Betting on Football Pools

by
Edward A. Bender

In a pool, one tries to guess the "winners" in a set of games. For example, one may have ten matches this weekend and one bets on who the "winners" will be. We've put winners in quotes because the pool may handicap the matches so that it is expected that each side has an equal chance. For example, if the Jets are playing the Sharks and the Sharks are a weaker team, then there will be point spread—the Jets must score some specified number of points more than the Jets to be declared the winner in the pool.

There a two common types of pools.

- One is among a group of friends and the person with the most correct guesses wins all the money that was bet. If several people have the same number of correct guesses, the money is divided evenly among them. We'll call this the *office* pool since that's where it's often done.
- The other type of pool is like the lotteries that are run in many states: How much you win depends on how many correct guesses you have and it is set up so that the organizers expect to pay out less than they take in. We'll call this the *for-profit* pool.

In both kinds of pools, a player normally does not know what winners the other players have chosen.

1 Can You Make Money in a Football Pool?

Suppose you play the football pool many times. Can you expect to come out ahead in the long run or must you certainly lose?

Suppose the handicapping is fair; that is, in each match, both teams have a 50% chance of being declared the (handicapped) winner. In that case, everyone in the pool may as well guess randomly at the outcome of each game and each person has an equal chance of winning.

- Since everything that comes in is paid out in the office pool, you should tend to break even there in the long run.
- Since not everything is paid out in the for profit pool, you should tend to lose there in the long run.

This seems to be the best you can do. Of course, you can do better if you somehow know that some teams are better or worse than believed. We'll explore some consequences of this in the for profit pool.

Amazingly, you can do better than break even in the office pool! How is this possible? The idea is to enter more than one bet each time, say two bets. Since each bet only breaks even in the long run, how can it help to make two bets, each of which seems to break even in the long run?

Before we go into this, which is a little complicated mathematically, let's look at a simpler situation which has the same strange property.

Three people are told that they will play a game as a team against the Mad Hatter. Here's how it works.

- The Mad Hatter will place either a red or a blue hat on each person's head. He'll choose the hats randomly.
- Each person can see the other two hats, but not his own. They will not be allowed to communicate. Each person must write on a slip of paper either a guess (red or blue) of his hat color or "no guess."
- If everyone writes "no guess" or if someone guesses the wrong color, the team loses. Conversely, if there is at least one guess and all guesses are correct, the team wins.

The team is told to work out a strategy and then they will play the game. One person says:

> It's clear that a guess can only be right half the time since seeing the other two hats is no information at all. Thus, if one person guesses, we have an even chance of winning. If two people guess, there is only one chance in four that both will be right. If all of us guess, there is only one chance in eight that all of us will be right. Therefore, it is *obvious* that we should simply choose one of us to guess and the other two should write "no guess."

Everything this person said is correct, but the strategy is not the best possible. In fact, there is a strategy gives a 75% chance of winning!

Here's the winning strategy. All three people do the same thing, namely: If the other two people have the same color hat, write the opposite color; otherwise write "no guess." Let's see what happens. There are eight possibilities which are listed in the following table along with each person's guess and whether it is a win or not.

HAT	guess	HAT	guess	HAT	guess	win?
RED	blue	RED	blue	RED	blue	no
RED	none	RED	none	BLUE	blue	yes
RED	none	BLUE	blue	RED	none	yes
RED	red	BLUE	none	BLUE	none	yes
BLUE	blue	RED	none	RED	none	yes
BLUE	none	RED	red	BLUE	none	yes
BLUE	none	BLUE	none	RED	red	yes
BLUE	red	BLUE	red	BLUE	red	no

Why did this work? Each person guessed half the time and half of those guesses were wrong, but the right and wrong guesses were distributed differently:

- When a person guessed correctly, the other two people did not guess. So every correct guess led to a win.
- When a person guessed incorrectly, everyone guessed incorrectly. So the wrong guesses "piled up."

2 The Office Pool

Can we use this idea somehow in the office pool? If all the entries in the pool are random, then sometimes one person will win and other times there are ties and the pool is shared. To do better than average, we should try to avoid ties. We might be able to do this by analyzing all the other bets and then making our choice, but we want a simpler strategy. The key idea is to make more than one bet is such a way that they will seldom if ever be tied. Suppose our bets can never be tied. Each is still as likely to win as a random bet; however, if it wins it will have to share with less than a random bet would since it will never have to share with our other bets.

