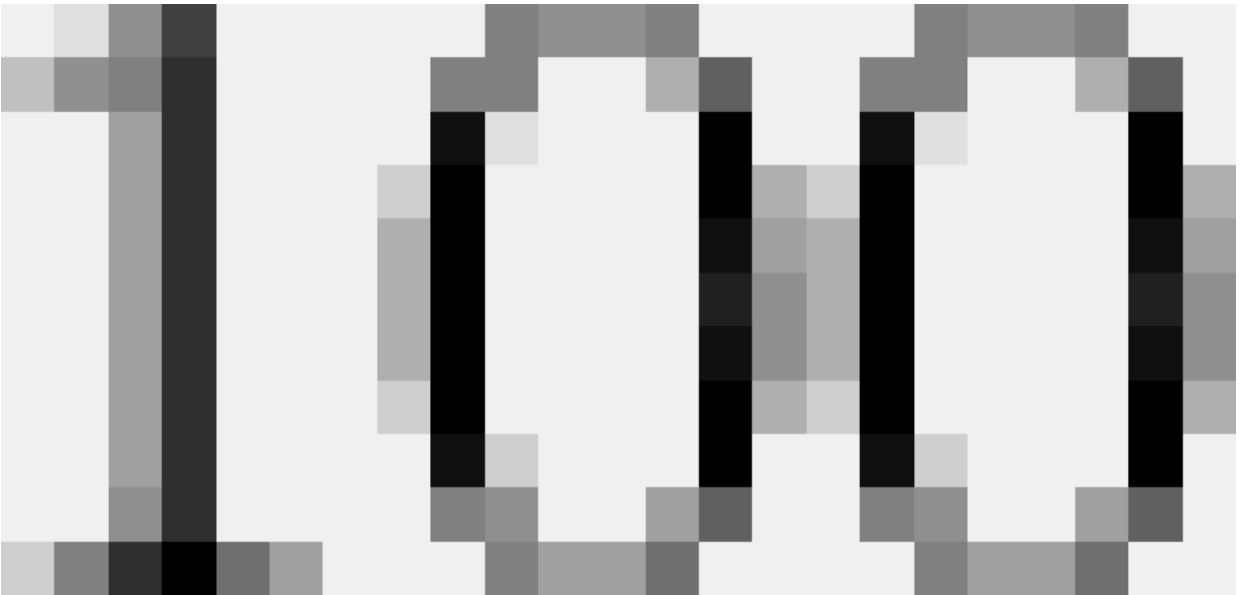
coordinates. I generate outputs



,

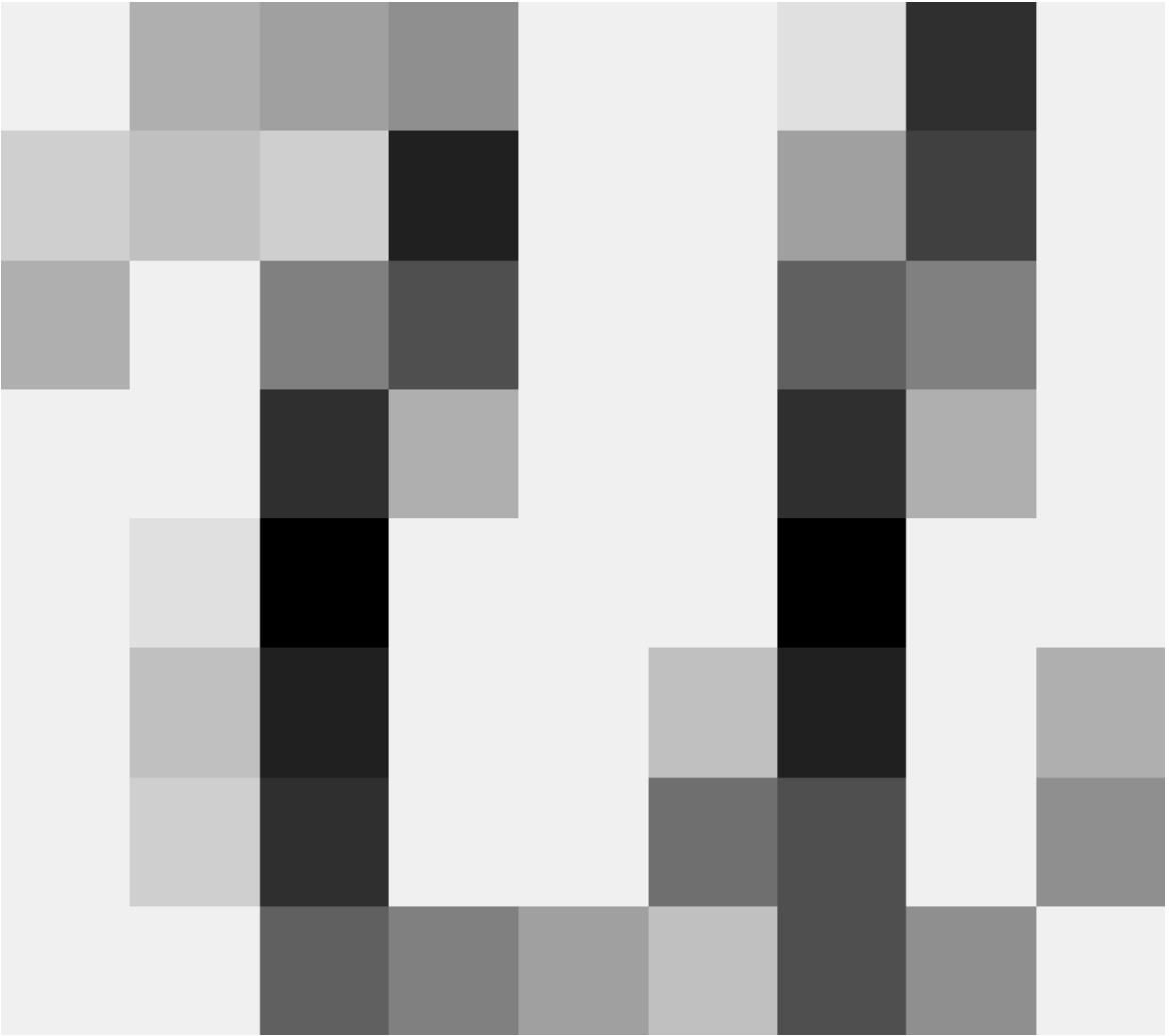


in the following way. First I globally select

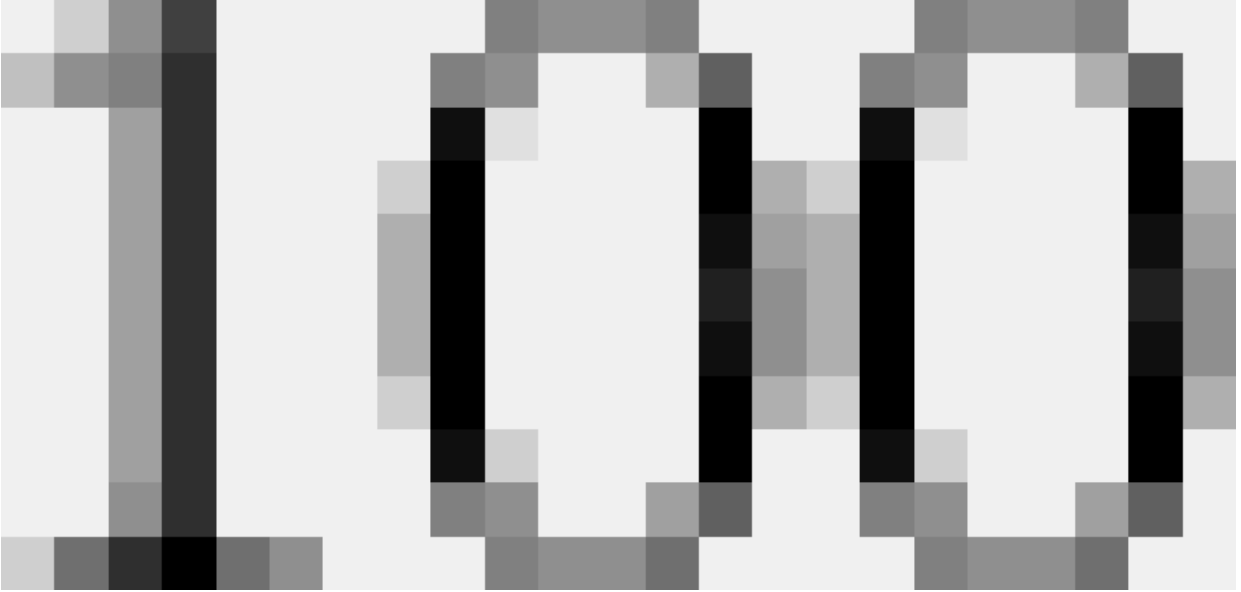


of the

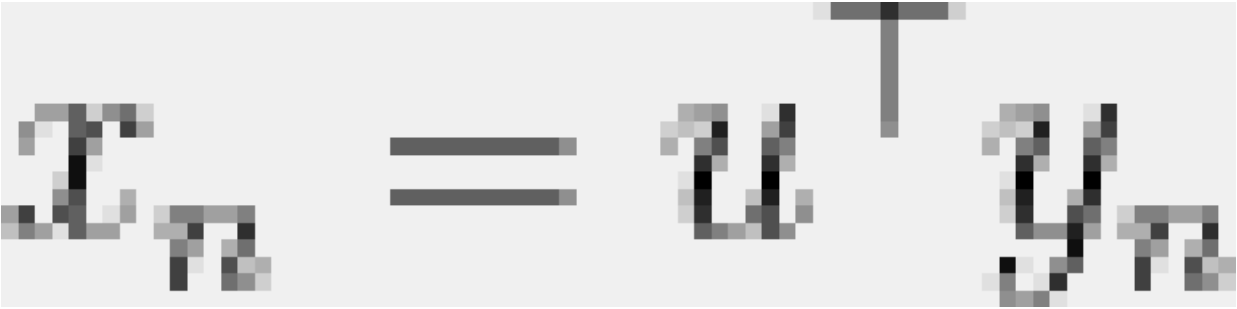
coordinates uniformly at random, then I select a fixed vector



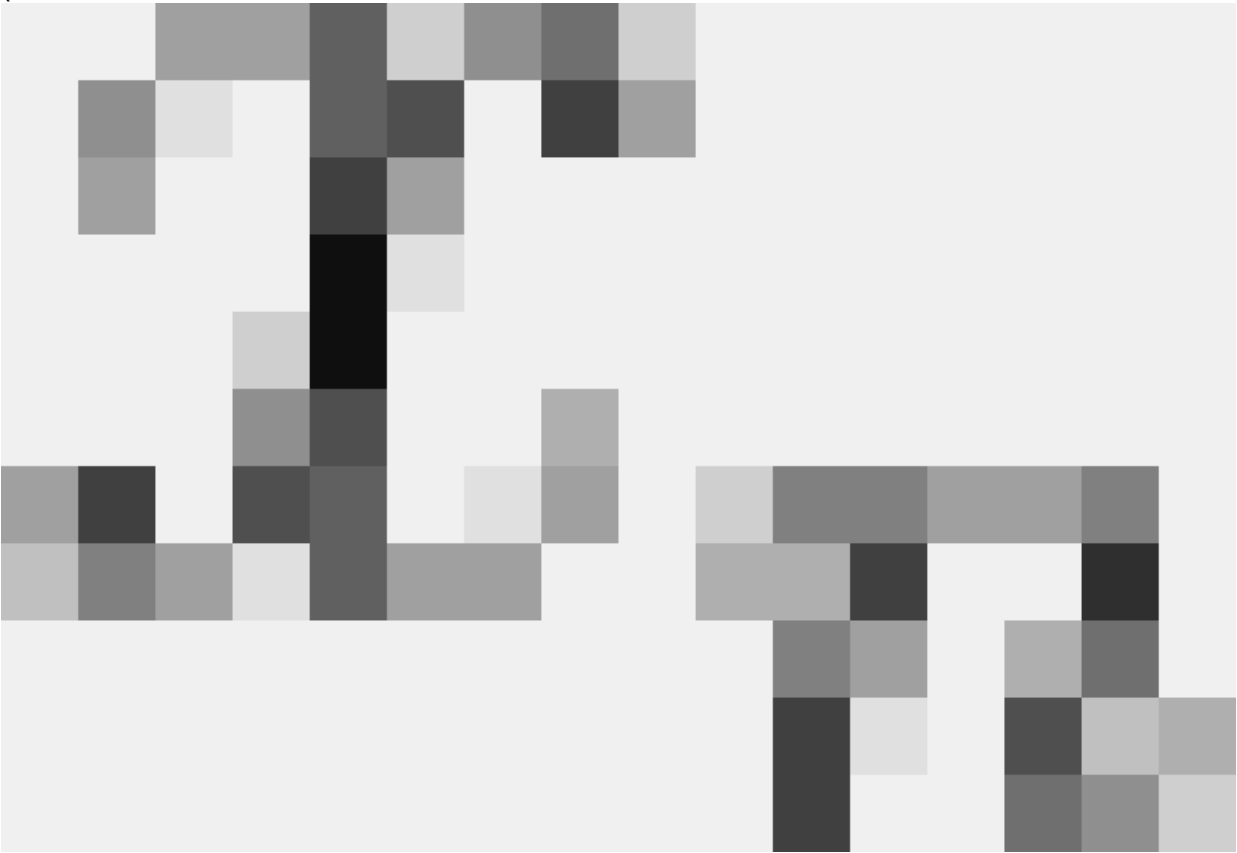
such that those



coordinates are drawn from i.i.d. Gaussians and the rest of the coordinates are zero.
Now I set

A pixelated, grayscale image of the equation $x_n = w^T y_n$. The characters are composed of small squares in various shades of gray, giving it a low-resolution, digital-art appearance.

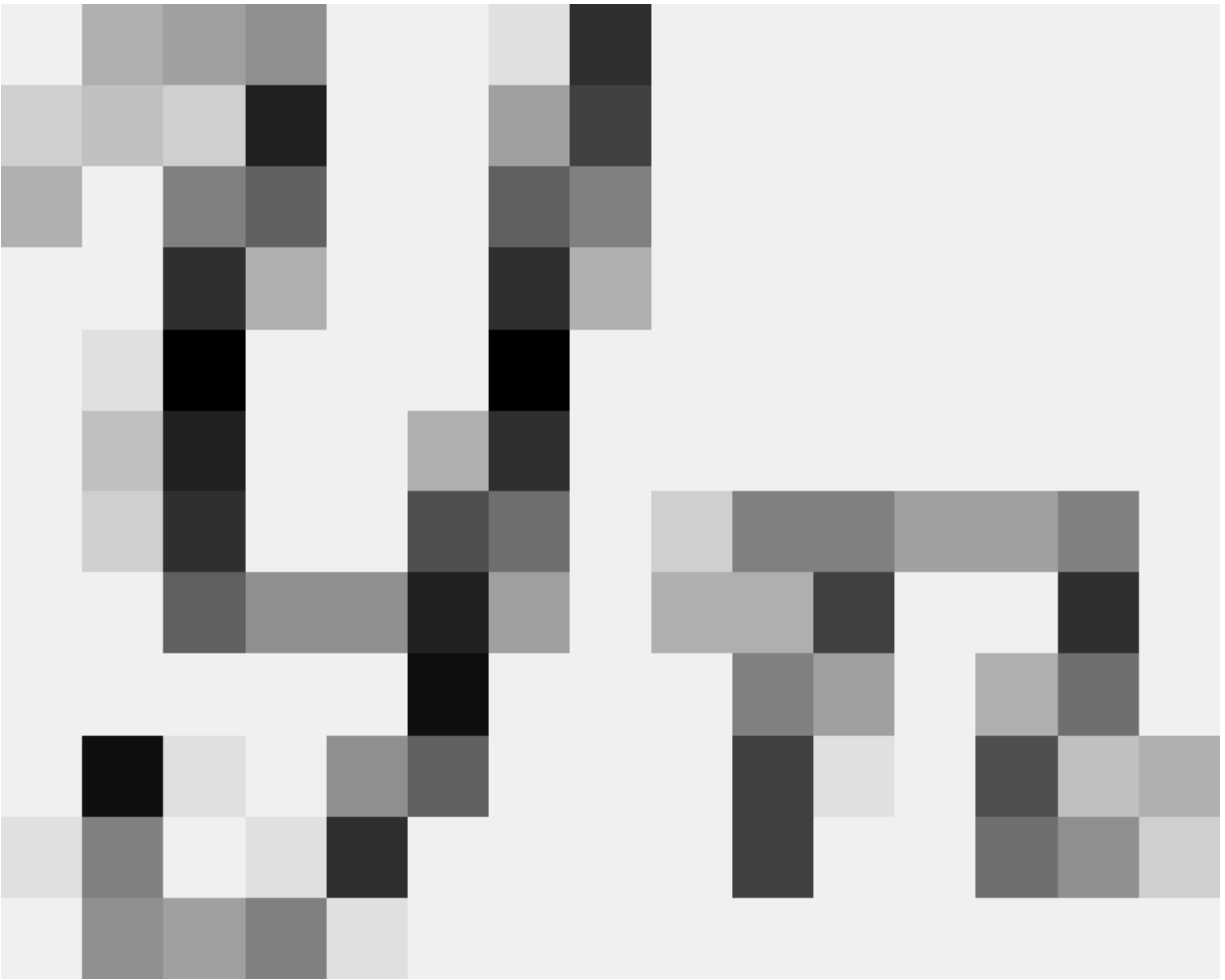
(i.e.



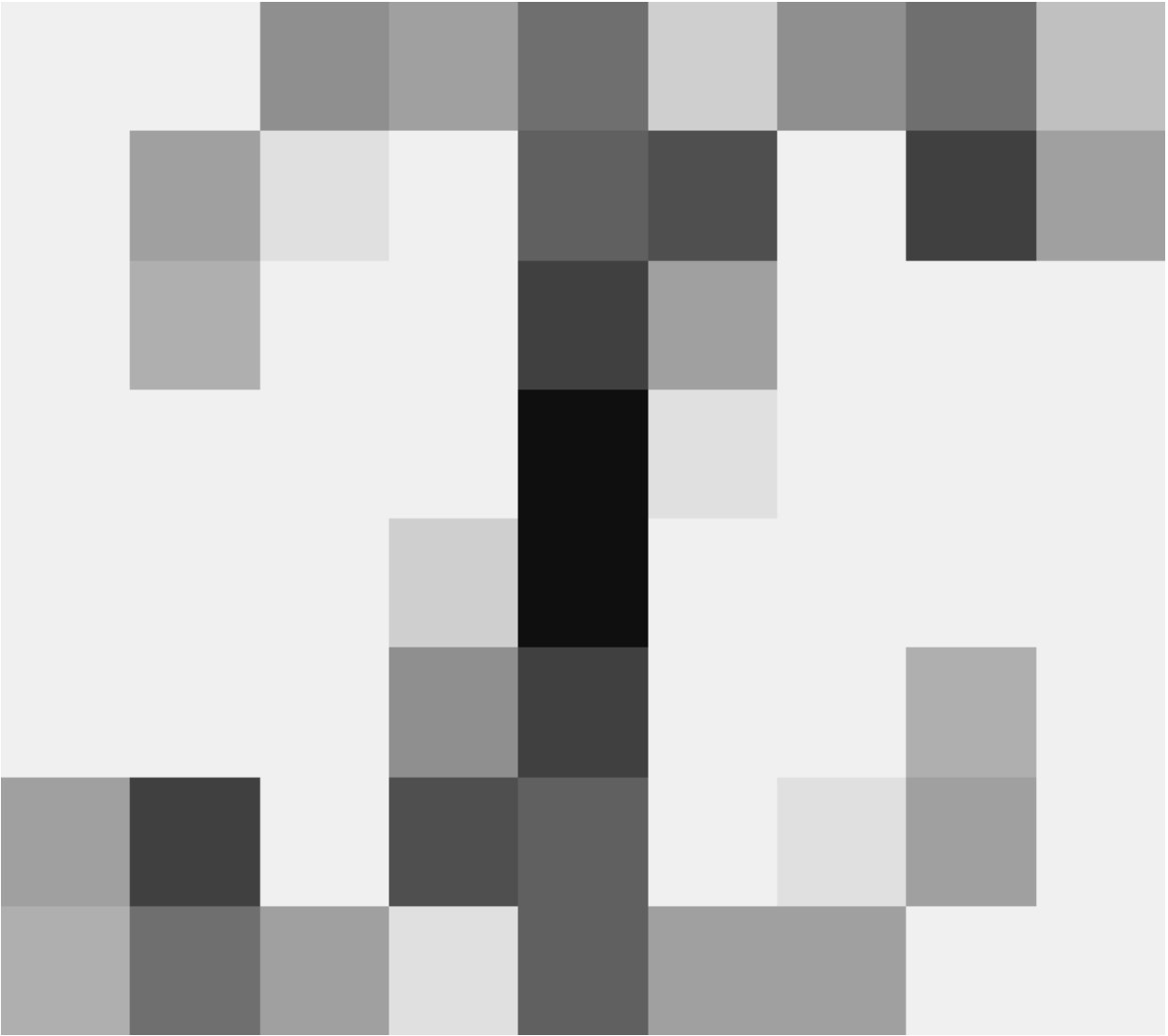
is the dot product of

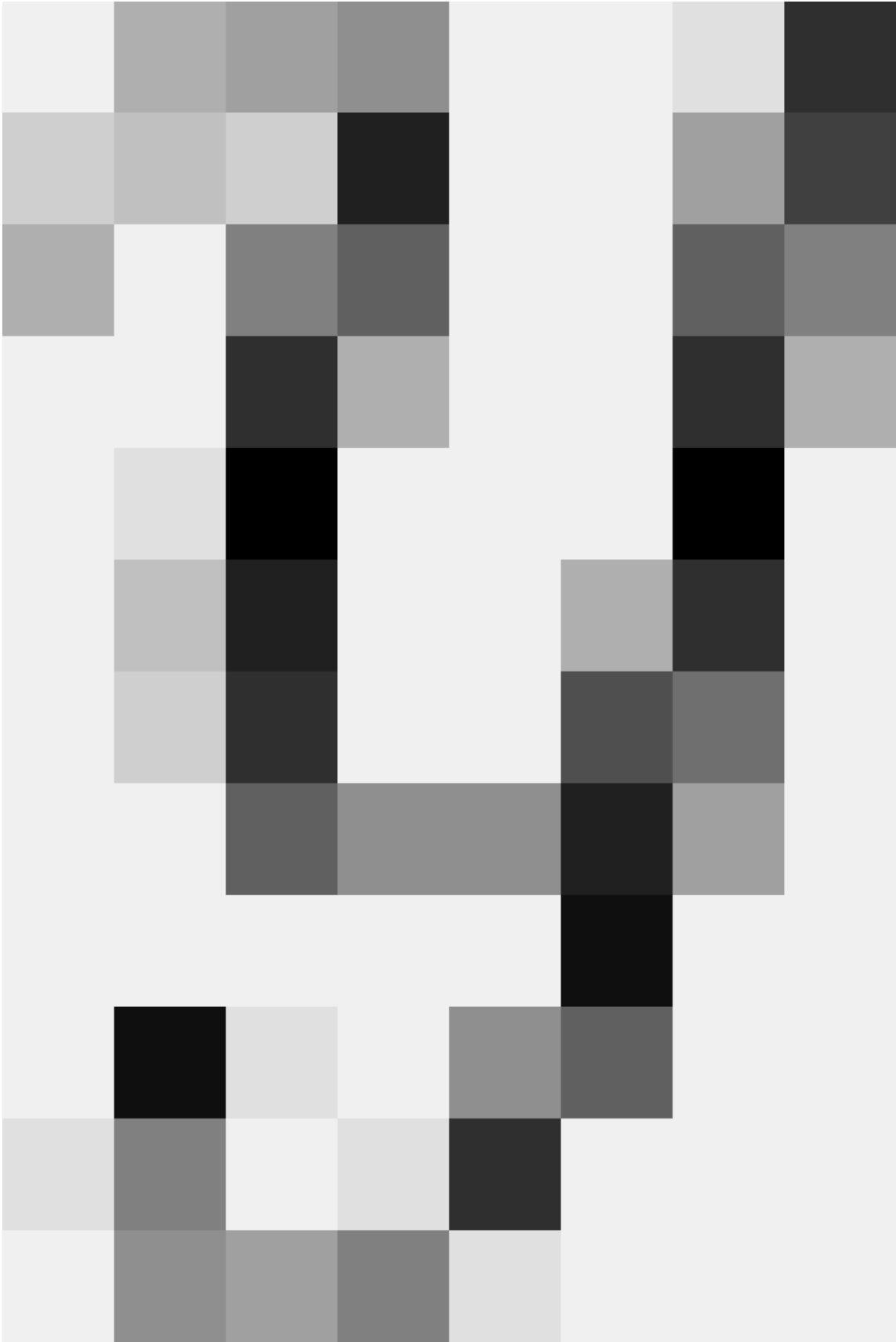


with



). You are given





and

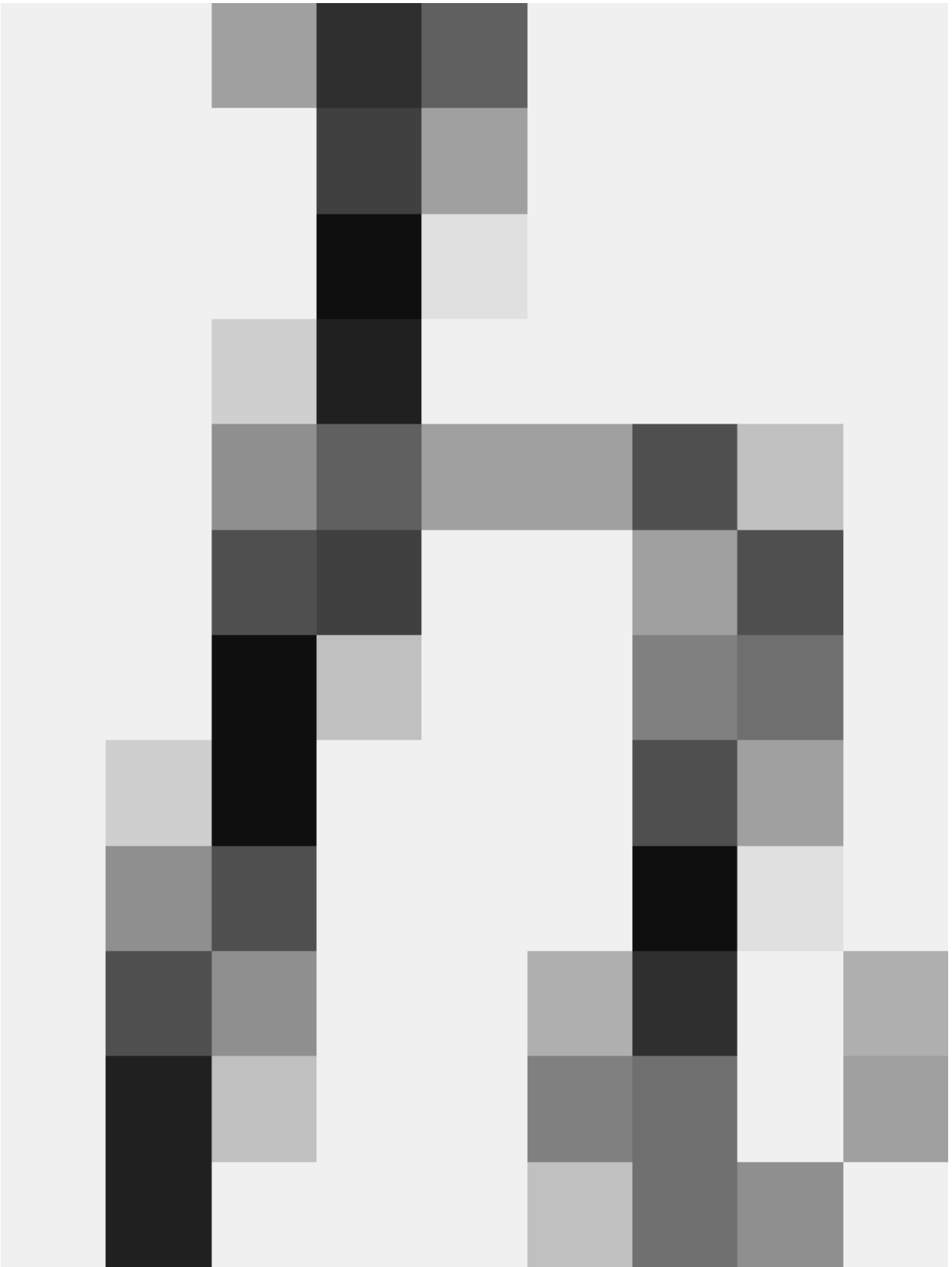
and your job is to infer



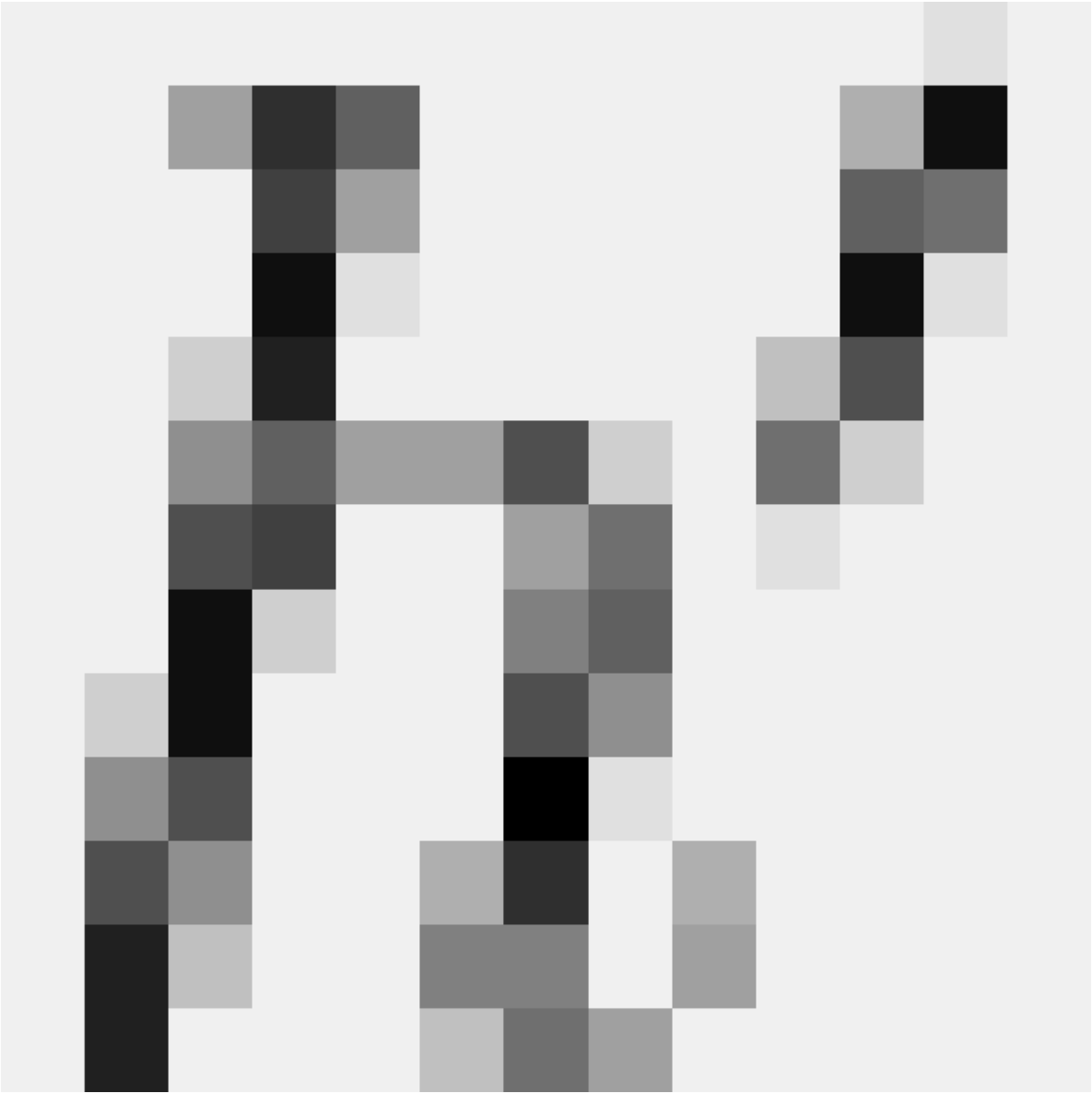
. This is a completely well-specified problem, the only task remaining is computational. I know people who have solved this problem using Bayesian methods with approximate inference. I have respect for these people, because doing so is no easy task. I think very few of them would say that “we can just approximate Bayesian updating and be fine”. (Also, this particular problem can be solved trivially with frequentist methods.)

A particularly egregious example of this is when people talk about “computable approximations to Solomonoff induction” or “computable approximations to AIXI” as if such notions were meaningful.

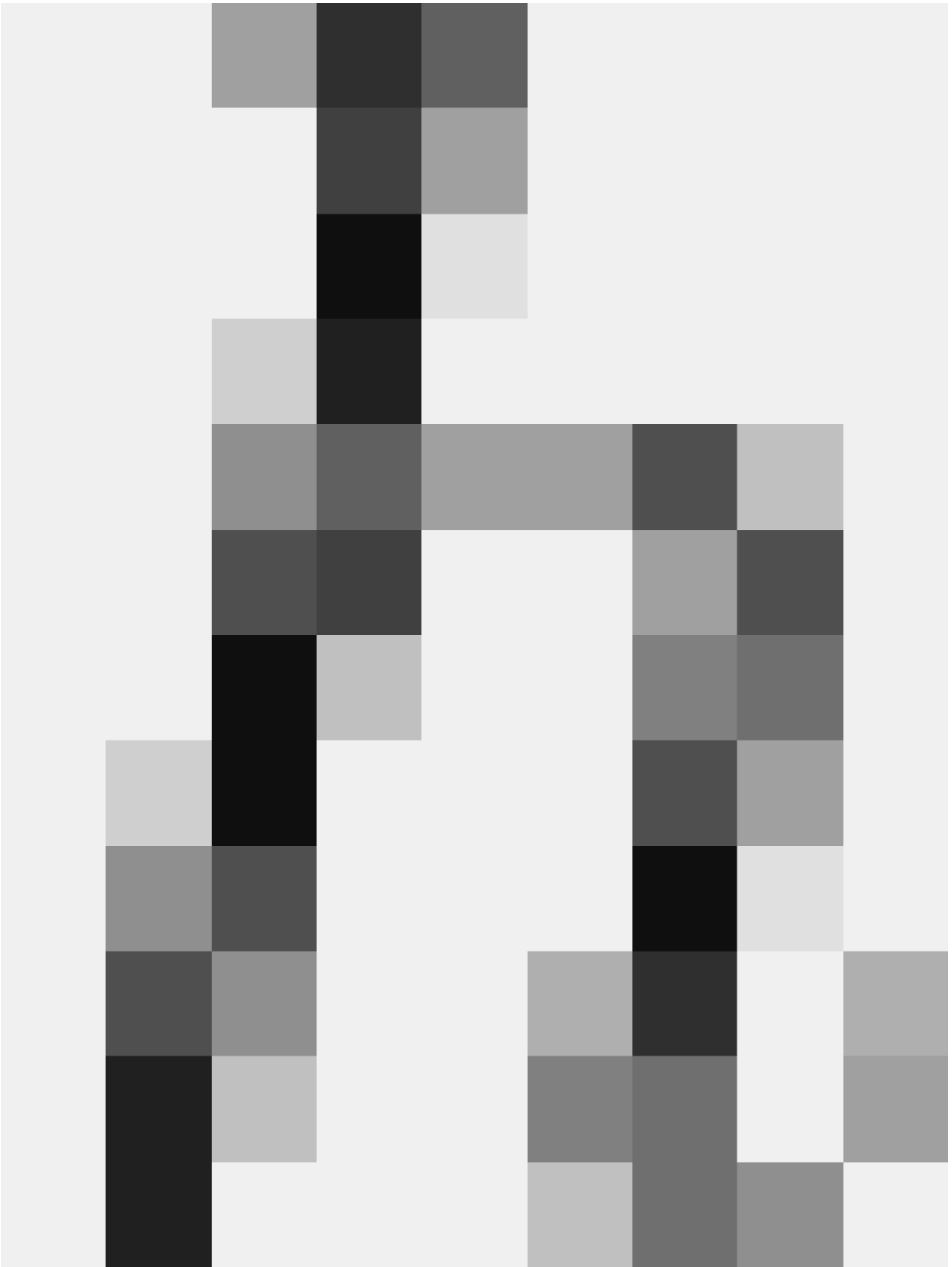
Myth 4: the prior isn't a big deal because Bayesians can always share likelihood ratios. Putting aside the practical issue that there would in general be an infinite number of likelihood ratios to share, there is the larger issue that for any hypothesis



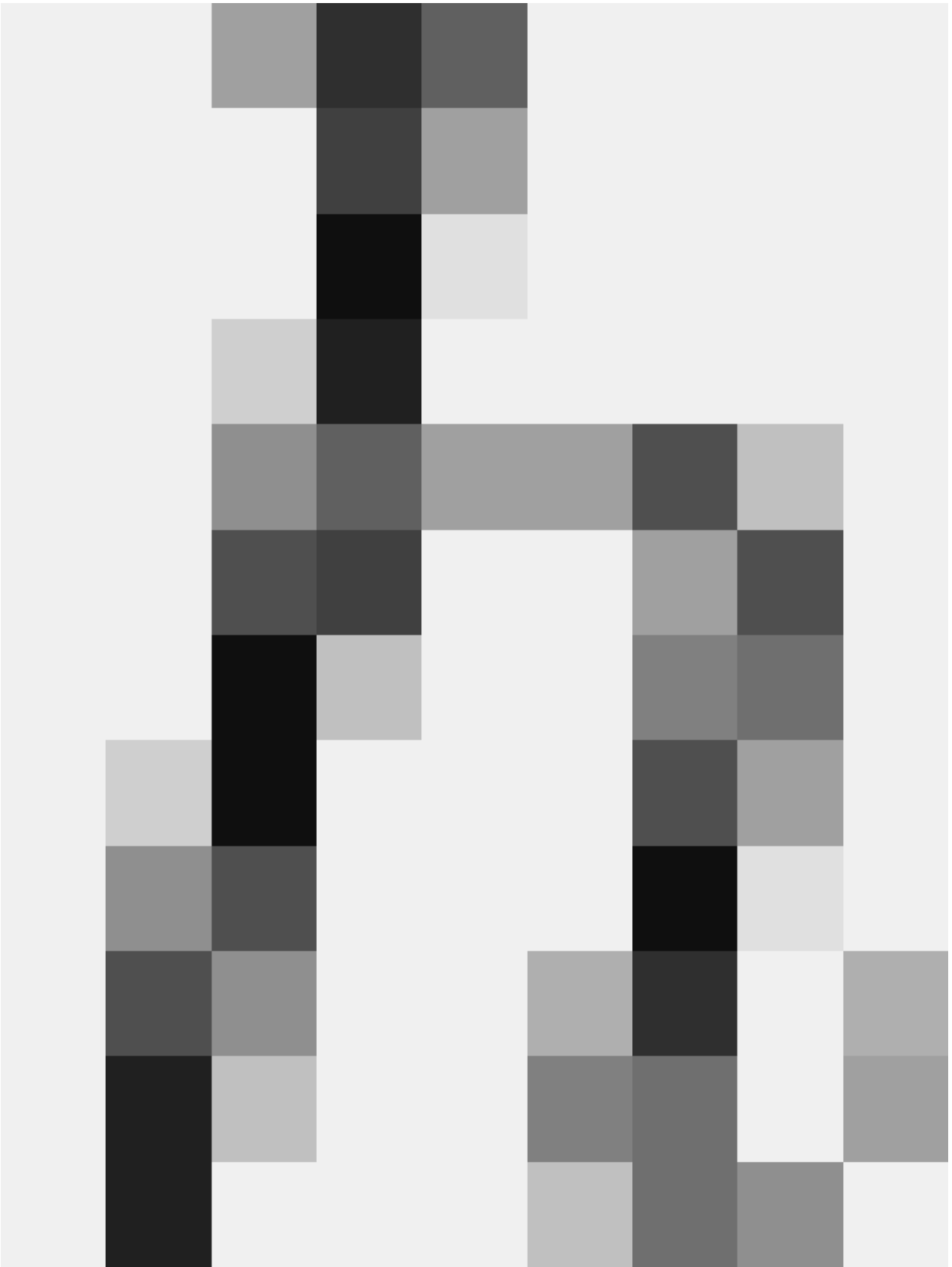
, there is also the hypothesis



that matches



exactly up to now, and then predicts the opposite of



at all points in the future. You have to constrain model complexity at some point, the question is about how. To put this another way, sharing my likelihood ratios without

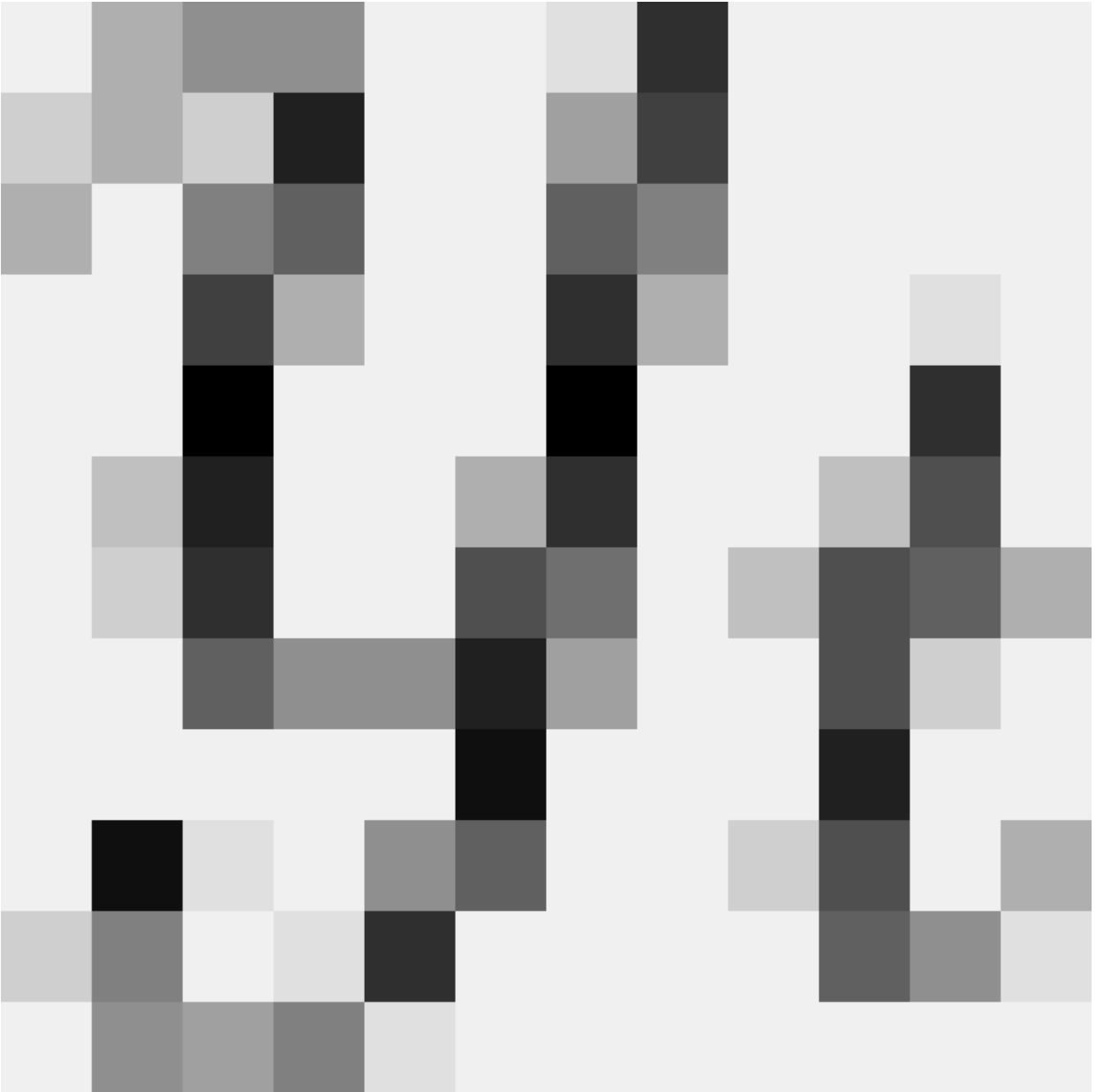
also constraining model complexity (by focusing on a subset of all logically possible hypotheses) would be equivalent to just sharing all sensory data I've ever accrued in my life. To the extent that such a notion is even possible, I certainly don't need to be a Bayesian to do such a thing.

