Yet Another Statistical Analysis
of the data of the (2015)

"Loophole-Free" Bell-CHSH Experiments (Part 1 of 3)

Richard D. Gill

Leiden University
Mathematical Institute

Combray
Causality Consultancy

http://www.math.leidenuniv.nl/~gill
http://richardgill.nl

¿QIRIF‽ Växjö, Wednesday 12 June 2019

Part 1

“Optimal” statistical data analysis of loophole-free experiments

Yet another statistical analysis of the data of the 'loophole free' experiments of 2015

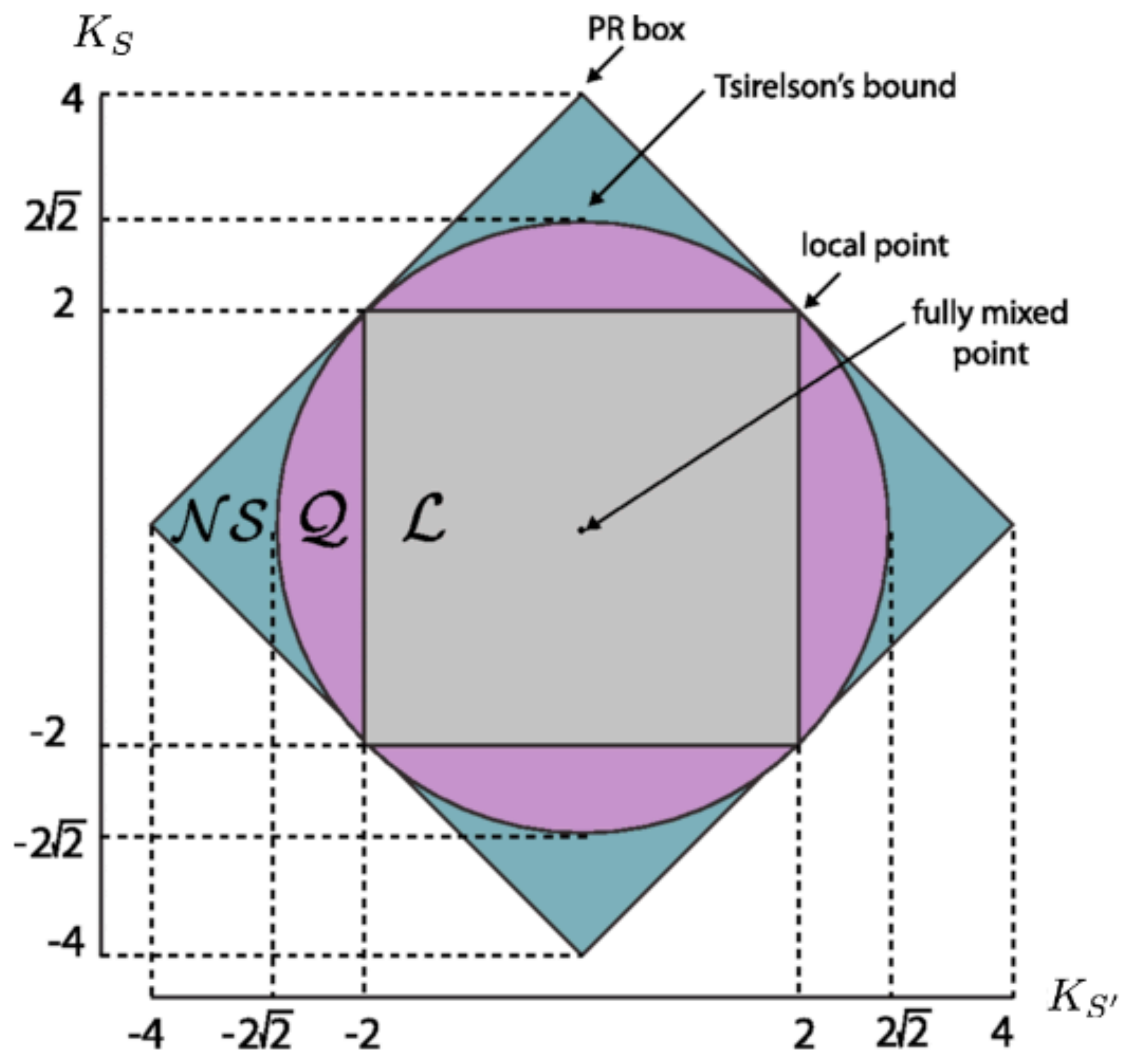
I present novel statistical analyses of the data of the famous Bell-inequality experiments of 2015 and 2016: Delft, NIST, Vienna and Munich. Every statistical analysis relies on statistical assumptions. I'll make the traditional, but questionable, i.i.d. assumptions. They justify a novel (?) analysis which is both simple and (close to) optimal.