Here's one way to achieve this. Make two bets. Choose the first any way at all. Choose the second so that all its guesses for the winners are just the reverse of the first bet. If there are N games and the first bet gets k of them correct, then the second will get $N-k$ correct, namely those the first bet gets wrong. There will be a tie if and only if $k = N-k$. Thus N must be even and $k = N/2$. Since someone else is likely to guess more than half the games correctly, it is unlikely that our two bets will have to share the pool.

This may all sound good, but is it correct or is there a flaw in our reasoning? We should be suspicious since the "obvious" strategy with the hats turned out to be wrong. The surest way to deal with this is to carefully calculate the result for some number of games and people in the pool.

Suppose there are three games, we make two bets as suggested above, only one other person is in the pool, and all games have 50:50 odds.

The following table shows the eight possible win/lose results for our two bets. For example, $\ell w \ell$ means the bet was correct (won) on the second game but was wrong (lost) on the the other two games. The bottom row shows the number of correct guesses for the best bet. Each of the eight outcomes is equally likely because the games have 50:50 odds.

1st	www	$ww\ell$	$w\ell w$	$w\ell\ell$	ℓww	$\ell w\ell$	$\ell\ell w$	$\ell\ell\ell$
2nd	$\ell\ell\ell$	$\ell\ell w$	$\ell w\ell$	ℓww	$w\ell\ell$	$w\ell w$	$ww\ell$	www
best	3	2	2	2	2	2	2	3

Thus we have 1 chance in 4 of getting all games correct and the remaining time we get two games correct.

The other player has eight possible outcomes, all of which are equally likely. They look like the top row of the preceding table: 1 in 8 times we expect him to guess all three games correctly, 3 in 8 times we expect him to guess exactly two games correctly and half the time we expect him to guess less than two games correctly. We need to pair his possibilities with ours. That is done in the following table. The column headings are his correct guesses and the chances of that happening. The row headings are our correct guesses and the chances of that happening. We have made row and column widths proportional to the chances. Thus the area of each rectangle is proportional to its chances. The number in each rectangle is the fraction of the pot that we win.

	all 1/8	2 right 3/8	less than 2 1/2
all 1/4	1/2	all	all
2 3/4	0	1/2	all

Since the area of the six rectangles totals 1, the chances of being in any particular rectangle equals its area. We multiply each area by the fraction of the pot we get and add up the results to get the fraction of the pot we expect in the long run:

$$\frac{1}{4} \times \left(\frac{1}{8} \times \frac{1}{2} + \frac{3}{8} \times 1 + \frac{1}{2} \times 1\right) + \frac{3}{4} \times \left(\frac{1}{8} \times 0 + \frac{3}{8} \times \frac{1}{2} + \frac{1}{2} \times 1\right) \;=\; \frac{3}{4}.$$

If a bet costs a dollar, then we put in \$2, the pot is \$3 and we get on average $\$3 \times (3/4) \;=\; \$2.25$. Thus we expect to make on average 25 cents per pool.

Of course, this is an unrealistic situation: only three games and only one other person in the pool. Let's see what we can say in general. For simplicity, we assume that then number of games N is odd.

In this case, exactly one of our bets will have more than $N/2$ correct guesses. Why is this? For each of the N games, exactly one of our bets will be correct. Since the two bets have N correct guesses, the average number of correct guesses is $N/2$. Given an average of numbers, one or more of the numbers is at least as large as the average and one or more is at least as small. One of our bets must be right at least $N/2$ times. Since $N/2$ is *not* an integer, the bet must be right more than $N/2$ times. Similarly the other is right less than $N/2$ times.

Look at the set $\mathcal{S}$ of all the bets that are right more than $N/2$ times. Except for the fact that they're right more than $N/2$ times, they are random. (Why? Because the only nonrandom feature of the bets was the fact that one of our bets was opposite the other but only one of our bets is in $\mathcal{S}$.)

Since the bets in $\mathcal{S}$ are random, each bet is equally likely to be a winner. Suppose there are p people in the pool besides us. Thus the pool has $p+2$ dollars. Suppose there are s bets in $\mathcal{S}$. Since each bet is equally likely to be a winner, in the long term each bet in $\mathcal{S}$ will return $1/s$ of the pool. Thus we expect to average $\frac{p+2}{s}$.