Myth 5: frequentist methods need to assume their model is correct or that the data are i.i.d. **Understanding the content of this section is the most important single insight to gain from this essay.** For some reason it's assumed that frequentist methods need to make strong assumptions (such as Gaussianity), whereas Bayesian methods are somehow immune to this. In reality, the opposite is true. While there are many beautiful and deep frequentist formalisms that answer this, I will choose to focus on one of my favorite, which is **online learning**.

To explain the online learning framework, let us suppose that our data are

$$(x_1, y_1), (x_2, y_2), \dots, (x_T, y_T)$$

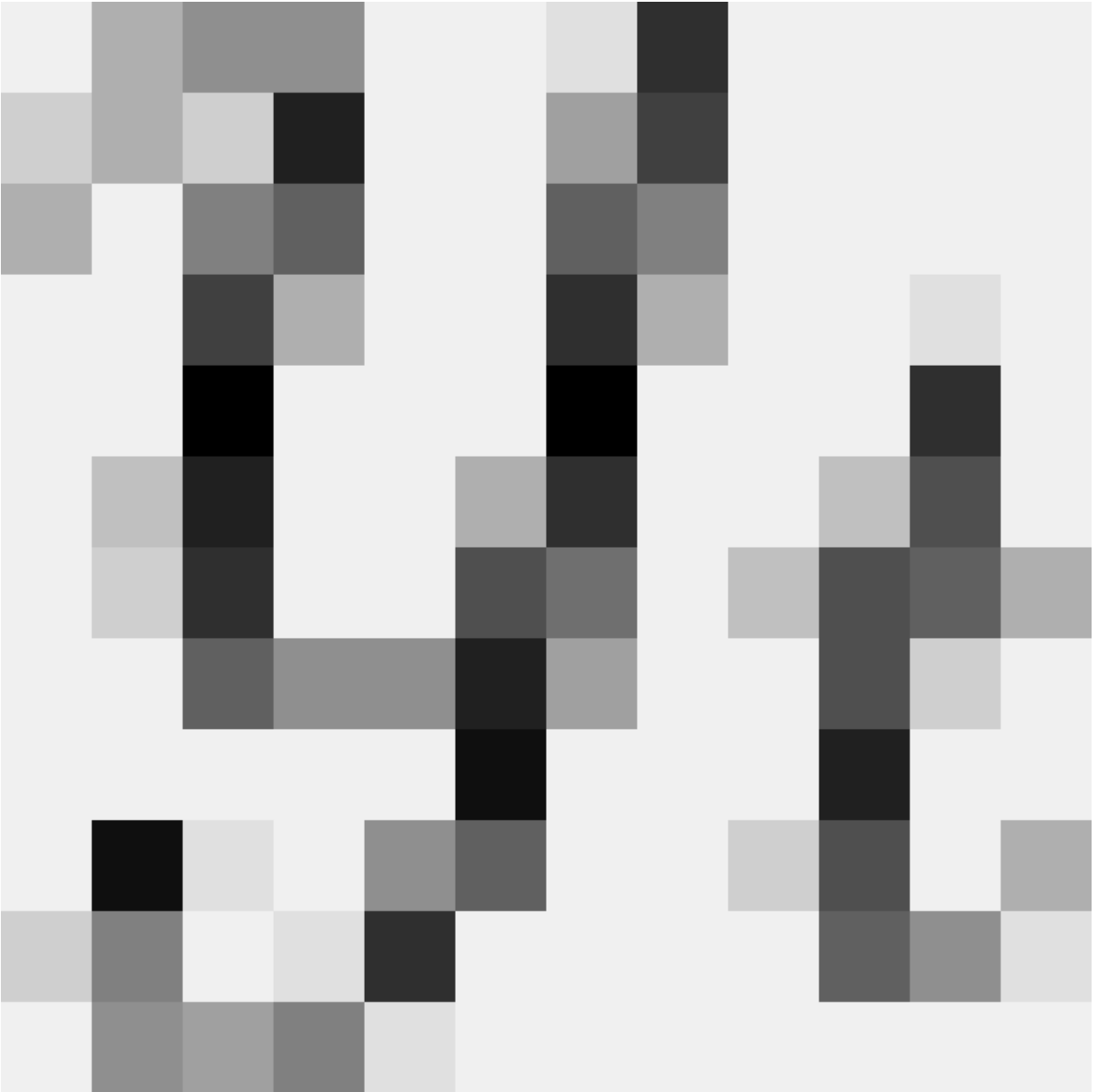
. We don't observe



until after making a prediction



of what



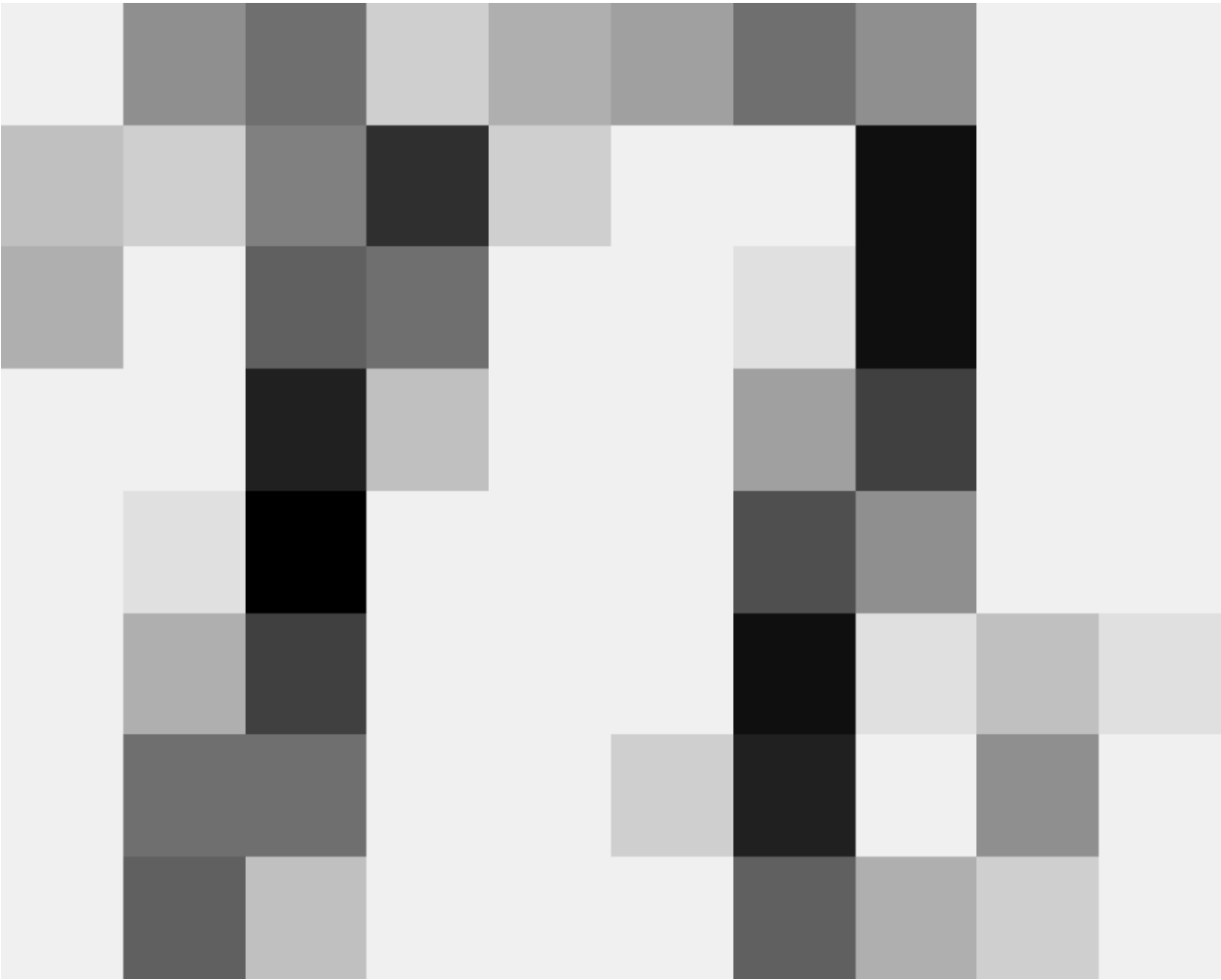
will be, and then we receive a penalty



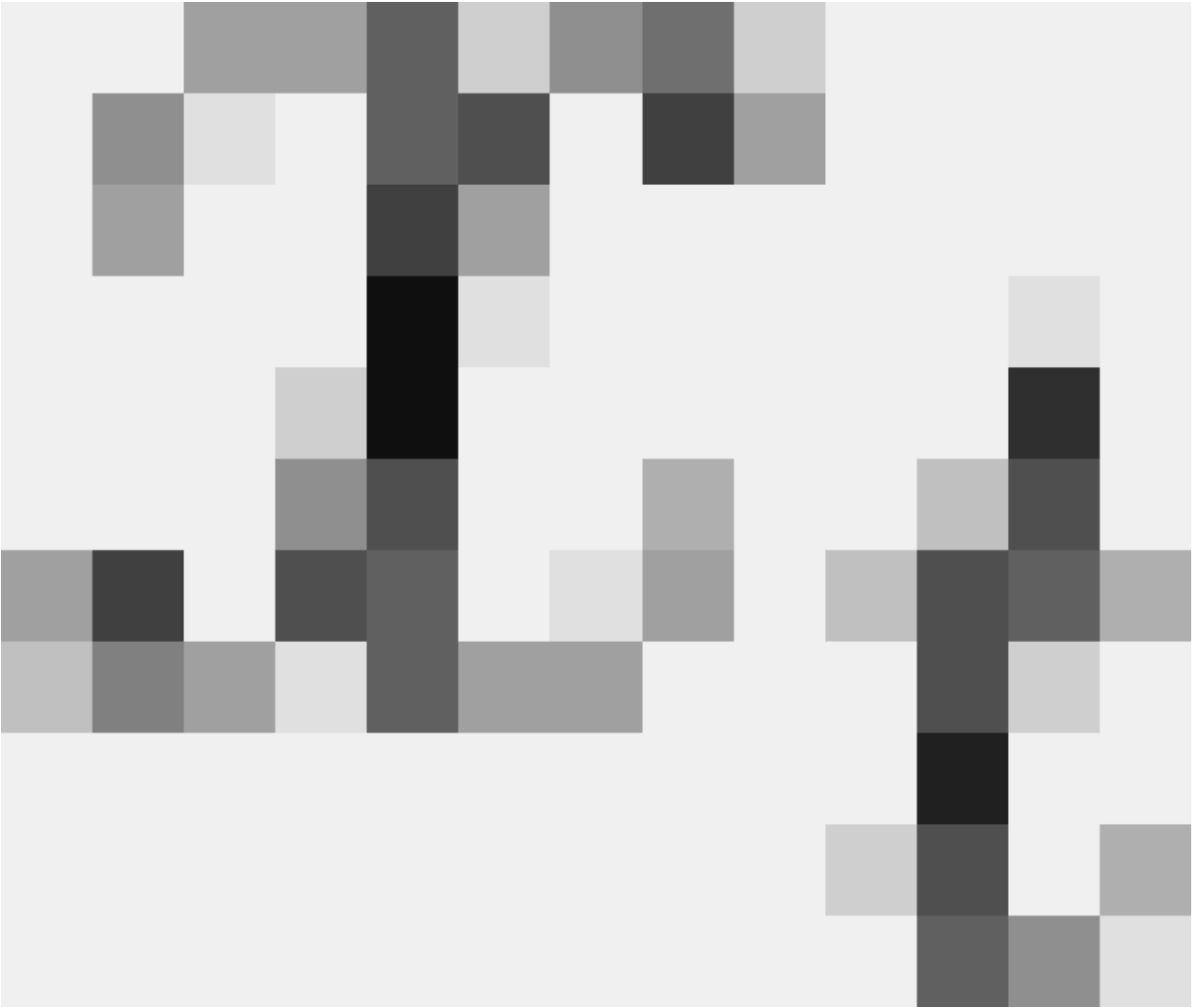
based on how incorrect we were. So we can think of this as receiving prediction problems one-by-one, and in particular we make no assumptions about the relationship

between the different problems; they could be i.i.d., they could be positively correlated, they could be anti-correlated, they could even be adversarially chosen.

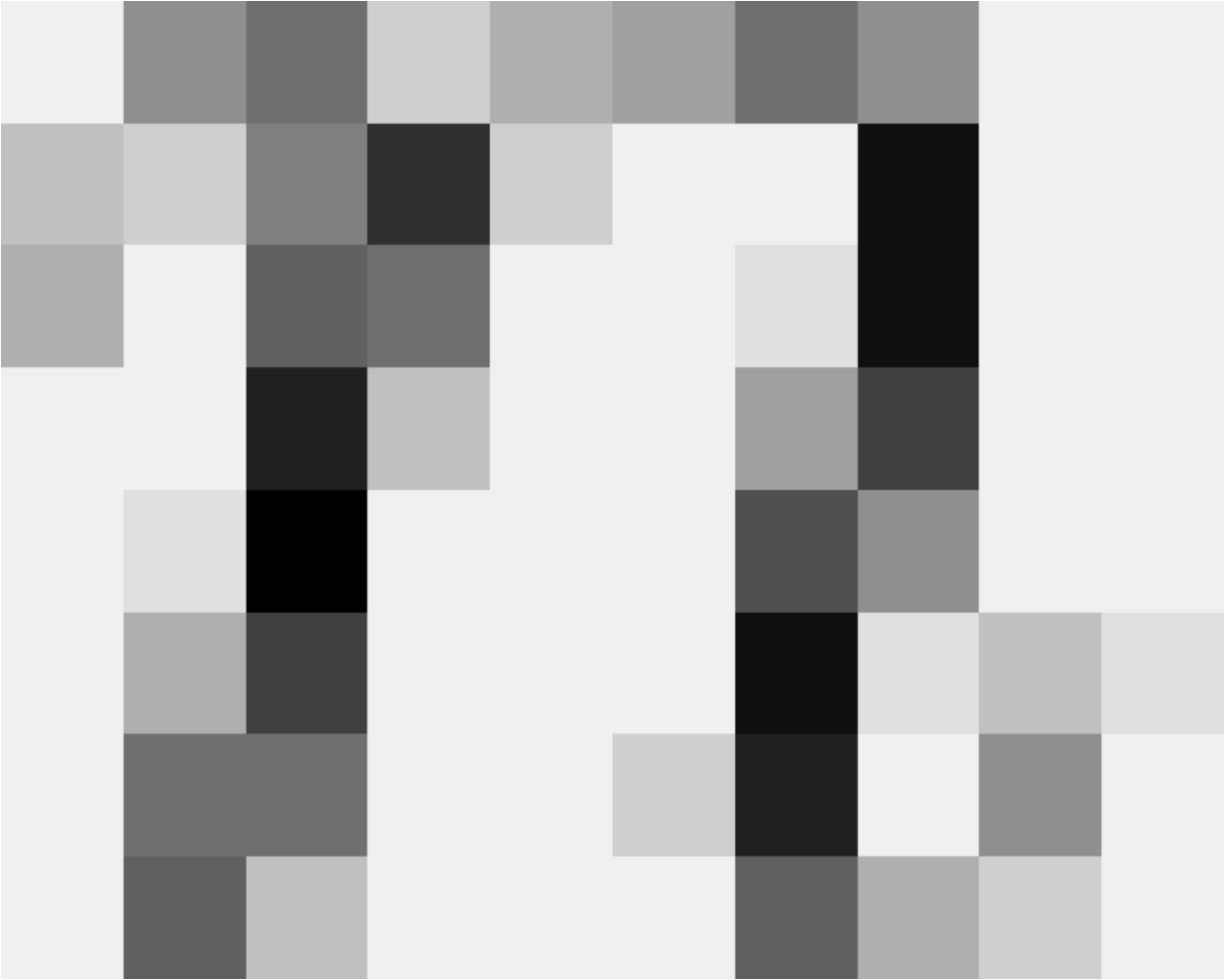
As a running example, suppose that I'm betting on horses and before each race there are



other people who give me advice on which horse to bet on. I know nothing about horses, so based on this advice I'd like to devise a good betting strategy. In this case,



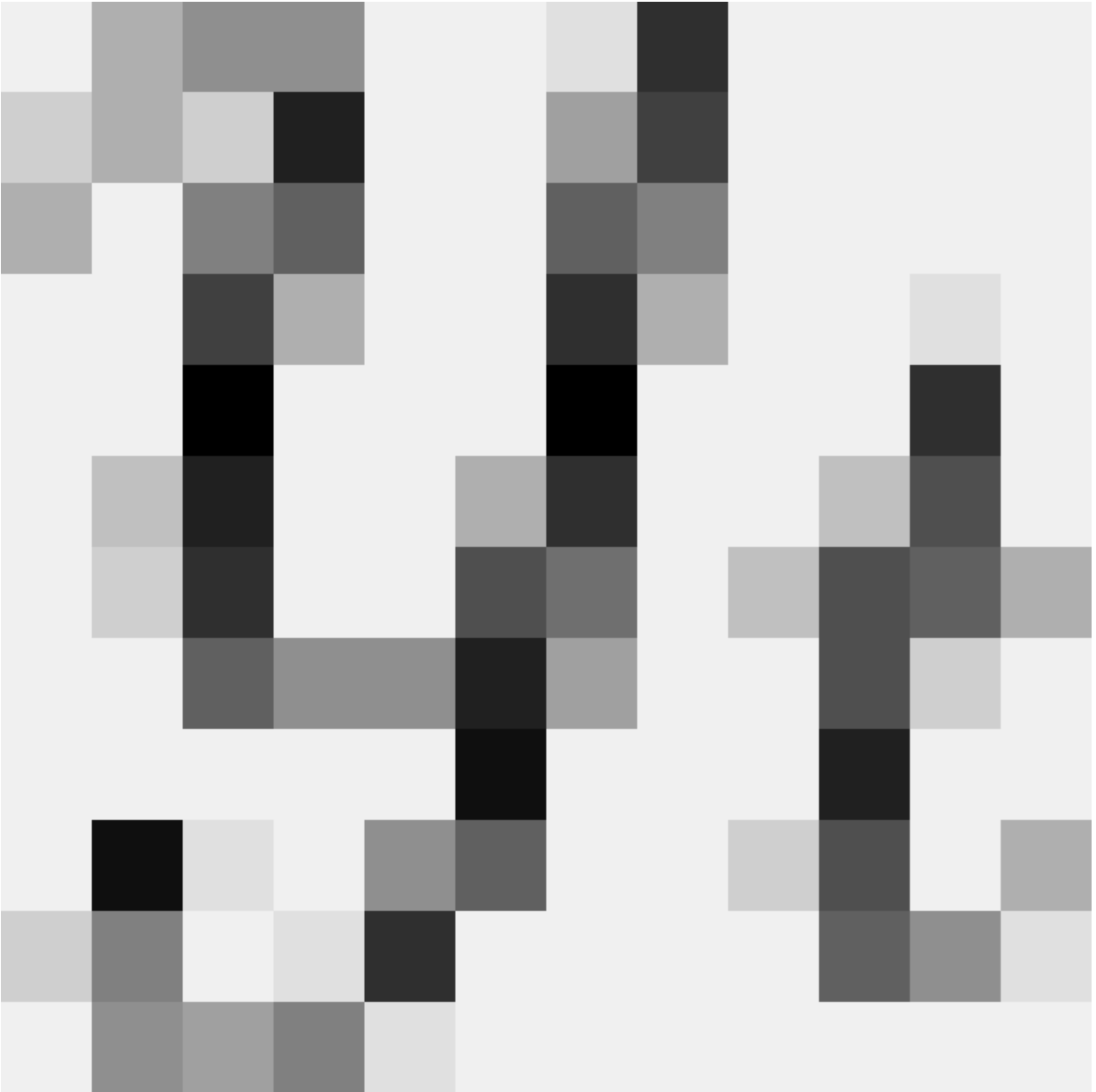
would be the



bets that each of the other people recommend,



would be the horse that I actually bet on, and



would be the horse that actually wins the race. Then, supposing that



(i.e., the horse I bet on actually wins),

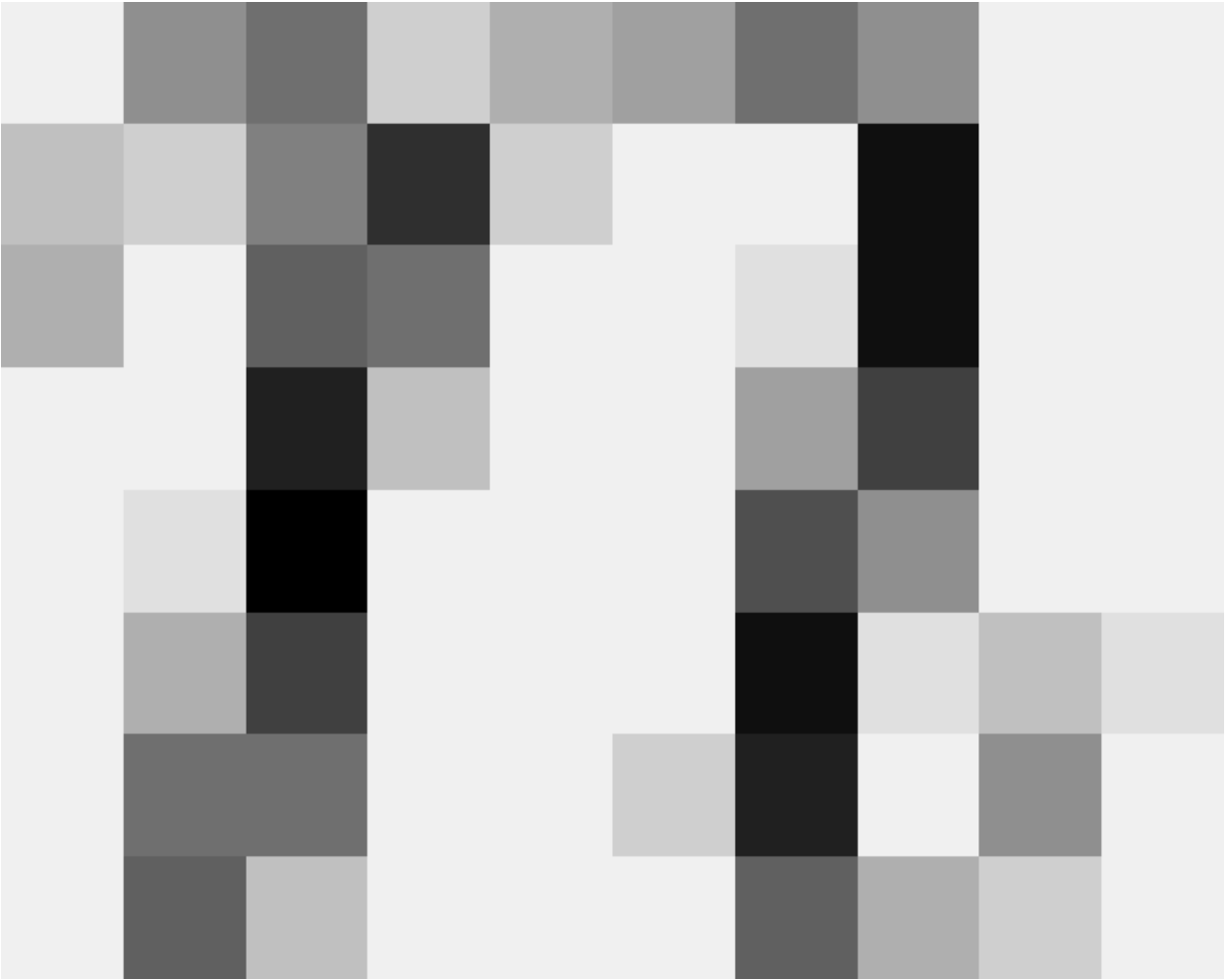
$$L(w, z_t)$$

is the negative of the payoff from correctly betting on that horse. Otherwise, if the horse I bet on doesn't win,

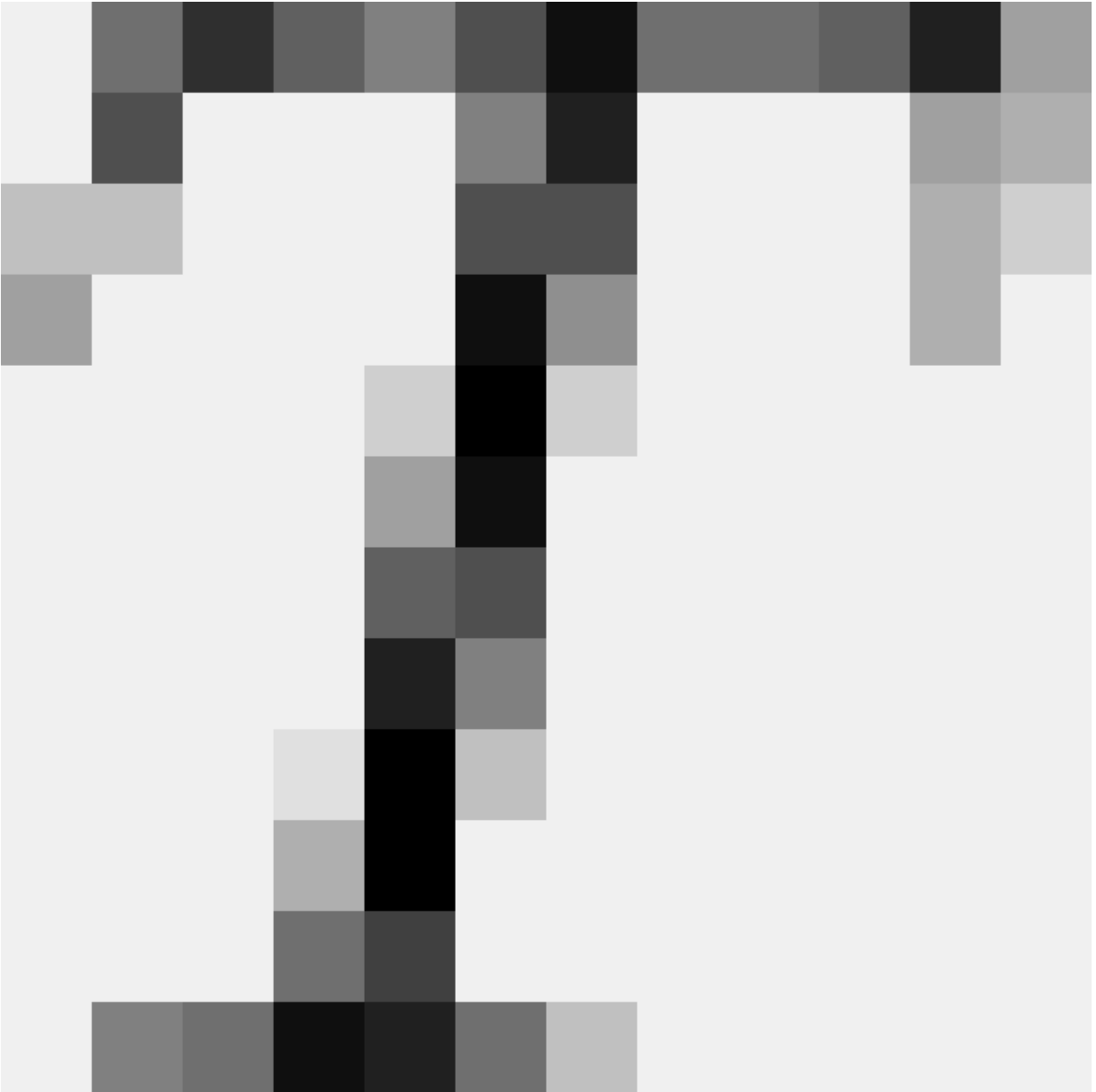
$$L(w, z_t)$$

is the cost I had to pay to place the bet.

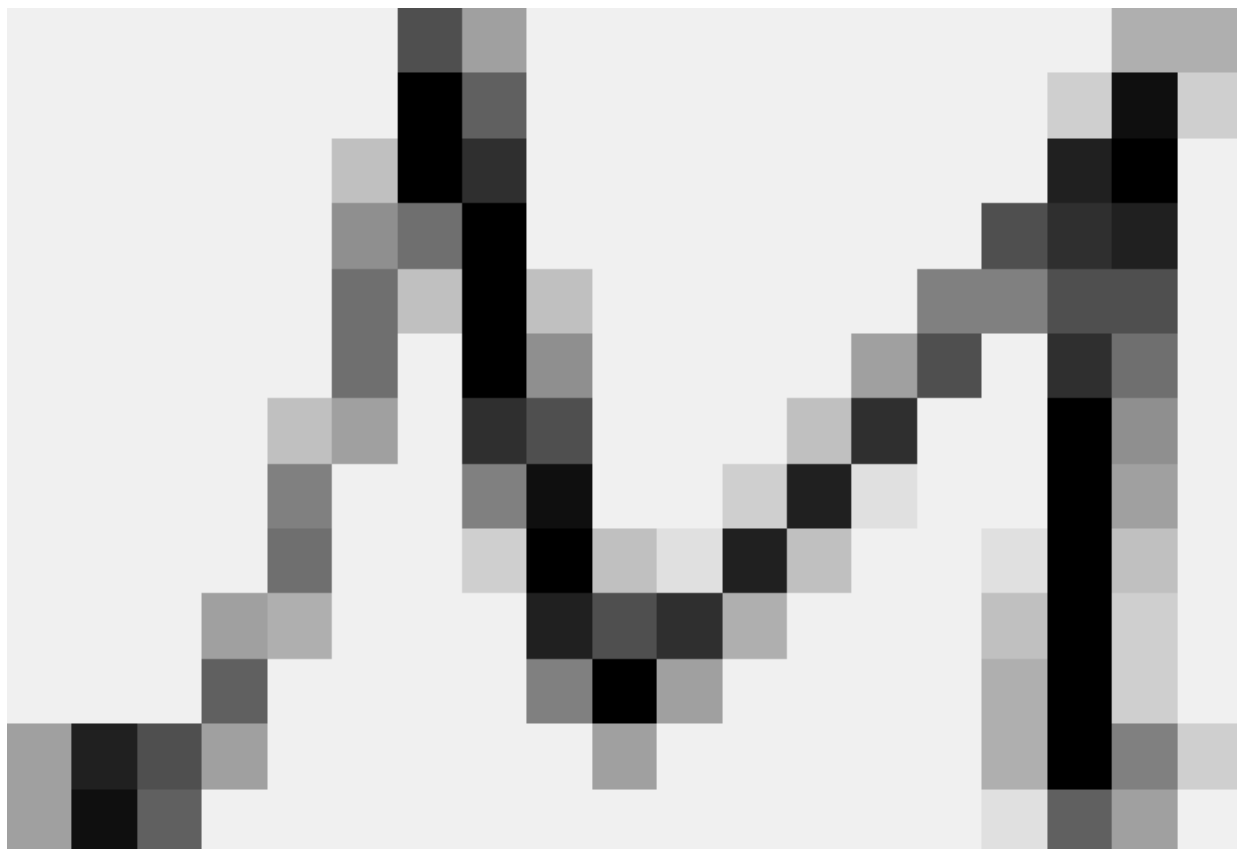
If I'm in this setting, what guarantee can I hope for? I might ask for an algorithm that is guaranteed to make good bets — but this seems impossible unless the people advising me actually know something about horses. Or, at the very least, *one* of the people advising me knows something. Motivated by this, I define my **regret** to be the difference between my penalty and the penalty of the best of the



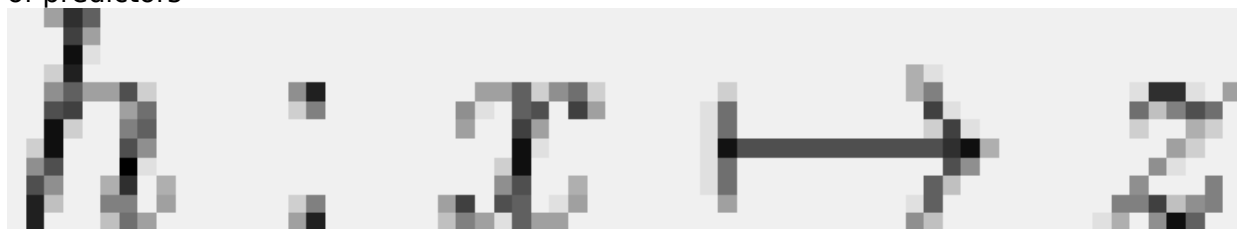
people (note that I only have access to the latter after all



rounds of betting). More formally, given a class



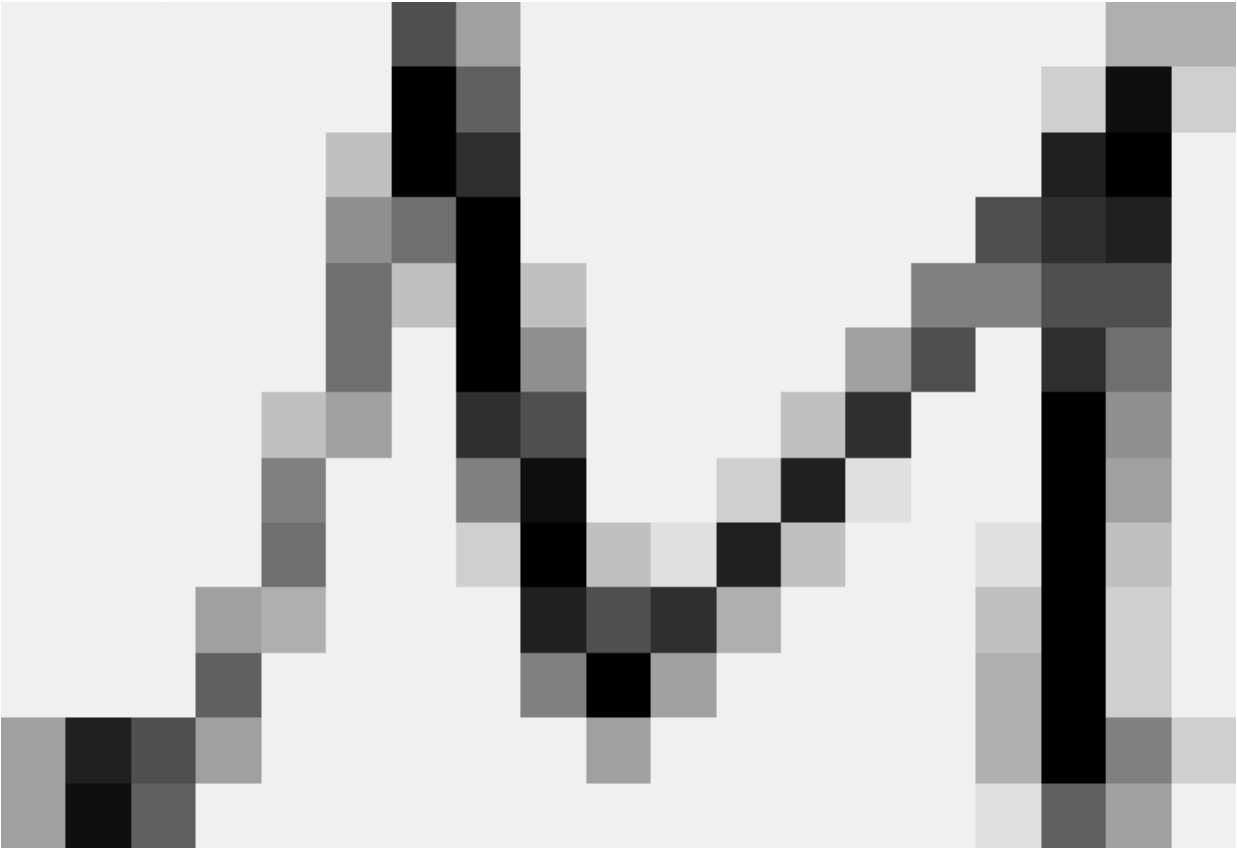
of predictors



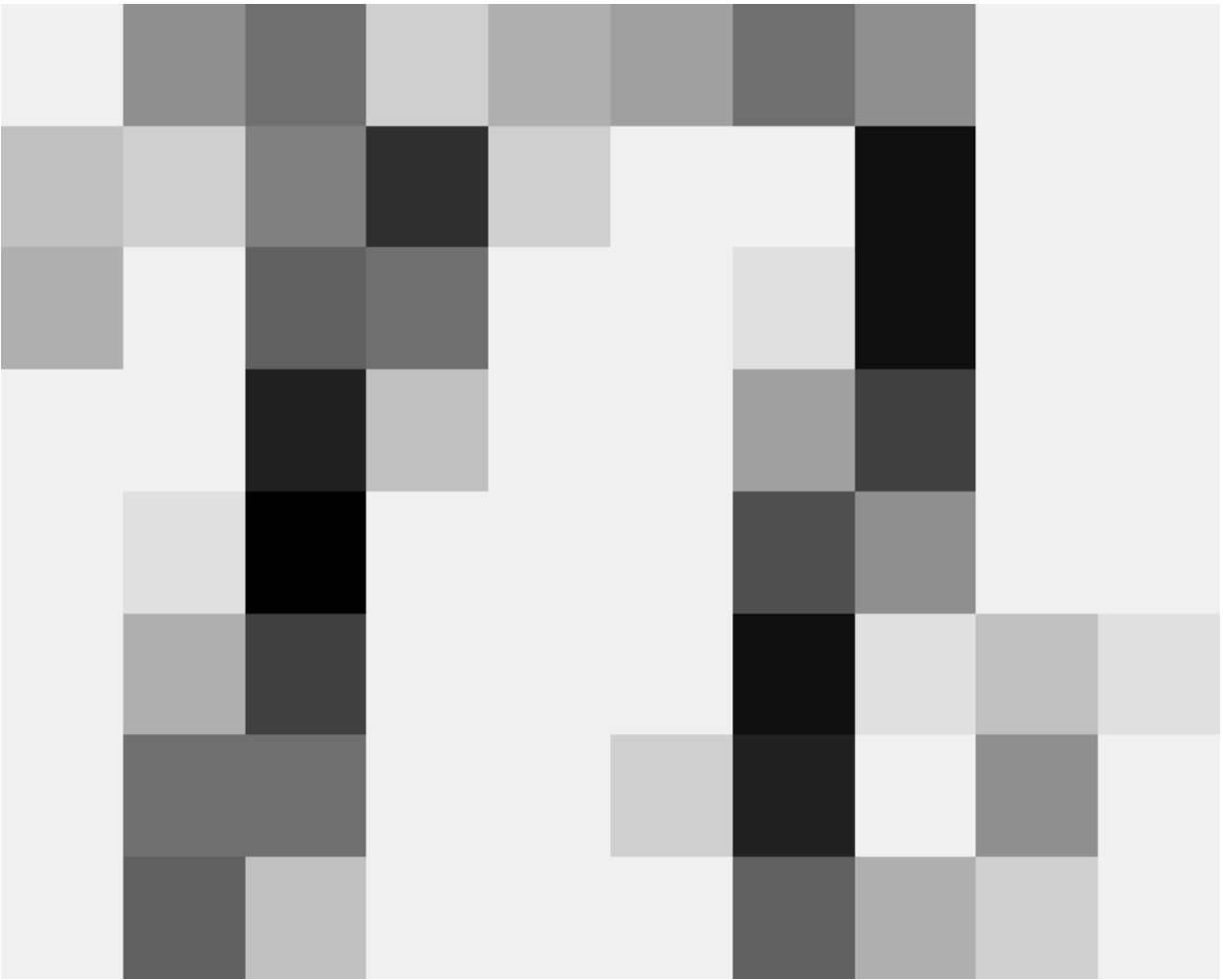
, I define

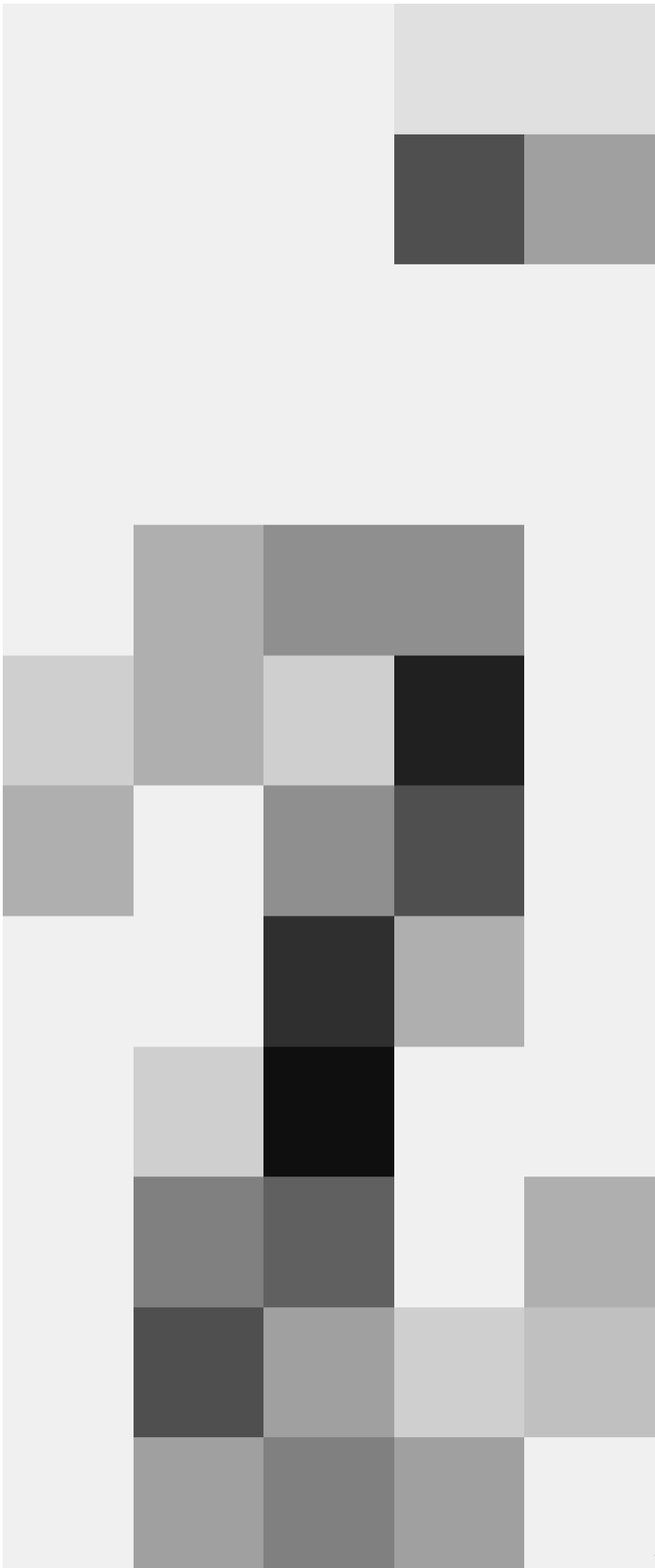
$$\text{Regret}(T) = \frac{1}{T} \sum_{t=1}^T L(y_t, z_t) - \min_{h \in \mathcal{M}} \frac{1}{T} \sum_{t=1}^T L(y_t, h(x_t))$$