It enables us to fairly compare the results of the two main types of experiments: NIST and Vienna CH-Eberhard "one-channel" experiment with settings and state chosen to optimise the handling of the detection loophole (detector efficiency > 66.7%); Delft and Munich CHSH "two channel" experiments based on entanglement swapping, with the state and settings which achieve the Tsirelson bound (detector efficiency ≈ 100%).

One cannot say which type of experiment is better without agreeing on how to compromise between the desires to obtain high statistical significance and high physical significance. Moreover, robustness to deviations from traditional assumptions is also an issue

The local polytope

- The local polytope of a 2x2x2 experiment has exactly 8 facets, A. Fine (1982).
- They are the 8 one-sided CHSH inequalities
- They are necessary and sufficient for LR.
  There are no other 2x2x2 inequalities!
- CH, Eberhard, J are therefore *just* different ways
  to write CHSH !
- Yet with experimental data they give different results !?



The diagram should be imagined as drawn on a plane in a higher dimensional space
The experimental data is a point close to, but not on, the plane

“One channel” experiment

VIENNA data

Raw counts

		Settings			
		11	12	21	22
Outcomes	dd	141.439	146.831	158.338	8.392
	dn	73.391	67.941	425.067	576.445
	nd	76.224	326.768	58.742	463.985
	nn	875.392.736	874.976.534	875.239.860	874.651.457
	Totals	875.683.790	875.518.074	875.882.007	875.700.279

