880 Lecture 1

January 24, 2012

We will be presenting frameworks for counting problems, and using them to get tractability or intractability results.

1 Holant

Given the graph $G = (V, E)$, we have the edges are variables taking either 0 or 1, and every vertex is a function over its edge values. The vertex function can also be called a constraint function, or a signature. So for vertex v, we have f_v. This is the Holant framework.

Imagine if we have, for all v, $f_v =$ EXACT-ONE. So $f_v(e_1, e_2, \ldots, e_{deg(v)}) = 1$ if the number of 1s is 1, and 0 otherwise. We say $f_v|_{E(v)}$ to mean the evaluation of f_v over the assigned value of the edges of v. Consider this equation

$$\sum_{\sigma: E \to \{0,\}1} \prod_{v \in V} f_v|_{E(v)} \tag{1}$$

where we sum over all possible edge assignments. What does this count? The number of perfect matching! In a sense, $\prod_{v \in V} f_v|_{E(v)}$ is a big conjunction when f_v is either 0 or 1.

So a Holant problem is $\Omega = (G = (V, E),$ assignment of functions at vertices$)$. We can consider what different kinds of functions to use, and if the problem becomes hard or easy.

2 Partition

Consider if we have $G = (V, E)$ and an assignment of $0, 1$ values to every *vertex* $\sigma : V \to \{0, 1\}$ and each *edge* has a function f_e. This is a case that arises in physics, it is called a Partition problem. Any such function f_e can be described in the form:

$$f = \begin{pmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \tag{2}$$

If the graph is undirected, then the matrix is symmetric. In that case, where $b = c$, either they are 0 or non-zero. Note that for the zero case,

$$f = \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix} \tag{3}$$

it is easy, because the only non-zero cases force all the vertices which are connected to one another to have the same values. We can scale by a constant factor, so for the non-zero case we can make the anti-diagonal entries 1.

$$f = \begin{pmatrix} a & 1 \\ 1 & d \end{pmatrix} \tag{4}$$

Then that's the spin model, and if $a = d$ that's the Ising model. Again we compute the sum of products:

$$\sum_{\sigma: V - \{0,1\}} \prod_{e \in E} f_e(u, v) \tag{5}$$

Though physicists often put this in the additive form: Define $H = \sum_{e \in E} g_e(u, v)$, then they compute e^H. Technically throughout this there are many constants, but we can always scale those. We will see that all this is a very special case of graph homomorphism.

3 Holant is More General than Partition

There is a straightforward way of converting from a Partition problem to a Holant problem: If in the partition problem we have two vertices u and v that are connected by an edge, then the two vertices each take a $\{0, 1\}$ value and the edge computes some function f_e. What we do is add a new vertex in-between, call this new vertex e. This gives us the path (u, e, v). Now the *edges* take $\{0, 1\}$ values, the vertex e has the function f_e, and u, v both take the equality function of the appropriate arity, so that all the edges connected to u are the same value, and all the edges connected to v are the same value.

You can see that the Holant is the same as the Partition function in this case. It has been shown that counting-perfect matchings, the example problem when we introduced the Holant framework, is *not* expressible as a Partition problem, and so the inclusion is strict.

Computing the number of perfect matchings is $\#P$-hard. Are there signatures where computing the Holant is tractable?

4 A Tractable Family of Signatures: Fibonacci Gates

Consider a function f that has n Boolean inputs where f has value f_i if i of those inputs are 1. The behavior of f depends solely on the hamming weight of its input. For such a function, we define a *symmetric* signature as $f = [f_0, f_1, \ldots, f_n]$.

A symmetric signature f is Fibonacci if $f_{k+2} = f_{k+1} + f_k$, for $k \geq 0$. An example is $f = [1, 0, 1, 1]$. This is tractable! That is quite interesting, because it is the "not", in a sense, of the same gate which counts the number of perfect matchings. The signature for perfect macthings is $f = [0, 1, 0, 0]$, and is $\#P$-hard.

This brings us to Ladner's theorem. There is a $\#P$ equivalent, that if $P \neq \#P$ then there are $\#P$-intermediate problems. However, these intermediate problems are artificial, they're constructed through diagonalization. But for these sum-of-product problems, which seem to capture most "natural" problems, it does not appear to be the case: there are dichotomy theorems, that all problems so expressed are either in P or $\#P$-hard, nothing in between.

In the papers handed out, we can read why Fibonacci gates are tractable. The proof (for a more general type of gate) is using holographic transformations. The steps are:

1. The basic Fibonacci gates are tractable.
2. Holographic transformations of those gates are tractable.
3. We can characterize all of the transformed signatures (via "basic transformations").
4. A "collapse theorem" says that "basic transformations" suffice.

These four steps seem like a somewhat esoteric result without the history. But, then we see the next result:

5. Hardness. That's it!
6. Technique for #2 is also used in hardness proofs.

So #5 brings us to a dichotomy theorem, either the problem is known-easy or it is $\#P$-hard.

The two techniques we use for dichotomy results are really #6 and the seventh: interpolation.

5 Holographic Transformations

What are holographic transformations? Consider the "full" vector of a signature (expanded from the symmetric notation). For example, we have NOT-ALL-EQUAL defined as $[0, 1, 1, 0]$ as a symmetric signature. That is for a three-input function—it is an abbreviated form of an eight-dimensional vector indexed by the Boolean value of the inputs. So we can write it as

$$\text{NAE} = (1,1)^{\otimes 3} - (1,0)^{\otimes 3} - (0,1)^{\otimes 3}. \tag{6}$$

Now consider multiplying by this matrix, which may look familiar from quantum computation

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes 3} \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}^{\otimes 3} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes 3} - \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes 3} \right). \tag{7}$$

Recall the remarkable fact about tensors

$$A^{\otimes n} B^{\otimes n} = AB^{\otimes n}. \tag{8}$$

So we can say the above equals

$$\text{NAE} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes 3} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}^{\otimes 3} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}^{\otimes 3} - \begin{pmatrix} 1 \\ -1 \end{pmatrix}^{\otimes 3} \tag{9}$$

Which, after doing the tensor powering, is (back in symmetric notation)

$$[8, 0, 0, 0] - [1, 1, 1, 1] - [1, -1, 1, -1] = [6, 0, -2, 0]. \tag{10}$$

And if you do the inverse transformation on the signature on the other side, which is the function $(=_2) = [1, 0, 1] = (1, 0, 0, 1) = (1, 0)^{\otimes 2} + (0, 1)^{\otimes 2}$, or

$$\left((1,0)^{\otimes 2} + (0,1)^{\otimes 2}\right) \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{11}$$

(Recall that this matrix is its own inverse, ignoring constants.) This equation, you can see, ultimately results in $[2, 0, 2]$. That is, given a constant, *still* the equality function! So that remains the same.

The important part of this transformation is that $[6, 0, -2, 0]$ fulfills the parity requirement, and is *matchgate realizable*, this is tractable for planar graphs by the FKT algorithm.