To complete this we need to know the chances that s has a particular value and add up all the possibilities. We don't have the mathematics to do this in general — see [1] — but we can do particular cases. If there is one other person in the pool, then the chances are 50:50 that he will guess more than half the games correctly. Thus

- Half the time $s = 1$ (ours is the only bet in $\mathcal{S}$) and we get \$3.
- Half the time $s = 2$ and we average \$3/2.

Thus our average return is

$$3 \times \frac{1}{2} + \frac{3}{2} \times \frac{1}{2} \;=\; 2.25,$$

an average gain of 25 cents per pool. Notice what happened here:

- Our earlier calculations were for three games and one other bettor. To work out the answer we needed some tables.
- Instead of just plunging ahead with the calculations, we first removed the one nonrandom situation — our paired opposite bets. Then the calculations were easier and we did more: Still just one other bettor, but now any number of games (as long as it's odd).

We've done $p = 1$ and could do other values of p the same way. For example, when $p = 2$ we have $s = 1$ on average $1/4$ of the time, $s = 2$ on average $1/2$ the time, and $s = 3$ on average $1/4$ of the time. Instead, we'll simply quote the general formula from [1].

> If the number of games is odd, each game has a 50:50 chance, there are p other bettors and each of them guess randomly, then our average profit per pool is
>
> $$\frac{2}{p+1}\left(1 - \frac{p+2}{2^{p+1}}\right) \quad \text{dollars.} \qquad (1)$$

3 Various Averages and a Partial Result

We said we lacked the tools to derive (1), but can we at least show that our double bet strategy wins money for us. It may seem clear that it should from the earlier discussion, but we'd better be careful about trusting intuition after playing with the Mad Hatter. Can we find a proof?

It all depends on knowing something about averages. As we observed, if $\mathcal{S}$ has s people, we expect to win on average $\frac{p+2}{s}$ dollars. To compute our expected winnings, we simply need to compute the average of $1/s$ over the various outcomes and multiply by $p + 2$. We don't have the tools to do this.

On the other hand, we can compute the average value of s. Every bet has a 50:50 chance of being in $\mathcal{S}$. Thus, on average, there are $\frac{1}{2}(p + 2)$ bets in $\mathcal{S}$. In other words, the average value of s is $\frac{p+2}{2}$. Of course, that's not the average we wanted, so what good does it do us?

We need to digress for a discussion of averages. Given n positive numbers $a_1, a_2, \ldots, a_n$. One possible average is the *arithmetic mean* A, defined by

$$A \;=\; \frac{a_1 + a_2 + \cdots + a_n}{n}.$$

Another average that is sometimes used is the *geometric mean* G, defined by

$$\ln G \;=\; \frac{\ln(a_1) + \ln(a_2) + \cdots + \ln(a_n)}{n},$$

Exponentiating this equation and doing a bit of algebra, we can rewrite the geometric mean in a more common form:

$$G \;=\; (a_1 a_2 \cdots a_n)^{1/n}.$$

Another average is the *harmonic mean* H, defined by

$$(1/H) \;=\; \frac{(1/a_1) + (1/a_2) + \cdots + (1/a_n)}{n}.$$

For example, suppose $n = 2$, $a_1 = 2$ and $a_2 = 8$. Then

$$\begin{aligned} A &= (2+8)/2 \;=\; 5 \\ G &= \sqrt{2 \times 8} \;=\; \sqrt{16} \;=\; 4 \\ H &= \frac{1}{(1/2 + 1/8)/2} \;=\; \frac{2}{5/8} \;=\; \frac{16}{5} \;=\; 3.2, \end{aligned}$$

three different numbers. Which is correct? There is no "correct" answer. Which average we want depends on what we are doing. Notice that $H < G < A$ in this example. The following fact is proved in Appendix A:

Unless all the a_i are the same, $H < A$. (2)

Back to our office pool. In that case, we want the average value of $\frac{p+2}{s}$, which is $p+2$, times the average value of $1/s$. The average value of $1/s$ is the reciprocal of the harmonic mean of the s values. Thus, our average return is

$$\frac{p+2}{\text{harmonic mean of } s \text{ values}} \;>\; \frac{p+2}{\text{arithmetic mean of } s \text{ values}},$$

by (2). We already noticed that the arithmetic mean is $\frac{p+2}{2}$. Thus

$$\text{average return} \;>\; \frac{p+2}{(p+2)/2} \;=\; 2,$$

which proves that we win.