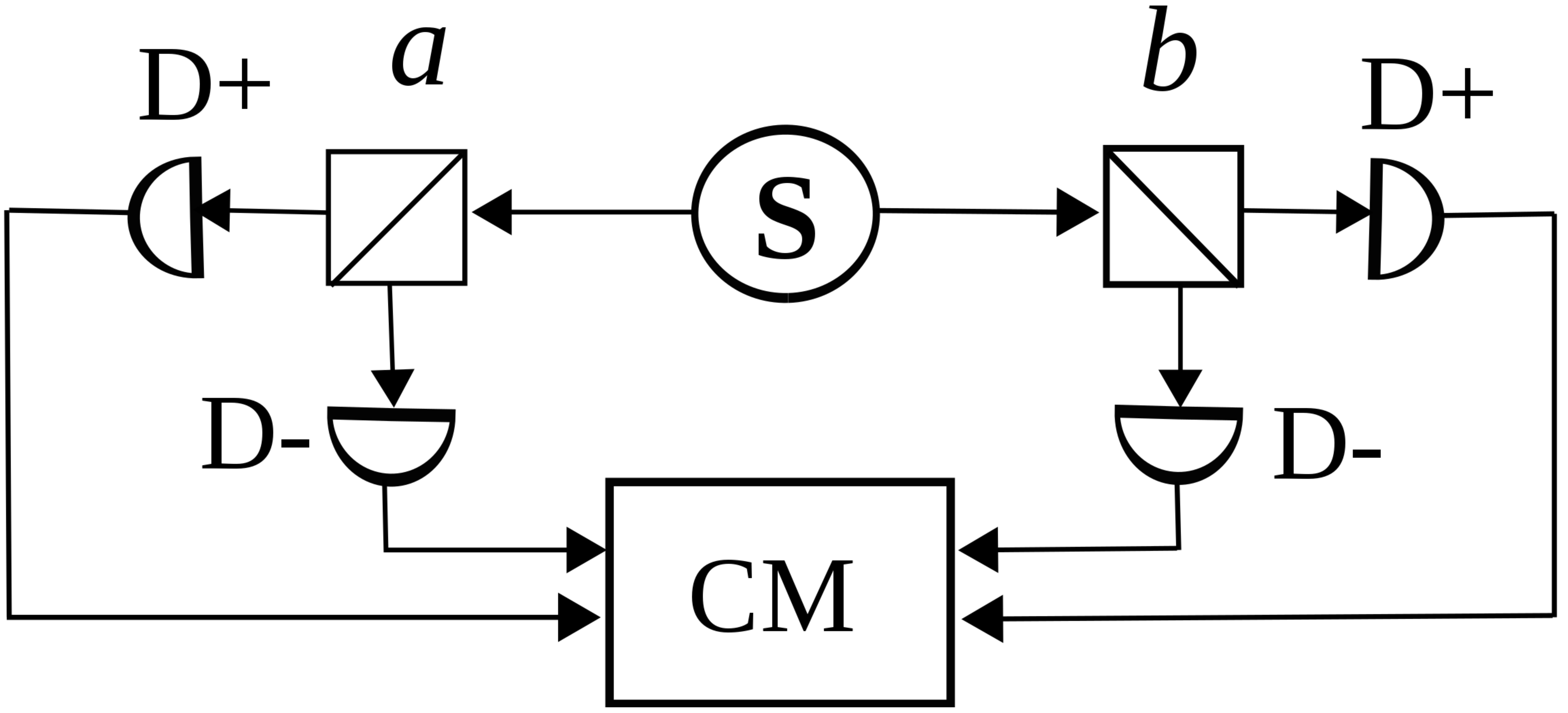
Normalised counts

		Settings			
		11	12	21	22
Outcomes	dd	162	168	181	10
	dn	84	78	485	658
	nd	87	373	67	530
	nn	999.668	999.381	999.267	998.802
	Totals	1.000.000	1.000.000	1.000.000	1.000.000

