

# Inference after model selection



Joshua Loftus (jloftus@turing.ac.uk)

2 December, 2016

Slides and markdown source at

<https://jloftus.github.io/turing/studenttalk/>

# Setting: regression model selection

## Linear model

$$y = X\beta + \epsilon$$

- $y$  vector of outcomes
- $X$  predictor/feature matrix
- $\beta$  parameters/weights to be estimated, assume most are “null,” i.e. equal 0 (sparsity)
- $\epsilon$  random errors, assume probability distribution  $N(0, \sigma^2 I)$
- Pick subset of predictors we think are non-null
- How good is the model using this subset?
- Are chosen predictors actually non-null, i.e. significant?

# Brief refresher on $p$ -values

- Common measure of “statistical significance”
- Due to Sir R. A. Fisher. (J. Leek and R. Irizarry estimated **3 million** citations, most highly cited research publication of all time)
- Assume null hypothesis is true, then how extreme is the data we observed?

## $p$ -value definition

$$P_{H_0}(|T| > |t_{\text{obs}}|)$$

**Type 1 error:** declaring a predictor significant when it is actually null.

# How do we use $p$ -values?

Under  $H_0$ , if  $p \sim \text{Unif}[0, 1]$  then...

<u>P-VALUE</u>	<u>INTERPRETATION</u>
0.001	HIGHLY SIGNIFICANT
0.01	
0.02	
0.03	
0.04	SIGNIFICANT
0.049	
0.050	OH CRAP. REDO CALCULATIONS.
0.051	ON THE EDGE OF SIGNIFICANCE
0.06	
0.07	HIGHLY SUGGESTIVE, SIGNIFICANT AT THE $P < 0.10$ LEVEL
0.08	
0.09	
0.099	HEY, LOOK AT THIS INTERESTING SUBGROUP ANALYSIS
$\geq 0.1$	

## Motivating example: forward stepwise

Data: California county health data...

Outcome: log-years of potential life lost.

Model: 5 out of 30 predictors chosen by FS with AIC.

```
model <- step(lm(y ~ .-1, df), k = 2, trace = 0)
print(summary(model)$coefficients[,c(1,4)], digits = 2)
```

##	Estimate	Pr(> t )
## Food.Environment.Index	0.342	0.0296
## `%.With.Access`	-0.036	0.0017
## `%.Excessive.Drinking`	0.090	0.0182
## Teen.Birth.Rate	0.026	0.0045
## Average.Daily.PM2.5	-0.225	0.0211

5 interesting effects, all significant. Time to publish!

What's wrong with this?

# What's wrong with this?

**The outcome was actually just noise, independent of the predictors**

```
set.seed(1)
df = read.csv("CaliforniaCountyHealth.csv")
df$y <- rnorm(nrow(df)) #!!!
```

(With apologies for deceiving you, I hope this makes the point...)

# What is going wrong?

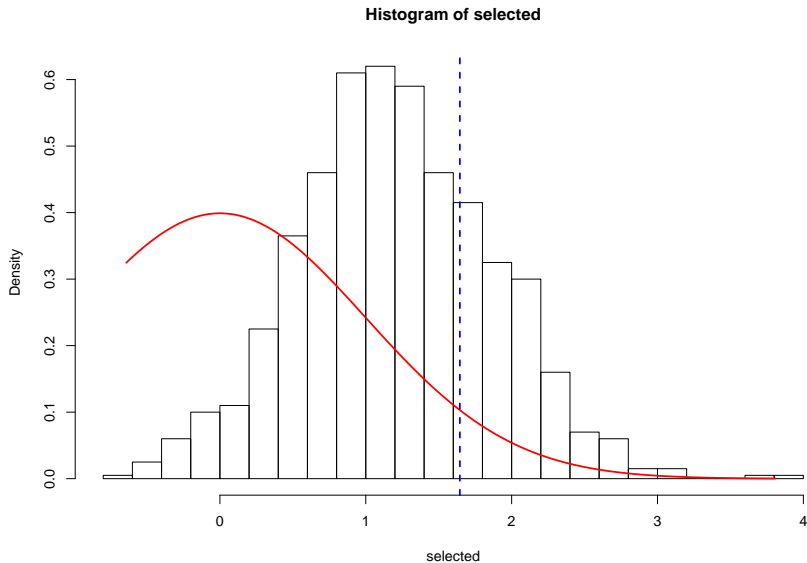
Simple example: choose the largest of 5 effects.

```
maxz <- function(n) return(max(rnorm(n)))  
selected <- replicate(1000, maxz(5))  
range <- seq(min(selected),  
             max(selected),  
             length.out = 1000)
```