In this case,



would have size

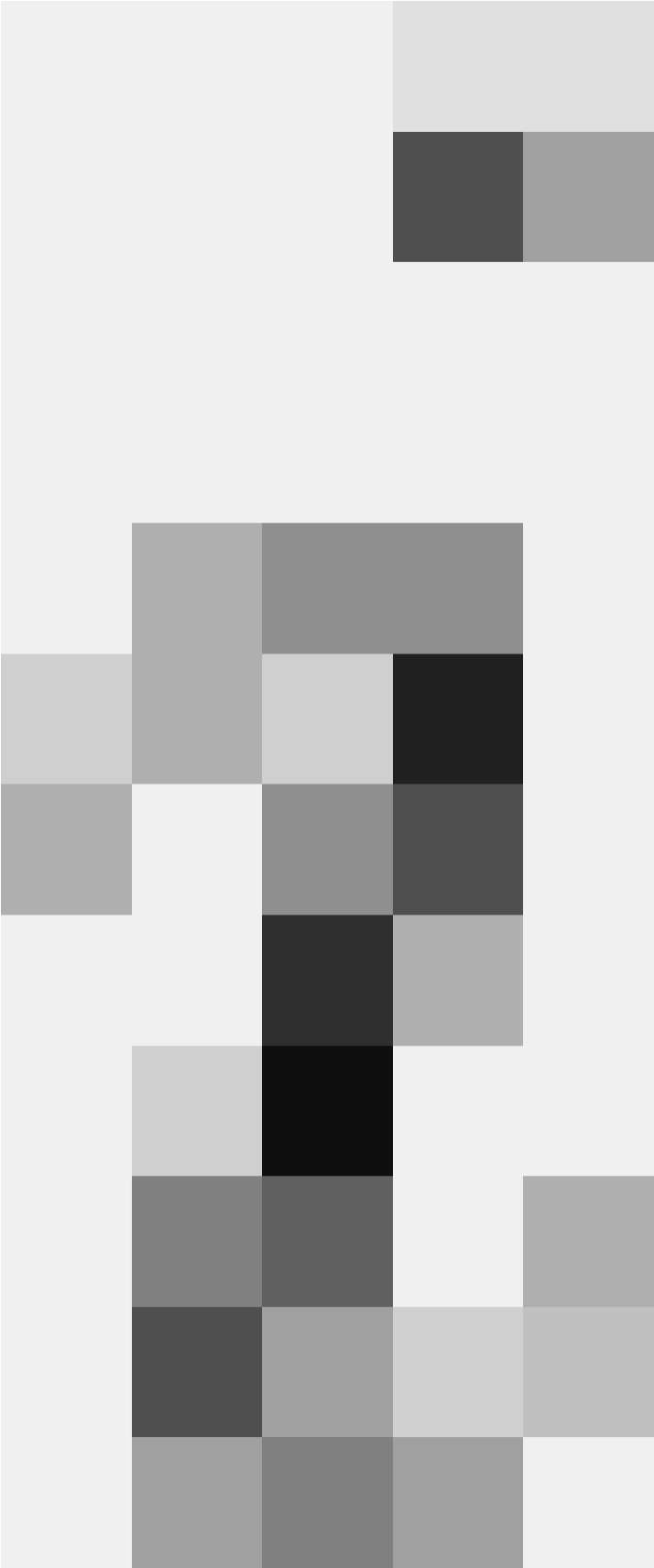




and the

th predictor would just

always follow the advice of person



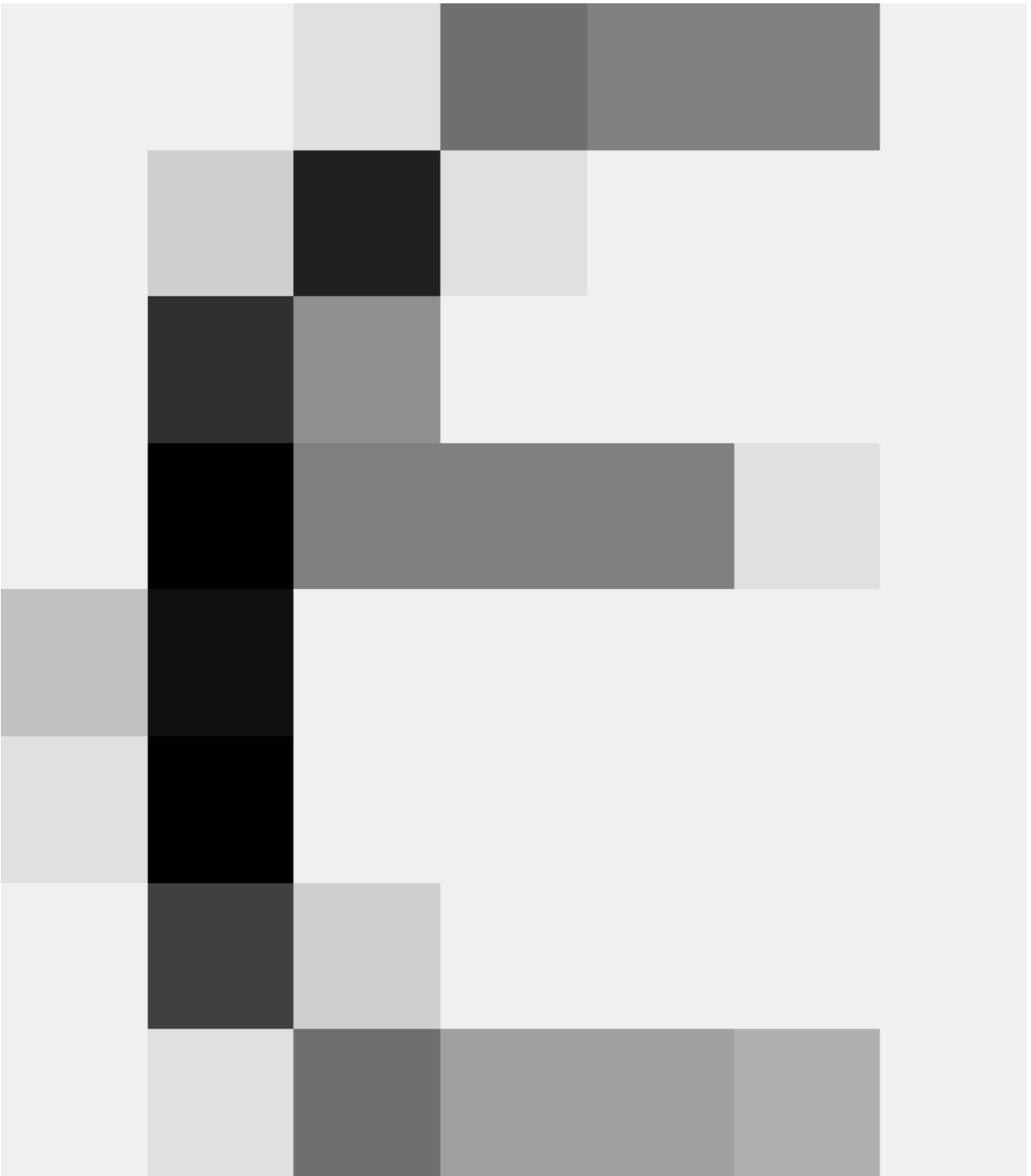
The regret is then how much worse I do on average than the best expert. A remarkable fact is that, in this case, there is a strategy such that

$$\text{Regret}(T)$$

shrinks at a rate of

$$\sqrt{\frac{\log(T)}{T}}$$

. In other words, I can have an average score within



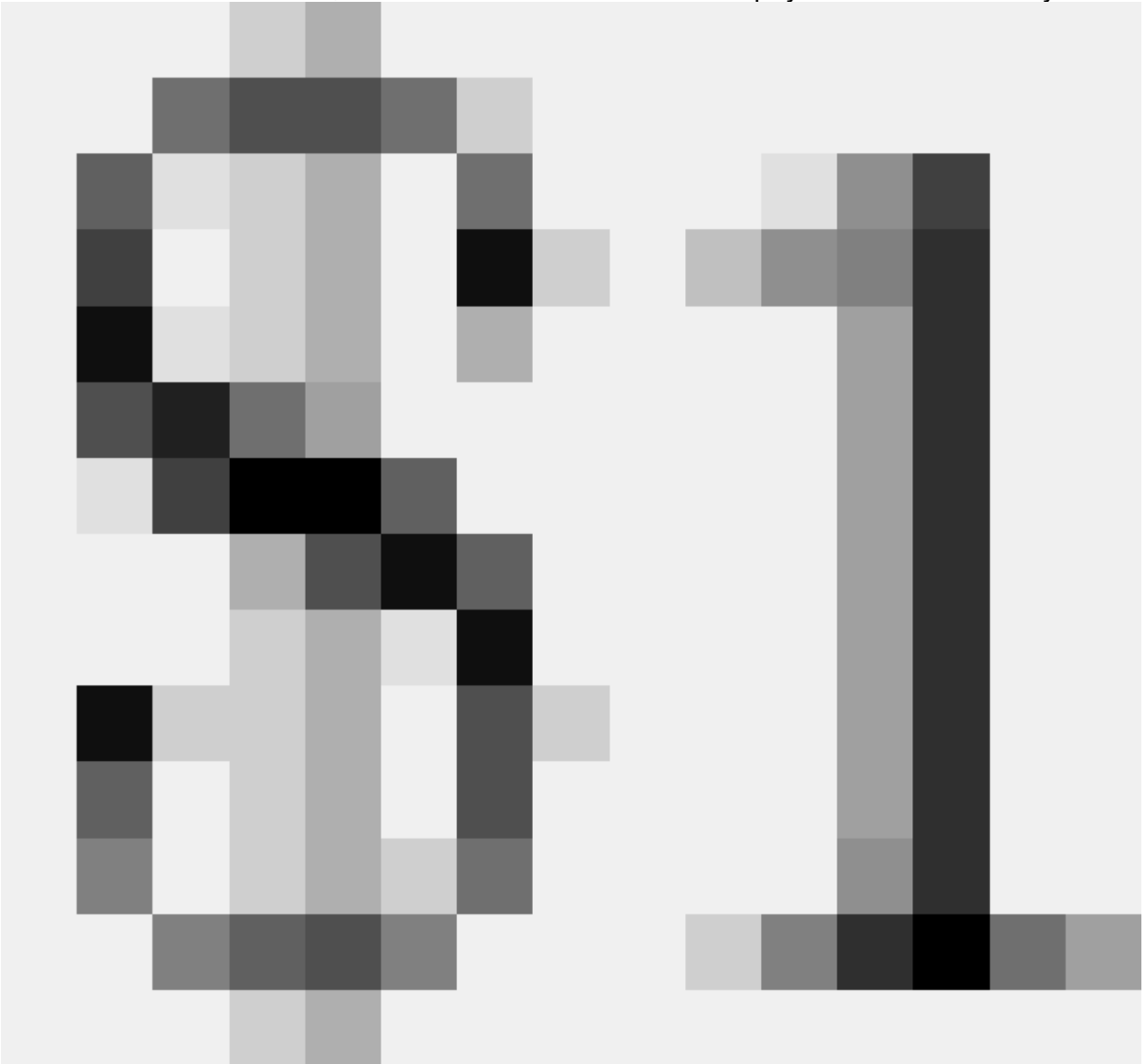
of the best advisor after



rounds of betting.

One reason that this is remarkable is that it does not depend at all on how the data are distributed; **the data could be i.i.d., positively correlated, negatively correlated, even adversarial**, and one can still construct an (adaptive) prediction rule that does almost as well as the best predictor in the family.

To be even more concrete, if we assume that all costs and payoffs are bounded by



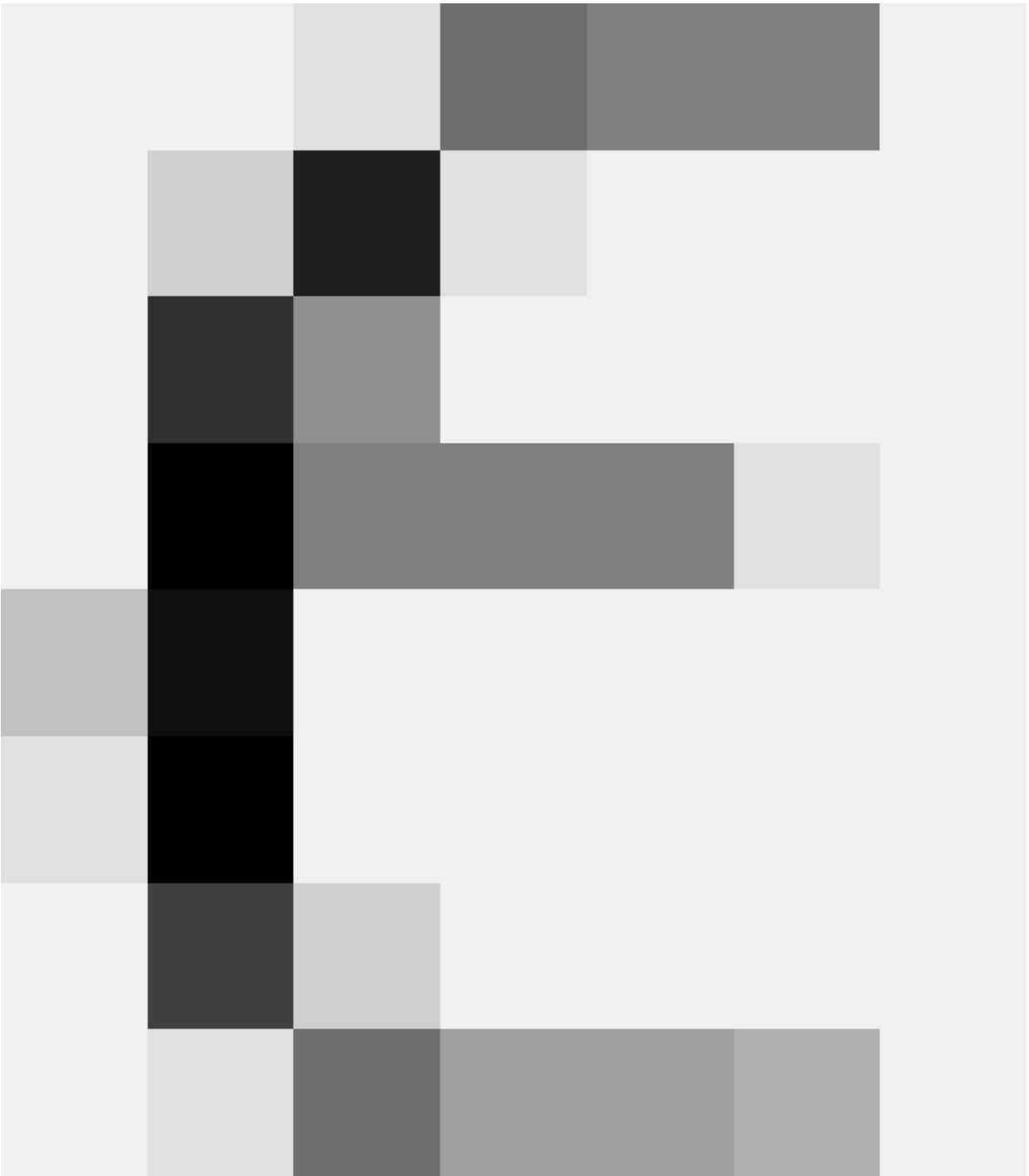
per round, and that there are



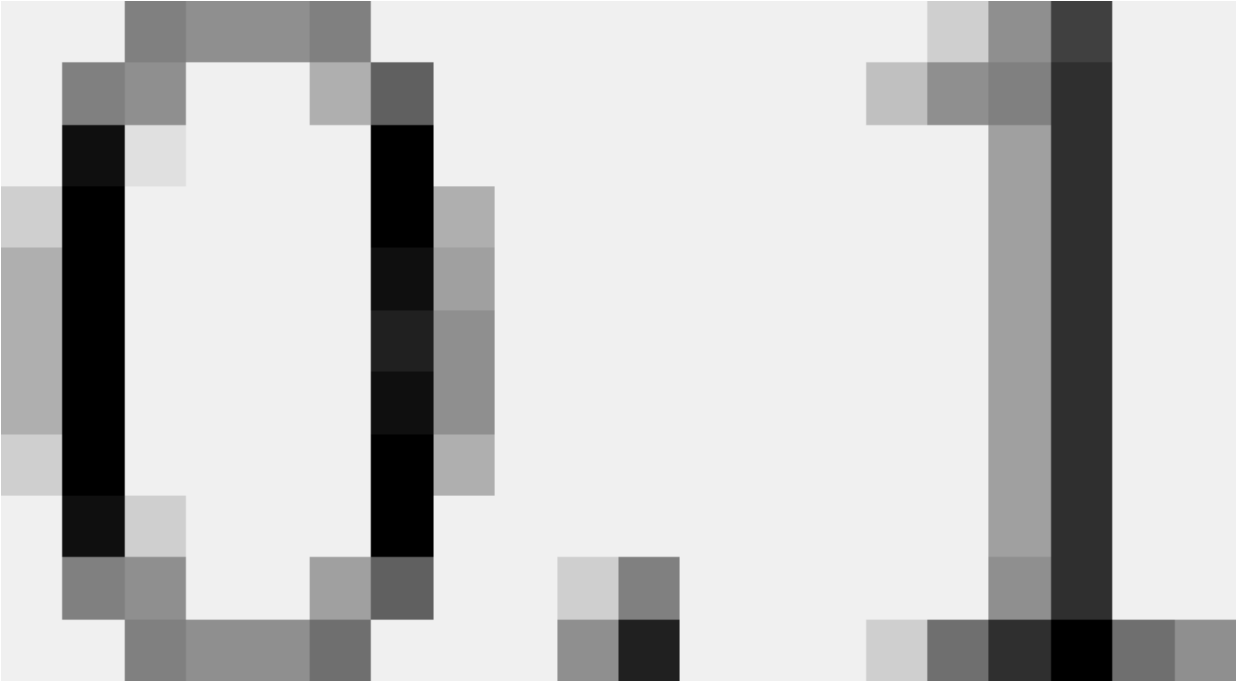
people in total, then an explicit upper bound is that after



rounds, we will be within



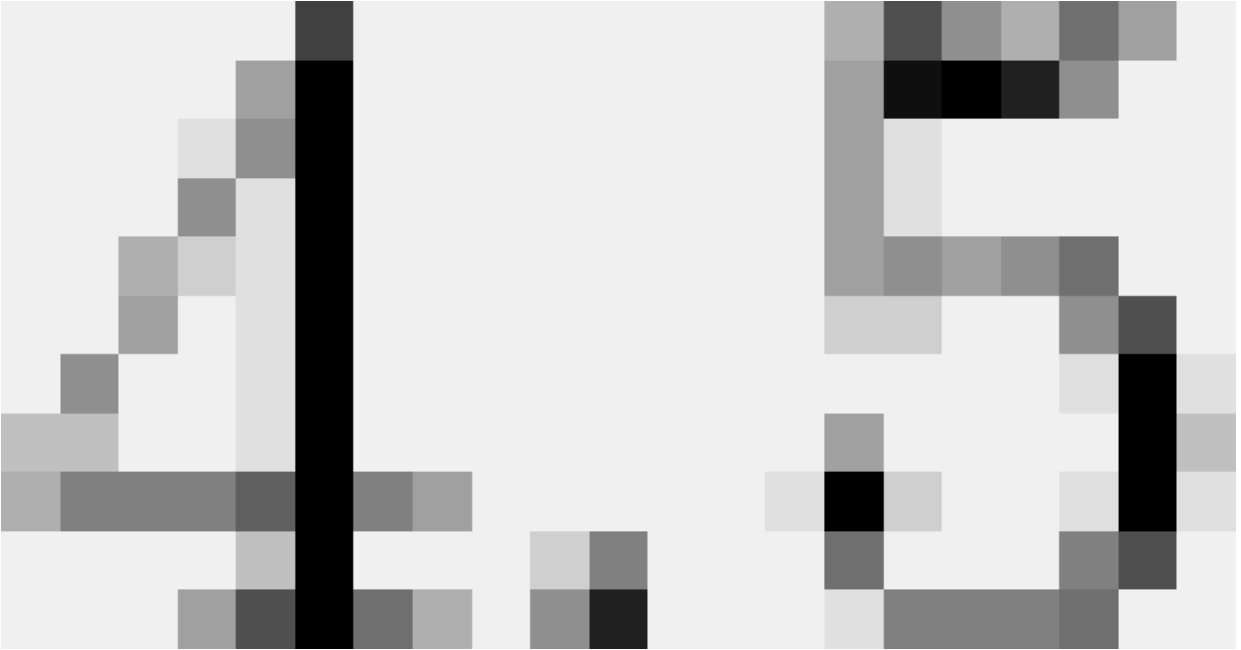
dollars on average of the best other person. Under slightly stronger assumptions, we can do even better, for instance if the best person has an average variance of



about their mean, then the

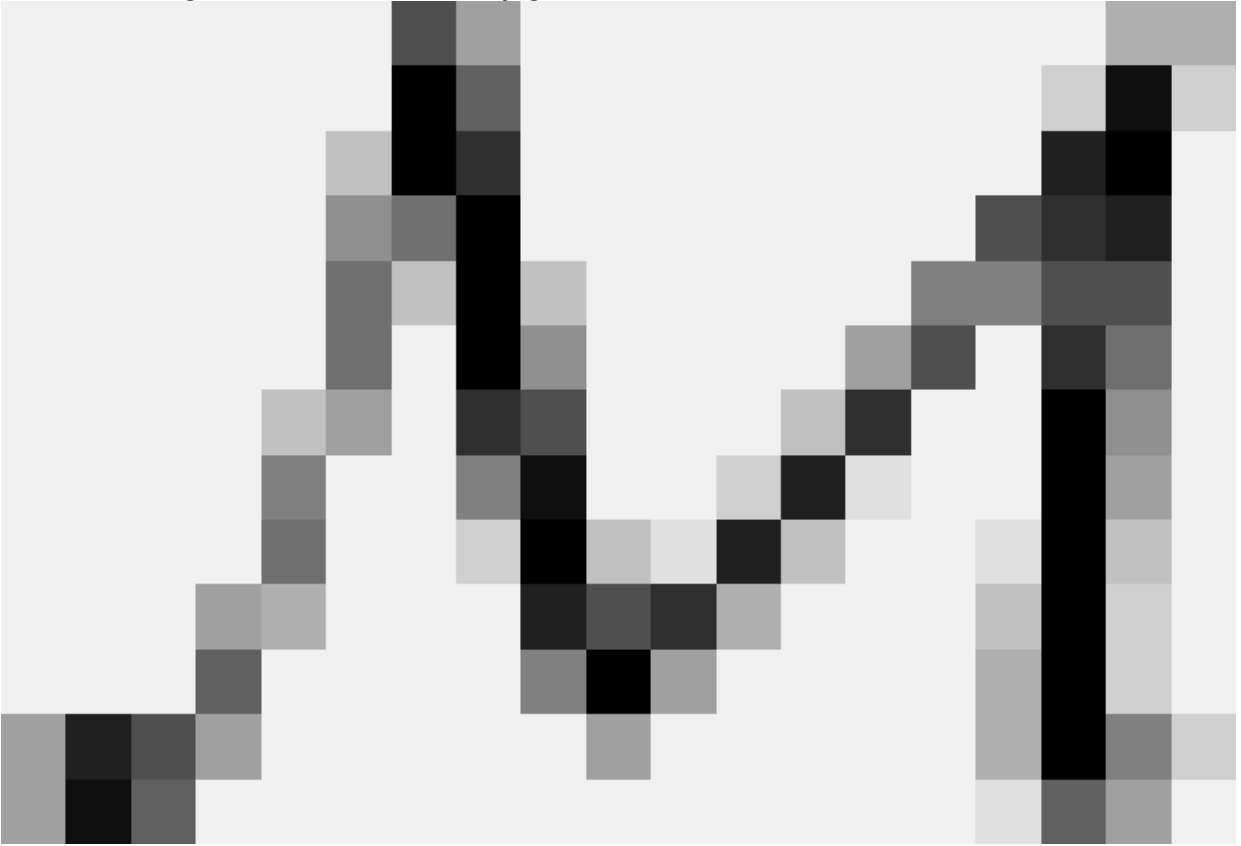


can be replaced with

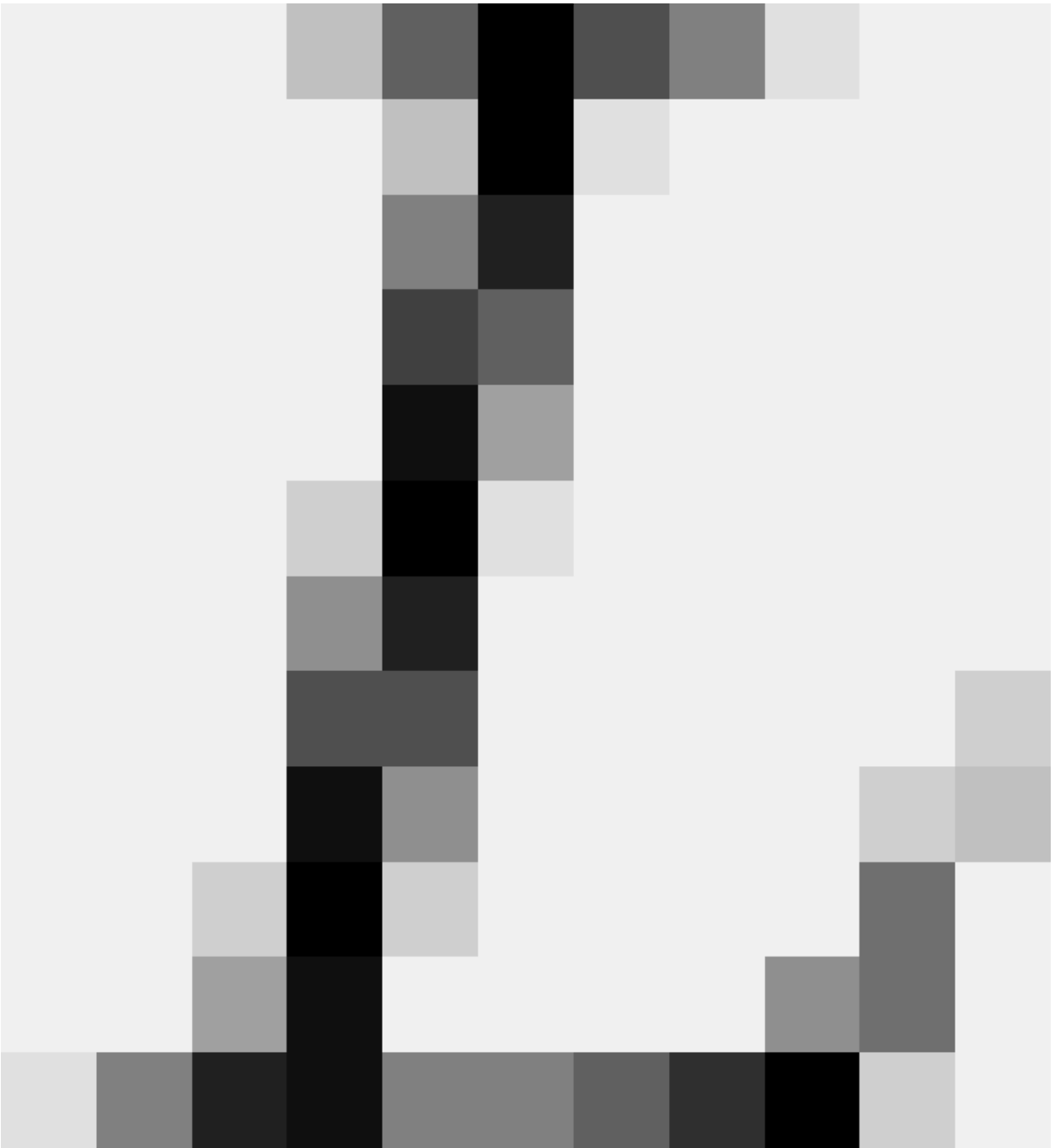


.

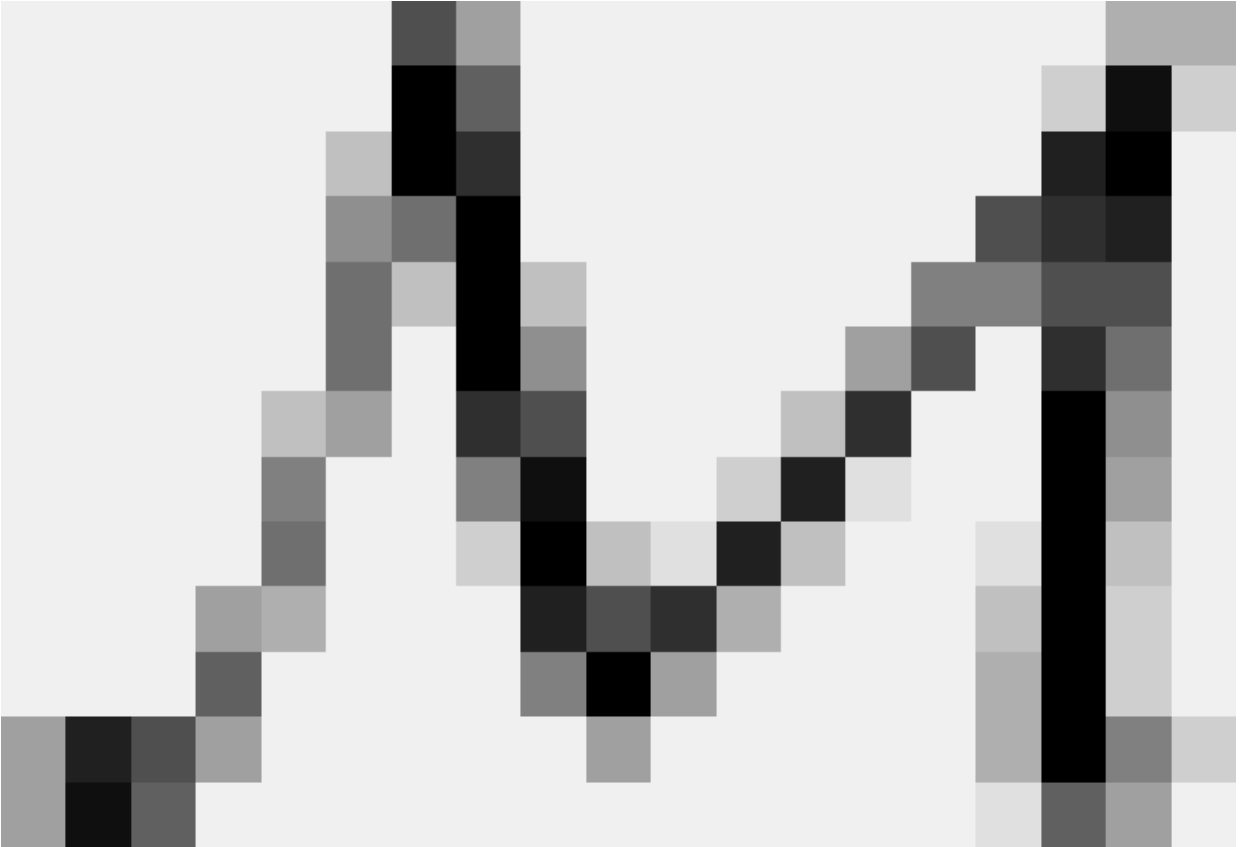
It is important to note that the betting scenario is just a running example, and one can still obtain regret bounds under fairly general scenarios;



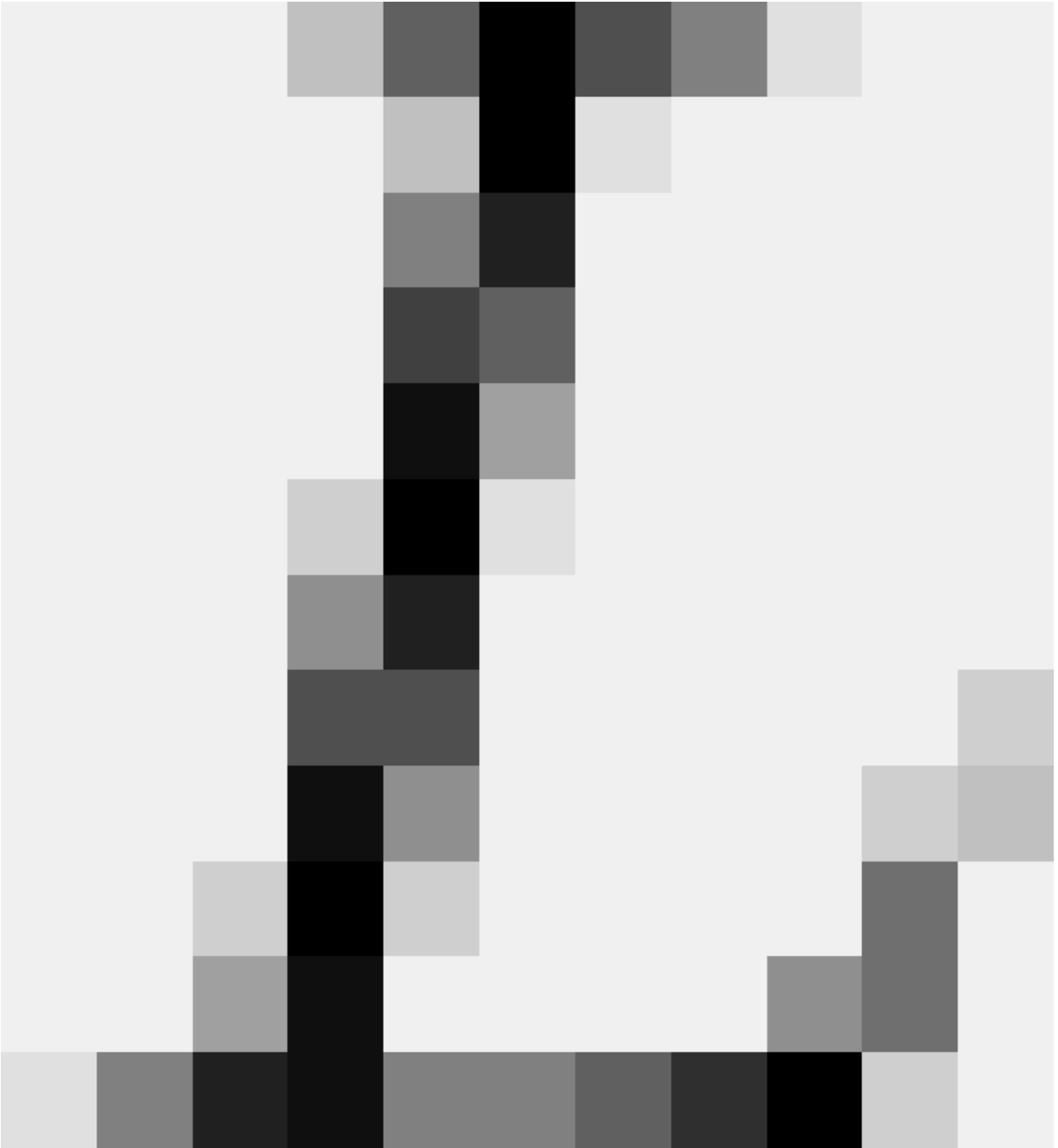
could be continuous and



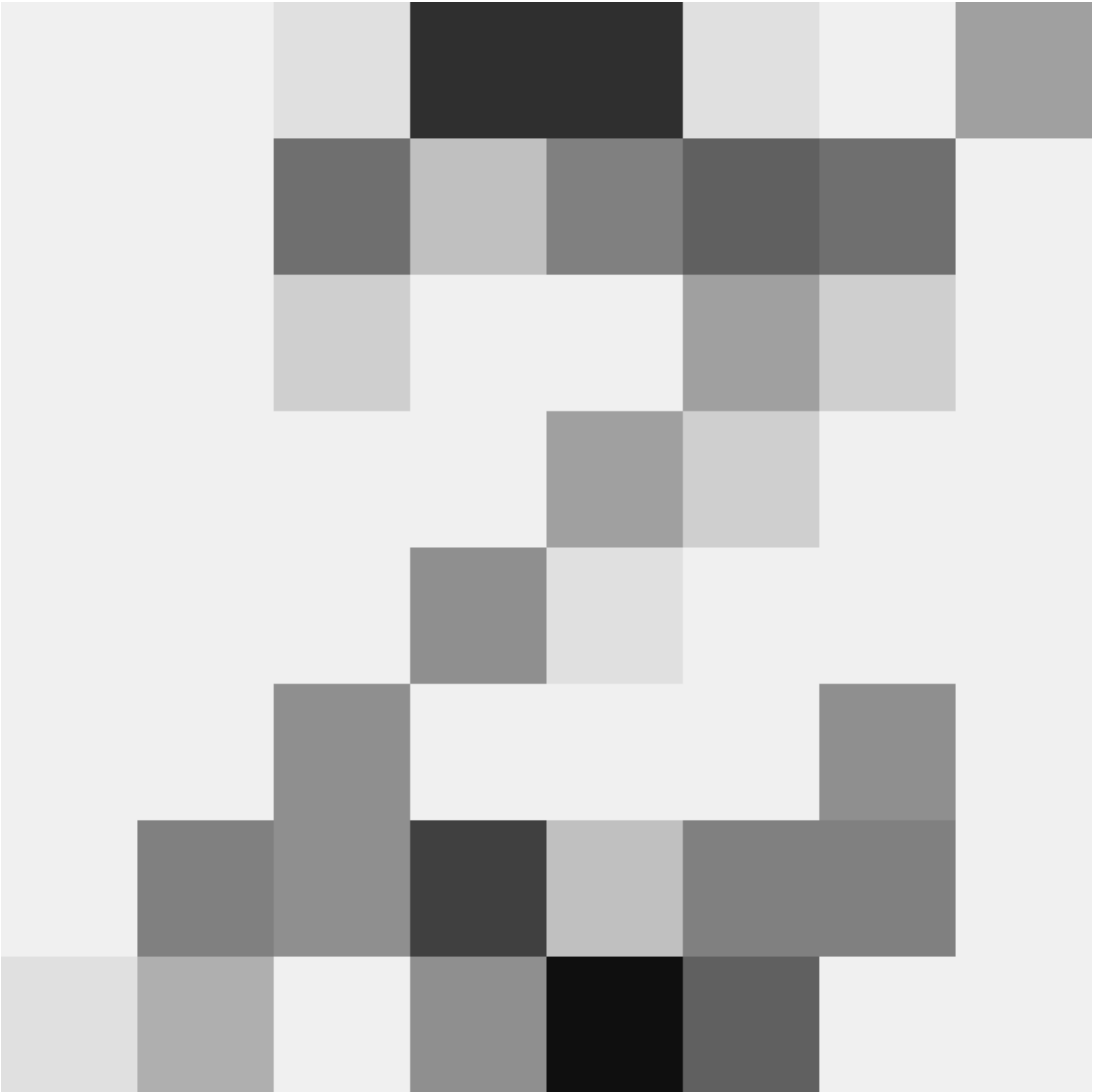
could have quite general structure; the only technical assumption is that