Normaliser		Normalised		
1.000.000	1.000.000	1.000.000	1.000.000	1.000.000

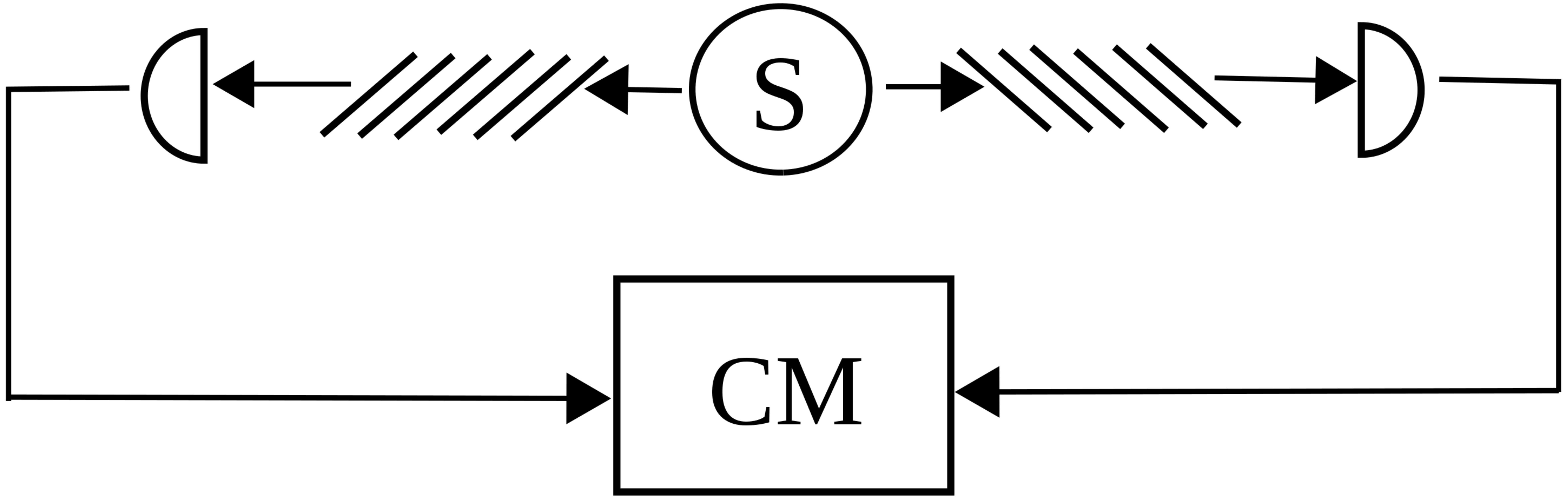
“d” = detection, “n” = no detection

“Two channel” experiment (CHSH - Aspect, Weihs, …, Delft, Munich)



Clocked experiment: outcomes on each side are “+”,”–“, or “0”

“One channel” experiment (Clauser-Horne, Eberhard, Vienna, NIST)



Outcomes on each side are “d” corresponding to “+” and “n” corresponding to “–” or “0”

Peter Bierhorst

Jan-Åke Larsson

$$S = 2 + 4\,J$$

$$J = (S - 2)/4$$

- The experiments in Vienna and at NIST (Boulder, Colorado) do *not* use the singlet state
- They exploit the fact that QM *can* violate CHSH from 66% detector efficiency
    - Clauser-Horne (1974)
    - Philippe H. Eberhard (1993) — **Experimental mathematics !!!**
    - Jan-Åke Larsson and Jason Semitecolos (2001) — **Proof !!!**
    - Peter Bierhorst (2016), “Geometric decompositions of Bell polytopes with practical applications”, *Journal of Physics A: Mathematical and Theoretical* — **Proof !!! (a very different one)**

Lawrence Berkeley National Laboratory

Location
Berkeley, United States

Department
Physics Division

Position
Retired Senior staff

Philippe Eberhard

Jason Semitecolos

P.H. Eberhard (1993)

The vector ψ turned out to be of the form

$$\psi = \frac{1}{2\sqrt{1+r^2}} \begin{vmatrix} (1+r)e^{-i\omega} \\ -(1-r) \\ -(1-r) \\ (1+r)e^{i\omega} \end{vmatrix} , \tag{31}$$

which can be reached in the two-photon experiment considered in this paper by first superposing states $|\leftrightarrow\updownarrow>$ and $|\updownarrow\leftrightarrow>$ in unequal amounts,

$$\psi_0 = (1/\sqrt{1+r^2})\left(|\leftrightarrow\updownarrow> + r\,|\updownarrow\leftrightarrow>\right) , \tag{32}$$

then rotating the planes of polarization of a and of b in setup (α_1, β_1) by the angles

$$\alpha_1 = (\omega/2) - 90^\circ , \tag{33}$$

$$\beta_1 = \omega/2 , \tag{34}$$

respectively, and using the values of r, ω, and $\alpha_1 - \alpha_2$ $(\equiv \beta_1 - \beta_2)$ given in Table II. Note that, for $\eta = 1$, the vector ψ_0 reduces to the value given by Eq. (1), and the angles α_1, α_2, β_1, and β_2 reduce to the values given by Eqs. (2)–(5).

TABLE II. Extreme conditions for a loophole-free experiment.

η (%)	ζ (%)	r	ω (deg)	$\alpha_1 - \alpha_2$ (deg)
66.7	0.00	0.001	0.0	2.2
70	0.02	0.136	3.4	21.4
75	0.31	0.311	9.7	32.0
80	1.10	0.465	14.9	37.9
85	2.48	0.608	18.6	41.5
90	4.50	0.741	20.9	43.6
95	7.12	0.871	22.1	44.7
100	10.36	1.000	22.5	45.0

Theoretical no-signalling probabilities, × 4 // Observed relative frequencies, × 10^6