This data is generated under a global null, all effects have 0 mean. What happens if we now compute  $p$ -values for the selected effects using standard normal tail areas?



# Selection bias: type 1 error 0.256 instead of 0.05



# Selection can make noise look like signal

Any time we use the data to make a decision (e.g. pick one model instead of some others), we introduce a selection effect (bias).

# Selection can make noise look like signal

Any time we use the data to make a decision (e.g. pick one model instead of some others), we introduce a selection effect (bias).

Forward stepwise, Lasso, elastic net with cross-validation, etc, all use the data in a way that would result in such bias.

# Selection can make noise look like signal

Any time we use the data to make a decision (e.g. pick one model instead of some others), we introduce a selection effect (bias).

Forward stepwise, Lasso, elastic net with cross-validation, etc, all use the data in a way that would result in such bias.

Significance tests, prediction error,  $R^2$ , goodness of fit tests, etc, will all suffer from selection bias

# Big contributor to reproducibility crisis

***We conducted replications** of 100 experimental and correlational studies published in three psychology journals using high-powered designs and original materials when available. . . . Thirty-six percent of replications had significant results; 47% of original effect sizes were in the 95% confidence interval of the replication effect size; **39% of effects were subjectively rated to have replicated the original result***

From *Estimating the reproducibility of psychological science* (Open Science Collaboration, 2015). See also *Why most published research findings are false* (Ioannidis, 2005).

What's the most common solution?

# What's the most common solution?

## Data splitting

Before doing any selection, set aside some **validation data**. Then, *after* the final model is chosen, use this validation set to compute prediction error, significance tests, etc.

# Survival example: Cox's PH model, regularized

- Data: 240 lymphoma patients, 7399 genes

```
train <- sample(nrow(x), 140)
x.train <- x[train,]
y.train <- Surv(y[train], status[train])
fit <- glmnet(x.train, y.train, family = "cox")
cv.fit <- cv.glmnet(x.train, y.train,
                    family = "cox")
coefs <- coef(fit, s = cv.fit$lambda.min)
active <- which(coefs != 0)
length(active)
```

```
## [1] 15
```



## Inference for the selected model

```
test <- setdiff(1:nrow(x), train)
x.test <- x[test, active]
y.test <- Surv(y[test], status[test])
fit.test <- coxph(y.test ~ x.test)
fit.test
```

```
## Call:
```

```
## coxph(formula = y.test ~ x.test)
```

```
##
```

##		coef	exp(coef)	se(coef)	z	p
##	x.test1	-0.2730	0.7611	0.2096	-1.30	0.193
##	x.test2	0.6954	2.0045	0.4394	1.58	0.114
##	x.test3	0.1218	1.1295	0.3748	0.32	0.745
##	x.test4	-0.0145	0.9856	0.3038	-0.05	0.962
##	x.test5	0.0755	1.0784	0.1918	0.39	0.694
##	x.test6	-0.1430	0.8668	0.0648	-2.21	0.027

# Data splitting

## Pros:

- Simple: only took a few lines of code
- Robust: requires few assumptions
- Controls type 1 error, no selection bias

# Data splitting

## Pros:

- Simple: only took a few lines of code
- Robust: requires few assumptions
- Controls type 1 error, no selection bias