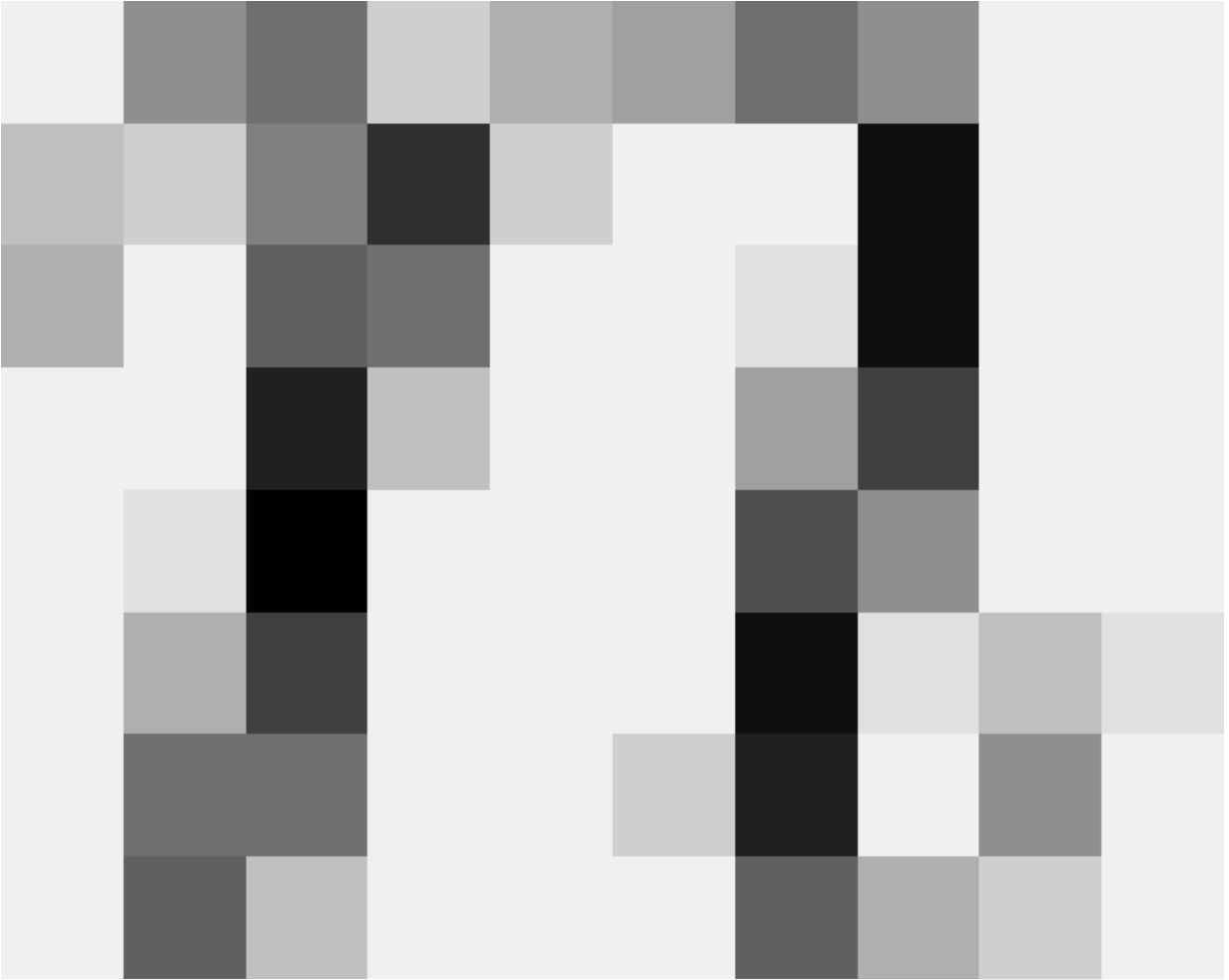
be a convex set and that



be a convex function of



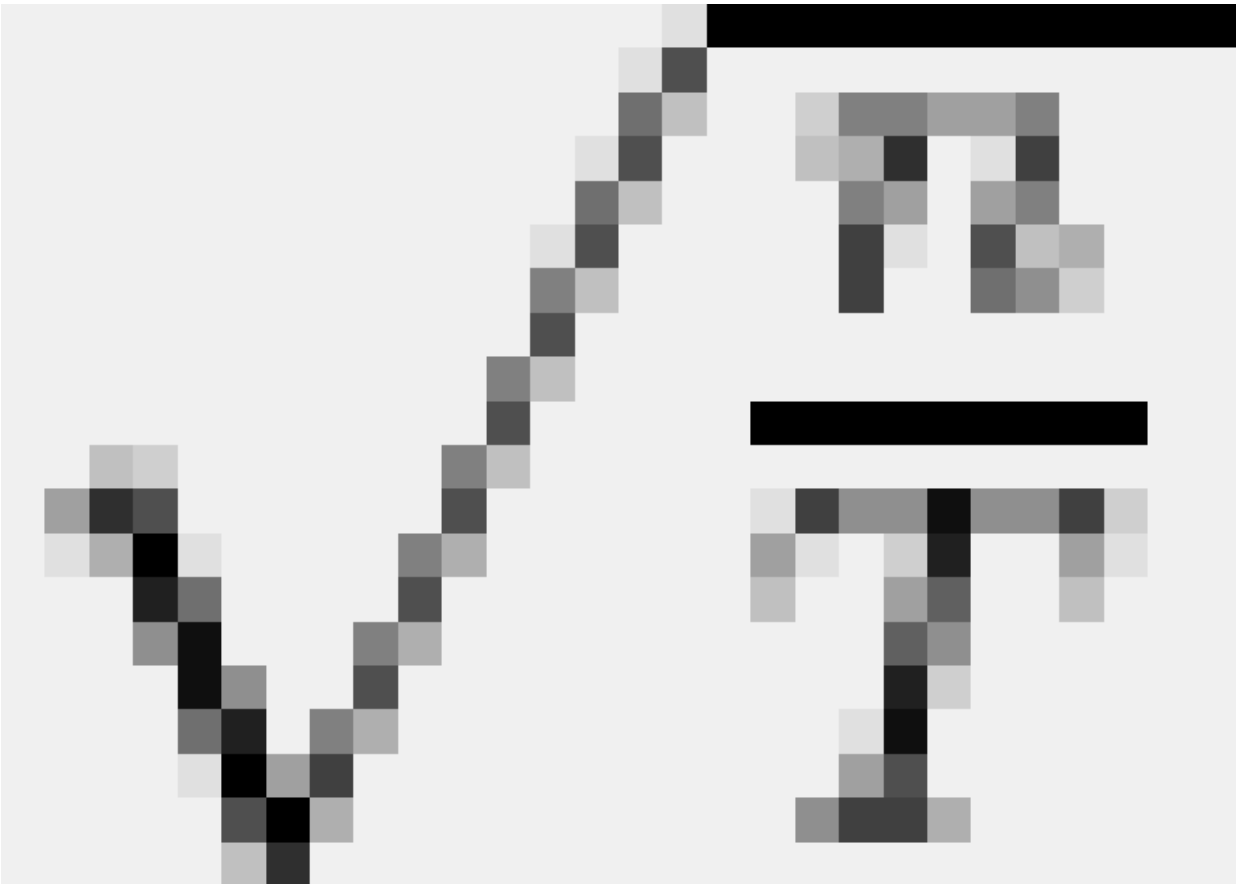
. These assumptions tend to be easy to satisfy, though I have run into a few situations where they end up being problematic, mainly for computational reasons. For an



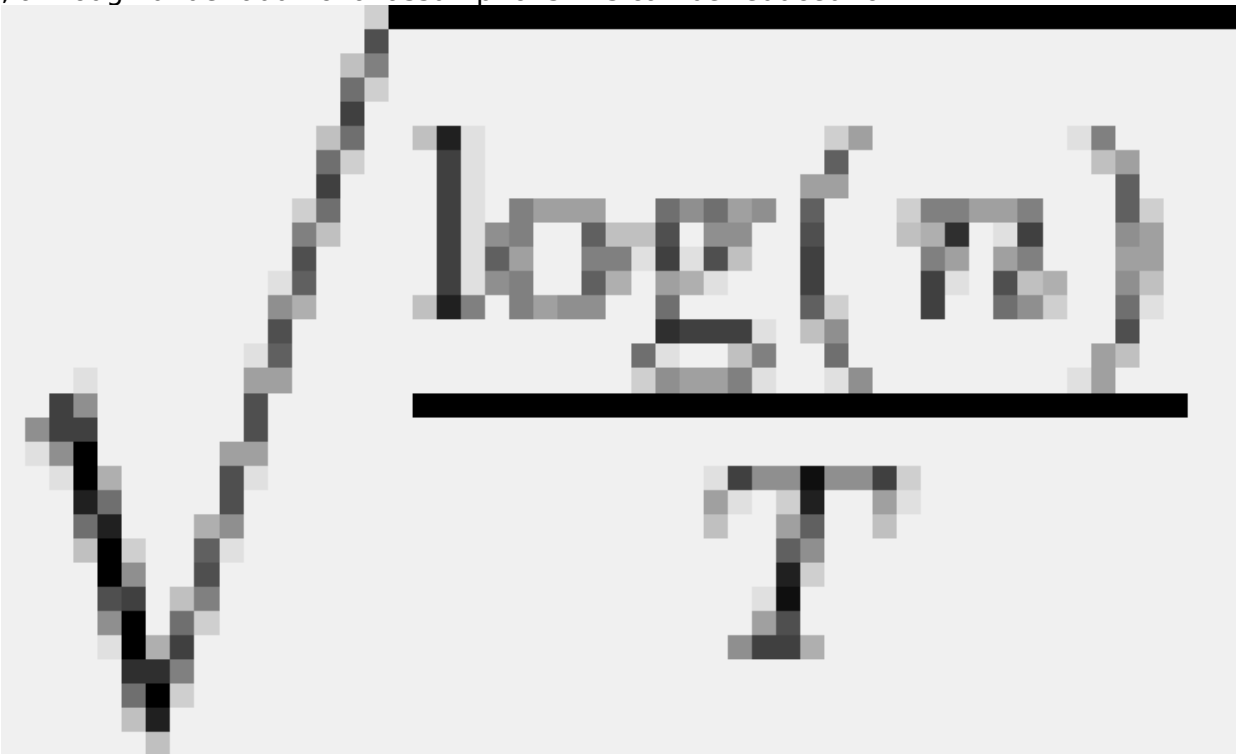
-dimensional model family, typically



decreases at a rate of



, although under additional assumptions this can be reduced to



, as in the betting example above. I would consider this reduction to be one of the crowning results of modern frequentist statistics.

Yes, these guarantees sound incredibly awesome and perhaps too good to be true. They actually are that awesome, and they are actually true. The work is being done by measuring the error relative to the best model in the model family. We aren't required to do well in an absolute sense, we just need to not do any worse than the best model. Of as long as at least one of the models in our family makes good predictions, that means we will as well. This is really what statistics is meant to be doing: you come up with everything you imagine could possibly be reasonable, and hand it to me, and then I come up with an algorithm that will figure out which of the things you handed me was most reasonable, and will do almost as well as that. As long as at least one of the things you come up with is good, then my algorithm will do well. Importantly, due to the

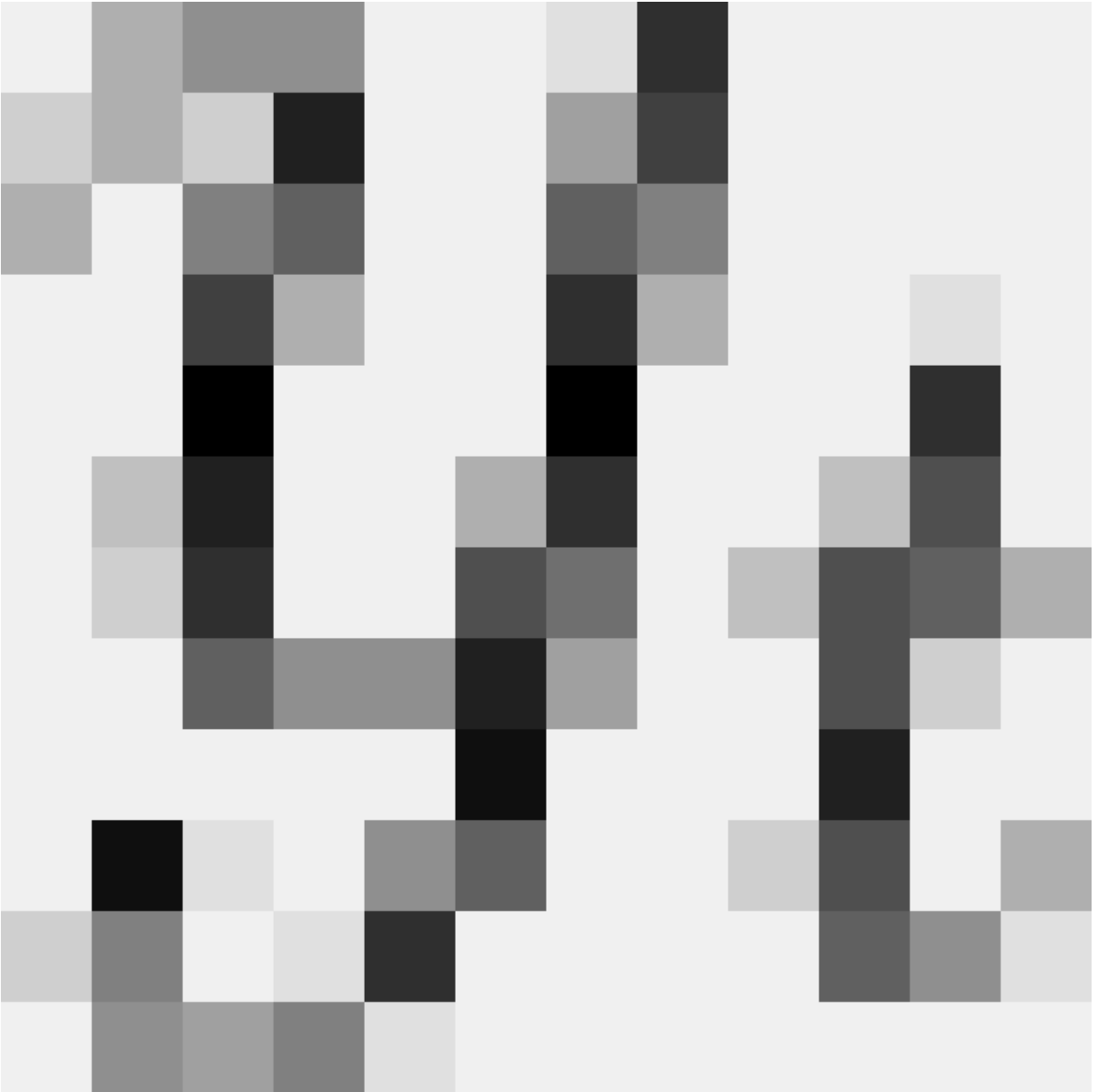


dependence on the dimension of the model family, you can actually write down extremely broad classes of models and I will still successfully sift through them.

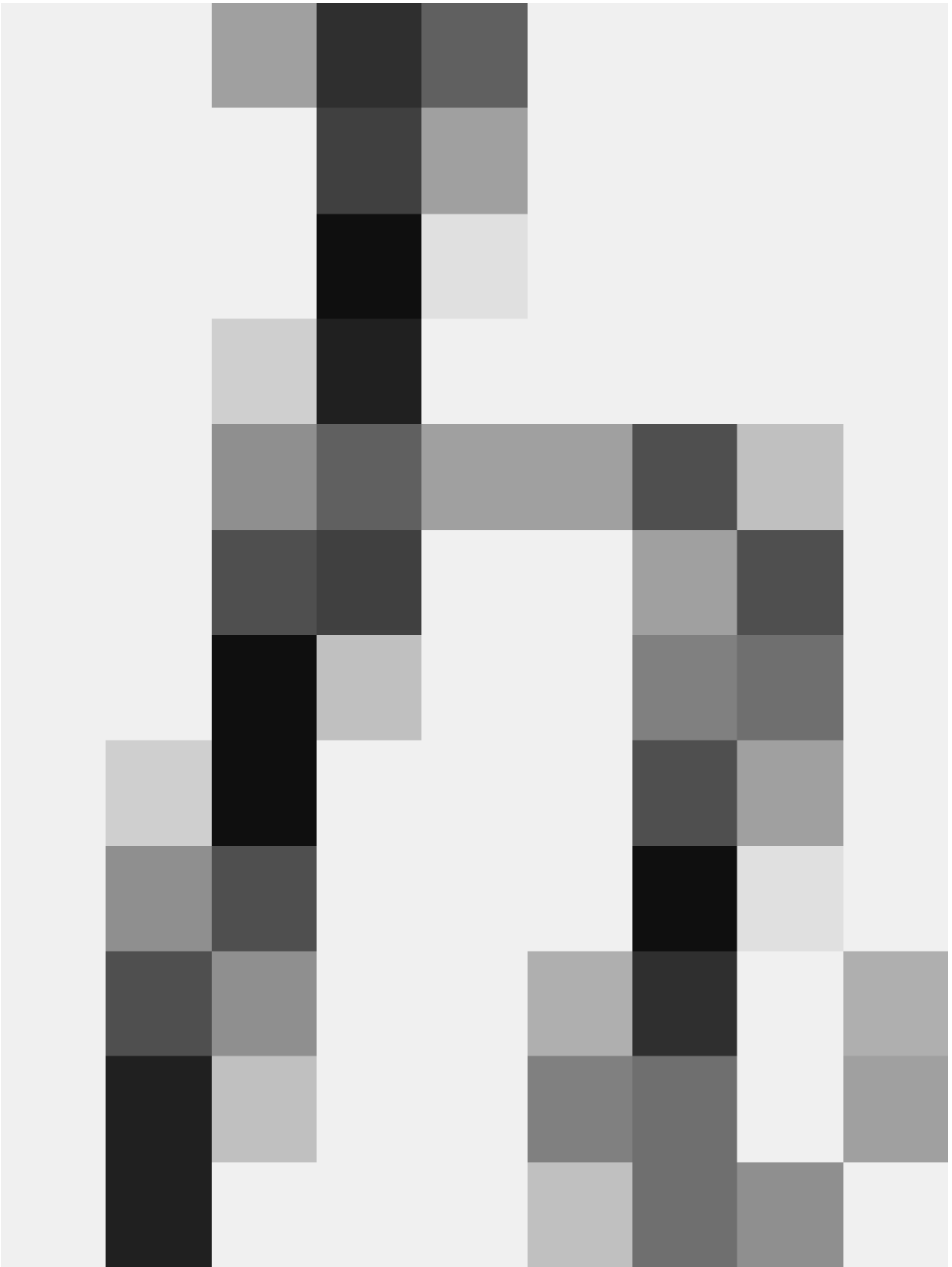
Let me stress again: regret bounds are saying that, no matter how the



and



are related, no i.i.d. assumptions anywhere in sight, we will do almost as well as any predictor



in

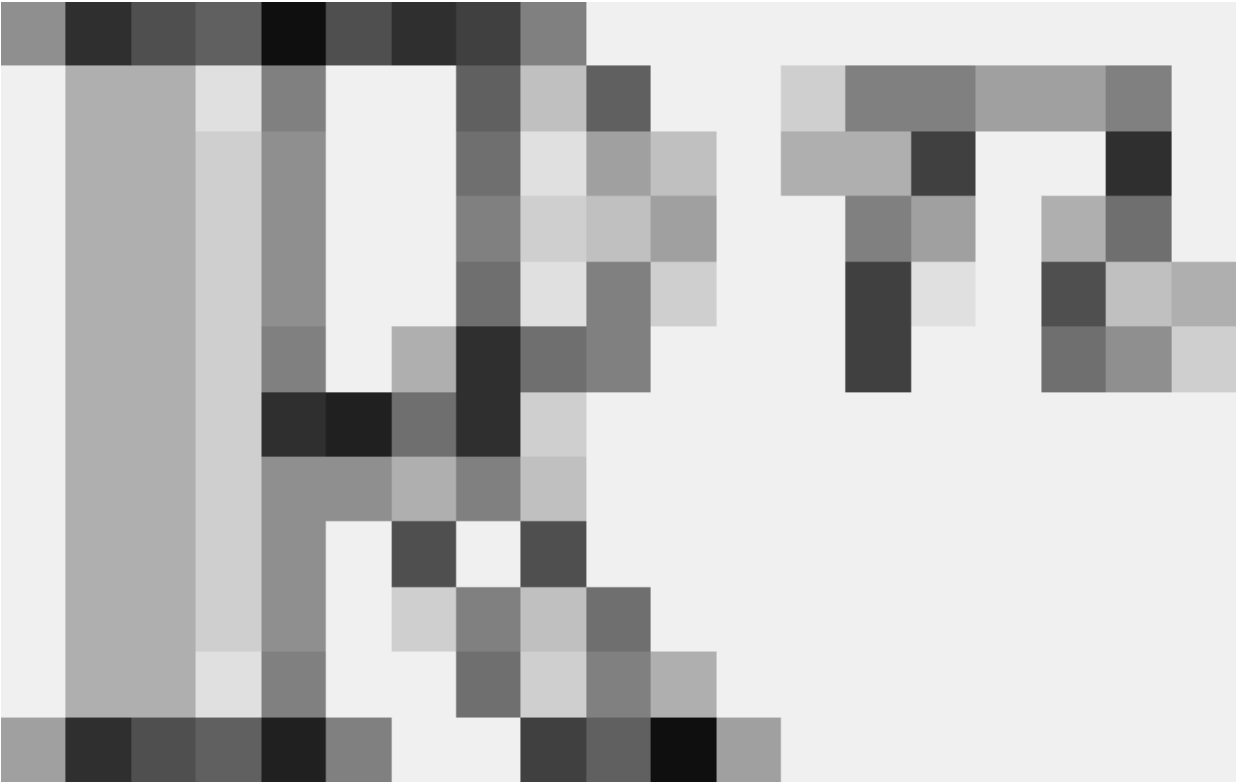


(in particular, almost as well as the best predictor).

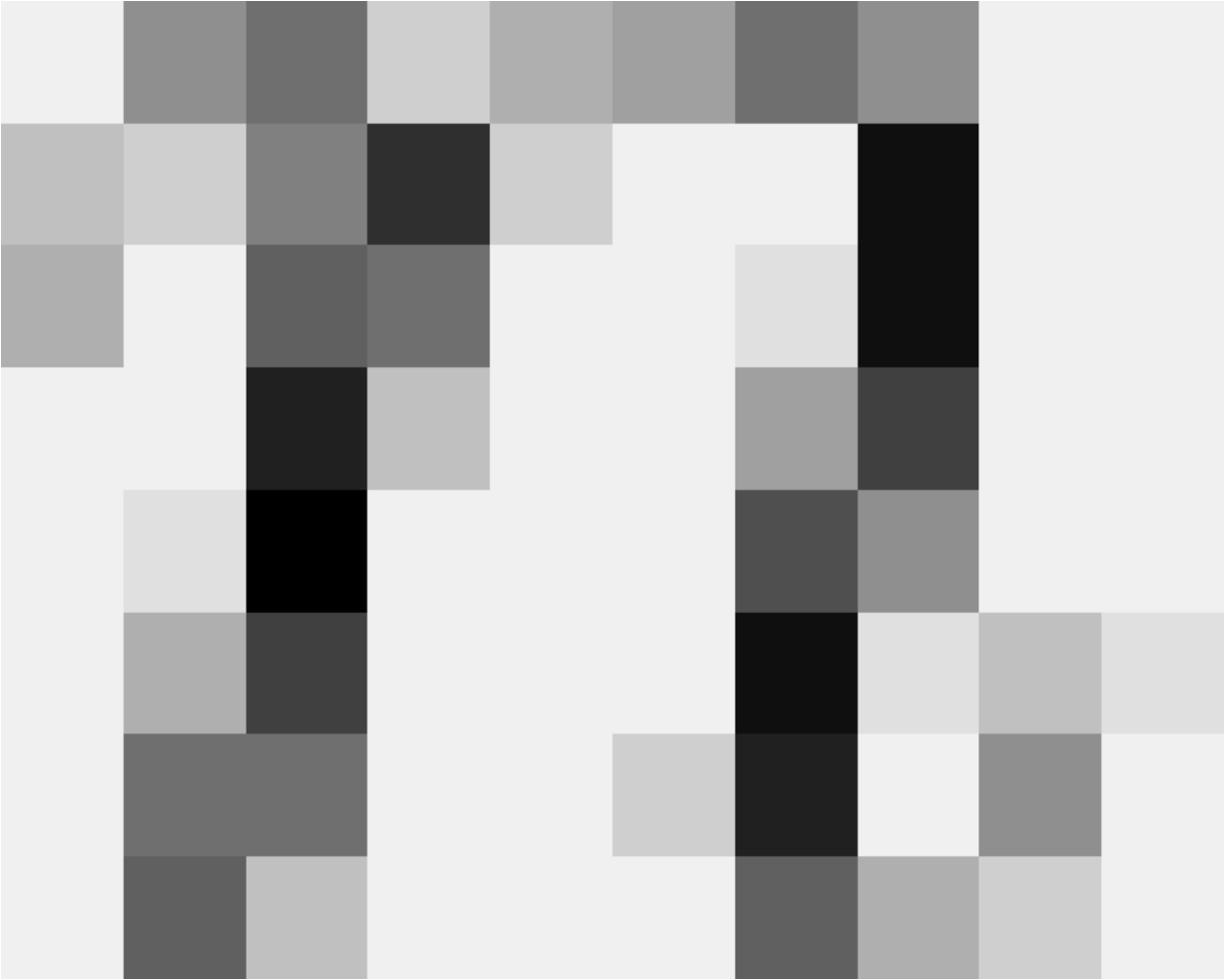
Myth 6: frequentist methods can only deal with simple models and need to make arbitrary cutoffs in model complexity. A naive perusal of the literature might lead one to believe that frequentists only ever consider very simple models, because many discussions center on linear and log-linear models. To dispel this, I will first note that there are just as many discussions that focus on much more general properties such as convexity and smoothness, and that can achieve comparably good bounds in many cases. But more importantly, the reason we focus so much on linear models is because **we have already reduced a large family of problems to (log-)linear regression.** The key insight, and I think one of the most important insights in all of applied mathematics, is that of **featurization**: given a *non-linear* problem, we can often embed it into a higher-dimensional *linear* problem, via a feature map

$$\phi : X \rightarrow \mathbb{R}^n$$

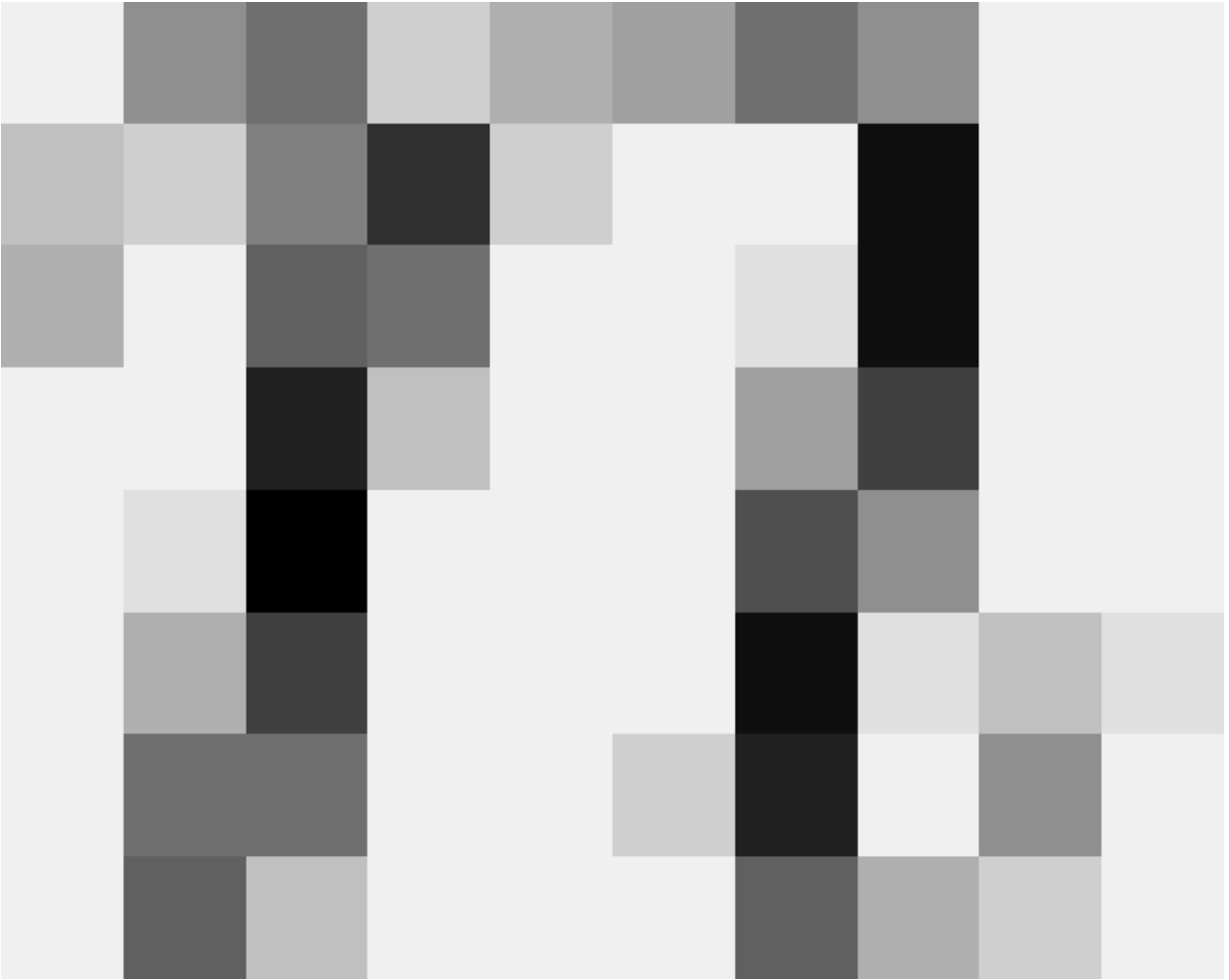
(



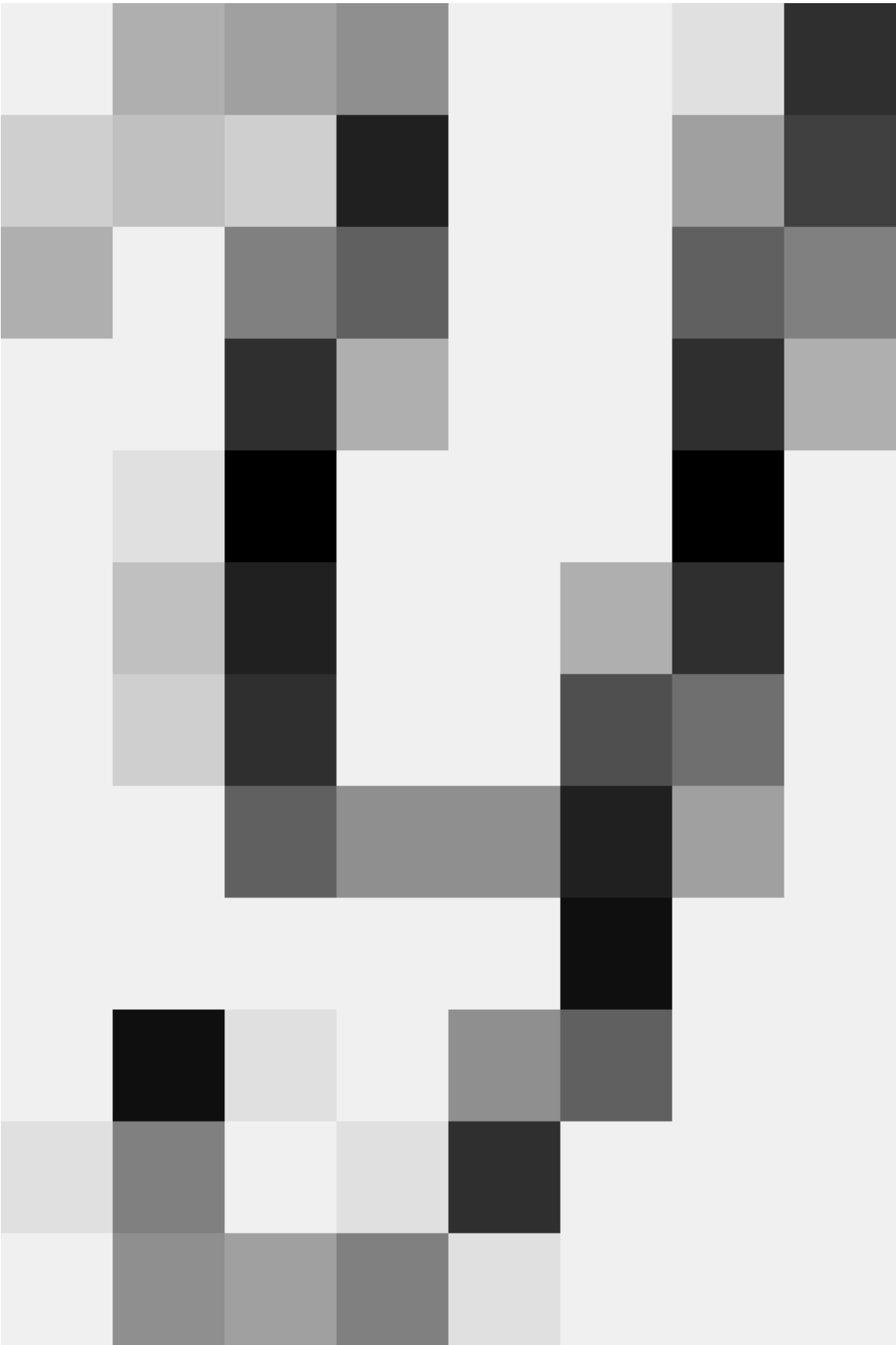
denotes



-dimensional space, i.e. vectors of real numbers of length

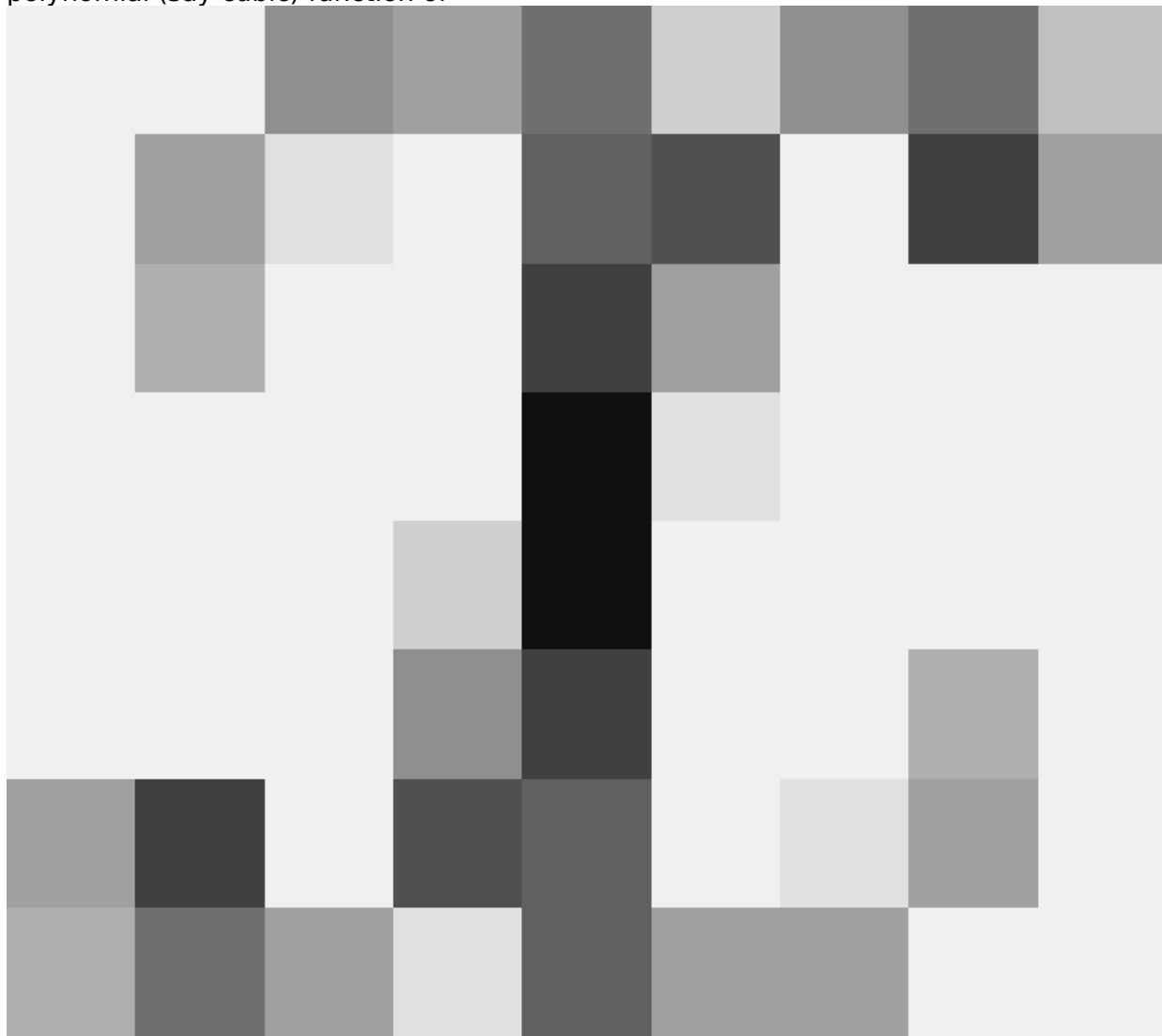


). For instance, if I think that



is a

polynomial (say cubic) function of



, I can apply the mapping

$$\phi(x) = (1, x, x^2, x^3)$$

, and now look for a *linear* relationship between

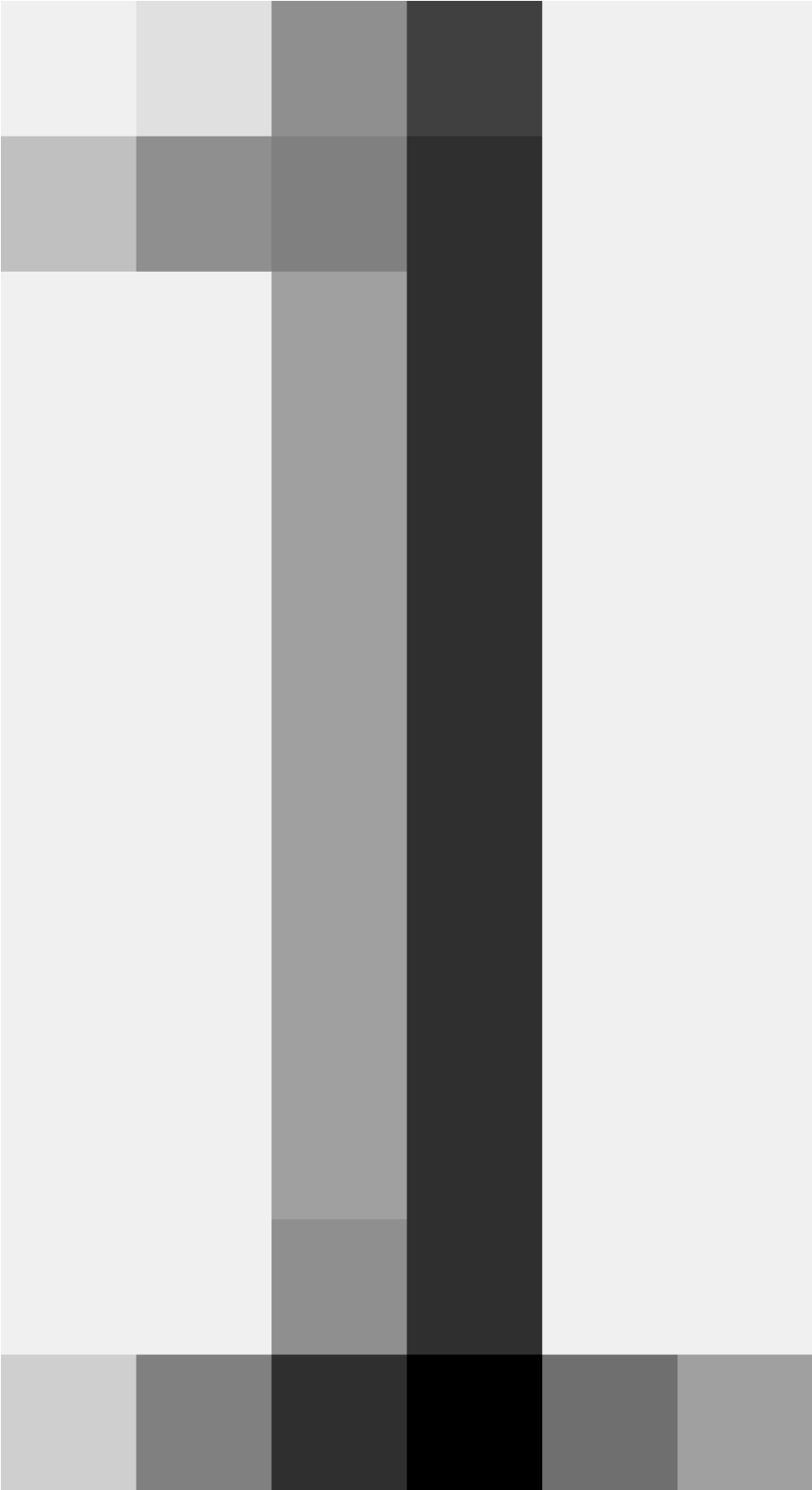


and



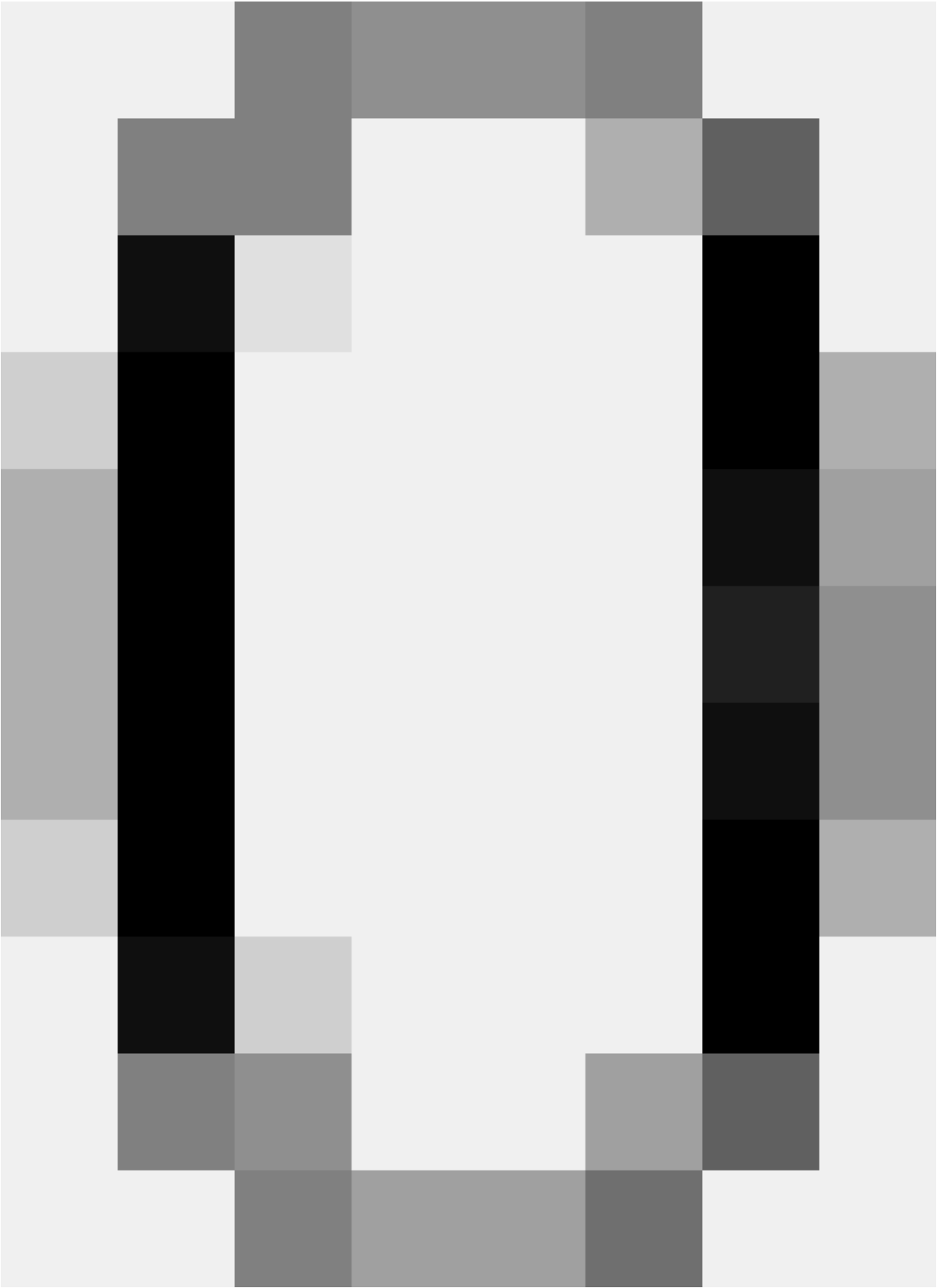
.