	Outcomes	Bob Setting 1 “ d ”	Bob Setting 1 “ n “		Bob Setting 2 “ d ”	Bob Setting 2 “ n “	
Alice Setting 1	“ d ”	1 + a1 + b1 + z11	1 + a1 – b1 – z11	2 + 2 a1	1 + a1 + b2 + z12	1 + a1 – b2 – z12	2 + 2 a1
	“ n “	1 – a1 + b1 – z11	1 – a1 – b1 + z11	2 – 2 a1	1 – a1 + b2 – z12	1 – a1 – b2 + z12	2 – 2 a1
		2 + 2 b1	2 – 2 b1	4	2 + 2 b2	2 – 2 b2	4
Alice Setting 2	“ d ”	1 + a2 + b1 + z21	1 + a2 – b1 – z21	2 + 2 a2	1 + a2 + b2 + z22	1 + a2 – b2 – z22	2 + 2 a2
	“ n “	1 – a2 + b1 – z21	1 – a2 – b1 + z21	2 – 2 a2	1 – a2 + b2 – z22	1 – a2 – b2 + z22	2 – 2 a2
		2 + 2 b1	2 – 2 b1	4	2 + 2 b2	2 – 2 b2	4

10 ^ 6 * VIENNA

162	84	168	78
87	999668	373	999381
181	485	10	658
67	999267	530	998802

J = 27

S = CHSH = 2,000108

4 rho 11
= (2 + 2 z11) – (2 – 2 z11)
= 4 z11

4 S = 4 CHSH
= 4 (z11 + z12 + z21 – z22)

S = z11 + z12 + z21 – z22
= 2 + 4 J

J = (S - 2) / 4

VIENNA

4 J = (1 + a1 + b1 + z11)
– (1 – a2 + b1 – z21)
– (1 + a1 – b2 – z12)
– (1 + a2 + b2 + z22)
= – 2 + (z11 + z21 + z12 – z22)

$4\ \rho_{11} = (2 + 2 z_{11}) - (2 - 2 z_{11}) = 4 z_{11}$

$4 S = 4\ CHSH = 4 (z_{11} + z_{12} + z_{21} - z_{22})$

$$
\begin{aligned}
4 J = &(1 + a_1 + b_1 + z_{11}) \\
&- (1 - a_2 + b_1 - z_{21}) \\
&- (1 + a_1 - b_2 - z_{12}) \\
&- (1 + a_2 + b_2 + z_{22}) \\
= &- 2 + (z_{11} + z_{21} + z_{12} - z_{22})
\end{aligned}
$$

$S = z_{11} + z_{12} + z_{21} - z_{22} = 2 + 4 J$

$J = (S - 2) / 4$

Modern approach:
algebraic geometry, computer algebra

Also possible: amusing hybrid solutions
***Also* asymptotically optimal**

Estimation, standard errors, p-values

Routine MLE (Sir R.A. Fisher 1921…)

Log Lik = N(dd|11)log(1 + a1 +b1 +z11) +
… [15 more terms]

Parameters: a1 a2 b1 b2 z11 z12 z21 z22

Get mle of z11 + z21 + z12 – z22

Get estimated standard error of z11 + z21 + z12 – z22 from Fisher information matrix

Asymptotically optimal

[Linear constraints?]

Poor man's solution:
two stage, generalised, least squares
Asymptotically just as good as MLE!

Next ≈6 slides:
Statistical theory

A standard Bell-type experiment with

- two parties,
- two measurement settings per party,
- two possible outcomes per measurement setting per party,

generates a vector of $16 = 4 \times 4$ numbers of outcome combinations per setting combination.

This can be applied to the two-channel experiments with no "no-shows",
and to the one-channel experiments,
and to the two-channel experiments with "-" and "no-show" combined

The four sets of four counts can be thought of as four observations each of a multinomially distributed vector over four categories.

Write X_{ij} for the number of times outcome combination j was observed, when setting combination i was in force.

Let n_i be the total number of trials with the ith setting combination.

The four random vectors $\vec{X}_i = (X_{i1}, X_{i2}, X_{i3}, X_{i4})$, $i = 1, 2, 3, 4$,
are independent each with a Multinomial$(n_i; \vec{p}_i)$ distribution,
where $\vec{p}_i = (p_{i1}, p_{i2}, p_{i3}, p_{i4})$.

