Calculating Group Characters Algorithmically

David Loeffler

Spring 2007

In this lecture, we'll see how, given enough knowledge about the structure of a finite group G, we can calculate the character table of G in a completely deterministic process, which is easy to automate on a computer. The method we'll use is called *Burnside's algorithm*.

1 A lemma on conjugacy classes

As input, we need one more piece of theory.

Lemma 1. *If (ρ, V) is any irreducible representation of G and R is a conjugacy class in G, then*

$$\sum_{r \in R} \rho(r)$$

is a scalar multiple of the identity.

Proof. For all $g \in G$, we have

$$\rho(g) \cdot \left(\sum_{r \in R} \rho(r) \right) = \left(\sum_{r \in R} \rho(grg^{-1}) \right) \cdot \rho(g) = \left(\sum_{r \in R} \rho(r) \right) \cdot \rho(g).$$

Hence $\sum_{r \in R} \rho(r)$ is a scalar by Schur's lemma. □

By calculating traces, it's clear what this scalar must be: it's $\frac{|R|\chi(r)}{d}$, where $d = \dim V$ and χ is the character of ρ. (Here we write $\chi(r)$ for the value of χ on any $r \in R$; since χ is a class function it doesn't depend on the choice of r. We shall use this convention throughout.)

Corollary 2. *For any two conjugacy classes R and S, we have*

$$\left(\frac{|R|\chi(r)}{d} \right) \cdot \left(\frac{|S|\chi(s)}{d} \right) = \sum_T c_{RST} \frac{|T|\chi(t)}{d},$$

where c_{RST} is the number of pairs $(r, s) \in R \times S$ such that $rs = t$, for any fixed $t \in T$ (which is clearly independent of the choice of t).

Proof. The right-hand sum is clearly just

$$\sum_{(r,s)\in R\times S} \frac{\chi(rs)}{d} = \mathrm{Tr}\left(\sum_{(r,s)\in R\times S} \frac{\rho(r)\rho(s)}{d}\right)$$
$$= \mathrm{Tr}\left(\sum_{s\in S} \frac{\sum_{r\in R}\rho(r)\cdot\rho(s)}{d}\right).$$

Since $\mathrm{Tr}(\lambda A) = \lambda\,\mathrm{Tr}(A)$ for any matrix A and scalar λ, and $\sum_{r\in R}\rho(r) = \frac{|R|\chi(r)}{d}$ is a scalar, we can pull it out, so this is

$$\left(\frac{|R|\chi(r)}{d}\right)\cdot \mathrm{Tr}\left(\sum_{s\in S}\frac{\rho(s)}{d}\right) = \left(\frac{|R|\chi(r)}{d}\right)\cdot\left(\frac{|S|\chi(s)}{d}\right)$$

as required. □

2 The algorithm

We now number the conjugacy classes as $R_1,\ldots,R_N$ for some N, and calculate, for each $k = 1\ldots N$, the $N\times N$ matrix M_k whose (i,j) entry is $c_{R_k R_i R_j}$.

Theorem 3. *The vector*

$$\begin{pmatrix}\chi(r_1)|R_1|/d\\ \vdots\\ \chi(r_N)|R_N|/d\end{pmatrix}$$

is an eigenvector for each of the matrices M_k, and the corresponding eigenvalue is $\chi(r_k)|R_k|/d$.

Proof. This is just a restatement of Corollary 2 above. □

But it's easy to calculate the matrices M_k from the multiplication table of G – we don't need to know anything about the characters in order to do this. Calculating the eigenvectors and eigenvalues of these matrices now allows us to work out the characters!

More precisely, we can find, by ordinary linear algebra, N vectors which are simultaneous eigenvectors for all the matrices. Since we know there exist N simultaneous eigenvectors such that no two of them have the same eigenvalues for all of the M_k's, any set of simultaneous eigenvectors will be rescaled versions of these ones. The scaling factor is uniquely determined by the requirement that for any irreducible character χ, $\chi(1)$ is a positive integer and $\langle\chi,\chi\rangle = 1$, so we can now read off the characters.

3 An example

Let's use this to work out the character table of a group. I'm not sure whether one should confess to having a favourite group, but I've always been rather fond of the group of order 20 with presentation $\langle a, b \mid a^5 = b^4 = e, b^{-1}ab = a^2\rangle$.

This can be realised as the subgroup of S_5 generated by (12345) and (2354). Group theorists will see this as the semidirect product of C_4 and C_5, while number theorists like myself might prefer to think of it as the Galois group of the splitting field of $X^5 - a$ where a is any element of $\mathbb{Q}$ which isn't a fifth power.

It's easy to calculate the conjugacy classes of G:

Class	1	2	3	4	5
Elt	Id	$(2,5)(3,4)$	$(1,2,3,4,5)$	$(2,3,5,4)$	$(2,4,5,3)$
Order	1	2	5	4	4
Size	1	5	4	5	5

The corresponding matrices are

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad M_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 5 & 0 & 5 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 5 & 0 \end{pmatrix} \quad M_3 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 4 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix}$$

$$M_4 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 5 & 0 & 5 & 0 & 0 \end{pmatrix} \qquad M_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 5 & 0 & 5 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \end{pmatrix}$$

All of this is easy to check by hand, given that we know how to multiply elements of G. We also know how to do linear algebra, so we get some eigenvectors:

$$\left\{ \begin{pmatrix} 1 \\ 5 \\ 4 \\ 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \\ 4 \\ -5 \\ -5 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ -5 \\ 4 \\ 5i \\ -5i \end{pmatrix}, \begin{pmatrix} 1 \\ -5 \\ 4 \\ -5i \\ 5i \end{pmatrix} \right\}$$

And now we're more or less finished; the scaling factors are easy to work out, and we deduce that the character table is:

Class	1	2	3	4	5
Elt	Id	$(2,5)(3,4)$	$(1,2,3,4,5)$	$(2,3,5,4)$	$(2,4,5,3)$
Order	1	2	5	4	4
Size	1	5	4	5	5
χ_1	1	1	1	1	1
χ_2	1	1	1	-1	-1
χ_3	1	-1	1	i	$-i$
χ_4	1	-1	1	$-i$	i
χ_5	4	0	-1	0	0

4 Concluding remarks

The example we've just done shows two things: firstly, that a character table algorithm exists; and secondly, that it's extremely tedious to do in practice.

The two main steps are calculating the constants c_{RST}, and doing the linear algebra to extract the eigenvectors. The first is easy if you have the whole multiplication table of G in front of you, but if the group is at all large (more than 500 elements or so) this is an awful lot of data. The second is even more of a problem: it's extremely difficult to calculate the eigenvectors of a very large matrix exactly, even if you know in advance that the eigenvalues are defined over some relatively small extension of $\mathbb{Q}$ (a cyclotomic one in this case).

Hence in practice one tends to look for short cuts and apply some "native wit", much as we did in class – this is what's sometimes known as a *heuristic* approach to the problem. There are computer programs that calculate character tables, and their approach is usually to use a selection of pre-programmed tricks to try and reduce the problem to a simpler one, only falling back on Burnside-type algorithms if nothing else works.