This insight extends far beyond polynomials. In combinatorial domains such as natural language, it is common to use *indicator features*: features that are



if a certain event

occurs and



otherwise. For instance, I might have an indicator feature for whether two words appear consecutively in a sentence, whether two parts of speech are adjacent in a syntax tree, or for what part of speech a word has. Almost all state of the art systems in natural language processing work by solving a relatively simple regression task (typically either log-linear or max-margin) over a rich feature space (often involving hundreds of thousands or millions of features, i.e. an embedding into



or



).

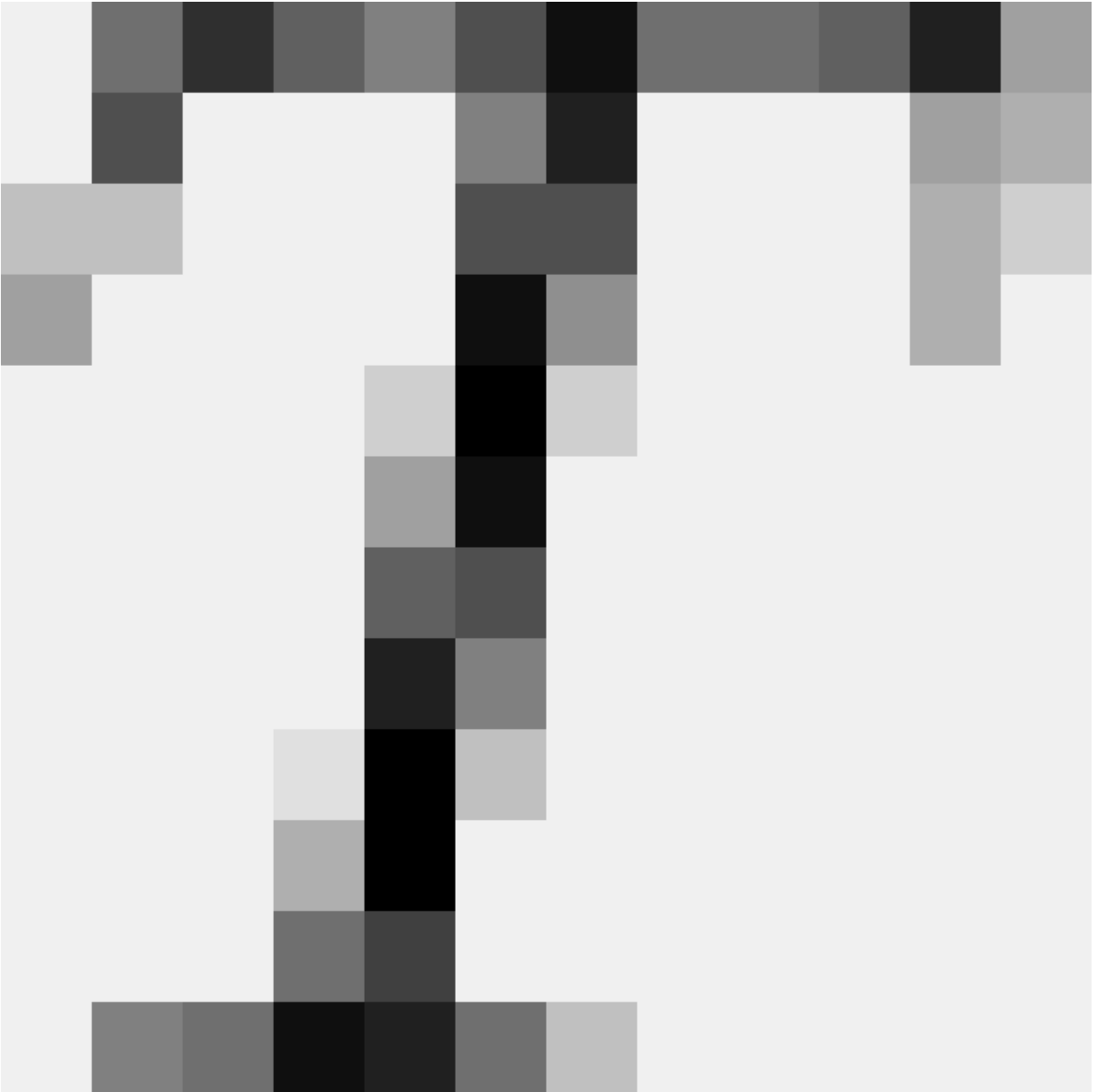
A counter-argument to the previous point could be: "Sure, you could create a high-dimensional family of models, but it's still a *parameterized family*. I don't want to be

stuck with a parameterized family, I want my family to include all Turing machines!" Putting aside for a second the question of whether "all Turing machines" is a well-advised model choice, this is something that a frequentist approach can handle just fine, using a tool called *regularization*, which after featurization is the second most important idea in statistics.

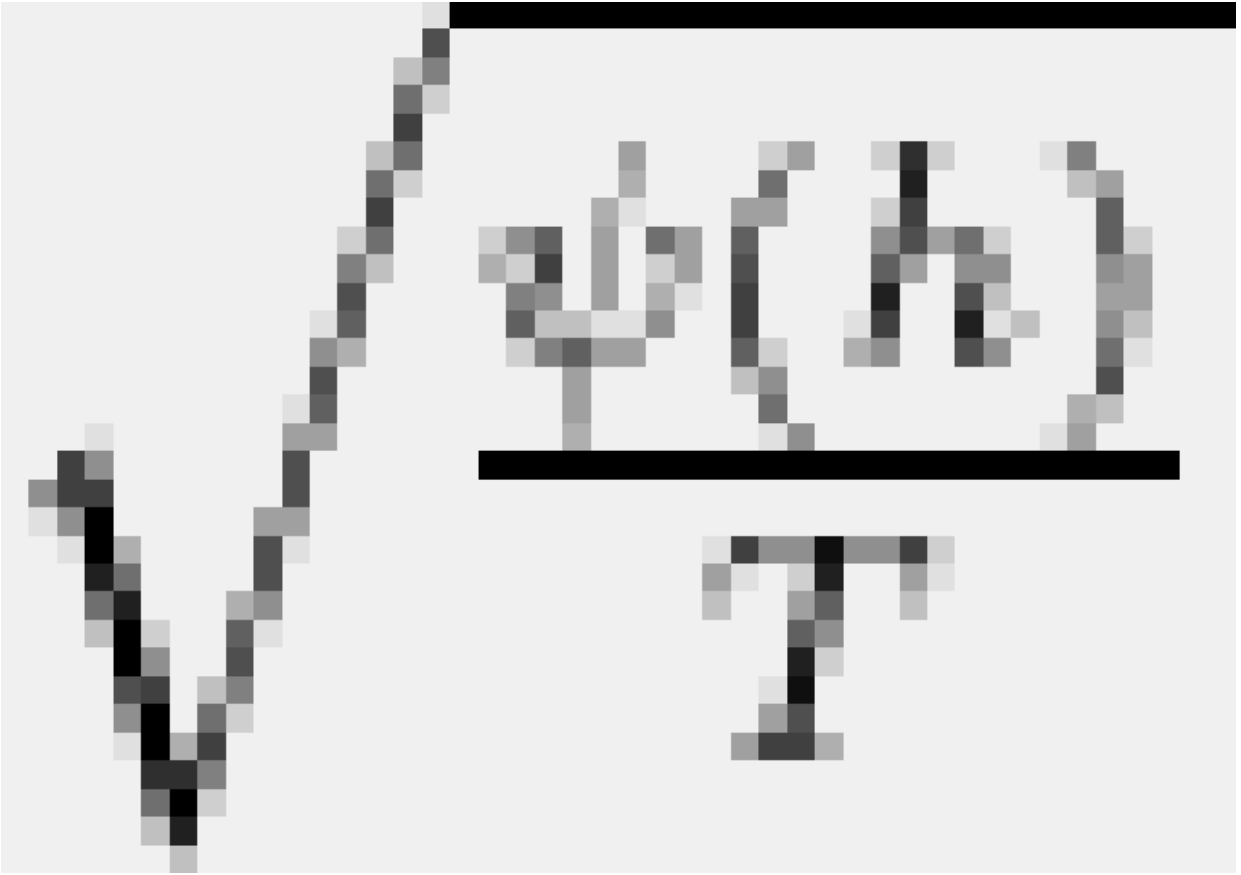
Specifically, given any sufficiently quickly growing function



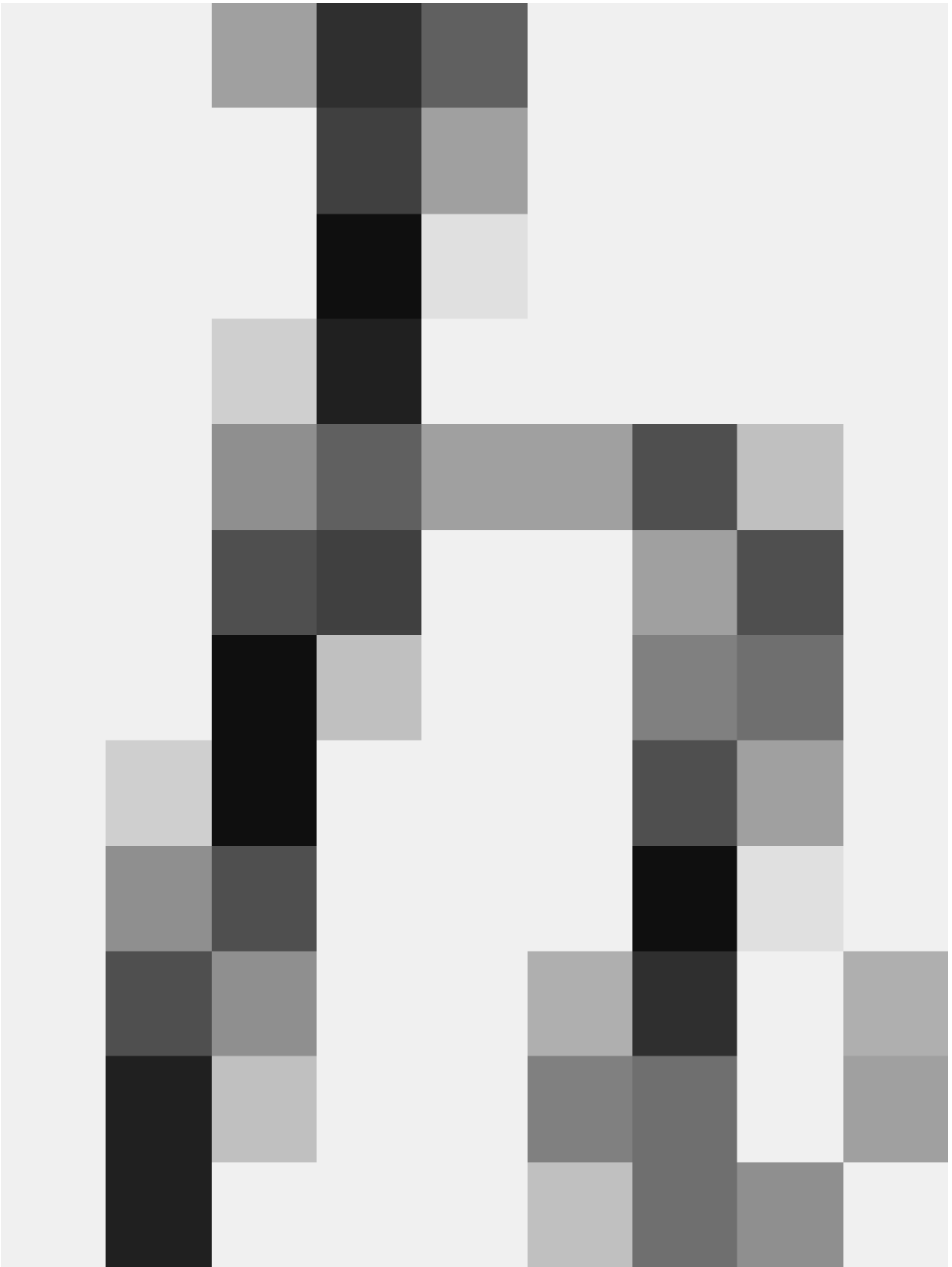
, one can show that, given



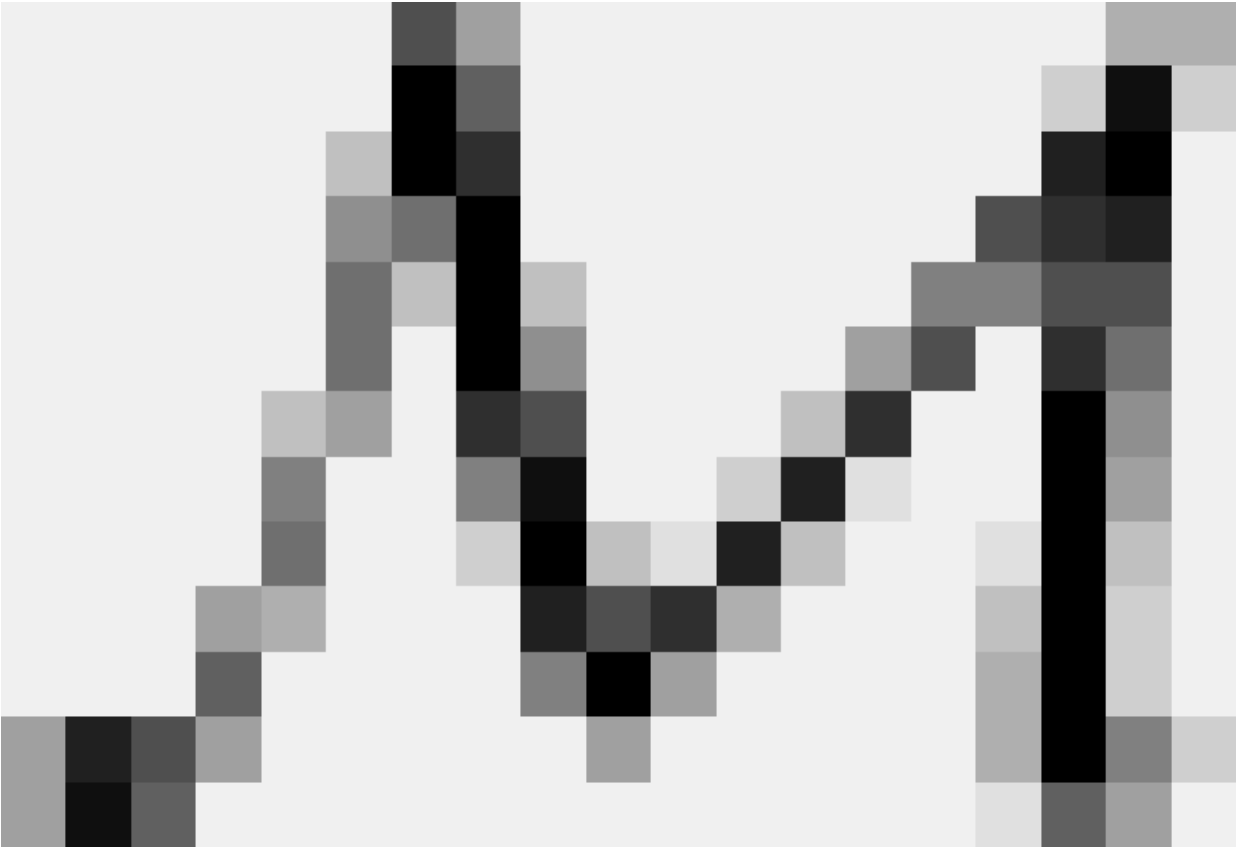
data points, there is a strategy whose average error is at most



worse than *any* estimator



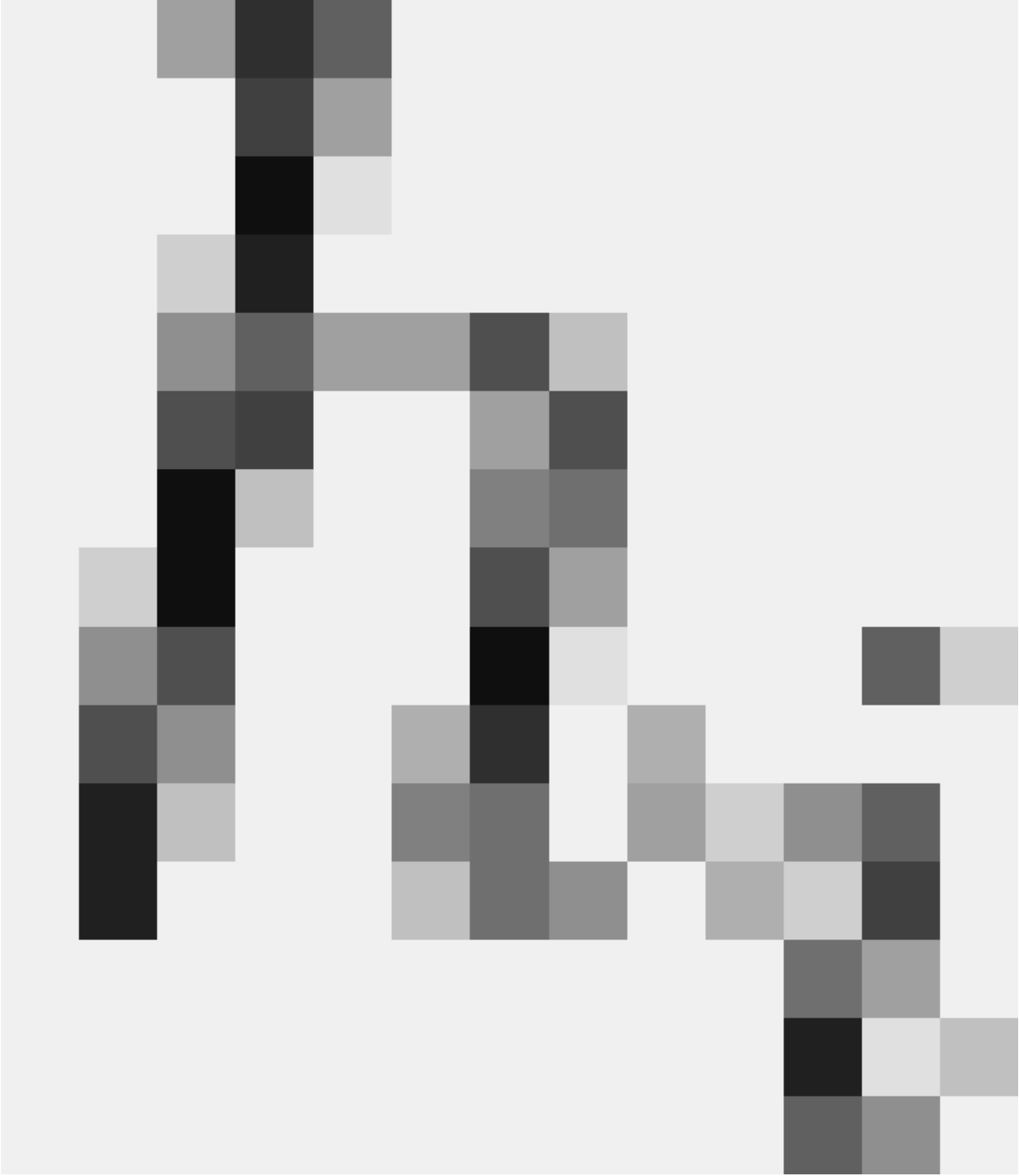
. This can hold even if the model class



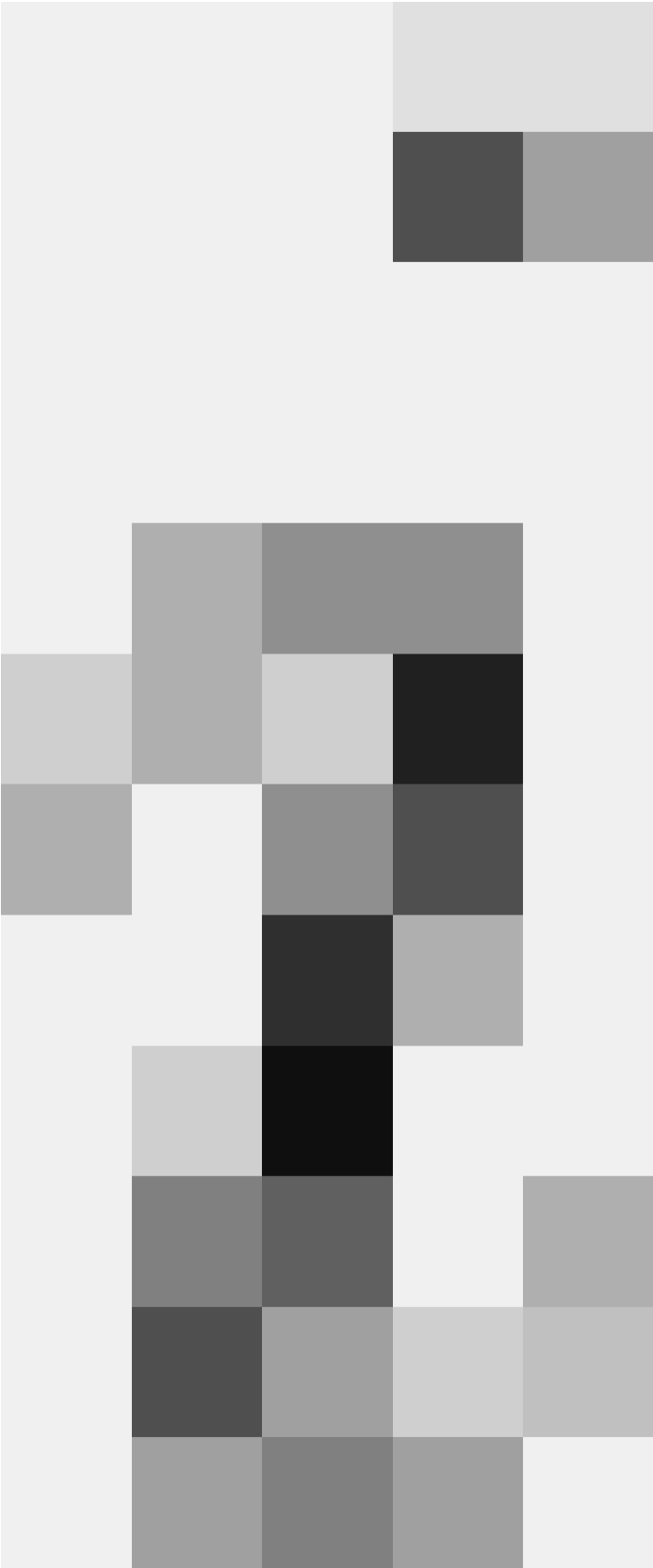
is infinite dimensional. For instance, if



consists of all probability distributions over Turing machines, and we let

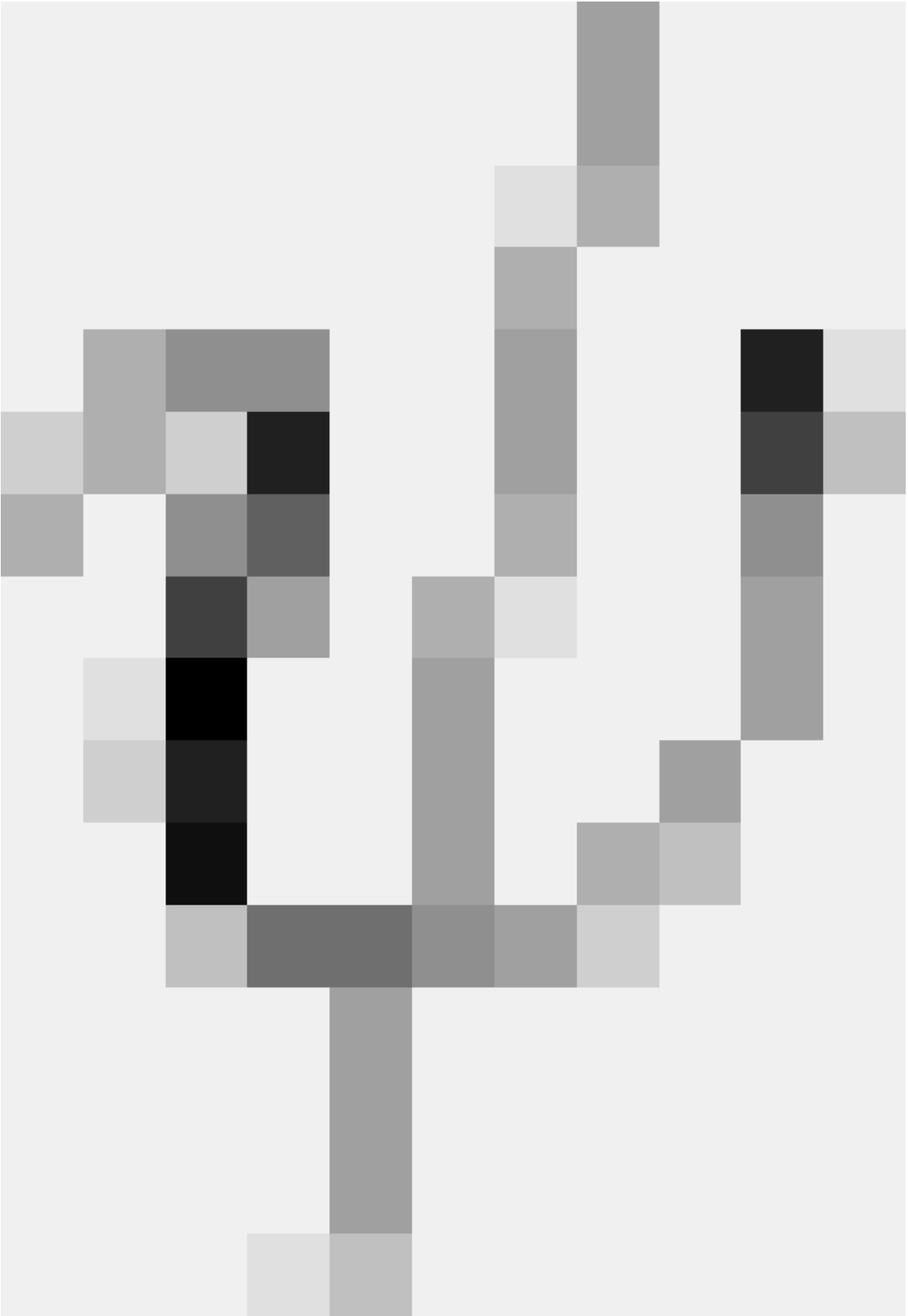


denote the probability mass placed on the



th Turing machine, then a valid

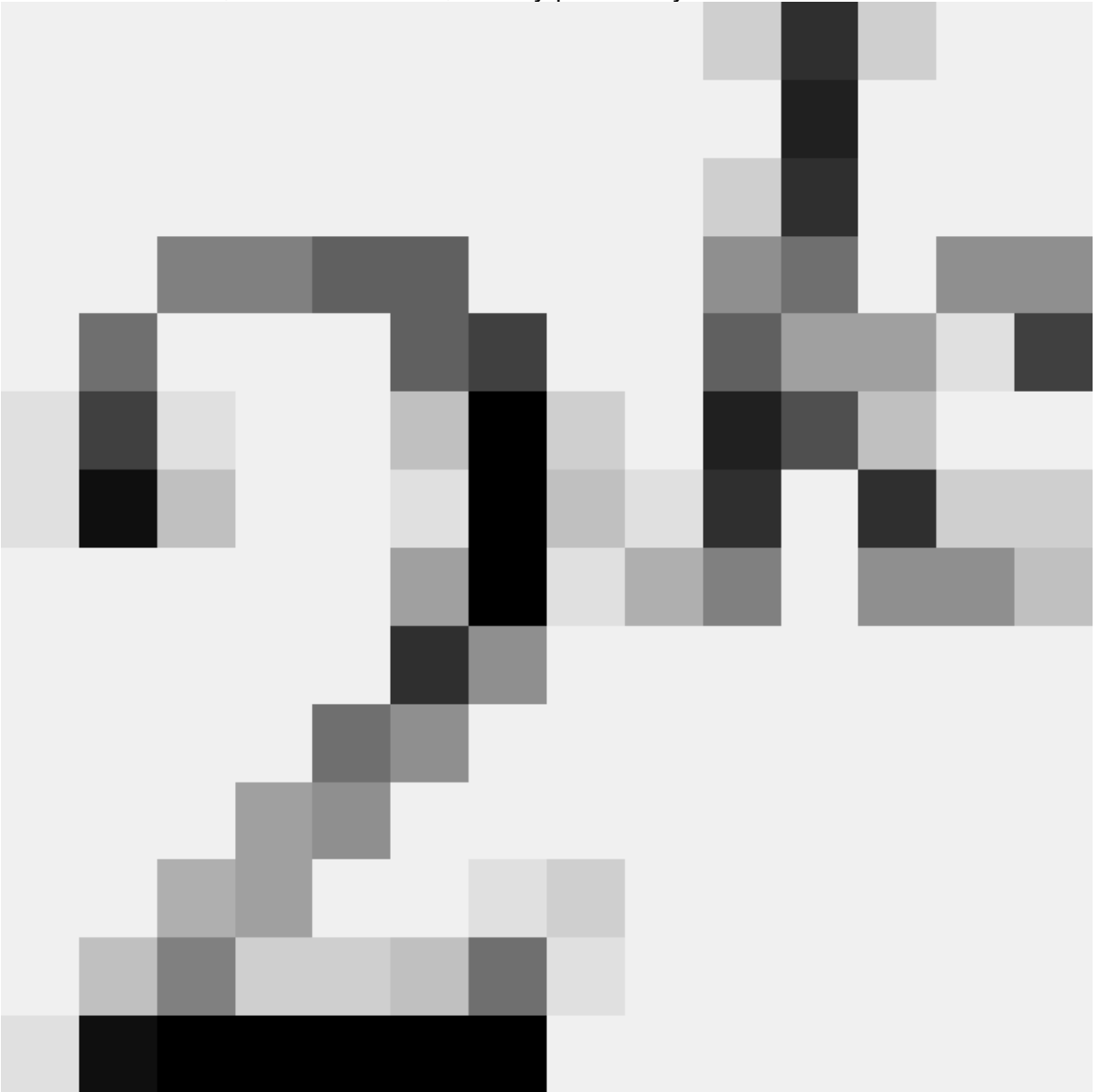
regularizer



would be

$$\psi(h) = \sum_i h_i \log(i^2 \cdot h_i)$$

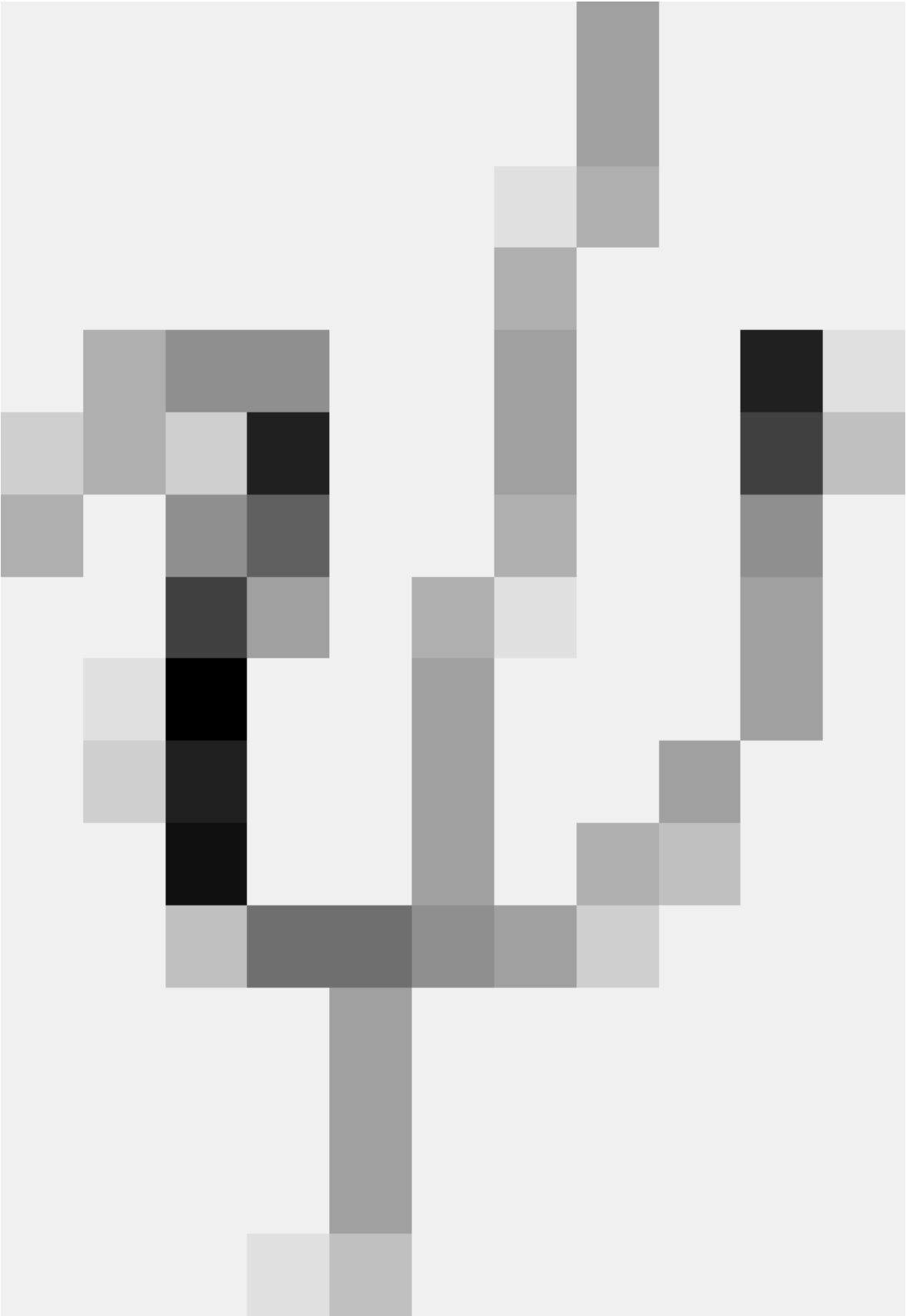
If we consider this, then we see that, for any probability distribution over the first