The 16 probabilities p_{ij} can be estimated by relative frequencies $\widehat{p}_{ij} = X_{ij}/n_i$ which have the following variances and covariances:

$$\operatorname{var}(\widehat{p}_{ij}) = p_{ij}(1 - p_{ij})/n_i,$$

$$\operatorname{cov}(\widehat{p}_{ij}, \widehat{p}_{ij'}) = -p_{ij}p_{ij'}/n_i \quad \text{for } j \neq j',$$

$$\operatorname{cov}(\widehat{p}_{ij}, \widehat{p}_{i'j'}) = 0 \quad \text{for } i \neq i'.$$

The variances and covariances can be arranged in a 16×16 block diagonal matrix Σ of four 4×4 diagonal blocks of non-zero elements.

Arrange the 16 estimated probabilities and their true values correspondingly in (column) vectors of length 16.

I will denote these simply by $\widehat{p}$ and p respectively.

We have $\mathrm{E}(\widehat{p}) = p \in \mathbb{R}^{16}$ and $\operatorname{cov}(\widehat{p}) = \Sigma \in \mathbb{R}^{16\times 16}$.

We are interested in the value of one particular linear combination of the p_{ij}, let us denote it by $\theta = a^\top p$.

We know that four other particular linear combinations are identically equal to zero: the so-called no-signalling conditions.

This can be expressed as $B^\top p = 0$ where the 16×4 matrix B contains, as its four columns, the coefficients of the four linear combinations.

We can sensibly estimate θ by $\widehat{\theta} = a^\top \widehat{p} - c^\top B^\top \widehat{p}$ where c is any vector of dimension 4. For whatever choice we make, $\mathrm{E}\widehat{\theta} = \theta$.

We propose to choose c so as to minimise the variance of the estimator. This minimization problem is a well-known problem from statistics and linear algebra ("least squares").

Define

$$\mathrm{var}(a^\top \widehat{p}) = a^\top \Sigma a =: \Sigma_{aa},$$
$$\mathrm{cov}(a^\top \widehat{p}, B^\top \widehat{p}) = a^\top \Sigma B =: \Sigma_{aB},$$
$$\mathrm{var}(B^\top \widehat{p}) = B^\top \Sigma B =: \Sigma_{BB};$$

then the optimal choice for c is

$$c_{\mathrm{opt}} := \Sigma_{aB} \Sigma_{BB}^{-1}$$

leading to the optimal variance

$$\Sigma_{aa} - \Sigma_{aB} \Sigma_{BB}^{-1} \Sigma_{Ba}.$$

In the experimental situation we do not know p in advance, hence also do not know Σ in advance. However we can estimate it in the obvious way ("plug-in") and for $n_i \to \infty$ we will have, just as in the previous section, an asymptotic normal distribution for our "approximately best" Bell inequality estimate, with an asymptotic variance which can be estimated by natural "plug-in" procedure, leading again to asymptotic confidence intervals, estimated standard errors, and so on.

The asymptotic width of **this** confidence intervals **is** the smallest possible and correspondingly the number of standard errors deviation from "local realism" the largest possible.

The fact that c is not known in advance does not harm these results.

"two stage (generalised) least squares"

Next ≈10 slides:
Work in progress: the practice

```
table11 <- matrix(c(141439,   73391,   76224,   875392736),
    2, 2, byrow = TRUE,
    dimnames = list(Alice = c("d", "n"), Bob = c("d", "n")))
table12 <- matrix(c(146831,   67941,   326768,   874976534),
    2, 2, byrow = TRUE,
    dimnames = list(Alice = c("d", "n"), Bob = c("d", "n")))
table21 <- matrix(c(158338,   425067,   58742,   875239860),
    2, 2, byrow = TRUE,
    dimnames = list(Alice = c("d", "n"), Bob = c("d", "n")))
table22 <- matrix(c(8392,   576445,   463985,   874651457),
    2, 2, byrow = TRUE,
    dimnames = list(Alice = c("d", "n"), Bob = c("d", “n")))
```

```
table11

##       Bob
## Alice      d         n
##     d 141439     73391
##     n  76224 875392736

table12

##       Bob
## Alice      d         n
##     d 146831     67941
##     n 326768 874976534

table21

##       Bob
## Alice      d         n
##     d 158338    425067
##     n  58742 875239860
```