4 The For-Profit Pool

The typical for-profit pool differs in three important respects from the typical office pool.

- There are usually many more bettors.
- Only a part of the money taken in is paid out, the rest being kept by those running the pool.
- Instead of the people or persons with the most correct guesses dividing the prize, the for-profit pool usually has prizes for no errors in the bettor's guesses (first prize), a lesser prize (second prize) for one error and perhaps other lesser prizes.

With assumptions as in the office pool (50:50 winning chances in a game and bettors guessing randomly), will the office pool strategy work? No. The number p of bettors in (1) will be large so that the expected gain per bet will be very small, *assuming all money is returned as a prize.* Unfortunately, the money taken by those running the pool will be much larger than the small gain in (1).

Is there any way to win? Not with our present assumptions; however, if we "know"[1] the outcomes of some of the games, then the situation is different. In the office-pool case, we would make two bets. They would agree on the games where we knew what the outcome would be. This situation could be analyzed, but we'll turn to a different scenario.

Suppose there are N games and we "know" what the outcome of k of them will be. We want to place bets so that, assuming our "knowledge" is correct we are certain to win either a first or second prize. How can we do this?

Because we want to *guarantee* a win regardless of what happens, we don't need to make any assumptions about how the other bettors behave or the odds in the games we're uncertain about.[2]

[1] We might know that some games are fixed or have inside information about a player injury. On the other hand, we might just be using intuition. Regardless, we'll treat this "knowledge" as if it is a certainty.

[2] We would need such assumptions if we wanted to compute how much money we could expect to get back from our bet — an important question since we could still be losing

Since we know how we'll bet on the k games we're sure of, we'll just look at how we should bet on the other $N-k$ games. Call this number n. A bet consists of a list of the n winners. We can reprsent this by a list of n zeros and ones: A one means the first-named team in a contest wins and a zero means that team loses. We want a collection of n-long lists of zeros and ones (the bets) so that, given any list A of zeros and ones (the outcomes), there is a list B in our collection that either agrees exactly with A or differs from it in only one position. For example, when $n=3$, the two bets $(0,0,0)$ and $(1,1,1)$ work.

Can we say anything about how many bets we must make? Yes. There are $1+n$ possible outcome lists for which a bet will win: a first prize if the outcome list agrees with it and a second prize if the outcom list differs from the bet in just one of the n games. Thus b bets can "cover" at most $(1+n)b$ outcome lists. (It may be less because two bets might cover the same outcome list.) Since there are 2^n possible outcome lists and we want to cover every outcome list, we must have $(1+n)b \geq 2^n$. In other words,

$$\text{We must make at least } \frac{2^n}{n+1} \text{ bets.} \tag{3}$$

For example, when $n=4$ we must make at least $2^4/5 = 16/5 = 3\frac{1}{5}$ bets. Since the number of bets must be an integer, this means we must make at least four bets.

This problem belongs to an area called "covering problems" and is related to error correcting codes, which we look at next. For more information on the for-profit pool, see [2].

5 Error Correcting Codes

It is convenient to introduce the notion of "distance" between two n-long strings of zeros and ones:

$$d(a_1,\ldots,a_n;\ b_1,\ldots,b_n) \text{ is the number of } i \text{ for which } a_i \neq b_i.$$

money. In this lecture, we'll just look at how we need to place our bets to guarantee a prize, not whether it's expected to be financially rewarding.

For example, $d(0,0,0;\ 1,1,1) = 3$. The previous football pool problem we were just looking at can be phrased as follows:

$$\text{Find a set } B \text{ of } n\text{-long strings of zeros and ones such that, given } \textit{any } n\text{-long string } s \text{ of zeros and ones, there is at } \textbf{least} \text{ one string } b \text{ in } B \text{ such that } d(b;s) < 2. \text{ Also, make } B \text{ as } \textbf{small} \text{ as possible.} \tag{4}$$

Of course, if we just want to make sure we one of the first k prizes, we can replace $d(b;s) < 2$ with $d(b;s) < k$.