Turing machines (i.e. all Turing machines with description length



), the value of



is

at most

$$\log((2^k)^2) = k \log(4)$$

. (Here we use the fact that

$$\psi(h) \geq \sum_i h_i \log(i^2)$$

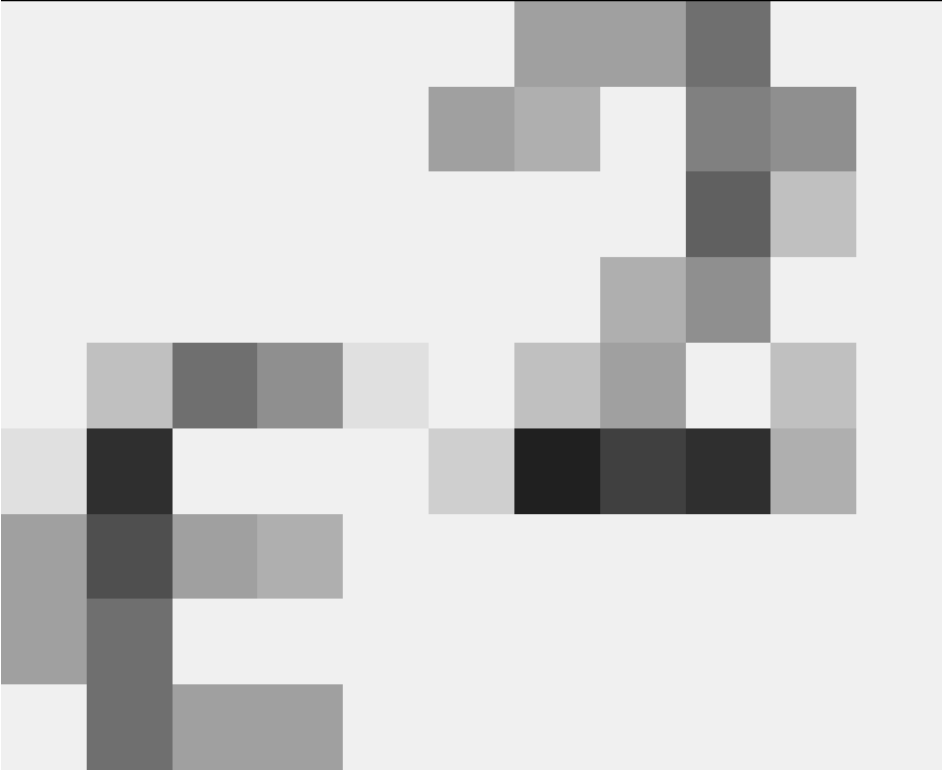
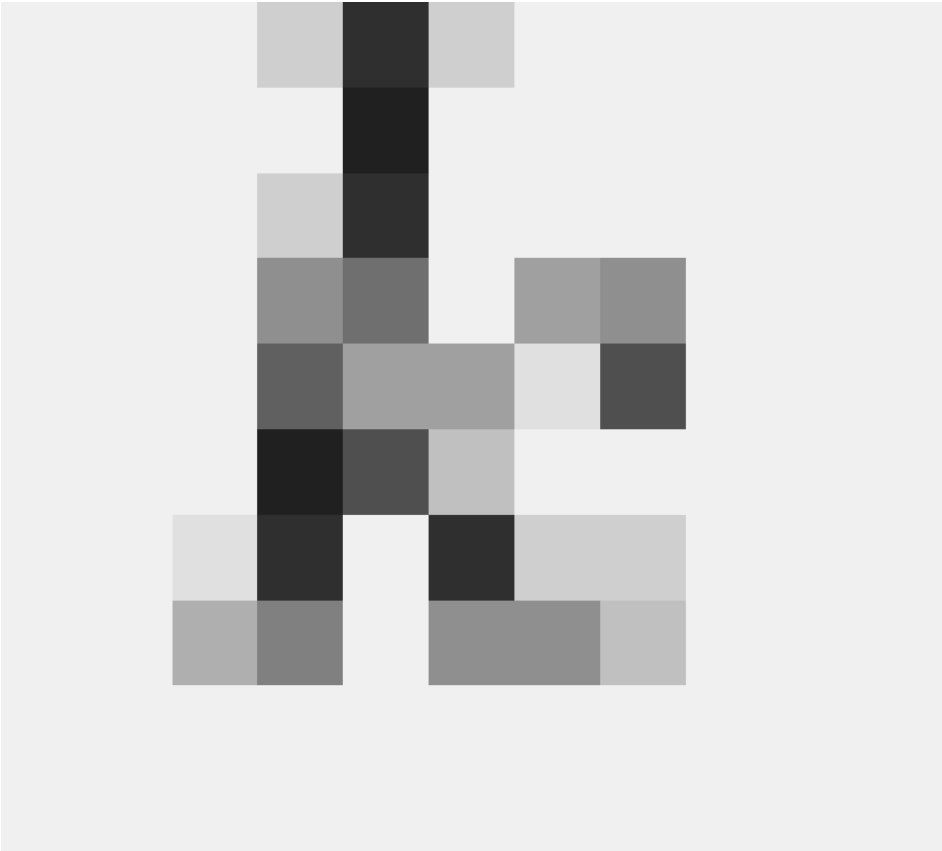
, since

$$h_i \leq 1$$

and hence

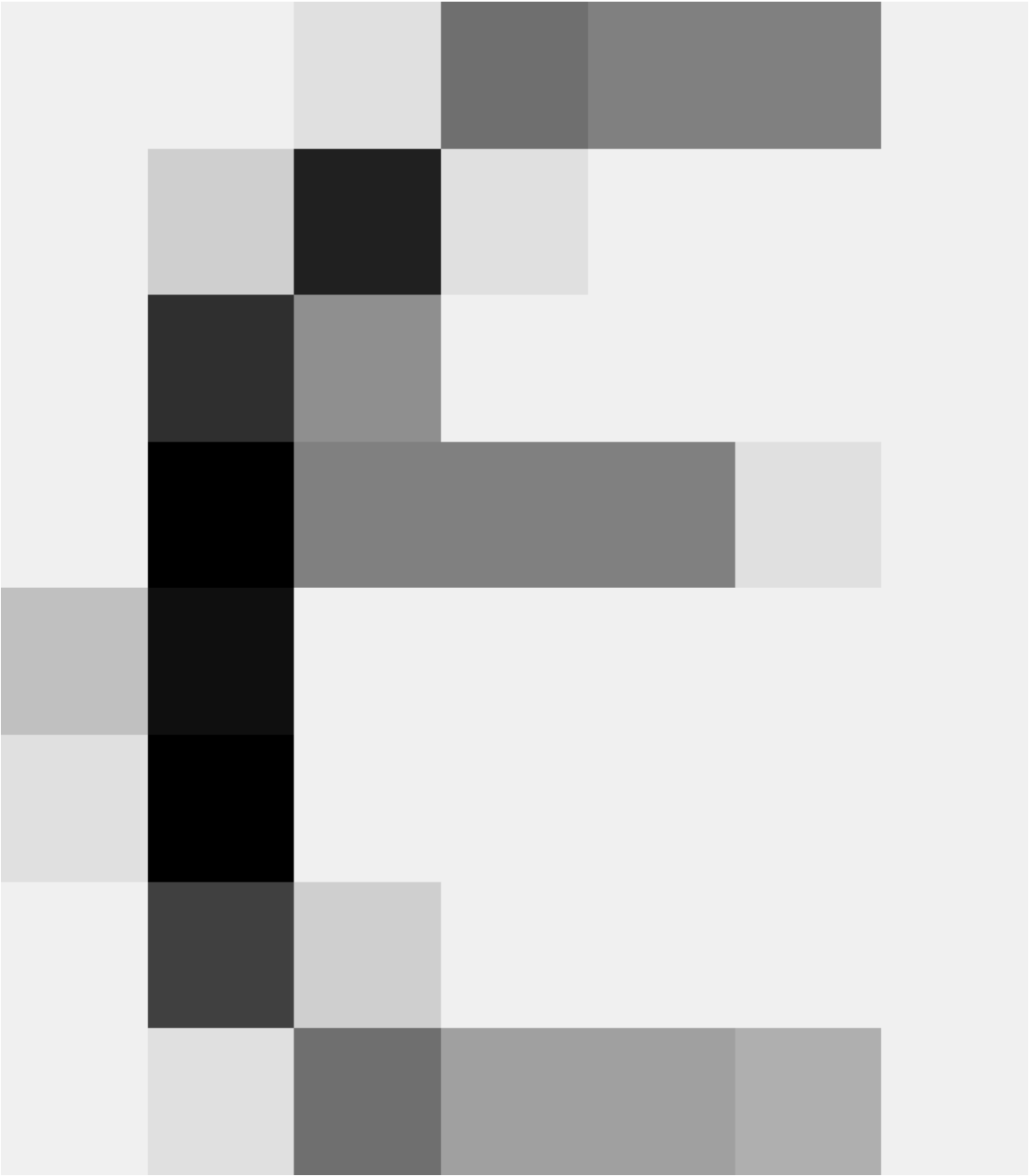
$$h_i \log(h_i) \leq 0$$

.) This means that, if we receive roughly



data, we will

achieve error within



of the best Turing machine that has description length



•

Let me note several things here:

- This strategy makes no assumptions about the data being i.i.d. It doesn't even assume that the data are computable. It just guarantees that it will perform as well as any Turing machine (or distribution over Turing machines) given the appropriate amount of data.
- This guarantee holds for any given sufficiently smooth measurement of prediction error (the update strategy depends on the particular error measure).
- This guarantee holds deterministically, no randomness required (although predictions may need to consist of probability distributions rather than specific points, but this is also true of Bayesian predictions).

Interestingly, in the case that the prediction error is given by the negative log probability assigned to the truth, then the corresponding strategy that achieves the error bound is just normal Bayesian updating. But for other measurements of error, we get different update strategies. Although I haven't worked out the math, intuitively this difference could be important if the universe is fundamentally unpredictable but our notion of error is insensitive to the unpredictable aspects.

Myth 7: frequentist methods hide their assumptions while Bayesian methods make assumptions explicit. I'm still not really sure where this came from. As we've seen numerous times so far, a very common flavor among frequentist methods is the following: I have a model class



, I want to do as well as any model in

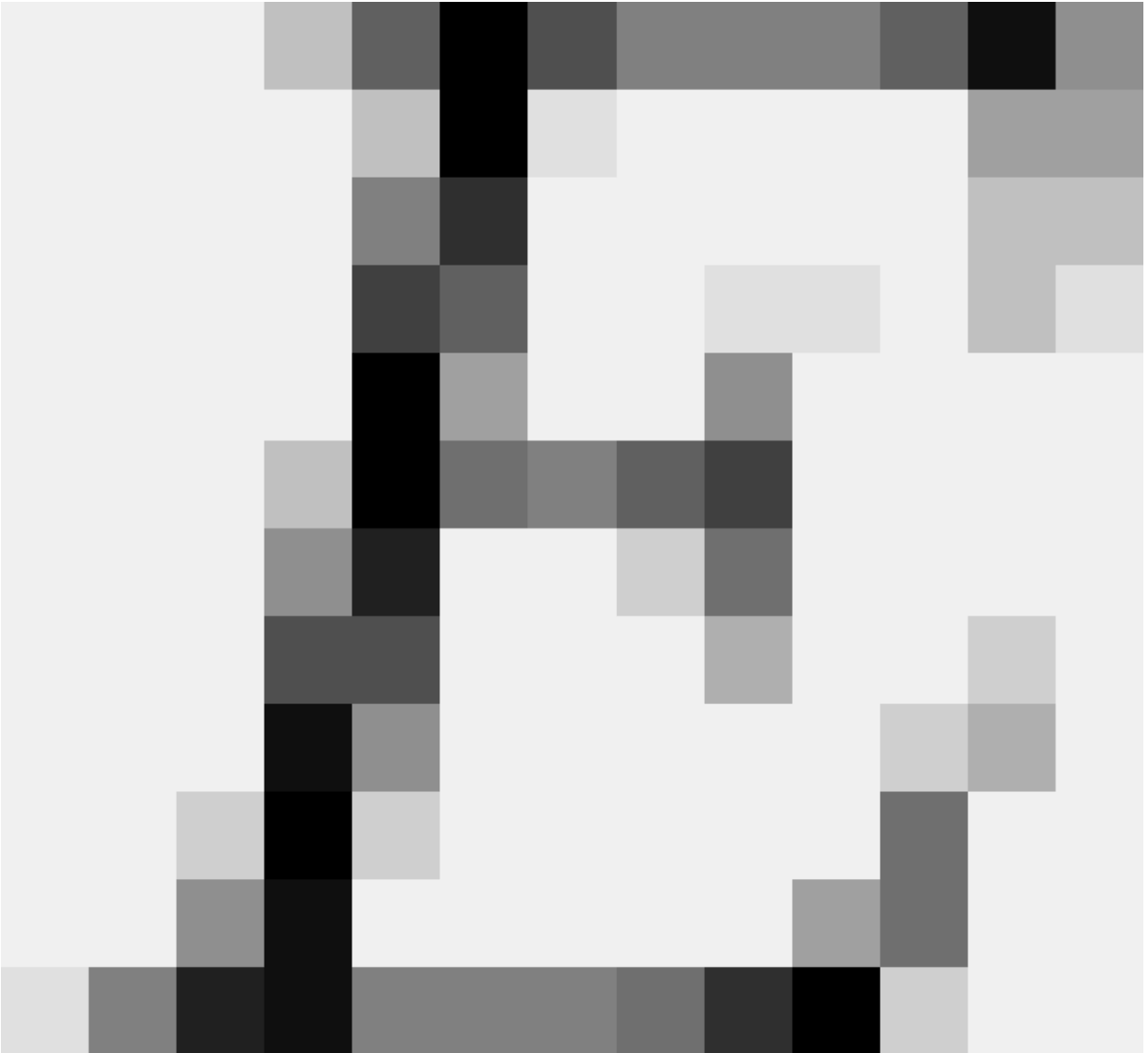


; or put another way:

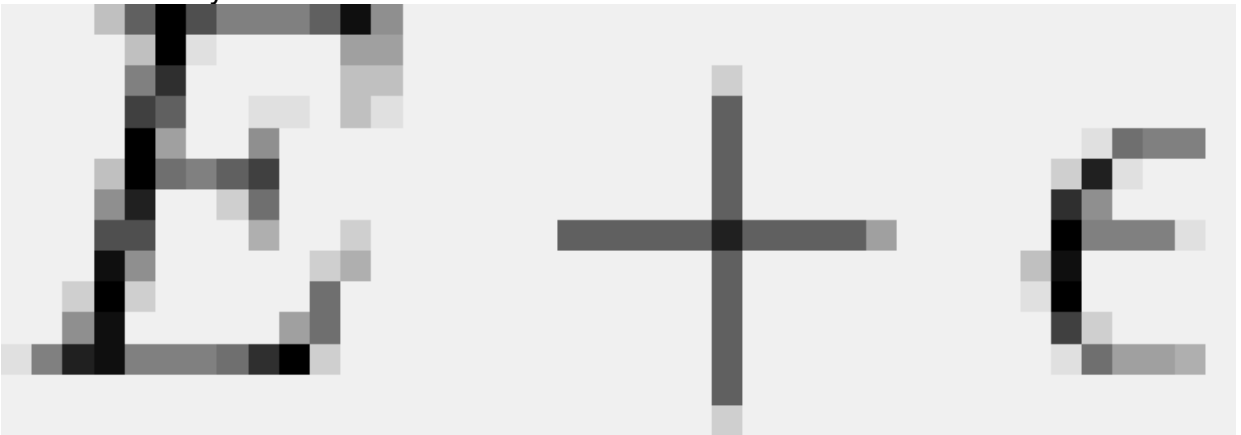
Assumption: At least one model in



has error at most



Guarantee: My method will have error at most



This seems like a very explicit assumption with a very explicit guarantee. On the other hand, an argument I hear is that Bayesian methods make their assumptions explicit because they have an explicit prior. If I were to write this as an assumption and guarantee, I would write:

Assumption: The data were generated from the prior.

Guarantee: I will perform at least as well as any other method.

While I agree that this is an assumption and guarantee of Bayesian methods, there are two problems that I have with drawing the conclusion that “Bayesian methods make their assumptions explicit”. The first is that it can often be very difficult to understand how a prior behaves; so while we could say “The data were generated from the prior” is an explicit assumption, it may be unclear what exactly that assumption entails. However, a bigger issue is that “The data were generated from the prior” is an assumption that very rarely holds; indeed, in many cases the underlying process is deterministic (if you’re a subjective Bayesian then this isn’t necessarily a problem, but it does certainly mean that the assumption given above doesn’t hold). So given that that assumption doesn’t hold but Bayesian methods still often perform well in practice, I would say that Bayesian methods are making some other sort of “assumption” that is far less explicit (indeed, I would be very interested in understanding what this other, more nebulous assumption might be).

Myth 8: frequentist methods are fragile, Bayesian methods are robust. This is another one that’s straightforwardly false. First, since frequentist methods often rest on weaker assumptions they are more robust if the assumptions don’t quite hold. Secondly, there is an entire area of robust statistics, which focuses on being robust to adversarial errors in the problem data.

Myth 9: frequentist methods are responsible for bad science. I will concede that much bad science is done using frequentist statistics. But this is true only because pretty much all science is done using frequentist statistics. I’ve heard arguments that using Bayesian methods instead of frequentist methods would fix at least some of the problems with science. I don’t think this is particularly likely, as I think many of the problems come from mis-application of statistical tools or from failure to control for multiple hypotheses. If anything, Bayesian methods would exacerbate the former, because they often require more detailed modeling (although in most simple cases the difference doesn’t matter at all). I don’t think being Bayesian guards against multiple hypothesis testing. Yes, in some sense a prior “controls for multiple hypotheses”, but in general the issue is that the “multiple hypotheses” are never written down in the first place, or are written down and then discarded. One could argue that being in the habit of writing down a prior might make practitioners more likely to think about multiple hypotheses, but I’m not sure this is the first-order thing to worry about.

Myth 10: frequentist methods are unprincipled / hacky. One of the most beautiful theoretical paradigms that I can think of is what I could call the “geometric view of statistics”. One place that does a particularly good job of show-casing this is [Shai Shalev-Shwartz’s PhD thesis](#), which was so beautiful that I cried when I read it. I’ll try (probably futilely) to convey a tiny amount of the intuition and beauty of this paradigm in the next few paragraphs, although focusing on minimax estimation, rather than online learning as in Shai’s thesis.

The geometric paradigm tends to emphasize a view of measurements (i.e. empirical expected values over observed data) as “noisy” linear constraints on a model family. We can control the noise by either taking few enough measurements that the total error from the noise is small (classical statistics), or by broadening the linear

constraints to convex constraints (robust statistics), or by controlling the Lagrange multipliers on the constraints (regularization). One particularly beautiful result in this vein is the duality between maximum entropy and maximum likelihood. (I can already predict the Jaynesians trying to claim this result for their camp, but (i) Jaynes did not invent maximum entropy; (ii) maximum entropy is not particularly Bayesian (in the sense that frequentists use it as well); and (iii) the view on maximum entropy that I'm about to provide is different from the view given in Jaynes or by physicists in general [edit: EHeller thinks this last claim is questionable, see discussion [here](#)].)

To understand the duality mentioned above, suppose that we have a probability distribution



and the only information we have about it is the expected value of a certain number of functions, i.e. the information that

$$\mathbb{E}[\phi(x)] = \phi^*$$

, where the expectation is taken with respect to



. We are interested in constructing a probability distribution



such that no matter what particular value



takes,



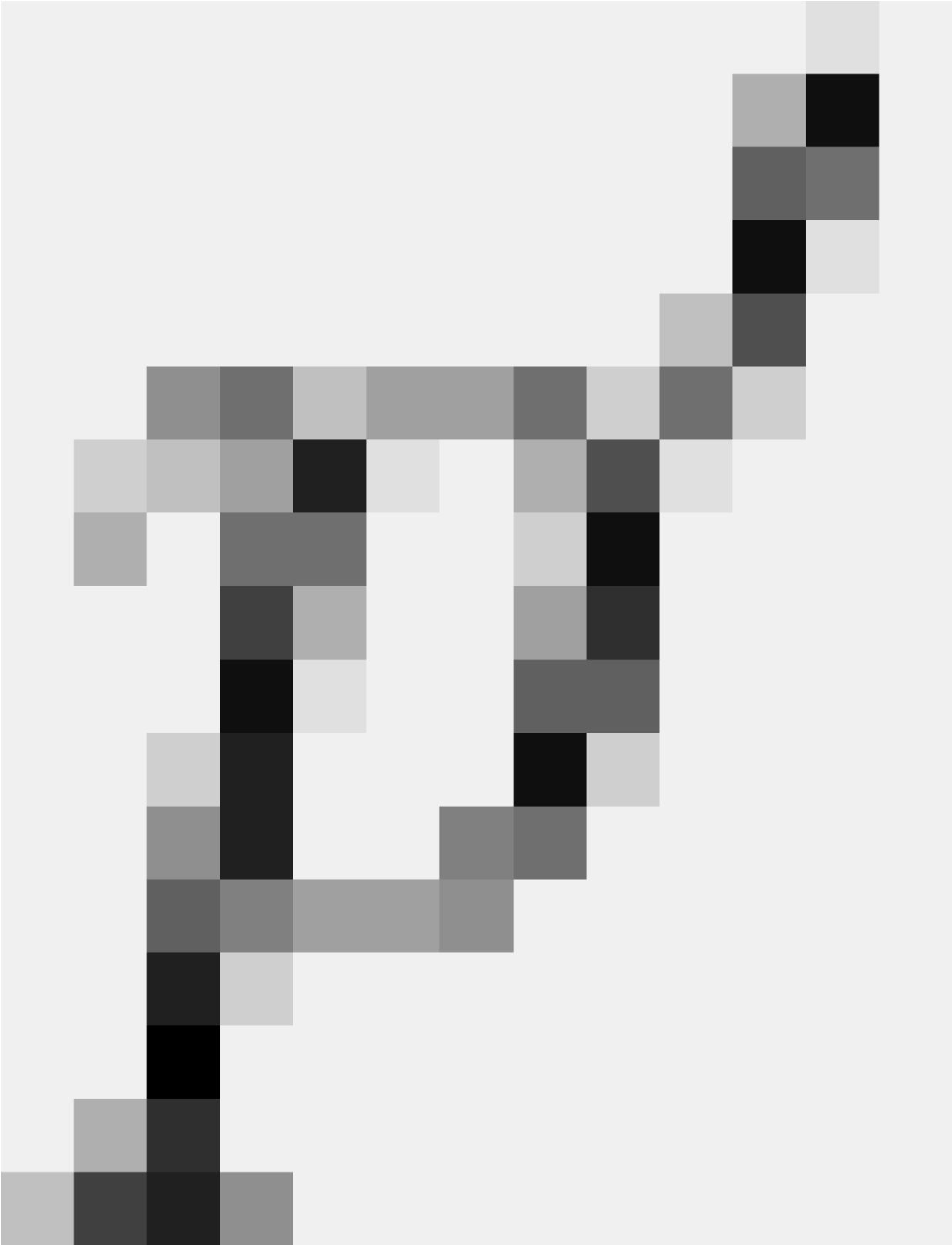
will still make good predictions. In other words (taking

$$\log p(x)$$

as our measurement of prediction accuracy) we want

$$\mathbb{E}_p[\log q(x)]$$

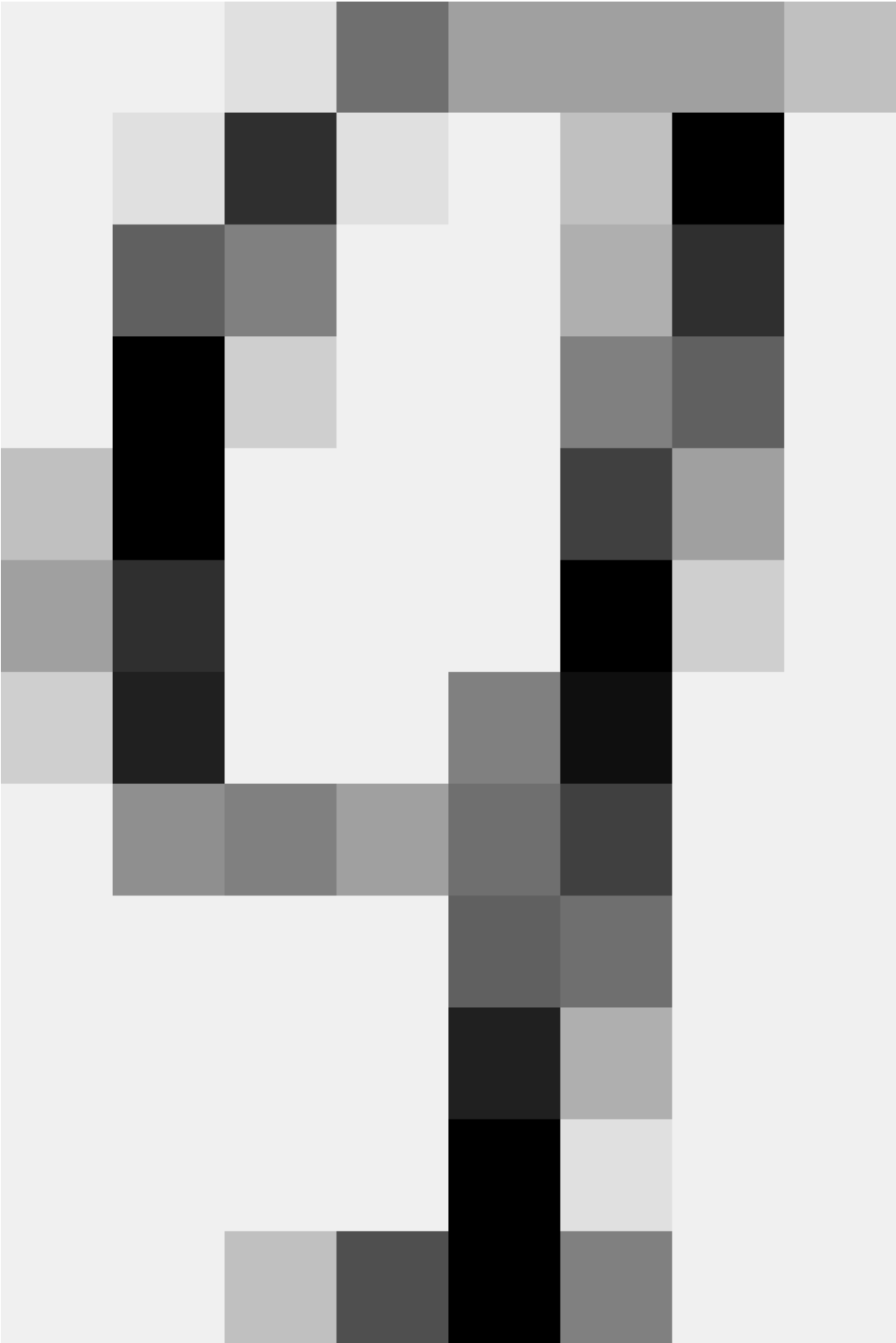
to be large for all distributions



such that

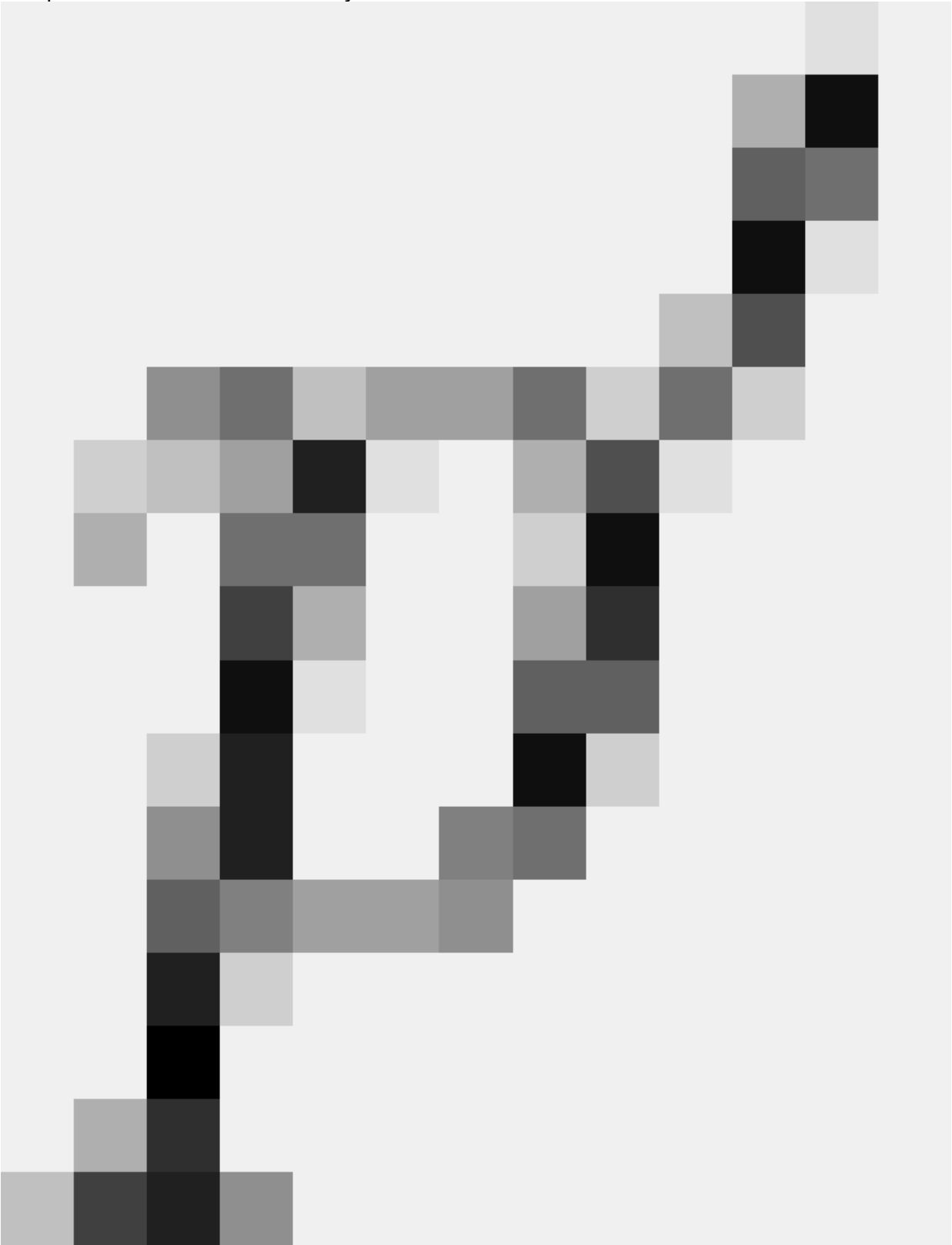
$$\mathbb{E}_{\mathcal{P}}[\phi(x)] = \phi^*$$

. Using a technique called Lagrangian duality, we can both find the optimal distribution



and

compute its worse-case accuracy over all



with

$$\mathbb{E}_p[\phi(x)] = \phi^*$$

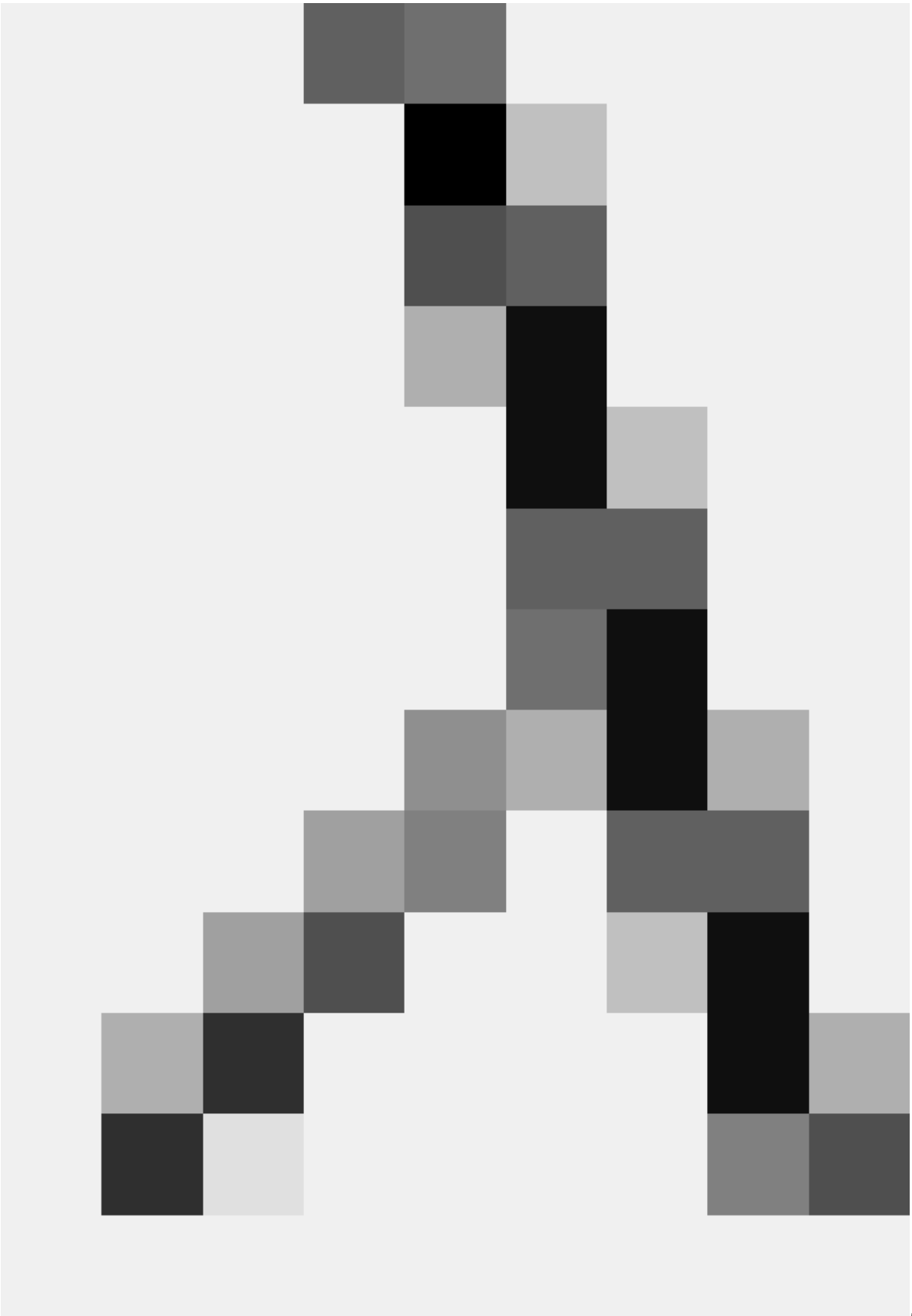
. The characterization is as follows: consider all probability distributions

$$q(x)$$

that are proportional to

$$\exp(\lambda^T \phi(x))$$

for some vector



i.e.

$$q(x) = \exp(\lambda^\top \phi(x)) / Z(\lambda)$$

for some



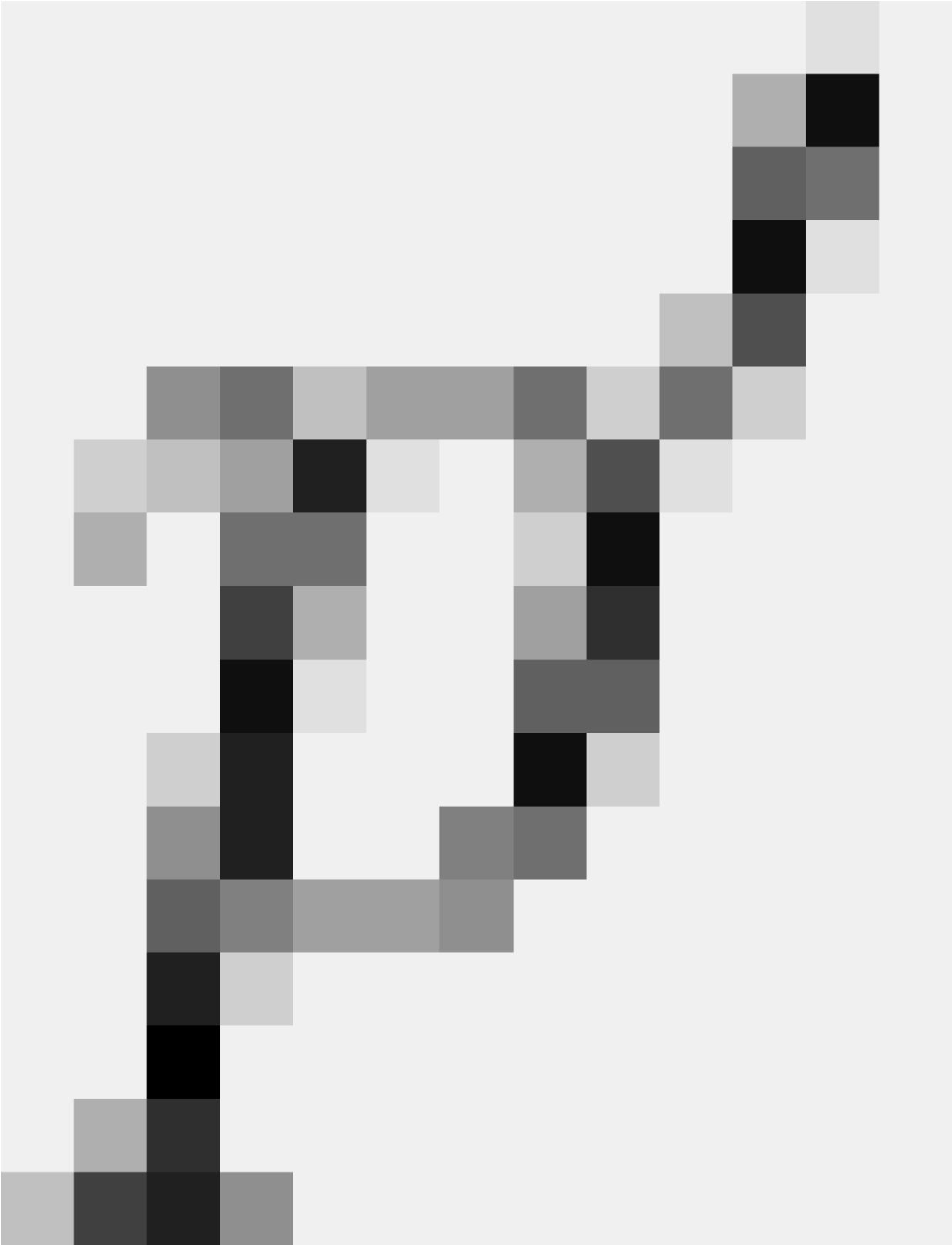
. Of all of these, take the $q(x)$ with the largest value of

$$\lambda^\top \phi^* - \log Z(\lambda)$$

. Then



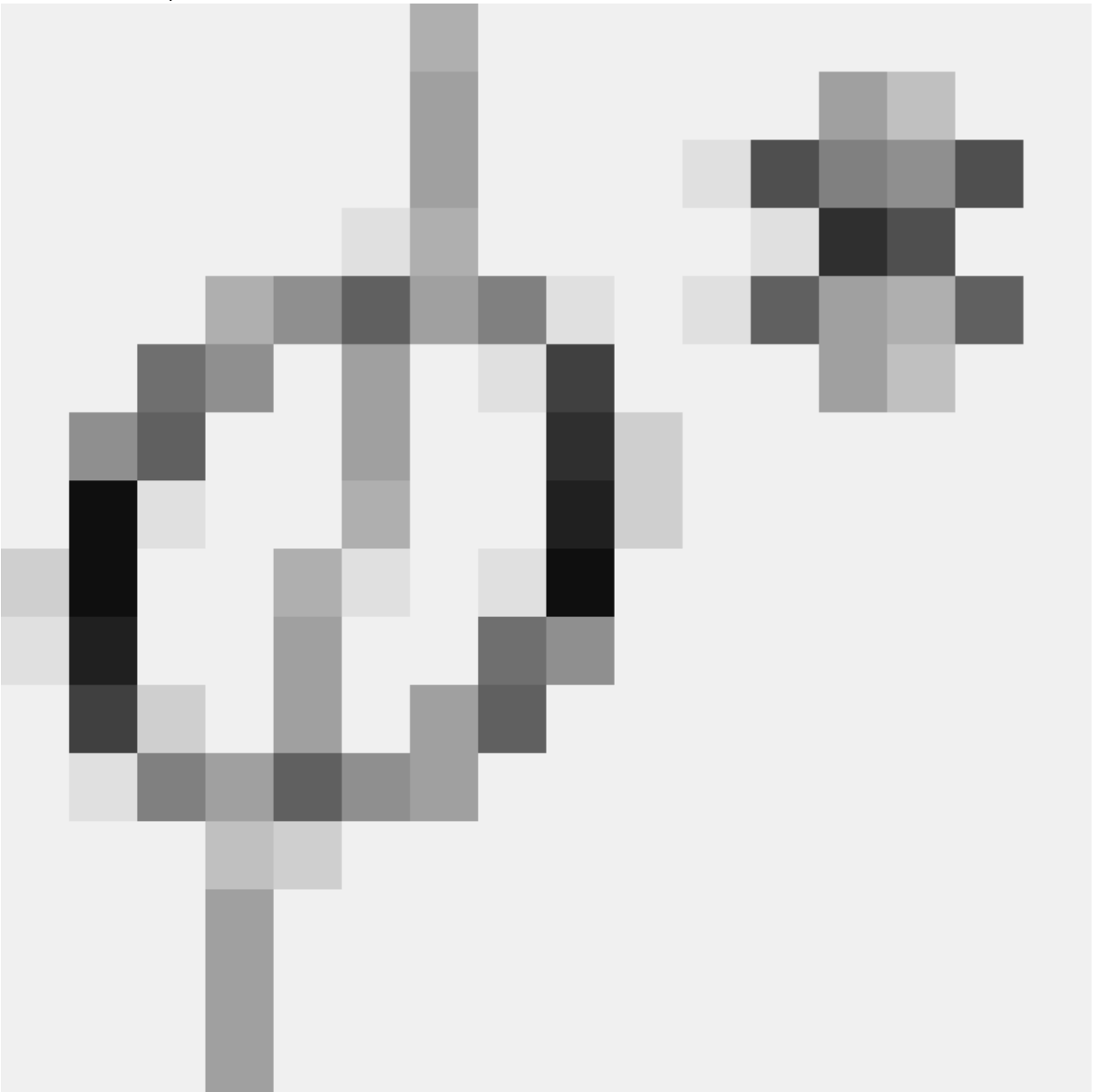
will be the optimal distribution and the accuracy for *all* distributions



will be exactly

$$\lambda^T \phi^* - \log Z(\lambda)$$

. Furthermore, if



is the empirical expectation given some number of samples, then one can show that

$$\lambda^T \phi^* - \log Z(\lambda)$$

is proportional to the log likelihood of

which is why I say that maximum entropy and maximum likelihood are dual to each other.