table22

```
##      Bob
## Alice      d         n
##   d   8392    576445
##   n 463985 874651457
```

```r
tables <- cbind(as.vector(t(table11)), as.vector(t(table12)),
                as.vector(t(table21)), as.vector(t(table22)))
tables
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,]    141439    146831    158338      8392
## [2,]     73391     67941    425067    576445
## [3,]     76224    326768     58742    463985
## [4,] 875392736 874976534 875239860 874651457
```

```r
dimnames(tables) = list(outcomes = c("dd", "dn", "nd", "nn"),
                        settings = c(11, 12, 21, 22))
```

```r
tables
```

```
##          settings
## outcomes         11         12         21         22
##       dd     141439     146831     158338       8392
##       dn      73391      67941     425067     576445
##       nd      76224     326768      58742     463985
##       nn  875392736  874976534  875239860  874651457
```

```r
Ns <- apply(tables, 2, sum)
Ns
```

```
##        11        12        21        22
## 875683790 875518074 875882007 875700279
```

```r
rawProbsMat <- tables / outer(rep(1,4), Ns)
rawProbsMat
```

```
##          settings
## outcomes            11            12            21            22
##       dd 1.615183e-04 1.677076e-04 1.807755e-04 9.583188e-06
##       dn 8.380993e-05 7.760091e-05 4.853017e-04 6.582675e-04
##       nd 8.704512e-05 3.732282e-04 6.706611e-05 5.298445e-04
##       nn 9.996676e-01 9.993815e-01 9.992669e-01 9.988023e-01
```

```r
VecNames <- as.vector(t(outer(colnames(rawProbsMat),
                              rownames(rawProbsMat), paste, sep = "")))
VecNames
```

```
##  [1] "11dd" "11dn" "11nd" "11nn" "12dd" "12dn" "12nd" "12nn" "21dd" "21dn"
## [11] "21nd" "21nn" "22dd" "22dn" "22nd" "22nn"
```

```r
rawProbsVec <- as.vector(rawProbsMat)
names(rawProbsVec) <- VecNames
VecNames
```

```
##  [1] "11dd" "11dn" "11nd" "11nn" "12dd" "12dn" "12nd" "12nn" "21dd" "21dn"
## [11] "21nd" "21nn" "22dd" "22dn" "22nd" “22nn"
```

```r
rawProbsVec
```

```
##         11dd         11dn         11nd         11nn         12dd
## 1.615183e-04 8.380993e-05 8.704512e-05 9.996676e-01 1.677076e-04
##         12dn         12nd         12nn         21dd         21dn
## 7.760091e-05 3.732282e-04 9.993815e-01 1.807755e-04 4.853017e-04
##         21nd         21nn         22dd         22dn         22nd
## 6.706611e-05 9.992669e-01 9.583188e-06 6.582675e-04 5.298445e-04
##         22nn
## 9.988023e-01
```

```r
Aplus <- c(1, 1, 0, 0)
Aminus <- - Aplus
Bplus <- c(1, 0, 1, 0)
Bminus <- - Bplus
zero <- c(0, 0, 0, 0)
NSa1 <- c(Aplus, Aminus, zero, zero)
NSa2 <- c(zero, zero, Aplus, Aminus)
NSb1 <- c(Bplus, zero, Bminus, zero)
NSb2 <- c(zero, Bplus, zero, Bminus)
NS <- cbind(NSa1 = NSa1, NSa2 = NSa2, NSb1 = NSb1, NSb2 = NSb2)
rownames(NS) <- VecNames
```

```
NS