Error correcting codes are used in digital communication. The idea is that we have certain n-long strings of zeros and ones that are allowed to be sent and, if a few of the zeros and ones are garbled, we want to be able to guess what was sent. Examples include communications between (a) cell phones and towers, (b) satellites and ground stations, and (c) computers. The idea is, if the string s is received we want to choose the allowed string b that is as close as possible. Naturally, we want as many "code words"—allowed n-long strings as possible since this means we can send information faster. Also, we don't want more than one allowed string to be close. This can be phrased as follows:

$$\text{Find a set } B \text{ of } n\text{-long strings of zeros and ones such that, given } \textit{any } n\text{-long string } s \text{ of zeros and ones, there is at } \textbf{most} \text{ one string } b \text{ in } B \text{ such that } d(b;s) < k. \text{ Also, make } B \text{ as } \textbf{large} \text{ as possible.} \tag{5}$$

Except for the replacement of 2 with k, there are only two differences between (4) and (5), printed in **bold**. When $k = 2$, we can rewrite (3) as

$$\text{We can have at most } \frac{2^n}{n+1} \text{ code words.} \tag{6}$$

Of course, since this must be an integer, we can round it down just as we were able to round (3) up. When $\frac{2^n}{n+1}$ is an integer, it might be possible to find a set B that works for both the football pool and error correcting code problem. This can be done. We'll discuss it now.

Since 2^n is a power of 2, $n+1$ must also be a power of 2 to divide it evenly. So let $n = 2^\ell - 1$. When $\ell = 2$, we have $n = 3$ and we found the solution with the $\frac{2^3}{3+1} = 2$ words 0,0,0 and 1,1,1.

The next case is $\ell = 3$. We have $n = 2^3 - 1 = 7$ and $\frac{2^7}{7+1} = 16$. Since the problem is getting larger, it would be nice to have a more systematic way of stating the answer. To do this, we need to discuss how to add together two strings of zeros and ones. We do that element by element. We're not allowed to have twos, so we define $1+1$ to equal 0. For example, 0,1,1,0 plus 1,1,0,0 equals 1,0,1,0. Here are sixteen bets (or code words) for $\ell = 3$:

- The string 0,0,0,0,0,0,0.
- The four strings $b_1 = 1,0,0,0,0,1,1 \quad b_2 = 0,1,0,0,1,0,1$
  $b_3 = 0,0,1,0,1,1,0 \quad b_4 = 0,0,0,1,1,1,1.$
- All possible sums of two or more of the strings b_1–b_4. For example $b_1+b_2+b_3 = 1,1,1,0,0,0,0$. (There are eleven such sums.)

There is a method for systematically constructing solutions for any value of ℓ. To explain the method and prove that it works would take us too long because we would need some background in "linear algebra" and "finite fields." Most texts for an applied algebra course like Math 103 discuss error correcting codes. There are also books devoted to the subject. For example, the book [3] can be read after taking a linear algebra course such as Math 20F.

A Inequalities for Means

Let's see how we can prove (2). Before doing so, we need a theorem about concave and convex functions:

Theorem 1 *Suppose $a_1, \ldots, a_n$ lie in some interval I and are not all equal. Let A be their arithmetic mean.*

- *If f is a function for which $f''(x) > 0$ for all $x \in I$, then $f(A)$ is less than the arithmetic mean of $f(a_1), \ldots, f(a_n)$.*
- *If f is a function for which $f''(x) < 0$ for all $x \in I$, then $f(A)$ is greater than the arithmetic mean of $f(a_1), \ldots, f(a_n)$.*

Proof: If you wish, you can just assume the theorem and skip the proof. Using the Fundamental Theorem of Calculus and then integrating by part with $u = f'(t)$ and $v = t - b$, we have

$$f(b)-f(a) \;=\; \int_a^b f'(t)\,dt \;=\; (t-b)f'(t)\big|_a^b - \int_a^b (t-b)f''(t)\,dt \;=\; 0-(a-b)f'(a)+\int_a^b (b-t)f''(t)\,dt.$$

With a little rearranging:

$$f(b) \;=\; f(a)+(b-a)f'(a)+E(b), \quad \text{where} \quad E(b) \;=\; \int_a^b (b-t)f''(t)\,dt. \tag{7}$$

Suppose $f''(t)$ is either always positive or always negative between a and b. If $a < b$, then $b - t > 0$ for $a < t < b$ and so $E(b)$ and $f''(t)$ have the same sign. If $a > b$, then $\int_a^b = -\int_b^a$ and you should be able to show that, again, $E(b)$ and $f''(t)$ have the same sign.