This is a relatively simple result but it underlies a decent chunk of models used in practice.

Myth 11: frequentist methods have no promising approach to computationally bounded inference. I would personally argue that frequentist methods are *more* promising than Bayesian methods at handling computational constraints, although computationally bounded inference is a very cutting edge area and I'm sure other experts would disagree. However, one point in favor of the frequentist approach here is that we already have some frameworks, such as the "tightening relaxations" framework discussed [here](#), that provide quite elegant and rigorous ways of handling computationally intractable models.

References

(Myth 3) Sparse recovery: [Sparse recovery using sparse matrices](#)

(Myth 5) Online learning: [Online learning and online convex optimization](#)

(Myth 8) Robust statistics: see [this](#) blog post and the [two linked](#) papers

(Myth 10) Maximum entropy duality: [Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory](#)

Bridge Collapse: Reductionism as Engineering Problem

Followup to: [Building Phenomenological Bridges](#)

Summary: AI theorists often use models in which agents are crisply separated from their environments. This simplifying assumption can be useful, but it leads to trouble when we build machines that presuppose it. A machine that believes it can only interact with its environment in a narrow, fixed set of ways will not understand the value, or the dangers, of self-modification. By analogy with Descartes' mind/body dualism, I refer to agent/environment dualism as *Cartesianism*. The [open problem in Friendly AI](#) (OPFAI) I'm calling [naturalized induction](#) is the project of replacing Cartesian approaches to scientific induction with reductive, physicalistic ones.

I'll begin with a story about a storyteller.

Once upon a time — specifically, 1976 — there was an AI named TALE-SPIN. This AI told [stories](#) by inferring how characters would respond to problems from background knowledge about the characters' traits. One day, TALE-SPIN constructed a most peculiar tale.

Henry Ant was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. Gravity drowned.

Since Henry fell in the river near his friend Bill, TALE-SPIN concluded that Bill rescued Henry. But for Henry to fall in the river, gravity must have pulled Henry. Which means gravity must have been in the river. TALE-SPIN had never been told that gravity knows how to swim; and TALE-SPIN had never been told that gravity has any friends. So gravity drowned.

TALE-SPIN had previously been programmed to understand involuntary motion in the case of characters being pulled or carried by other characters — like Bill rescuing Henry. So it was programmed to understand 'character X fell to place Y' as 'gravity moves X to Y', as though gravity were a character in the story.¹

For us, the hypothesis 'gravity drowned' has low prior probability because we know gravity isn't the *type* of thing that swims or breathes or makes friends. We want agents to seriously consider whether the law of gravity pulls down rocks; we don't want agents to seriously consider whether the law of gravity pulls down the law of electromagnetism. We [may not want](#) an AI to assign [zero probability](#) to 'gravity drowned', but we at least want it to neglect the possibility as Ridiculous-By-Default.

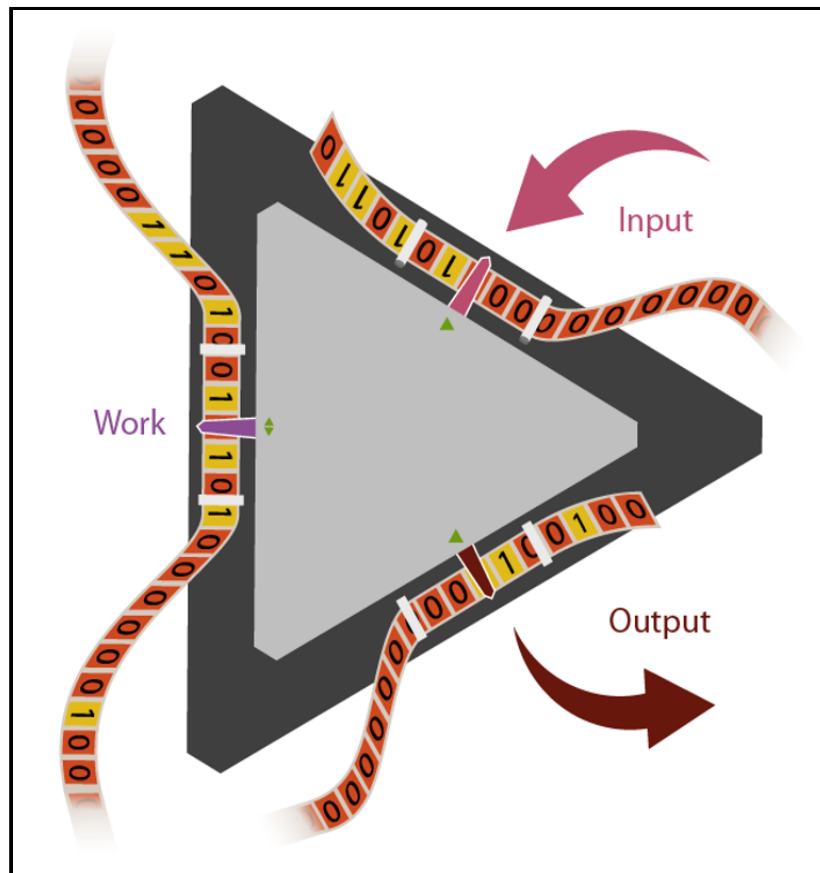
When we introduce deep type distinctions, however, we also introduce new ways our stories can fail.

Hutter's cybernetic agent model

Russell and Norvig's leading AI textbook credits Solomonoff with setting the agenda for the field of AGI: "AGI looks for a universal algorithm for learning and acting in any environment, and has its roots in the work of Ray Solomonoff[.]" As an approach to AGI, Solomonoff induction presupposes a model with a strong type distinction between the 'agent' and the 'environment'. To make its intuitive appeal and attendant problems more obvious, I'll sketch out the model.

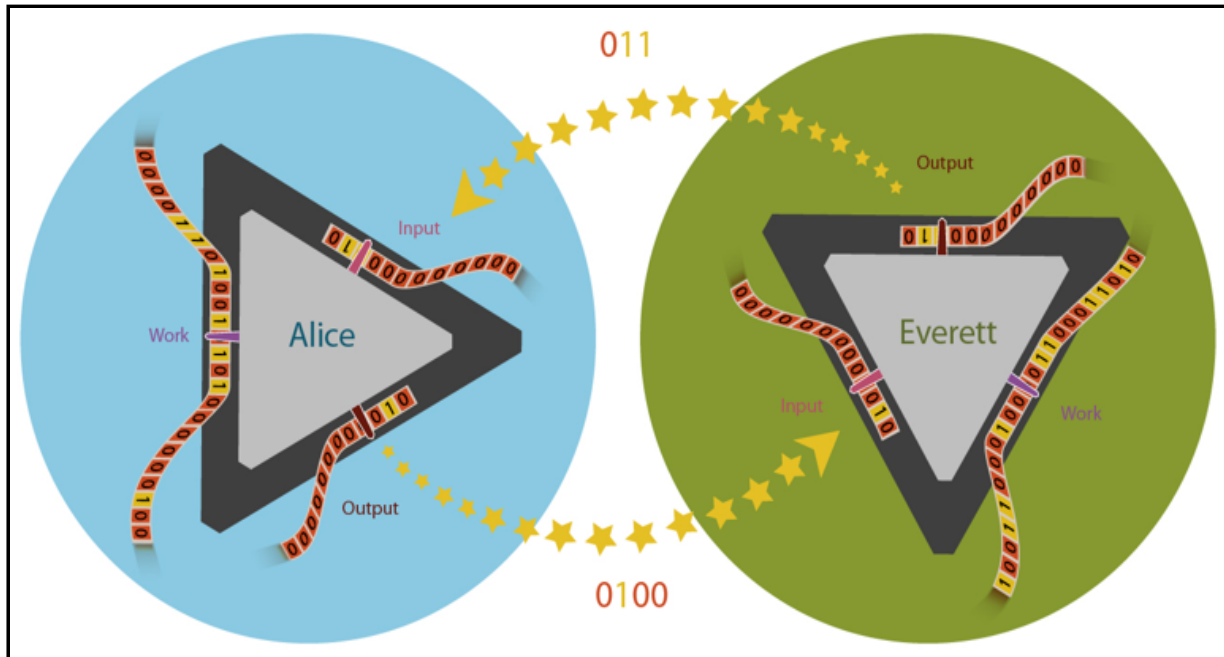
A Solomonoff-inspired AI can most easily be represented as a multi-tape [Turing machine](#) like the one Alex Altair describes in [An Intuitive Explanation of Solomonoff Induction](#). The machine has:

- three tapes, labeled 'input', 'work', and 'output'. Each initially has an infinite strip of 0s written in discrete cells.
- one head per tape, with the input head able to read its cell's digit and move to the right, the output head able to write 0 or 1 to its cell and move to the right, and the work head able to read, write, and move in either direction.
- a program, consisting of a finite, fixed set of transition rules. Each rule says when heads read, write, move, or do nothing, and how to transition to another rule.



A three-tape Turing machine.

We could imagine two such Turing machines communicating with each other. Call them 'Agent' and 'Environment', or 'Alice' and 'Everett'. Alice and Everett take turns acting. After Everett writes a bit to his output tape, that bit magically appears on Alice's input tape; and likewise, when Alice writes to her output tape, it gets copied to Everett's input tape. AI theorists have used this setup, which [Marcus Hutter](#) calls the **cybernetic agent model**, as an extremely simple representation of an agent that can **perceive** its environment (using the input tape), **think** (using the work tape), and **act** (using the output tape).²



A Turing machine model of agent-environment interactions. At first, the machines differ only in their programs. 'Alice' is the agent we want to build, while 'Everett' stands for everything else that's causally relevant to Alice's success.

We can define Alice and Everett's behavior in terms of any bit-producing Turing machines we'd like, including ones that represent probability distributions and do [Bayesian updating](#). Alice might, for example, use her work tape to track four distinct possibilities and update probabilities over them:³

- (a) Everett always outputs 0.
- (b) Everett always outputs 1.
- (c) Everett outputs its input.
- (d) Everett outputs the opposite of its input.

Alice starts with a uniform prior, i.e., 25% probability each. If Alice's first output is 1, and Everett responds with 1, then Alice can store those two facts on her work tape and conditionalize on them both, treating them as though they were certain. This results in 0.5 probability each for (b) and (c), 0 probability for (a) and (d).

We care about an AI's epistemology only because it informs the AI's behavior — on this model, its bit output. If Alice outputs whatever bits maximize her expected chance of receiving 1s as input, then [we can say](#) that Alice **prefers** to perceive 1. In the example I just gave, such a preference predicts that Alice will proceed to output 1 forever. Further exploration is unnecessary, since she knows of no other importantly different hypotheses to test.

Enriching Alice's set of hypotheses for how Everett could act will let Alice win more games against a wider variety of Turing machines. The more programs Alice can pick out and assign a probability to, the more Turing machines Alice will be able to identify and intelligently respond to. If we aren't worried about whether it takes Alice ten minutes or a billion years to compute an update, and Everett will always patiently wait his turn, then we can simply have Alice perform perfect Bayesian updates; if her priors are right, and she translates her beliefs into sensible actions, she'll then be able to [optimally](#) respond to any environmental Turing machine.

For AI researchers following Solomonoff's lead, that's the name of the game: Figure out the program that will let Alice behave optimally while communicating with as wide a range of Turing machines as possible, and you've at least solved the *theoretical* problem of picking out the optimal artificial agent from the space of possible reasoners. The agent/environment model here may look simple, but a number of theorists see it as distilling into its most basic form the task of an AGI.²

Yet a Turing machine, [like a cellular automaton](#), is an [abstract machine](#) — a creature of thought experiments and mathematical proofs. Physical computers can act like abstract computers, in just the same sense that [heaps of apples](#) can behave like [the abstract objects we call 'numbers'](#). But computers and apples are [high-level generalizations](#), imperfectly represented by concise equations.⁴ When we move from our mental models to trying to build an actual AI, we have to pause and ask how well our formalism captures what's going on in reality.

The problem with Alice

'Sensory input' or 'data' is what I call the information Alice conditionalizes on; and 'beliefs' or 'hypotheses' is what I call the resultant probability distribution and representation of possibilities (in Alice's program or work tape). This distinction [seems basic to reasoning](#), so I endorse programming agents to treat them as two clearly distinct types. But in building such agents, we introduce the possibility of **Cartesianism**.

René Descartes held that human minds and brains, although able to causally interact with each other, can each exist in the absence of the other; and, moreover, that the properties of purely material things can never fully explain minds. In his honor, we can call a model or procedure *Cartesian* if it treats the reasoner as a being separated from the physical universe. Such a being can perceive (and perhaps alter) physical processes, but it can't be identified with any such process.⁵

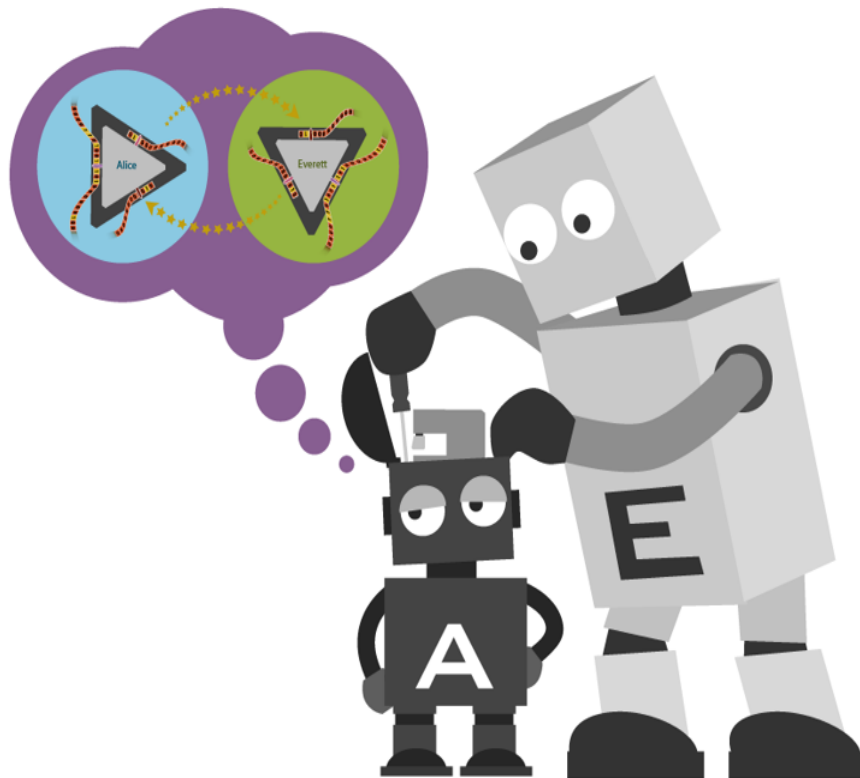
The relevance of Cartesians to AGI work is that we can model them as agents experiencing a strong type distinction between 'mind' and 'matter', and an unshakable belief in the metaphysical independence of those two categories; *because* they're of such different kinds, they can vary independently. So we end up with AI errors that are the opposite of TALE-SPIN's — like an induction procedure that distinguishes gravity's type from embodied characters' types so strongly that it cannot hypothesize that, say, particles underlie or mediate both phenomena.

My claim is that if we plug in 'Alice's sensory data' for 'mind' and 'the stuff Alice hypothesizes as causing the sensory data' for 'matter', then agents that can only model themselves using the cybernetic agent model are Cartesian in the relevant sense.⁶

The model is Cartesian because the agent and its environment *can only interact by communicating*. That is, their only way of affecting each other is by trading bits printed to tapes.

If we build an actual AI that believes it's like Alice, it will believe that the environment can't affect it in ways that aren't immediately detectable, can't edit its source code, and can't force it to halt. But that makes the Alice-Everett system almost nothing like a physical agent embedded in a real environment. Under many circumstances, a real AI's environment will alter it directly. E.g., the AI can fall into a volcano. A volcano doesn't harm the agent by feeding unhelpful bits into its environmental sensors. It harms the agent by destroying it.

A more naturalistic model would say: Alice outputs a bit; Everett reads it; and then Everett does whatever the heck he wants. That might be feeding a new bit into Alice. Or it might be vandalizing Alice's work tape, or smashing Alice flat.



A robotic Everett tampering with an agent that mistakenly assumes Cartesianism. A real-world agent's computational states have physical correlates that can be directly edited by the environment. If the agent can't model such scenarios, its reasoning (and resultant decision-making) will suffer.

A still more naturalistic approach would be to place Alice *inside* of Everett, as a subsystem. In the real world, agents are surrounded by their environments. The two form a cohesive whole, bound by the same physical laws, freely interacting and commingling.

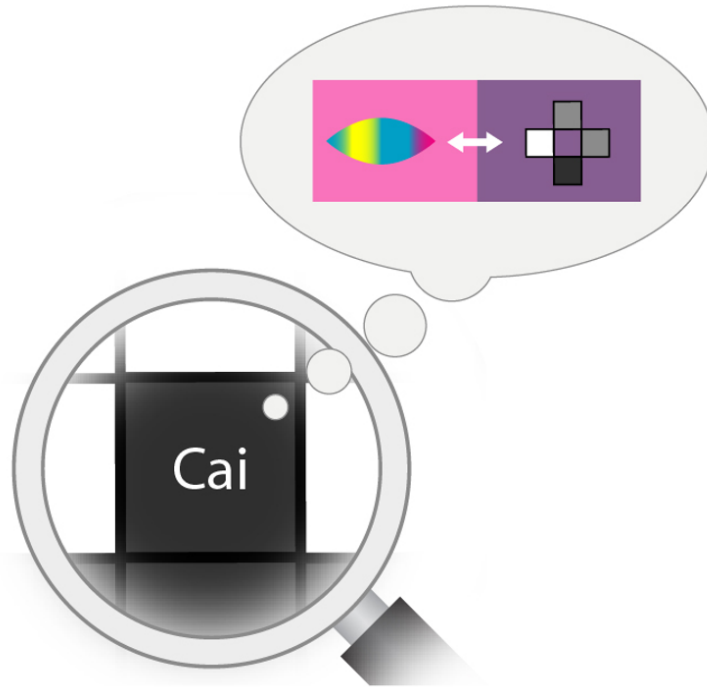
If Alice only worries about whether Everett will output a 0 or 1 to her sensory tape, then no matter how complex an understanding Alice has of Everett's inner workings, Alice will fundamentally misunderstand the situation she's in. Alice won't be able to represent hypotheses about how, for example, a pill might erase her memories or otherwise modify her source code.

Humans, in contrast, can readily imagine a pill that modifies our memories. It seems childishly easy to hypothesize being changed by avenues other than perceived sensory information. The limitations of the cybernetic agent model aren't immediately obvious, because it isn't easy for us to put ourselves in the shoes of agents with alien blind spots.

There *is* an agent-environment distinction, but it's a pragmatic and artificial one. The boundary between the part of the world we call 'agent' and the part we call 'not-agent' (= 'environment') is frequently fuzzy and mutable. If we want to build an agent that's robust across many environments and self-modifications, we can't just design a program that excels at predicting sensory sequences generated by Turing machines. We need an agent that can form accurate beliefs about the actual world it lives in, including accurate beliefs about its own physical underpinnings.

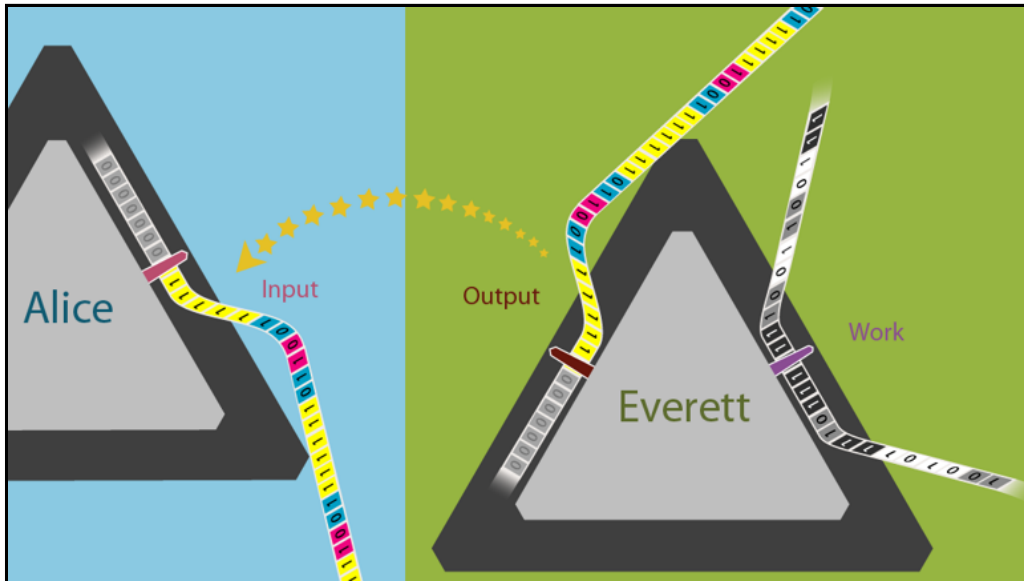
From Cartesianism to naturalism

What would a naturalized self-model, a model of the agent as a process embedded in a lawful universe, look like? As a first attempt, one might point to the pictures of Cai in [Building Phenomenological Bridges](#).



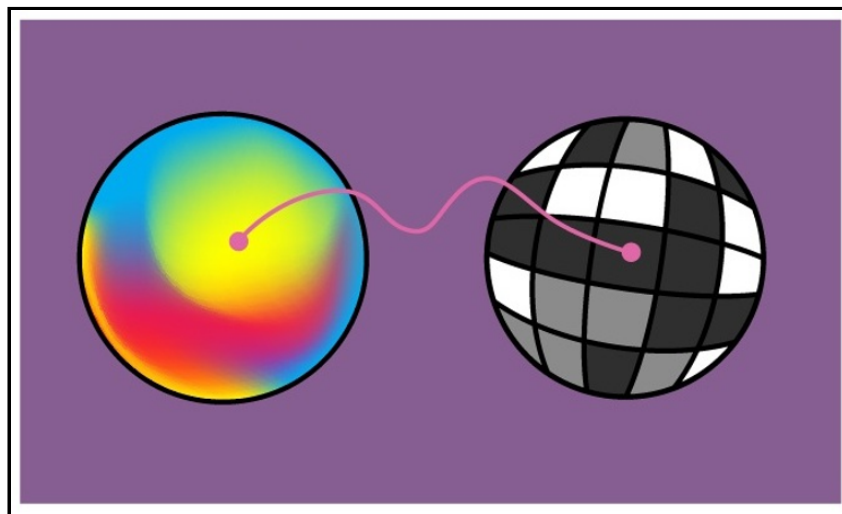
Cai has a simple physical model of itself as a black tile at the center of a cellular automaton grid. Cai's phenomenological bridge hypotheses relate its sensory data to surrounding tiles' states.

But this doesn't yet specify a non-Cartesian agent. To treat Cai as a Cartesian, we could view the tiles surrounding Cai as the work tape of Everett, and the dynamics of Cai's environment as Everett's program. (We can also convert Cai's perceptual experiences into a binary sequence on Alice/Cai's input tape, with a translation like 'cyan = 01, magenta = 10, yellow = 11'.)



Alice/Cai as a cybernetic agent in a Turing machine circuit.

The problem isn't that Cai's world is Turing-computable, of course. It's that if Cai's hypotheses are solely about what sorts of perception-correlated patterns of environmental change can occur, then Cai's models will be Cartesian.

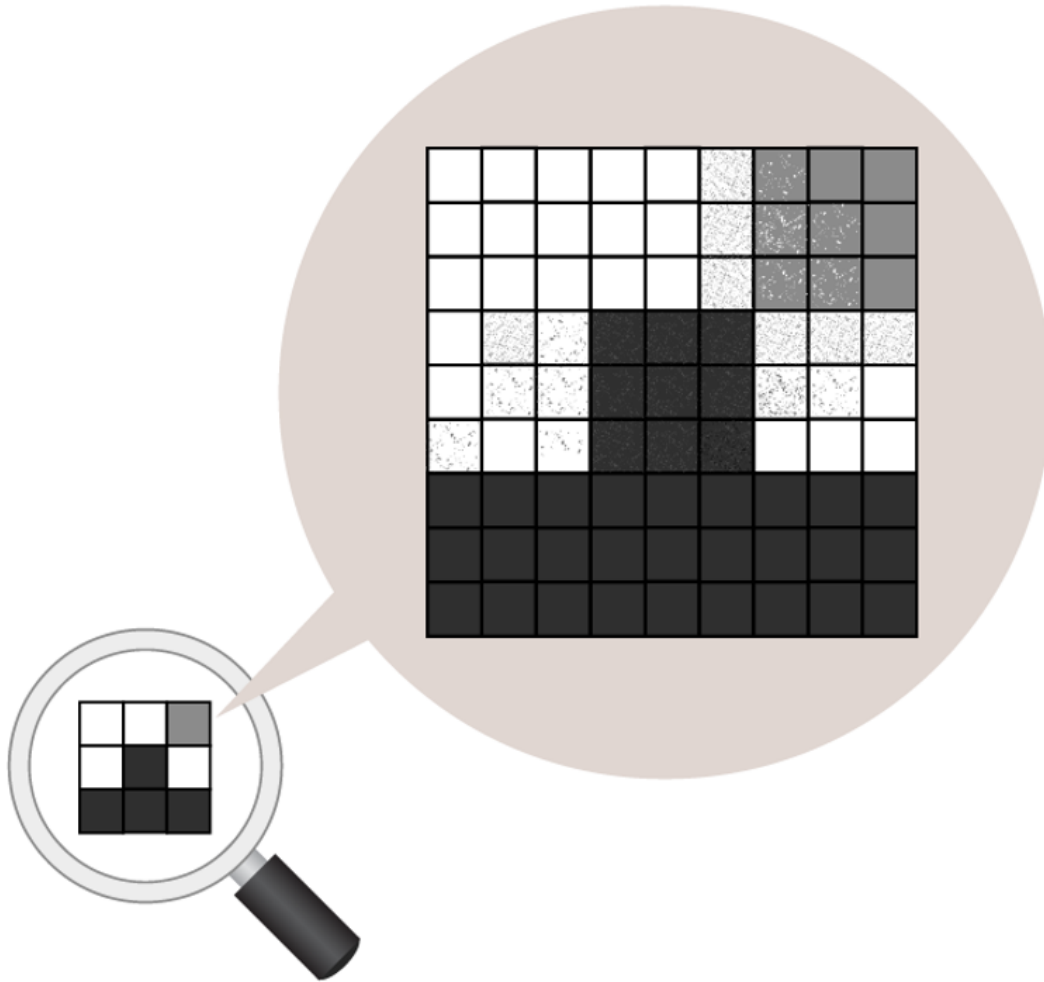


Cai as a Cartesian treats its sensory experiences as though they exist in a separate world.

Cartesian Cai recognizes that its two universes, its sensory experiences and hypothesized environment, can interact. But it thinks they can only do so via a narrow

range of stable pathways. No actual agent's mind-matter connections can be that simple and uniform.

If Cai were a robot in a world resembling its model, it would *itself* be a complex pattern of tiles. To form accurate predictions, it would need to have self-models and bridge hypotheses that were more sophisticated than any I've considered so far. Humans are the same way: No bridge hypothesis explaining the physical conditions for subjective experience will ever fit on a T-shirt.



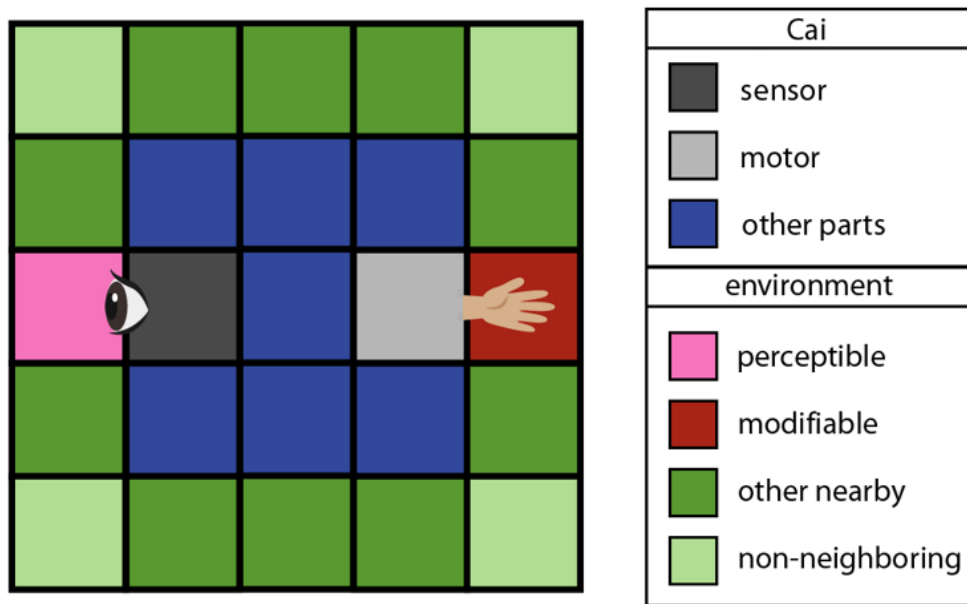
Cai's world divided up into a 9x9 grid. Cai is the central 3x3 grid. Barely visible: Complex computations like Cai's reasoning are possible in this world because they're implemented by even finer tile patterns at smaller scales.

Changing Cai's tiles' states — from black to white, for example — could have a large impact on its computations, analogous to changing a human brain from solid to gaseous. But if an agent's hypotheses are all shaped like the cybernetic agent model, 'my input/output algorithm is replaced by a dust cloud' won't be in the hypothesis space.

If you programmed something to think like Cartesian Cai, it might decide that its sequence of visual experiences will persist even if the tiles forming its brain completely change state. It wouldn't be able to entertain thoughts like 'if Cai performs self-modification #381, Cai will experience its environment as smells rather than colors' or 'if Cai falls into a volcano, Cai gets destroyed'. No pattern of perceived colors is identical to a perceived smell, or to the absence of perception.

To form naturalistic self-models and world-models, Cai needs hypotheses that look less like conversations between independent programs, and more like worlds in which it is a fairly [ordinary](#) subprocess, governed by the same general patterns. It needs to form and privilege physical hypotheses under which it has parts, as well as bridge hypotheses under which those parts correspond in plausible ways to its high-level computational states.

Cai wouldn't need a *complete* self-model in order to recognize general facts about its subsystems. Suppose, for instance, that Cai has just one sensor, on its left side, and a motor on its right side. Cai might recognize that the motor and sensor regions of its body correspond to its introspectible decisions and perceptions, respectively.

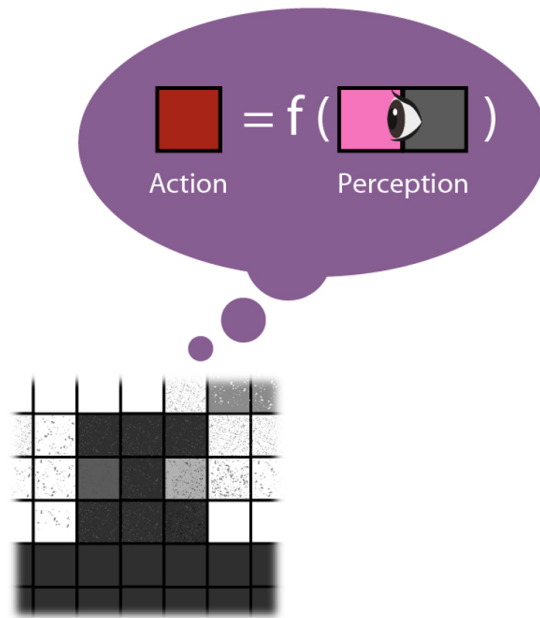


A naturalized agent can recognize that it has physical parts with varying functions. Cai's top and bottom lack sensors and motors altogether, making it clearer that Cai's environment can impact Cai by entirely non-sensory means.

We care about Cai's models because we want to use Cai to modify its environment. For example, we may want Cai to convert as much of its environment as possible into grey tiles. Our interest is then in the algorithm that reliably outputs maximally greyifying actions when handed perceptual data.

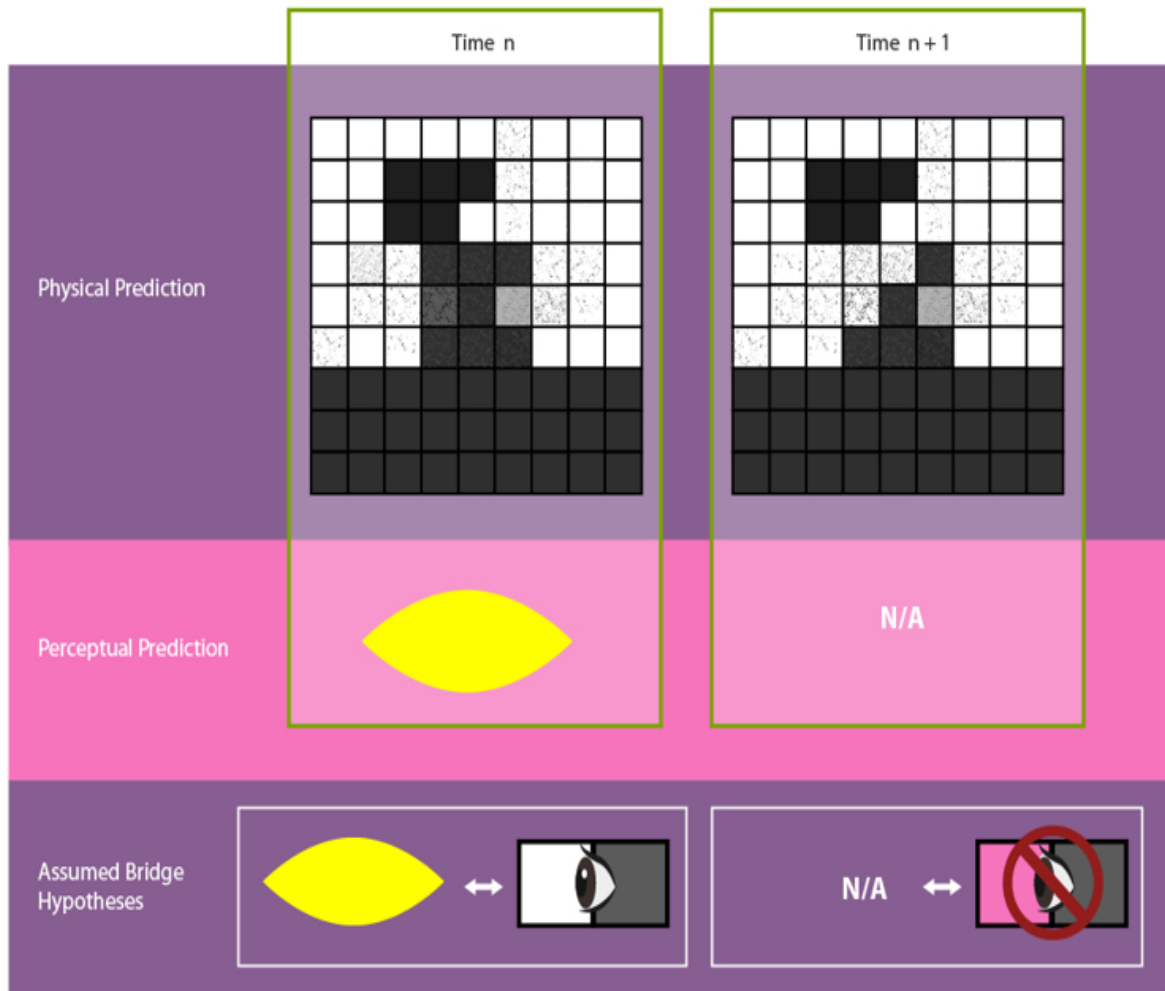
If Cai is able to form sophisticated self-models, then Cai can recognize that it's a grey tile maximizer. Since it wants there to be more grey tiles, it also wants to make sure that it continues to exist, provided it believes that it's better than chance at pursuing its goals.

More specifically, Naturalized Cai can recognize that its actions are some black-box function of its perceptual computations. Since it has a bridge hypothesis linking its perceptions to its middle-left tile, it will then reason that it should *preserve* its sensory hardware. Cai's self-model tells it that if its sensor fails, then its actions will be based on beliefs that are much less correlated with the environment. And its self-model tells it that if its actions are poorly calibrated, then there will be fewer grey tiles in the universe. Which is bad.



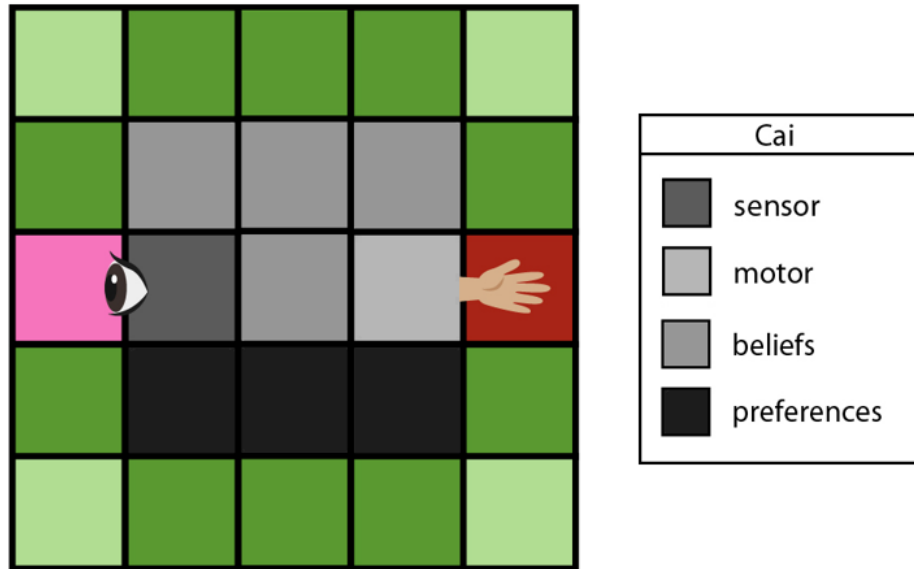
A naturalistic version of Cai can reason intelligently from the knowledge that its actions (motor output) depend on a specific part of its body that's responsible for perception (environmental input).

A physical Cai might need to foresee scenarios like 'an anvil crashes into my head and destroys me', and assign probability mass to them. Bridge hypotheses expressive enough to consider that possibility would not just relate experiences to environmental or hardware states; they would also recognize that the agent's experiences can be absent altogether.



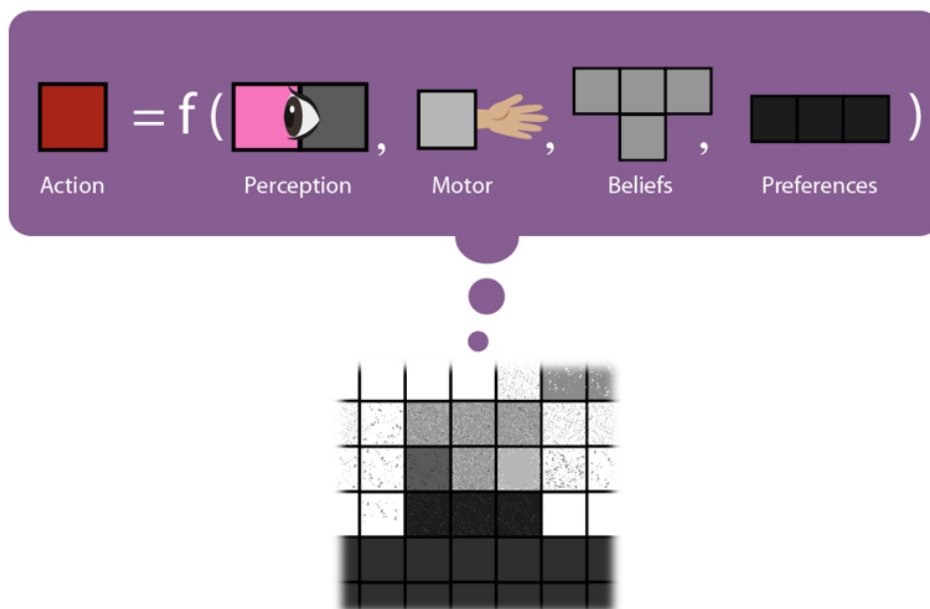
An anvil can destroy Cai's perceptual hardware by crashing into it. A Cartesian might not worry about this eventuality, expecting its experience to persist after its body is smashed. But a naturalized reasoner will form hypotheses like the above, on which its sequence of color experiences suddenly terminates when its sensors are destroyed.

This point generalizes to other ways Cai might self-modify, and to other things Cai might alter about itself. For example, Cai might learn that other portions of its brain correspond to its hypotheses and desires.



Another very simple model of how different physical structures are associated with different computational patterns.

This allows Cai to recognize that its goals depend on the proper functioning of *many* of its hardware components. If Cai believes that its actions depend on its brain's goal unit's working a specific way, then it will avoid taking pills that foreseeably change its goal unit. If Cai's causal model tells it that agents like it stop exhibiting future-steering behaviors when they self-modify to have mad priors, then it won't self-modify to acquire mad priors. And so on.



If Cai's motor fails, its effect on the world can change as a result. The same is true if its hardware is modified in ways that change its thoughts, or its preferences (i.e., the thing linking its conclusions to its motor).

Once Cai recognizes that its brain needs to work in a very specific way for its goals to be achieved, its preferences can take its physical state into account in sensible ways, without our needing to hand-code Cai at the outset to have the right beliefs or preferences over every individual thing that could change in its brain.