## Cons:

- Reproducibility issues: different random splits, different split proportions
- Efficiency: using less data for model selection, also less power
- Feasibility: categorical variables with rare levels (e.g. rare variants)

# Selective error control

New and active research area; Taylor, Tibshirani, Fithian, many others. To adjust for the selection effect, *condition* on the selected model. Mathematically, if we select  $M$ , want test a null hypothesis  $H_0$  about  $M$  (e.g. significance test for a variable in  $M$ ), we want tests that control

## Selective type 1 error

$$P_{M, H_0}(\text{reject } H_0 | M \text{ selected}) \leq \alpha$$

# Selective error control

New and active research area; Taylor, Tibshirani, Fithian, many others. To adjust for the selection effect, *condition* on the selected model. Mathematically, if we select  $M$ , want test a null hypothesis  $H_0$  about  $M$  (e.g. significance test for a variable in  $M$ ), we want tests that control

## Selective type 1 error

$$P_{M, H_0}(\text{reject } H_0 | M \text{ selected}) \leq \alpha$$

If a variable “surprises” us enough to be *included in the model*, it must surprise us *again* in order to be *declared significant*

- Data splitting controls this error trivially
- Controlling this would fix reproducibility problems

## A simple example: marginal screening rule

Observe many (independent) means, select the effects which might be large, say  $> 1$

```
Zs <- rnorm(10000)
screen <- Zs > 1
Zscreened <- Zs[screen]
mean(screen)
```

```
## [1] 0.1637
```

## A simple example: marginal screening rule

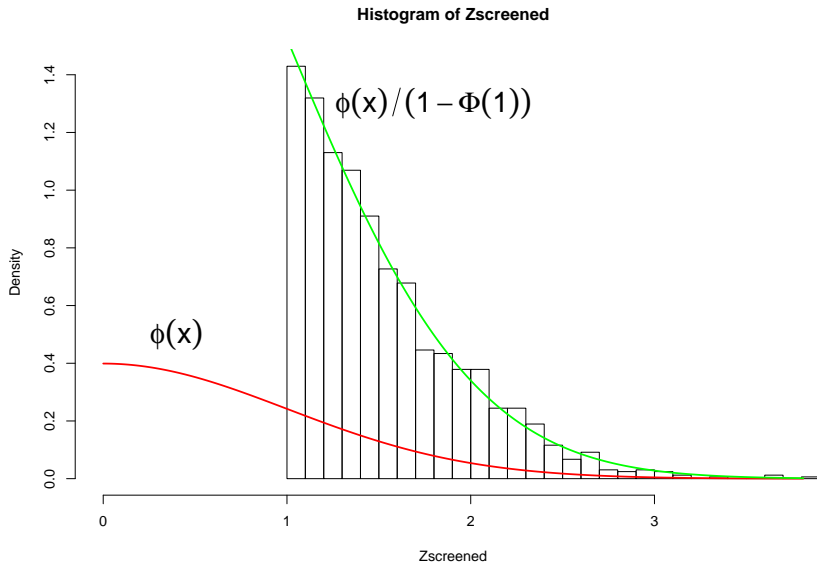
Observe many (independent) means, select the effects which might be large, say  $> 1$

```
Zs <- rnorm(10000)
screen <- Zs > 1
Zscreened <- Zs[screen]
mean(screen)
```

```
## [1] 0.1637
```

These are all null. What distribution can we compare them to, and still control type 1 error?

