Lecture 1: Weighing

September 6, 2006

0.1 Problem

Suppose we have n coins; one is bad, it has weight different from that of the others, which are all the same.

We want to find the bad coin.

We have a balance device, and can put coins on either side of it. It will balance only when there are the same number of coins on each side of it, and they are all good ones.

Our task is to discover which of our coins is counterfeit, as efficiently as possible. We measure efficiency by the number of times we have to use our balance.

There is a standard puzzle that asks: how can we handle 12 coins with 3 weighings? We will answer this and much more.

0.2 Lower bound on the number of weighings needed

First we are going to look for a lower bound on the number of weighings we need, with n coins. This will give us a way of estimating how good any procedure we develop is.

The standard way to get a lower bound is what is called the *Pigeonhole Principle:* It says, if you have more pigeons (or objects you want to stick in pigeonholes) than pigeonholes, then at least two of them must share a hole.

This is relevant here because our task is to use the outcomes of weighings to distinguish the bad coin. Thus, the number of outcomes of weighings must be greater than or equal to the number of coins. In other words, if coin 5 say is bad, it must give rise to a different weighing outcome from what would happen if coin 7, or any other of the n coins, is bad.

Thus the outcomes of the weighings are like pigeonholes, and the possible bad coins are like pigeons. There is one subtlety however. If any of our coins is ever weighed, and it is the bad one, it can give rise to two different weighing outcomes. One if it is heavy, and one if it is light. If the coin is heavy, its side of the balance will go down; if light it will go up, and these are different outcomes.

Note that at most one coin can be kept off the balance for all weighings. (Otherwise, if two coins were off the scale at every weighing, we could not distinguish one coin being bad from the other coin being bad.)

This tells us that the number of weighing outcomes has to be at least $2n - 1$ if we are to have a separate outcome for each possibly bad coin. We have $2n - 1$ pigeons with n potentially bad coins.

So how many outcomes do we get with k weighings?

With one weighing we get three possible outcomes. Which we can denote as (left side) heavy, (left side) light, or balance. Which we describe by the numerals 1, -1 and 0 respectively.

With k weighings we get a sequence of such numbers; thus with three weighings we can write a typical outcome as 1, -1,-1, which corresponds to the right side going down in the second and third weighing, and the left side going down in the first.

The number of such sequences is 3^k. This number is the number of pigeon

holes. The first few values of which are $3, 9, 27, 81, \ldots$

The pigeonhole principle then tells us, that we must have

$$3^k \geq 2n - 1$$

and this determines our lower bound on the number of weighings needed.

The first few values we get for the largest number n possible with k weighings are: $2, 5, 14, 41, \ldots$

This number in general is $((3^k) + 1)/2$.

0.3 The non-adaptive problem, matrix notation and conditions on the weighing matrix

So now we turn to the question, can we produce a weighing scheme that will always find the bad coin, that meets this bound. That is, can we find a procedure for putting coins on the balance that will allow us to find the bad coin with $n = 2, 5, 14, 41$ coins and $1, 2, 3, 4, \ldots$ weighings?

The answer is both yes and no. If we are given an additional $(n + 1)^{\text{st}}$ coin that we know to be good, then we can. Otherwise, we can not.

First, Why not?

The pigeonhole principle tells us that not only can we distinguish at most 3^k cases using k weighings, but after the first weighing, no matter what happens, we can distinguish from at most $3^{(k-1)}$ cases for each outcome of the first weighing. And 3 raised to any power is an odd number.

On the other hand, if we have no definitely good coin, all the coins we put on the balance must be potentially bad; and we must put the same number of coins on either side of it.

This means that the left hand side will go down if any of the x coins on it is heavy or if any of the x coins on the right side is light. This is $2x$

cases which has to be an even number. No matter what we do, after the first weighing then, the number of cases corresponding to the left being heavy (or light) must be an even number less than $3^{(k-1)}$ or at most $3^{(k-1)} - 1$. This lowers the number of possible cases by 2, and the number of coins we can handle to at most 1, 4, 13, 40, in 1 to 4 weighings. And to $((3^k) - 1)/2$ in general. This number respresents the the largest number n possible coins with k weighings when we are not given a definite good coin.

We will assume we have a good coin, because this allows us to accommodate one extra coin, and because it allows us to do something useful in the one weighing case.

The standard way to attack this problem is to see what the cases are if we put x coins on each side, arrange so that there are $3^{(k-1)}$ possibilities for each first outcome, and continue in the same way to figure out what to do on the second weighing, and so on. (If this paragraph makes no sense I apologise and ignore it.)

We are going to create a harder problem in that there are more constraints on the solution, which actually makes it easier to solve. We will look for a scheme of weighing that is fixed in advance (non-adaptive algorithm). We will not look at what happened on previous weighings in determining what to do next (adaptive algorithm); instead we will seek as scheme for putting coins on the balance that can be written down in advance for all weighings at once.

We need some notation to describe such a scheme. And that notation is a matrix.

Along the top of the matrix we list the coins, and in each row we list where that coin goes on the corresponding weighing. If coin y is put on the left on weighing j we put a 1 in the jth row and the column marked y.