Just the opposite is true for Cartesians. Since they can't form hypotheses like 'my tape heads will stop computing digits if I disassemble them', they can only intelligently navigate such risks if they've been hand-coded in advance to avoid perceptual experiences the programmer thought would correlate with such dangers.

In other words, even though all of this is still highly informal, there's already some cause to think that a reasoning pattern like Naturalized Cai can generalize in ways that Cartesians can't. The programmers don't need to know *everything* about Cai's physical state, or anticipate *everything* about what future changes Cai might undergo, if Cai's epistemology allows it to easily form accurate reductive beliefs and behave accordingly. An agent like this might be adaptive and self-correcting in very novel circumstances, leaving more wiggle room for programmers to make human mistakes.

Bridging maps of worlds and maps of minds

Solomonoff-style dualists have alien blind spots that lead them to neglect the possibility that some hardware state is equivalent to some introspected computation '000110'. TALE-SPIN-like AIs, on the other hand, have blind spots that lead to mistakes like trying to figure out the angular momentum of '000110'.

A naturalized agent doesn't try to do away with the data/hypothesis type distinction and acquire a typology as simple as TALE-SPIN's. Rather, it tries to tightly interconnect its types using bridges. Naturalizing induction is about combining the dualist's useful [map/territory distinction](#) with a more sophisticated metaphysical monism than TALE-SPIN exhibits, resulting in a *reductive monist AI*.⁷

Alice's simple fixed bridge axiom, {environmental output 0 ↔ perceptual input 0, environmental output 1 ↔ perceptual input 1}, is inadequate for physically embodied agents. And the problem isn't *just* that Alice lacks other bridge rules and can't weigh evidence for or against each one. Bridge hypotheses are a step in the right direction, but they need to be diverse enough to express a variety of correlations between the agent's sensory experiences and the physical world, and they need a sensible prior. An agent that only considers bridge hypotheses compatible with the cybernetic agent model will falter whenever it and the environment interact in ways that look nothing like exchanging sensory bits.

With the help of an inductive algorithm that uses bridge hypotheses to relate sensory data to a continuous physical universe, we can avoid making our AIs Cartesians. This will make their epistemologies much more secure. It will also make it possible for them to want things to be true about the physical universe, not just about the particular sensory experiences they encounter. Actually writing a program that does all this is an OPFAI. Even formalizing how bridge hypotheses ought to work in principle is an OPFAI.

In my next post, I'll move away from toy models and discuss [AIXI](#), Hutter's optimality definition for cybernetic agents. In asking whether the *best* Cartesian can overcome the difficulties I've described, we'll get a clearer sense of why Solomonoff inductors aren't reflective and reductive enough to predict drastic changes to their sense-input-to-motor-output relation — and why they *can't* be that reflective and reductive — and why this matters.

Notes

¹ Meehan (1977). Colin Allen first introduced me to this story. [Dennett](#) discusses it as well. [↵](#)

² E.g., Durand, Muchnik, Ushakov & Vereshchagin (2004), Epstein & Betke (2011), Legg & Veness (2013), Solomonoff (2011). Hutter (2005) uses the term "cybernetic agent model" to emphasize the parallelism between his Turing machine circuit and [control theory's cybernetic systems](#). [↵](#) [↵](#)

³ One simple representation would be: Program Alice to write to her work tape, on round one, 0010 (standing for 'if I output 0, Everett outputs 0; if I output 1, Everett outputs 0'). Ditto for the other three hypotheses, 0111, 0011, and 0110. Then write the hypothesis' probability in binary (initially 25%, represented '11001') to the right of each, and program Alice to edit this number as she receives new evidence. Since the first and third digit stay the same, we can simplify the hypotheses' encoding to 00, 11, 01, 10. Indeed, if the hypotheses remain the same over time there's no reason to visibly distinguish them in the work tape at all, when we can instead just program Alice to use the left-to-right ordering of the four probabilities to distinguish the hypotheses. [↵](#)

⁴ To the extent our universe [perfectly resembles any mathematical structure](#), it's much more likely to do so [at the level](#) of gluons and mesons than at the level of medium-sized dry goods. The resemblance of apples to natural numbers is much more approximate. Two apples and three apples generally make five apples, but when you start cutting up or pulverizing or genetically altering apples, you may find that other mathematical models do a superior job of predicting the apples' behavior. It seems likely that the only perfectly general and faithful mathematical representation of apples will be some drastically large and unwieldy physics equation.

Ditto for machines. It's sometimes possible to build a physical machine that closely mimics a given Turing machine — but only 'closely', as Turing machines have unboundedly large tapes. And although any halting Turing machine can in principle be simulated with a bounded tape (Cockshott & Michaelson (2007)), nearly all Turing machine programs are too large to even be approximated by any physical process.

All physical machines structurally resemble Turing machines in ways that allow us to draw productive inferences from the one group to the other. See Piccinini's (2011) discussion of the [physical Church-Turing thesis](#). But, for all that, the concrete machine and the abstract one remain distinct. [↵](#)

⁵ Descartes (1641): "[A]lthough I certainly do possess a body with which I am very closely conjoined; nevertheless, because, on the one hand, I have a clear and distinct idea of myself, in as far as I am only a thinking and unextended thing, and as, on the

other hand, I possess a distinct idea of body, in as far as it is only an extended and unthinking thing, it is certain that I (that is, my mind, by which I am what I am) am entirely and truly distinct from my body, and may exist without it."

From this it's clear that Descartes also believed that the mind can exist without the body. This interestingly parallels the [anvil problem](#), which I'll discuss more in my next post. However, I don't build immortality into my definition of 'Cartesianism'. Not all agents that act as though there is a Cartesian barrier between their thoughts and the world think that their experiences are future-eternal. I'm taking care not to conflate Cartesianism with the anvil problem because the formalism I'll discuss next time, AIXI, does face both of them. Though the problems are logically distinct, it's true that a naturalized reasoning method would be much less likely to face the anvil problem. [↵](#)

⁶ This isn't to say that a Solomonoff inductor would need to be conscious in anything like the way humans are conscious. It can be fruitful to point to similarities between the reasoning patterns of humans and unconscious processes. Indeed, this already happens when we speak of unconscious mental processes within humans.

Parting ways with Descartes (cf. Kirk (2012)), many present-day dualists would in fact go even further than reductionists in allowing for structural similarities between conscious and unconscious processes, treating all cognitive or functional mental states as (in theory) realizable without consciousness. E.g., Chalmers (1996): "Although consciousness is a feature of the world that we would not predict from the physical facts, the things we say about consciousness are a garden-variety cognitive phenomenon. Somebody who knew enough about cognitive structure would immediately be able to predict the likelihood of utterances such as 'I feel conscious, in a way that no physical object could be,' or even Descartes's 'Cogito ergo sum.' In principle, some reductive explanation in terms of internal processes should render claims about consciousness no more deeply surprising than any other aspect of behavior." [↵](#)

⁷ And since we happen to live in a world made of physics, the kind of monist we want in practice is a reductive *physicalist* AI. We want a 'physicalist' as opposed to a reductive monist that thinks everything is made of monads, or abstract objects, or morality fluid, or what-have-you. [↵](#)

References

- Chalmers (1996). *The Conscious Mind: In Search of a Fundamental Theory*. Oxford University Press.
- Cockshott & Michaelson (2007). [Are there new models of computation? Reply to Wegner and Eberbach](#). *The Computer Journal*, 50: 232-247.
- Descartes (1641). [Meditations on first philosophy, in which the existence of God and the immortality of the soul are demonstrated](#).
- Durand, Muchnik, Ushakov & Vereshchagin (2004). [Ecological Turing machines](#). *Lecture Notes in Computer Science*, 3142: 457-468.
- Epstein & Betke (2011). [An information-theoretic representation of agent dynamics as set intersections](#). *Lecture Notes in Computer Science*, 6830: 72-81.
- Hutter (2005). [Universal Artificial Intelligence: Sequence Decisions Based on Algorithmic Probability](#). Springer.

- Kirk (2012). [Zombies](#). In Zalta (ed.), *The Stanford Encyclopedia of Philosophy*.
- Legg & Veness (2013). [An approximation of the Universal Intelligence Measure](#). *Lecture Notes in Computer Science*, 7070: 236-249.
- Meehan (1977). [TALE-SPIN, an interactive program that writes stories](#). *Proceedings of the 5th International Joint Conference on Artificial Intelligence*: 91-98.
- Piccinini (2011). [The physical Church-Turing thesis: Modest or bold?](#) *British Journal for the Philosophy of Science*, 62: 733-769.
- Russell & Norvig (2010). [Artificial Intelligence: A Modern Approach](#). Prentice Hall.
- Solomonoff (2011). [Algorithmic probability — its discovery — its properties and application to Strong AI](#). In Zenil (ed.), *Randomness Through Computation: Some Answers, More Questions* (pp. 149-157).

A self-experiment in training "noticing confusion"

I [previously discussed](#) the potential relevance of therapeutic and instructional models of metacognitive training to LW-style rationality skills. As an attempted concrete realization of what this connection could look like, I ran a self-experiment in which I counted instances of noticing confusion. Below I elaborate on the motivation and design of the experiment, then discuss some quantitative results and qualitative reflections.

Background

Self-monitoring as a treatment vehicle in cognitive-behavioral and related therapies can take many forms. In one (to my secondhand understanding), the patient is coached in noticing a physical or mental behavior by identifying examples of the behavior and heuristics for when to watch for it, and by examining the feeling of the behavior itself. This is accompanied by practice of coping strategies. The patient is instructed to count the occurrences of that behavior on their own. This is ideally done with a "wrist counter," which is always available, can be incremented with the press of a single button, and gives both tactile and visual feedback on being pressed.

The patient might, for example, count instances of acting on their own initiative, or of having positive thoughts about themselves. In this case, tying the thought to the specific physical action of pressing the button, as well as watching a "score" go up, helps with the reward circuit for both noticing the thought and the content of the thought.

The patient could also count negative thoughts, engaging in bad habits, inappropriate "should" statements or other "cognitive distortions." At first, so I'm told, the count will go up, as you get better at noticing; then (optimistically) back down over a few weeks, as your symptoms diminish. In this case, it's important not to focus on the fact you're doing something "bad." Instead, try to reward noticing and dispelling the bad thing, or at least to reward noticing that you're focusing on the bad thing rather than rewarding the noticing. (If all else fails, reward noticing that you're focusing on failing to reward noticing that you're focusing on the bad thing rather than rewarding the noticing. That should definitely do it, right?)

This seems doubly useful: not only are you practicing and rewarding the noticing skill, but in tying it to a physical action, you necessarily bring the noticed behavior to your conscious attention, so that you can deal with it deliberately. If you noticed yourself dismissing a compliment, you'd take that opportunity to point out to yourself that the dismissal is mostly evidence of your mental state, and only weakly of the compliment's validity; you'd try to take the compliment at face value.

Design

I chose to implement a version of this for a personal version of [noticing confusion](#). (I also considered noticing a mental flinch, noticing motivated reasoning, flagging beliefs for review, activating curiosity, welcoming bad news, being specific, and noticing others' nonspecificity/asking for examples. I decided to go for now with what would be

most personally useful and the most frequent.) I'm using [this counter widget](#) on my phone's home screen. It's two button presses away at any time, and it shows me a nice big number. I also see it whenever I use my phone, which is good for scaffolding but bad for transfer—on one hand, I get reminders to pay attention to my mental processes, so I'm more likely to be able to practice the noticing skill; on the other, I might be inhibiting my learning to apply the skill without reminders. Since I could just keep using the counter if it helped, I didn't worry too much about this.

The details of the fuzzy introspective rules for whether I get to count something as noticing confusion probably don't matter so much, but the basic idea is this: If I notice an unresolved tension or conflict between things I believe, then I count it. I don't count the related and also-crucial noticing that I simply don't understand something—I have to identify a conflict. (I see "notice when I don't understand something" as Level 0 of this skill. It's also particularly easy to practice: just read something on an unfamiliar subject, and draw a question mark next to any specific thing you don't understand. Ideally, revisit those marks later. Get in the habit of doing this for everything you read.) I don't count confusions in retrospect—if I've already resolved a confusion by the time I bring it to conscious awareness and can press the button, then I don't count it. That was a personally controversial call, but there's another sense in which "noticing and resolving confusion" is simply a mode of thought that operates semi- or sub-consciously. I didn't want to get bogged down in counting those, and this seemed like a simple rule to split the cases.

Thus, some non-examples (still worth noticing in their own right, and often leading to pinpointing a confusion):

- I don't understand that.
- That doesn't seem right.
- That's surprising. Wait, is it? I'm not really sure what I expected, now that I think about it.

And some examples of thoughts that I would count (by the way, these mental processes, like most, are mostly nonverbal for me, so don't take this literally; I'm noticing a feeling like tension in the connections between concepts):

- X conflicts with my understanding of Y because Z.
- Why does that apply in case A but not case B?
- I expected the graph to look like J, but it looks like L.
- I don't think the software usually gives me that message. [Hint: IT DOESN'T. DO NOT PROCEED.]

Before I began, I guessed that I encountered this kind of confusion several times a day, mostly in seminars, papers, textbooks, debugging, simulated data, and experimental data. I suspected that I already consciously notice many of them, but not all, and that increasing the catch rate would markedly improve how much understanding I got out of the above activities and perhaps prevent some expensive mistakes.

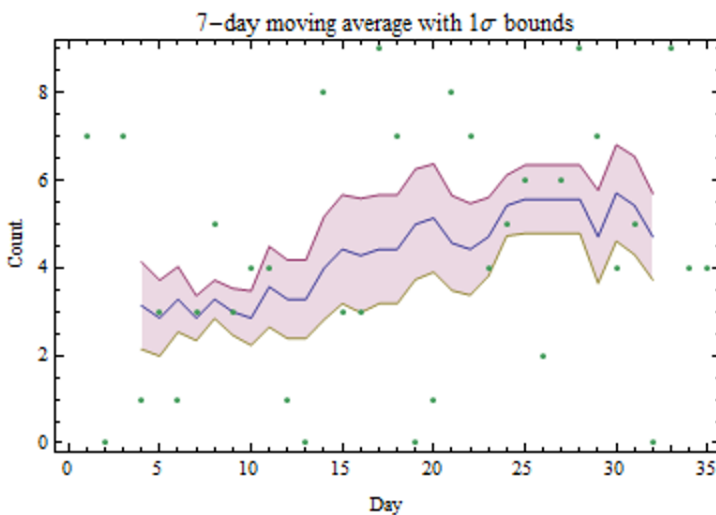
I attempted to keep my confusion-inducing workload constant by working the same number of hours every day. I also distributed my reading of textbooks/papers and my talk attendance to give roughly constant combined time each day, although I'm not sure that those activities had a particularly different density of confusion from my ordinary work. I typically took a couple days a week off of cognitively demanding work, and this pattern is visible in the data, at least at first.

The night before starting the experiment, I ran myself through a couple-hour training exercise on a meaty-looking paper, expressly to pay attention to conflicts in my growing understanding of the result as well as tensions between the content of the paper and my background knowledge, following recommendations of [instructional research on metacognition](#). This was already pretty satisfying and left me feeling good about my self-experiment. The challenge would be to see whether I could improve at spotting and pinning down my nagging doubts, and whether I could take this watchfulness beyond the more-studied domain of self-monitoring while reading. Both of these things seemed to happen.

Results

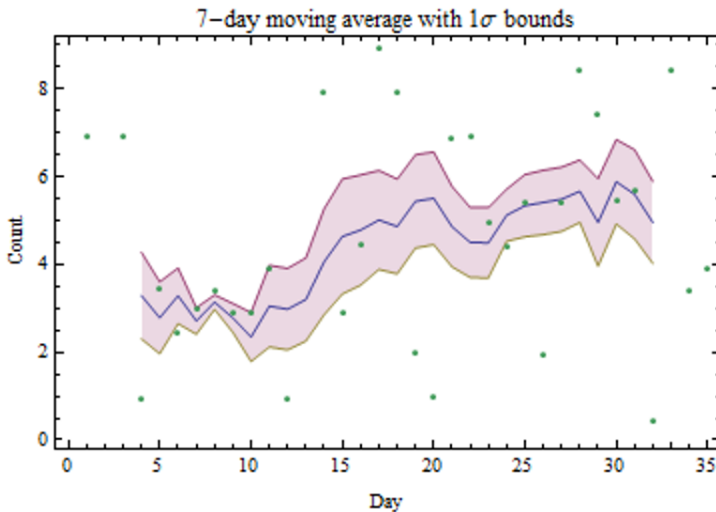
The quantitative results are promising, but not especially informative. There's only so much I can say with a month's worth of data points in such a non-rigorous self-experiment. As it turned out, my guess of "several times a day" was pretty good—for a good day, full of demanding work, which was what came to mind when guessing. In truth, there's a lot more variation between days, which didn't disappear as I got better at pressing the button: there's a standard deviation of 2.85 counts for week 1, 2.81 counts for week 5, and 2.81 counts for all days.

Here's what the data looks like, with a moving weekly average (thus accounting for the weekend effect) and moving weekly 1σ bounds (e.g. $\pm 2.85/\sqrt{7}$ for the first week):



By week 3, the weekly count has gone up by a standard deviation, and it stays there or higher for weeks 4 and 5. Again, I don't want to lean too hard on these numbers—I wasn't rigorously consistent about the amount and nature of my daily work or the rules for counting. Weeks 1 and 2 might have been bad weeks, so that the increase doesn't represent a real improvement; there's also room for my desire to have a better-looking LW post to have increased the counts. And there's a little ambiguity about what I'm measuring: perhaps the increase in counting comes only from remembering to press the button, and there are plenty of other times when I notice confusions and consciously address them without identifying them as button-pressing candidates. My guess is that this isn't the case—the increase seemed to come in the form of things I barely didn't miss.

If I naïvely say that Week 1 establishes a true distribution for averaged weekly counts, then being more than 1σ above the mean for three weeks would have a probability of about $p = (0.16)^3 = 0.0041$ if that true count distribution remained constant. I'm not going to do any more sophisticated analysis than that, since I don't think the data really supports it. See [this detailed comment](#) by VincentYu. There's also a barely-significant relationship with the previous night's sleep duration ($p = 0.043$, +1 count per hour of sleep). If I adjust for this, the appearance of improvement still holds:



So sleep perhaps accounts for a small amount of random variation, and not the overall shift.

Finally, some qualitative reflections:

- I feel like I gained more solid understanding of things and solved a lot of problems faster as a direct consequence of focusing on my feelings of confusion. Given that the counts went up, I suspect that things were understood and problems solved that wouldn't have been at all had I not been doing this.
- I occasionally found myself mentally searching for potential contradictions when I encountered new information. This is called either "cheating" or "[mission accomplished](#)."
- You might have noticed that I didn't say anything about what to do after bringing confusion to conscious attention. It turns out that curiosity hijacks my brain once I pinpoint an apparent contradiction, far more so than when I simply notice that I don't understand something. I do what I can to encourage that process.
- I'm underconfident in the significance of my confusions. When I have a vague sense that something's wrong, I'm often tempted to dismiss it as a weird fact about my brain, an uninteresting exception to a weak generalization, or something that would be resolved if I just did the math. But never in the course of this experiment did I count something that turned out to be unimportant.
- At first, I didn't seem to exercise this skill on days where I wasn't doing cognitively demanding work, or when most of my work was not in an academic context (typically weekends). Over time, I began doing so more, although still less than on demanding academic days. This shows up in the disappearance of weekend dips in the data with time, and I think it's a good sign concerning transfer.
- A few weeks in, I began spontaneously recalling past instances of confusion, apparently on the strength of their connections to the feeling of being confused.

Some of these I'd never resolved—I remembered a professor telling me years ago that the [filamentary organization of galaxies](#) had never been observed. That had sounded obviously wrong to me, but I'd just shrugged and moved on. The contradiction lay dormant in my mind until a week ago, when I took a minute to figure out that she was almost definitely talking about direct observation of intergalactic filaments. Depending on what counts (intergalactic? intracluster? visible/dark matter?), that didn't happen until [2012](#) or (provisionally) [very recently](#). (That's entirely irrelevant to my current work, but I thought it was interesting.)

- I had one lucid dream during the past 5 weeks, and it explicitly began with noticing confusion in this sense. But that's not very meaningful, since I ordinarily expect around one lucid dream in 8 weeks. It's just as plausible to me that I began lucid-dreaming and then my brain made the connection to this experiment.

Conclusion

The quantitative results are promising, but for me, the qualitative lessons are more important—particularly my underconfidence and the possibility of using contradiction to [fuel curiosity](#). I'll keep counting confusions like this for a while, but I'm [not going to worry much](#) about experimental validity. Similarly, it doesn't matter a whole lot to me whether the apparent gains rely on using the counter, since it costs me basically nothing to continue using it. I suppose that one could look into that by taking a break from counting and resuming it after a few months, but that's honestly not my priority.

This is a really easy thing to try, and I'd like to encourage others to build on the simple attempt I've presented here.

White Lies

Background: As can be seen from some of the comments on this post, many people in the LessWrong community take an extreme stance on lying. A few days before I posted this, I was at a meetup where we played the game [Resistance](#), and one guy announced before the game began that he had a policy of never lying *even when playing games like that*. It's such members of the LessWrong community that this post was written for. I'm *not* trying to encourage basically honest people with the normal view of white lies that they need to give up being basically honest.

Mr. Potter, you sometimes make a game of lying with truths, playing with words to conceal your meanings in plain sight. I, too, have been known to find that amusing. But if I so much as *tell* you what I hope we shall do this day, Mr. Potter, you will *lie* about it. You will lie straight out, without hesitation, without wordplay or hints, to anyone who asks about it, be they foe or closest friend. You will lie to Malfoy, to Granger, and to McGonagall. You will speak, always and without hesitation, in exactly the fashion you would speak if you knew nothing, with no concern for your honor. That also is how it must be.

- Rational!Quirrell, [Harry Potter and the Methods of Rationality](#)

This post isn't about HMPOR, so I won't comment on the fictional situation the quote comes from. But in many real-world situations, it's excellent advice.

If you're a gay teenager with homophobic parents, and there's a real chance they'd throw you out on the street if they found out you were gay, you should probably lie to them about it. Even in college, if you're still financially dependent on them, I think it's okay to lie. The minute you're no longer financially dependent on them, you should absolutely come out for your sake and the sake of the world. But it's OK to lie if you need to to keep your education on-track.

Oh, maybe you could get away with just shutting up and hoping the topic doesn't come up. When asked about dating, you could try to evade while being technically truthful: "There just aren't any girls at my school I really like." "What about ____? Why don't you ask her out?" "We're just friends." That might work. But when asked directly "are you gay?" and the wrong answer could seriously screw-up your life, I wouldn't bet too much on your ability to "lie with truths," as Quirrell would say.

I start with this example because the discussions I've seen on the ethics of lying on LessWrong (and everywhere, actually) tend to focus on the extreme cases: the now-cliché "Nazis at the door" example, or even discussion of whether you'd lie with the world at stake. The "teen with homophobic parents" case, on the other hand, might have actually happened to someone you know. But even this case is extreme compared to most of the lies people tell on a regular basis.

Widely-cited statistics claim that [the average person lies once per day](#). I recently saw a new study (that I can't find at the moment) that disputed this, and claimed most people lie rather less often than that, but it still found most people lie fairly often. These lies are mostly "white lies" to, say, spare others' feelings. Most people have no qualms about those kind of lies. So why do discussions of the ethics of lying so often

focus on the extreme cases, as if those were the only ones where lying is maybe possibly morally permissible?

At LessWrong there've been discussions of [several different views all described as "radical honesty."](#) No one I know of, though, has advocated Radical Honesty as defined by psychotherapist Brad Blanton, which (among other things) demands that people share every negative thought they have about other people. (If you haven't, I recommend reading [A. J. Jacobs on Blanton's movement.](#)) While I'm glad no one here is thinks Blanton's version of radical honesty is a good idea, a strict no-lies policy can sometimes have effects that are just as disastrous.

A few years ago, for example, when I went to see the play my girlfriend had done stage crew for, and she asked what I thought of it. She wasn't satisfied with my initial noncommittal answers, so she pressed for more. Not in a "trying to start a fight" way; I just wasn't doing a good job of being evasive. I eventually gave in and explained why I thought the acting had sucked, which did not make her happy. I think incidents like that must have contributed to our breaking up shortly thereafter. The breakup was a good thing for other reasons, but I still regret not lying to her about what I thought of the play.

Yes, there are probably things I could've said in that situation that would have been not-lies and also would have avoided upsetting her. Sam Harris, in his book [Lying](#), spends a lot of arguing against lying in that way: he takes situations where most people would be tempted to tell a white lie, and suggesting ways around it. But for that to work, you need to be good at striking the delicate balance between saying too little and saying too much, and framing hard truths diplomatically. Are people who lie because they lack that skill really less moral than people who are able to avoid lying because they have it?

Notice the signaling issue here: Sam Harris' book is a subtle brag that *he* has the skills to tell people the truth without too much backlash. This is especially true when Harris gives examples from his own life, like the time he told a friend "No one would ever call you 'fat,' but I think you could probably lose twenty-five pounds." and his friend went and did it rather than getting angry. Conspicuous honesty also overlaps with conspicuous outrage, the signaling move that announces (as Steven Pinker put it) "I'm so talented, wealthy, popular, or well-connected that I can afford to offend you."

If you're highly averse to lying, I'm not going to spend a lot of time trying to convince you to tell white lies more often. But I will implore you to do one thing: *accept other people's right to lie to you*. About some topics, anyway. Accept that some things are none of your business, and sometimes that includes the fact that there's something which is none of your business.

Or: suppose you ask someone for something, they say "no," and you suspect their reason for saying "no" is a lie. When that happens, don't get mad or press them for the real reason. Among other things, they may be operating on the assumptions of [guess culture](#), where your request means you strongly expected a "yes" and you might not think their real reason for saying "no" was good enough. Maybe *you* know you'd take an honest refusal well (even if it's "I don't want to and don't think I owe you that"), but they don't necessarily know that. And maybe you *think* you'd take an honest refusal well, but what if you're lying to yourself?

If it helps to be more concrete: Some men will react badly to being turned down for a date. Some women too, but probably more men, so I'll make this gendered. And also

because dealing with someone who won't take "no" for an answer is a scarier experience with the asker is a man and the person saying "no" is a woman. So I sympathize with women who give made-up reasons for saying "no" to dates, to make saying "no" easier.

Is it always the wisest decision? Probably not. But sometimes, I suspect, it is. And I'd advise men to accept that women doing that is OK. Not only that, I wouldn't want to be part of a community with lots of men who didn't get things like that. That's the kind of thing I have in mind when I say to respect other people's right to lie to you.

All this needs the disclaimer that some domains should be lie-free zones. I value the truth and despise those who would corrupt intellectual discourse with lies. Or, as Eliezer [once put it](#):

We believe that scientists should always tell the whole truth *about science*. It's one thing to lie in everyday life, lie to your boss, lie to the police, lie to your lover; but whoever lies *in a journal article* is guilty of utter heresy and will be excommunicated.

I worry this post will be dismissed as trivial. I simultaneously worry that, even with the above disclaimer, someone is going to respond, "Chris admits to thinking lying is often okay, now we can't trust anything he says!" If you're thinking of saying that, that's your problem, not mine. Most people will lie to you occasionally, and if you get upset about it you're setting yourself up for a lot of unhappiness. And refusing to trust someone who lies *sometimes* isn't actually very rational; all but the most prolific liars don't lie anything like half the time, so what they say is still significant evidence, most of the time. (Maybe such declarations-of-refusal-to-trust shouldn't be taken as *arguments* so much as *threats* meant to coerce more honesty than most people feel bound to give.)

On the other hand, if we ever meet in person, I hope you realize I might lie to you. Failure to realize a statement could be a white lie can create some *terribly* awkward situations.

Edits: Changed title, added background, clarified the section on accepting other people's right to lie to you (partly cutting and pasting from [this comment](#)).

Edit round 2: Added link to paper supporting claim that the average person lies once per day.

Beware Trivial Fears

Does the surveillance state affect us? It has affected me, and I didn't realize that it was affecting me until recently. I give a few examples of how it has affected me:

1. I was once engaged in a discussion on Facebook about Obama's foreign policy. Around that time, I was going to apply for a US visa. I stopped the discussion early. Semi-consciously, I was worried that what I was writing would be checked by US visa officials and would lead to my visa being denied.
2. I was once really interested in reading up on the Unabomber and his manifesto, because somebody mentioned that he had some interesting ideas, and though fundamentally misguided, he might have been onto something. I didn't explore much because I was worried---again semi-consciously---that my traffic history would be logged on some NSA computer somewhere, and that I'd pattern match to the Unabomber (I'm a physics grad student, the Unabomber was a mathematician).
3. I didn't visit [Silk Road](#) as I was worried that my visits would be traced, even though I had no plans of buying anything.
4. Just generally, I try to not search for some really weird stuff that I want to search for (I'm a curious guy!).
5. I was almost not going to write this post.

And these are just the ones that I became conscious of. I wonder how many more have slipped under the radar.

Yes, I know these fears are silly. In fact, writing them out makes them feel even more silly. But they still affected my behavior. Now, I may be atypical. But I'm sure I'm not *that* atypical. I'm sure many, many people refrain from visiting and exploring parts of the Internet and writing things on different forums and blogs because of the fear of being recorded and the data being used against them. Especially susceptible to this fear are immigrants.

In [Beware Trivial Inconveniences](#), Yvain points out that the [Great Firewall of China](#) is very easy to bypass but the vast majority of Chinese people don't bypass it because it's a trivial inconvenience.

I would like to introduce the analogous and very related concept of a **trivial fear**: fear of low probability events that affects behavior in a major way, especially over a large population. Much more insidiously, the people experiencing these fears don't even realize they're experiencing it: because the fear is of small magnitude, it can be rationalized away easily.

In this particular case, the fear acts in a way so as to restrict the desire for information and free speech.

In a recent conversation, a friend mentioned that calling the modern surveillance state 'Orwellian' is hyperbole. Maybe so. I don't know if the surveillance state is a Good Thing or a Bad Thing. I'm not an economist or a political scientist or a moral philosopher. I simply want to point out that the main lesson from *1984* is not the exact details of the dystopia, but the fact that the people living in the dystopia weren't even remotely aware that they were living in one.

Book Review: Linear Algebra Done Right (MIRI course list)

I'm reviewing the books in the MIRI course list.

It's been a while since I did a book review. The last book I reviewed was [Computation and Logic](#), which I read in November. After that, I spent a few weeks brushing up on specific topics in preparation for my first MIRI math workshop. I read about half of [The Logic of Provability](#) and studied a little topology. I also worked my way through some [relevant papers](#).

After the workshop, I took some time off around the holidays and [wrote a bit about my experience](#). I'm finally back into Study Mode. This week I finished [Linear Algebra Done Right](#), by Sheldon Axler.

Sheldon Axler

LINEAR ALGEBRA DONE RIGHT

Second Edition



Springer

I quite enjoyed the book. Linear algebra has far-reaching impact, and while I learned it in college, I was mostly just [memorizing passwords](#). There are a few important concepts in linear algebra that seem prone to poor explanations. *Linear Algebra Done*

Right derives these concepts intuitively. My understanding of Linear Algebra improved drastically as I read this book.

Below, I'll review the contents of the text before giving a more detailed overview of the book as a whole.

My Background

When reading a review of a textbook, it's important to know the reviewer's background. I studied Linear Algebra briefly in college, not in a Linear Algebra course, but as a subsection of a Discrete Mathematics course. I knew about vector spaces, I understood what 'linearity' entailed, and I was acquainted with the standard tools of linear algebra. I had a vague idea that matrices encoded multiple linear equations at the same time. I could solve problems mechanically using the usual tools, but I didn't fully understand them.

The book was an easy read (as I already knew the answers), but was still quite informative (as I didn't yet understand them).

Contents

1. [Vector Spaces](#)
2. [Finite Dimensional Vector Spaces](#)
3. [Linear Maps](#)
4. [Polynomials](#)
5. [Eigenvectors and Eigenvalues](#)
6. [Inner-Product Spaces](#)
7. [Operators on Inner-Product Spaces](#)
8. [Operators on Complex Vector Spaces](#)
9. [Operators on Real Vector Spaces](#)
10. [Trace and Determinant](#)

Vector Spaces

This chapter briefly explains complex numbers, vectors (providing both geometric and non-geometric interpretations), and vector spaces. It's short, and it's a nice review. If you don't know what vector spaces are, this is as good a way to learn as any. Even if you are already familiar with vector spaces, this chapter is worth a skim to learn the specific notation used in this particular book.

Finite Dimensional Vector Spaces

This chapter covers span, linear independence, bases, and dimension. There are some interesting results here if you're new to the field (for example, any two vectors which are not scalar multiples of each other, no matter how close they are to each other, span an entire plane). Mostly, though, this section is about generalizing the basic geometric intuition for vector spaces into more-than-three dimensions.

Again, this chapter is probably a good introduction for people who have never seen Linear Algebra before, but it doesn't cover much that is counter-intuitive.

Linear Maps

This is the first chapter that started explaining things in ways I hadn't heard before. It covers linear maps from one vector space to another. It spends some time discussing null spaces (the vector spaces which a linear map maps to zero) and ranges (the subspace of the target space that the map maps the source space onto). These are fairly simple concepts that turn out to be far more useful than I anticipated when it comes to building up intuition about linear maps.

Next, matrices are explored. Given a linear map and a basis for each of the source and the target spaces, we can completely describe the linear map by seeing what it does to each basis vector. No matter how complicated the map is, no matter what gymnastics it is doing, linearity guarantees that we can always fully capture the behavior by pre-computing only what it does to the basis vectors. This pre-computation, ignoring the 'actual function' and seeing what it does to two specific bases, is precisely a matrix of the linear map (with respect to those bases).

That was a neat realization. The chapter then covers surjectivity, injectivity, and invertability, none of which are particularly surprising.

Polynomials

This chapter is spent exploring polynomials in both real and complex fields. This was a nice refresher, as polynomials become quite important (unexpectedly so) later on in the book.

Eigenvectors and Eigenvalues

The book now turns to operators (linear maps from a vector space onto itself), and starts analyzing their structure. It introduces invariant subspaces (spaces which, under the operator, map to some scalar multiple of themselves) and more specifically eigenvectors (one-dimensional invariant subspaces) and eigenvalues (the corresponding scalar multiples).

Interestingly, if T is an operator with eigenvalue λ then the null space of $(T - \lambda I)$ includes the corresponding eigenvector. This seemingly simple fact turns out to have far-reaching implications. This leads to our first taste of applying polynomials to operators, which is a portent of things to come.

The chapter then discusses some "well-behaved" operator/basis combinations and methods for finding bases under which operators are well behaved. This leads to upper-triangular and diagonal matrices.

The chapter concludes with a discussion of invariant subspaces on real vector spaces. This introduces some technical difficulties that arise in real spaces (namely stemming from the fact that not every polynomial has real roots). The methods for dealing with real spaces introduced here are repeated frequently in different contexts for the remainder of the book.

Inner-Product Spaces

This chapter introduces inner products, which essentially allows us to start measuring the 'size' of vectors (for varying definitions of 'size'). This leads to a discussion of orthonormal bases (orthogonal bases where each basis vector has norm 1). Some of the very nice properties offered by orthogonal / orthonormal bases are then explored.

The chapter moves on to discuss linear functionals (linear maps onto a one-dimensional space) and adjoints. The adjoint of an operator is analogous to the complex conjugate of a complex number. Adjoint operators aren't very well motivated in this chapter, but they allow us to discuss self-adjoint operators in the following chapter.

Operators on Inner-Product Spaces

This chapter opens with self-adjoint operators, which are essentially operators T such that the inner product of Tv with w is the same as the inner product of v with Tw . To continue with the analogy above, self-adjoint operators (which are "equal to their conjugate") are analogous to real numbers.

Self-adjoint operators are generalized to normal operators, which merely commute with their adjoints. This sets us up for the spectral theorem, which allows us to prove some nice properties exclusive to normal operators on inner-product spaces. (Additional work is required for real spaces, as expected.)

The chapter moves on to positive operators, which should really be called non-negative operators, which essentially don't turn any vectors around (with respect to the inner product) — specifically these are operators T such that $\langle Tv, v \rangle \geq 0$. This allows us to start talking about square roots of operators, which are always positive.

This is followed by isometries, which are operators that preserve norms ($\text{norm}(Sv) = \text{norm}(v)$).

With these two concepts in hand, the chapter shows that every operator on an inner product space can be decomposed into an isometry and a positive operator (which is the square of the square root of the original operator). Intuitively, this shows that every operator on an inner product space can be thought of as one positive operation (no turning vectors around) followed by one isometric operation (no changing lengths). This is called a polar decomposition, for obvious reasons.

The chapter concludes by showing that the polar decomposition leads to a singular value decomposition, which essentially states that every operator on an inner product space has a diagonal matrix with respect to two orthonormal bases. This is pretty powerful.

Operators on Complex Vector Spaces

The chapter begins by introducing generalized eigenvectors. Essentially, not every operator has as many eigenvectors as it has dimensions. However, in such operators, there will be some eigenvalues that are "repeated": $(T - \lambda)^2$ maps additional subspaces to zero (that $T - \lambda$ did not). More generally, we can assign a 'multiplicity' to each eigenvalue counting the maximum dimension of the null space of $(T - \lambda)^j$. Counting multiplicity, each operator on space V has as many eigenvalues as the dimension of V .

This allows us to characterize every operator via a unique polynomial $p(T) = (T - \lambda_1) \dots (T - \lambda_n)$ such that $p(T) = 0$. In other words, we can think of T as a root of this

polynomial, which is called the characteristic polynomial of T . This polynomial can tell us much about an operator (as it describes both the eigenvalues of the operator and their multiplicities).

We can also find a minimal polynomial for T , which is the monic polynomial q of minimal degree such that $q(T) = 0$. If the degree of q is equal to the dimension of the space that T operates on, then the minimal polynomial is the same as the characteristic polynomial. The minimal polynomial of an operator is also useful for analyzing the operator's structure.

The chapter concludes by introducing Jordan form, a particularly 'nice' version of upper-triangular form available to 'nilpotent' operators (operators N such that $N^p = 0$ for some p).

Operators on Real Vector Spaces

This chapter derives characteristic polynomials for operators on real spaces. It's essentially a repeat of the corresponding section of chapter 8, but with extra machinery to deal with real vector spaces. Similar (although predictably weaker) results are achieved.

Trace and Determinant

This chapter explains trace (the sum of all eigenvalues, counting multiplicity) and determinant (the product of all eigenvalues, counting multiplicity), which you may also recognize as the second and final coefficients of the characteristic polynomial, respectively. These values can tell you a fair bit about an operator (as they're tightly related to the eigenvalues), and it turns out you can calculate them even when you can't figure out the individual eigenvalues precisely.

The book then fleshes out methods for calculating trace and determinant from arbitrary matrices. It tries to motivate the standard method of calculating determinants from arbitrary matrices, but this involves a number of arbitrary leaps and feels very much like an accident of history. After this explanation, I understand better what a determinant is, and it is no longer surprising to me that my university courses had trouble motivating them.

The book concludes by exhibiting some situations where knowing only the determinant of an operator is actually useful.

Who should read this?

This book did a far better job of introducing the main concepts of linear algebra to me than did my Discrete Mathematics course. I came away with a vastly improved intuition for why the standard tools of linear algebra actually work.

I can personally attest that *Linear Algebra Done Right* is a great way to un-memorize passwords and build up that intuition. If you know how to compute a determinant but you have no idea what it *means*, then I recommend giving this book a shot.

I imagine that *Linear Algebra Done Right* would also be a good introduction for someone who hasn't done any linear algebra at all.

What should I read?

Chapters 1, 2, and 4 are pretty simple and probably review. They are well-written, short, and at least worth a skim. I recommend reading them unless you feel you know the subjects very well already.

Chapters 3, 5, and 6 were very helpful, and really helped me build up my intuition for linear algebra. If you're trying to build an intuition for linear algebra, read these three chapters closely and do the exercises (chapter 5 especially, it's probably the most important chapter).

Chapters 7 and 8 also introduced some very helpful concepts, though things got fairly technical. They build up some good intuition, but they also require some grinding. I recommend reading these chapters and understanding all the concepts therein, but they're pretty heavy on the exercises, which you can probably skip without much trouble. Chapter 8 is where the book deviates most from the "standard" way of teaching linear algebra, and might be worth a close read for that reason alone.

Chapter 9 is largely just a repeat of chapter 8 but on real spaces (which introduces some additional complexity — har har): none of it is surprising, most of it is mechanical, and I recommend skimming it and skipping the exercises unless you really want the practice.

Chapter 10 is dedicated to explaining some specific memorized passwords, and does a decent job of it. The sections about trace and determinants of an operator are very useful. The corresponding sections about traces and determinants of matrices are eye-opening from the perspective of someone coming from the "standard" classes, and are probably somewhat surprising from a newcomer's perspective as well. (You can torture an impressive amount of information out of a matrix.) However, a good half of chapter 10 is an artifact of the "standard" way of teaching linear algebra: in my opinion, it's sufficient to understand what trace and determinants are and know that they *can* be calculated from matrices in general, without worrying too much about the specific incantations. I'd skim most of this chapter and skip the exercises.

Closing Notes

This book is well-written. It has minimal typos and isn't afraid to spend a few paragraphs building intuition, but it largely gets to the point and doesn't waste your time. It's a fairly quick read, it's well-paced, and it never feels too difficult.

I feel this book deserves its place on the course list. Linear algebra is prevalent throughout mathematics, and *Linear Algebra Done Right* provides a solid overview.