# Truncated probability law





## Details vary depending on procedure

Most of the work in this field is in understanding the geometry of truncation for each particular selection procedure

My work focuses on procedures with complicated geometry (quadratic constraints), and includes a few important special cases (groups of variables, cross-validation)

## groupfs simulation: $n = 100$ , $p = 100$ , $P = 50$

Model size chosen with BIC. Groups 1-4 have  $\|\beta\|_2 = .84$  within groups, all else 0

```
> set.seed(1)
```

```
...
```

```
> fit <- groupfs(x, y, ...)
```

```
> pvals <- groupfsInf(fit)
```

```
> pvals
```

	Group	Pvalue	TF	df	Size	Ints	Min	Max
1	3	0.088	49.913	2	67.811	1	44.949	112.760
2	1	0.000	98.077	1	54.267	1	68.151	122.418
3	2	0.003	69.266	1	28.659	1	50.423	79.082
4	4	0.000	37.099	2	28.803	1	20.194	48.997
5	47	0.319	5.143	1	3.887	1	3.518	7.406

Ignoring selection, first 4  $p$ -values are  $< 0.001$ , for 47 it's 0.024

# Remarks

Technical details in the papers, but beware:

- Tests not independent (with one notable exception)
- Some examples are computationally expensive (cross-validation)
- May be low powered against some alternatives

Software implementation: `selectiveInference` R package on CRAN

Github repo: <https://github.com/selective-inference/>

# Which method to use for a given problem?

- If  $n$  is very large, might just use data splitting (simple)
- Otherwise, consider the conditional approach, especially if  $p > n$  or bottlenecks like rare observations limit effective sample size
- If  $p$  is small, more robust/conservative method (“PoSI”) is available, see Berk et. al. (2013).

# Most general takeaway message

## **Selection** is a source of uncertainty

Data science pipelines must **address all sources of uncertainty**.  
Otherwise we might just be fooling ourselves. . .

# Some details of my research

- Model selection map  $M : \mathbb{R}^n \rightarrow \mathcal{M}$ , with  $\mathcal{M}$  space of potential models.
- Observe  $E_m = \{M(y) = m\}$ , want to condition on this event.
- For many model selection procedures

$$\underbrace{\mathcal{L}(y|M(y) = m)}_{\text{what we want}} = \mathcal{L}(y|\underbrace{A(m)y \leq b(m)}_{\text{simple geometry}}) \quad \text{on } \{M(y) = m\}$$

MVN constrained to a polytope.

# Quadratic model selection framework

For some model selection procedures (e.g. forward stepwise with groups, cross-validation), event can be decomposed as

## Quadratic selection event

$$E_m := \{M(y) = m\} = \bigcap_{j \in J_m} \{y : y^T Q_j y + a_j^T y + b_j \geq 0\}$$

- These  $Q, a, b$  are constant on  $E_m$ , so conditionally they are constants
- For conditional inference, need to compute this intersection of quadratics

# Geometry problem: intersection of quadratic regions

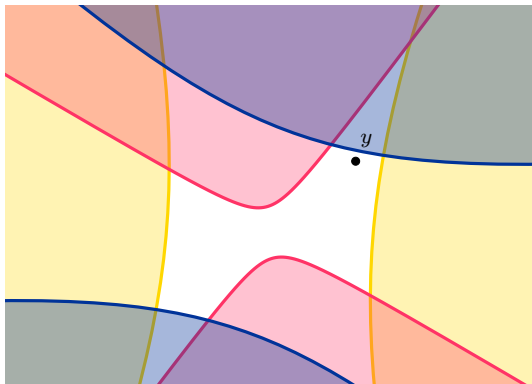


Figure 1: The *complement* of each quadratic is shaded with a different color. The unshaded, white region is  $E_m$ .



# Geometry problem: intersection of quadratic regions

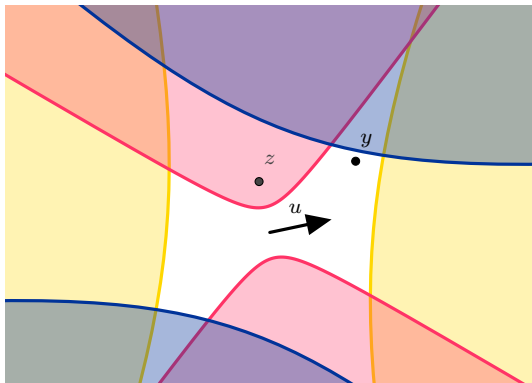


Figure 1: The *complement* of each quadratic is shaded with a different color. The unshaded, white region is  $E_m$ .