We now return to the statement of the theorem. In (7), let $a = A$, let $b = a_i$ and sum over i:

$$\sum_{i=1}^n f(a_i) \;=\; nf(A) + \sum_{i=1}^n (a_i - A)f'(A) + E, \quad \text{where} E \;=\; \sum_{i=1}^n E(a_i). \tag{8}$$

Note that E has the same sign as f'' and that

$$\sum_{i=1}^n (a_i - A)f'(A) \;=\; \left(\sum_{i=1}^n a_i - \sum_{i=1}^n A\right)f'(A) \;=\; (nA - nA)f'(A) \;=\; 0.$$

Using this in (8) and dividing by n, we obtain

$$\frac{1}{n}\sum_i^n f(a_i) \;=\; f(A) + \frac{E}{n},$$

which gives the theorem. This completes the proof.

We can now return our attention to (2).

It's natural to work with just H and A, but it's possible to do something more general. You may have guessed that there are lots of possible averages — take your favorite invertible function f and define the f-mean F by

$$f(F) \;=\; \frac{f(a_1) + f(a_2) + \cdots + f(a_n)}{n}. \tag{9}$$

To insure that f is invertible (so that we can compute F), we'll require that f' never be zero. To compare F with A, we need one technical fact, which we won't prove:

For the harmonic mean H, $f(x) = 1/x$ and so $f''(x) = 2/x^3 > 0$. By Theorem 1, $f(A)$ is less than the arithmetic mean of $f(a_1), \ldots, f(a_n)$. By definition, the latter mean is $f(H)$. Thus $f(A) < f(H)$. Since f is a decreasing function, $A > H$.

You should be able to use the theorem to prove that $G < A$.

Can we compare the harmonic and geometric means? Yes. In fact, let's see how to compare an f-mean and a g-mean. Let F and G be the f-mean and g-mean of $a_1, \ldots, a_n$. Let $h(x) = f(g^{-1}(x))$ and look at the h mean of $b_1 = g(a_1)$, …, $b_n = g(a_n)$. Call this number H. Also, let A be the arithmetic mean of $b_1, \ldots, b_n$. By the definition (9),

$$\begin{aligned}
h(H) &= \frac{h(b_1) + \cdots + h(b_n)}{n} \\
&= \frac{h(g(a_1)) + \cdots + h(g(a_n))}{n} \\
&= \frac{f(g^{-1}((g(a_1)))) + \cdots + f(g^{-1}((g(a_n))))}{n} \\
&= \frac{f(a_1) + \cdots + f(a_n)}{n} \\
&= f(F).
\end{aligned}$$

Also,

$$h(A) = h\left(\frac{g(a_1) + \cdots + g(a_n)}{n}\right) = h(g(G)) = f(g^{-1}(g(G))) = f(G).$$

We've just shown that $h(A) = f(G)$ and $h(H) = f(F)$. Now use Theorem 1 to compare $h(H)$ and $h(A)$:

(a) If $h'' > 0$, then $h(A) < h(H)$. In other words, $f(G) < f(F)$.

(b) If $h'' < 0$, then $f(G) > f(F)$.

For the harmonic and geometric means, let $f(x) = 1/x$ and $g(x) = \ln x$. Then F is the harmonic mean and G is the geometric mean. Also, $g^{-1}(x) = e^x$ and $h(x) = 1/g^{-1}(x) = 1/e^x = e^{-x}$. Since $h''(x) = e^{-x} > 0$, (a) tells us that $f(G) < f(F)$; that is, $1/G < 1/F$, and so $G > F$ — the geometric mean is greater than the harmonic mean.

References

[1] Joseph DeStefano, Peter Doyle and J. Laurie Snell, The evil twin strategy for a football pool, *The American Mathematical Monthly*, **100** (1993) 341–343.

[2] Heikki Hämäläinen, Iiro Honkala, Simon Litsyn and Patric Östergård, Football pools—A game for mathematicians, *The American Mathematical Monthly*, **102** (1995) 579–588.

[3] Vera Pless, *Introduction to the Theory of Error Correcting Codes*, 2nd ed., J. Wiley and Sons (1990).