If it goes on the right in that weighing we put a -1 in that place, and it stays off the balance, we put a 0 there.

Thus, with this notation, if we have only one weighing, and have a good

coin, we can employ the weighing scheme defined by the following one-row matrix:

$$
\begin{array}{ccc}
g & 1 & 2 \\
1 & -1 & 0
\end{array}
\tag{1}
$$

We now ask: what conditions must a matrix like this satisfy to represent a weighing scheme that allows us to find the counterfeit coin, no matter which one it is?

- First, every row sum must be 0. This expresses the condition that the same number of coins must be on each side of the balance, in every weighing.

- Second, no two columns, neither one representing the good coin, can be identical.

- Third, no two columns can be the exact negative of one another.

Why these conditions?

Obviously, if two columns are identical, the corresponding coins will give rise to the same set of outcomes, if either one is bad. And if one is the negative of the other, one coin being heavy gives rise to the same set of outcomes as the other being light. So these conditions are necessary for a successful weighing scheme. In other words, each pigeonhole can contain at most one pigeon.

Actually, these conditions are sufficient as well. For, if a matrix obeys these conditions, any particular outcome can be the same as at most one column or the negative of at most one column, and not both. So every outcome can correspond to only one bad coin.

0.4 Construction procedure for a solution

Now we turn to the problem of actually constructing a good weighing scheme of this predetermined, (or non-adaptive) kind.

We will do so by induction. We will start with the simple one weighing scheme, make three copies of it, putting 1's under the first copy, -1's under the next copy, and 0's under the third. What we get will not be the two weighing scheme we seek, but will be easily changeable into a good two weighing scheme.

So what do we get?

Here it is:

$$
\begin{array}{ccccccccc}
g1 & 1 & 2 & g2 & 3 & 4 & g3 & 5 & 6 \\
1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 \\
1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0
\end{array}
\tag{2}
$$

And what is wrong with this scheme?

First it has three good coins instead of 1. We must therefore get rid of two of them, say g2 and g3. Actually we can get rid of either g2 or 3 and either g3 or 5; that is we can switch the names of g2 and 3 or the names of g3 and 5 and get rid of the new g2 and g3, if we want. Who cares which we do?

Second there is a pair of columns that are the oppostite of one another. These are columns 2 and 4. So to get a good scheme we must get rid of one of these two columns as well.

In short, the two row scheme above has three pairs of columns that are opposites of one another. (not counting the g1 column.) These are :

$$
\begin{array}{cccccc}
1 & -1 & 1 & -1 & 0 & 0 \\
-1 & 1 & 0 & 0 & 1 & -1
\end{array}
\tag{3}
$$

The scheme does have 0 row sums, so to make it into a good scheme we must eliminate one column from each of these pairs, and what we eliminate

must also have 0 row sums.

Can we do it?

Yes we can. Take the first fourth and fifth of these, (or the other three) and throw them away. The resulting scheme will be a good one. And here it is: (with the first, fourth, and fifth in the three pairs above gone, and the coin names changed:

$$
\begin{array}{ccccccc}
g & 1 & 2 & 3 & 4 & 5 \\
1 & -1 & -1 & 0 & 1 & 0 \\
1 & 1 & -1 & -1 & 0 & 0
\end{array}
\tag{4}
$$

The nice thing about the procedure we used to go from one row to two works identically to go from two rows to three and three to four and so on.

How come?

The procedure is: take three copies of this scheme, add 1's , -1's and 0's under them, keep g1 but find the pairs of other columns that are negatives of one anothter; two of these will be pairs including the second and third copies of the good coin. The third will come from the all 0's column with 1 and -1 beneath it.

Why no others?

Because any two other columns can be opposite one another (or identical) only if their tops (without the last row) are opposite one another (or identical) and this happens only with good coins, or with the coin that never goes on the scale; which is the negative of itself.

And how do we eliminate one from each of the three ± pairs?

Exactly as in the one row to two row case; Let us do it explicitly to go from two to three rows.

Here are the bad pairs

$$\begin{array}{cccccc} 1 & -1 & 1 & -1 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & -1 \end{array} \qquad (5)$$

We can get rid of the first fourth and fifth of these again, and get a good scheme.

And the same thing works every time; the only difference is that the length of the columns increases. All entries except the last are always the same within each column, so the column length is of no relevance to the procedure.

And how good is the scheme we get? It meets our bound!

That is, the number of columns for k rows, call it $s(k)$, obeys $s(k) = 3s(k-1)-1$. Which is exactly the recursion of the bound (Why $3s(k-1)-1$? Well we triple our columns and get rid of three, two of which correspond to duplicates of the good coin; thus the number of risky coins triples minus one.)

The actual formula for the maximum number of coins in k weighings is $((3^k) + 1)/2$, which obviously obeys the same recursion.

And what happens if we have no good coin?

Then we can throw away the good column and its negative and we have a scheme for $((3^k) - 1)/2$ at risk coins.

And if as well, we want to know whether the bad coin is heavy or light? Then we cannot have an all-0 column which reduces us to $(3^k)/2 - 3/2$ coins in our scheme.

Suppose we want a good scheme for $(3^k)/2 - 5/2$ coins?

That is a good question. Figure out such a scheme for homework!