# Geometry problem: intersection of quadratic regions

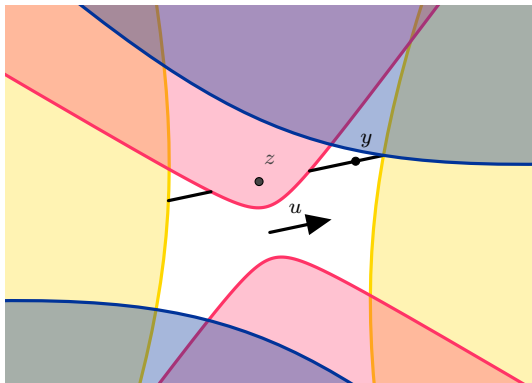


Figure 1: The *complement* of each quadratic is shaded with a different color. The unshaded, white region is  $E_m$ .

# Geometry problem: intersection of quadratic regions

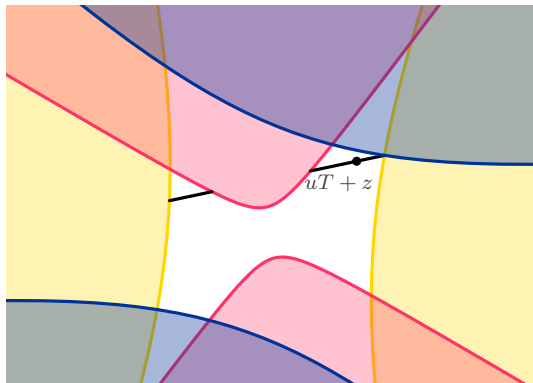


Figure 1: The *complement* of each quadratic is shaded with a different color. The unshaded, white region is  $E_m$ .

## Reduction to one-dimension, $\sigma^2$ known

Test statistics have form  $T^2 = \|Py\|_2^2$ , with  $P$  a projection matrix

Write  $y = Py + z$  where  $z = (I - P)y$ ,  $u = Py/\|Py\|_2$

Find  $S_m = \{t \geq 0 : tu + z \in E_m\} \subseteq \mathbb{R}$  by solving for  $t$  in

$$(tu + z)^T Q_j(tu + z) + a_j^T(tu + z) + b_j \geq 0$$

### One-dimensional test (recall $y \sim N(\mu, \sigma^2)$ )

For a fixed  $P$ , under  $H_0 : P\mu = 0$  we have  $T^2 \sim \sigma^2 \chi_r^2$ , where  $r = \text{rank}(P)$ .

Conditional on  $E_m, z$ , and  $u$ , only remaining variation is  $T = \|Py\|_2$  with truncated support  $S_m$ .

# Pros and cons of this approach

- Can also do  $\sigma^2$  unknown case
- Model selection procedures like cross-validation (previously unsolved)
- Can be computationally expensive (particularly cross-validation)
- Most usual limitations of model selection still apply

# References

- Taylor, Tibshirani (2015). Statistical learning and selective inference. **PNAS**.
- Benjamini, (2010). Simultaneous and selective inference: current successes and future challenges. Biometrical Journal.
- Berk et al, (2010). Statistical inference after model selection. Journal of Quantitative Criminology.
- Berk et al, (2013). Valid post-selection inference. Annals of Statistics.
- Simon et al, (2011). Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. Journal of Statistical Software.
- Loftus, (2015). Selective inference after cross-validation. arXiv Preprint.
- Loftus and Taylor, (2015). Selective inference in regression models with groups of variables. arXiv Preprint.

# Thanks for your attention!

## Questions?

Message ([jloftus@turing.ac.uk](mailto:jloftus@turing.ac.uk)) or find me at the Alan Turing Institute.