Adam Smith: Dumb Channels

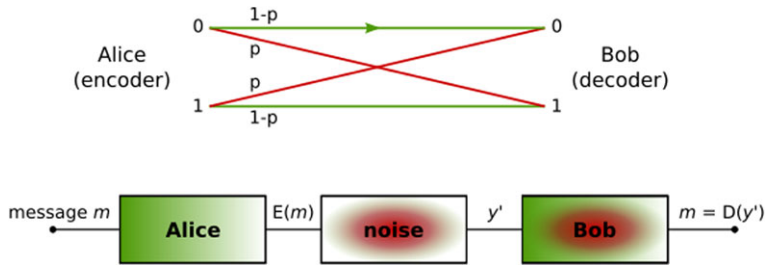
14

Adam Smith is a complexity theorist who works in the area of coding theory, and especially the interface between information theory and cryptography.

We describe a result of his with Venkatesan Guruswami. They both are masters of using cryptography and information theory, and they use these tools to prove a pretty theorem in information theory. This result appeared at the FOCS 2010 conference. Guruswami gave a detailed talk on it at the 2010 Eastern Great Lakes Theory of Computation Workshop, which was held on the University at Buffalo campus. The workshop was sponsored by NSF and the Transborder Regional University Network.

The central idea is one that I find extremely appealing. In their paper they study, among other things, the consequences of limiting the power of the communication channel. This is not a new idea, information theory has long studied binary symmetric channels. These are channels that have a probability p of "flipping a bit," and each bit is flipped with this probability and **independently**. The reason these channels are so popular is two-fold: (i) they are the simplest non-trivial channels and usually are the easiest to understand, and (ii) they sometimes can be used to solve even more complex situations.

Our figure taken from Wikipedia shows Alice sending bits to Bob over such a channel.



R.J. Lipton, K.W. Regan, *People, Problems, and Proofs*,
DOI 10.1007/978-3-642-41422-0_14, © Springer-Verlag Berlin Heidelberg 2013

I discussed the idea of limiting the power of the communication channel in a post from 2009.

An aside: are you all getting tired of Alice and Bob, they seem to be everywhere? I recently was told to stop using them to explain things. What do you think? Here is a great comment about them from John Gordon:

> So you see Alice has a whole bunch of problems to face. Oh yes, and there is one more thing I forgot to say—Alice doesn't trust Bob. We don't know why she doesn't trust him, but at some time in the past there has been an incident.
> Now most people in Alice's position would give up. Not Alice. She has courage which can only be described as awesome. Against all odds, over a noisy telephone line, tapped by the tax authorities and the secret police, Alice will happily attempt, with someone she doesn't trust, whom she cannot hear clearly, and who is probably someone else, to fiddle her tax returns and to organize a coup d'etat, while at the same time minimizing the cost of the phone call.
> A coding theorist is someone who doesn't think Alice is crazy.

14.1 Computationally Limited Channels

A major idea in their paper is to restrict the communication channel between Alice and Bob—yes I will stay with them—to be a logspace-limited computation. Actually they study a number of other results, but this is the one that I will discuss. The restriction on the *space* used by the channel to decide which bits to change is pretty reasonable in my opinion.

But I am biased. Almost twenty years ago I worked on a related problem and got weaker results. I did however introduce—I believe—the idea of limiting the computational power of the channel. My results were for the case where the channel was polynomial-time limited, which is of course more powerful than a logspace-limited one. At least we believe that is true: recall we still do not know whether logspace equals polynomial-time.

Shortly after my paper was out, Zvi Galil, Moti Yung, and Xiangdong Yu wrote a paper on limited channels, changing my assumption to logspace. I made a terrible mistake. They asked me to join them and we eventually wrote a joint paper which we never published. It was titled: "Computational Error-Correcting Codes Achieve Shannon's Bound Explicitly."

My terrible mistake was joining them. Their paper without me had a line in the introduction about the "great idea of Lipton." But once I was added to the paper that line was removed, as it should have been. Rule:

> Never join a paper that gives you a great reference.

14.2 The New Results

Adam and Venkatesan consider more generally space-limited channels. They showed that as long as the channel is restricted to logspace they have a coding method that is efficient and optimal up to the channel rate of $1 - H(p)$ and recovers a short list that contains the correct message with high probability.

I will not attempt to give any details of how they prove their theorem, take a look at their paper. I will make a few high-level comments about their wonderful paper. It makes use of Noam Nisan's famous pseudorandom generator. This is one of the great results in complexity theory: Noam shows that there is a pseudorandom number generator that "fools" logspace computations, and does this unconditionally. Noam's proof uses no unproved assumptions, no need to assume that one-way functions exist.

Their paper also extends the previous work in several ways. The major one, in my opinion, is they get rid of the assumption of a shared secret that was used in the previous work. In my paper and in the paper of Galil, Yu, and Yung the assumption was that Alice and Bob had a shared secret. Clearly removing this is extremely important. They call these "setup" assumptions. The proof that they can do this is quite a subtle one. The "obvious" idea is to send the shared secret over the channel, but this must be done with great care. Recall they are making no assumptions about the existence of public-key systems. Take a look at how they make all these work—it is quite impressive.

14.3 Open Problems

Suppose that you have a result that uses an error-correcting code somewhere. In order to make your result work against worst-case adversaries, you would likely need to use a classic code that can handle an all-powerful adversary. What would happen if you replaced the worst-case code by the code of Adam and Venkatesan? Of course your result would now likely *fail*. There might be a situation that would invalidate your result. The open question is: can your result still be proved to be true if the adversary has limited computational power? I would love to see such a result.

Another question is a version of this for the SAT problem, my favorite problem. Suppose that a logspace machine M must create instances of SAT problems. If we allow the machine M to be random it is unclear why the generated instances would be any easier than "general" ones. However, suppose that we require that the machine M must know the answer to the instances it generates:

(1) If M generates a satisfiable formula ϕ, then M must also generate a satisfying assignment.
(2) If M generates an unsatisfiable formula ϕ, then M must also generate a short proof of this.

What is the complexity of SAT for formulas generated in this way?

14.4 Notes and Links

Original post:
http://rjlipton.wordpress.com/2010/10/14/strong-codes-for-weak-channels/
Guruswami–Smith paper:
http://arxiv.org/abs/1004.4017

FOCS 2010 conference page:
http://www.egr.unlv.edu/~larmore/FOCS/focs2010/
Talk by Guruswami:
http://www.cse.buffalo.edu/events/theory-III/slides/venkat.pdf
Eastern Great Lakes (EaGL) workshop:
http://www.cse.buffalo.edu/events/theory-III/
Transborder Regional University Network:
http://wings.buffalo.edu/intled/trun/
Binary symmetric channel:
http://en.wikipedia.org/wiki/Binary_symmetric_channel
Referenced post on channels:
http://rjlipton.wordpress.com/2009/05/06/worst-case-complexity/
John Gordon on Alice and Bob:
http://downlode.org/Etext/alicebob.htm
Nisan generator:
http://en.wikipedia.org/wiki/Pseudorandom_generator#Pseudorandom_generators_for_logarithmic_space

Picture credits:
Binary symmetric channel:
http://en.wikipedia.org/wiki/Binary_symmetric_channel
http://upload.wikimedia.org/wikipedia/commons/thumb/8/8e/Binary_symmetric_channel_(en).svg/600px-Binary_symmetric_channel_(en).svg.png