##      NSa1 NSa2 NSb1 NSb2
## 11dd    1    0    1    0
## 11dn    1    0    0    0
## 11nd    0    0    1    0
## 11nn    0    0    0    0
## 12dd   -1    0    0    1
## 12dn   -1    0    0    0
## 12nd    0    0    0    1
## 12nn    0    0    0    0
## 21dd    0    1   -1    0
## 21dn    0    1    0    0
## 21nd    0    0   -1    0
## 21nn    0    0    0    0
## 22dd    0   -1    0   -1
## 22dn    0   -1    0    0
## 22nd    0    0    0   -1
## 22nn    0    0    0    0
```

```r
cov11 <- diag(rawProbsMat[ , "11"]) - outer(rawProbsMat[ , "11"], rawProbsMat[ , "11"])
cov12 <- diag(rawProbsMat[ , "12"]) - outer(rawProbsMat[ , "12"], rawProbsMat[ , "12"])
cov21 <- diag(rawProbsMat[ , "21"]) - outer(rawProbsMat[ , "21"], rawProbsMat[ , "21"])
cov22 <- diag(rawProbsMat[ , "22"]) - outer(rawProbsMat[ , "22"], rawProbsMat[ , "22"])
Cov <- matrix(0, 16, 16)
rownames(Cov) <- VecNames
colnames(Cov) <- VecNames
Cov[1:4, 1:4] <- cov11/Ns["11"]
Cov[5:8, 5:8] <- cov12/Ns["12"]
Cov[9:12, 9:12] <- cov21/Ns["21"]
Cov[13:16, 13:16] <- cov22/Ns["22"]
J <- c(c(1, 0, 0 ,0), - c(0, 1, 0 ,0), - c(0, 0, 1, 0), - c(1, 0, 0, 0))
names(J) <- VecNames
sum(J * rawProbsVec)

## [1] 7.26814e-06

varJ <- t(J) %*% Cov %*% J
covNN <- t(NS) %*% Cov %*% NS
covJN <- t(J) %*% Cov %*% NS
covNJ <- t(covJN)
```

```r
## Estimated variance of optimal test based on J
varJ - covJN %*% solve(covNN) %*% covNJ
```

```
##               [,1]
## [1,] 1.594636e-13
```

```r
## Estimated variance of Eberhard's J
varJ
```

```
##               [,1]
## [1,] 3.605539e-13
```

```r
sqrt(varJ / (varJ - covJN %*% solve(covNN) %*% covNJ))
```

```
##          [,1]
## [1,] 1.503676
```

```r
covJN %*% solve(covNN)
```

```
##          NSa1       NSa2      NSb1       NSb2
## [1,] 0.395483 0.05436871 0.3516065 0.06982674
```

```r
Jopt <- J - covJN %*% solve(covNN) %*% t(NS)
```

```
> J <- 0.00000726814
> J
[1] 7.26814e-06
> varJ <- 3.605539e-13
> J / sqrt(varJ)
[1] 12.10426
> pnorm(- J / sqrt(varJ))
[1] 5.013575e-34
```

```
Jopt

##            11dd      11dn        11nd 11nn      12dd      12dn         12nd
## [1,] 0.2529105 -0.395483 -0.3516065    0 0.3256562 -0.604517 -0.06982674
##      12nn      21dd        21dn       21nd 21nn       22dd       22dn
## [1,]    0 0.2972378 -0.05436871 -0.6483935    0 -0.8758045 0.05436871
##             22nd 22nn
## [1,] 0.06982674    0
```

```
> Jhatopt <- 7.26814e-06
> varJhatopt <- varJ/2.261042
> Jhatopt / sqrt(varJhatopt)
[1] 18.20088
> pnorm(- Jhatopt / sqrt(varJhatopt))
[1] 2.539047e-74
```

```
sum(J * rawProbsVec)

## [1] 7.26814e-06
```

```
sum(Jopt * rawProbsVec)

## [1] 6.997615e-06
```

```
varJ / (varJ - covJN %*% solve(covNN) %*% covNJ)

##          [,1]
## [1,] 2.261042
```

```
(varJ - covJN %*% solve(covNN) %*% covNJ) / varJ

##          [,1]
## [1,] 0.442274
```

```
sqrt( (varJ - covJN %*% solve(covNN) %*% covNJ) / varJ )

##           [,1]
## [1,] 0.6650368
```