Post-selection inference for models characterized by quadratic constraints

The Alan Turing Institute

UNIVERSITY OF CAMBRIDGE Statistical Laboratory

Joshua Loftus (jloftus@turing.ac.uk)

6 December, 2016

Slides and markdown source at https://joftius.github.io/turing

Setting: regression model selection

Linear model

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

- y vector of outcomes
- X predictor/feature matrix
- β parameters/weights to be estimated, assume most are "null," i.e. equal 0 (sparsity)
- ϵ 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?

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

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. . .)

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)

Related work

There has been much recent work in this area. Following Berk et al. (2013), we consider two categories:

Full-model: inference about the parameters β_j

In the (possibly undetermined/singular) linear model

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

Sub-model: inference about the parameters $\beta(A_0)_j$

In the (sparse, nonsingular) linear model

$$y = X(A_0)\beta(A_0) + \epsilon$$

for some $A_0 \subset \{1, \ldots, p\}$.

Inference in the full model $\mu = X\beta$

FDR control or similar

- Screen & clean Wasserman and Roeder (2009)
- Stability selection Meinshausen and Bühlmann (2010)
- Empirical Bayes Efron (2011)
- SLOPE Bogdan et al. (2014)
- Knockoffs Barber and Candès (2015)

Type 1 error

- Univariate treatment Belloni, Chernozhukov, and Hansen (2014)
- Debiasing methods Bühlmann (2013), Javanmard and Montanari (2014), Zhang and Zhang (2014)

Inference in the sub-model $\mu = X(A_0)\beta(A_0)$

- PoSI: simultaneous inference — Berk et al. (2013)
- Selective inference, FCR — Benjamini & Yekutieli (2005)
- Answer must be valid given that the question was asked
- Conditional approach: conditions the model selection event and uses corresponding truncated probability distributions

Literature on the conditional approach

- Frequentist interpretation Hurvich & Tsai (1990)
- Lasso, sequential Lockhart et al. (2014)
- General penalty, global null, geometry Taylor, Loftus, and Tibshirani (2015), Azaïs, Castro, and Mourareau (2015)
- Forward stepwise, sequential Loftus and Taylor (2014)
- Fixed λ Lasso / conditional Lee et al. (2015), Fithian, Sun, and Taylor (2014)
- Forward stepwise and LAR Tibshirani et al. (2014)
- Asymptotics Tian and Taylor (2015a)
- Unknown σ Tian, Loftus, and Taylor (2015), Gross, Taylor, and Tibshirani (2015)
- Group selection / unknown σ Loftus and Taylor (2015)
- Cross-validation Tian and Taylor (2015b), Loftus (2015)
- Unsupervised learning Blier, Loftus, and Taylor (2016)

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, 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, 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

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

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

A simple example: marginal screening rule

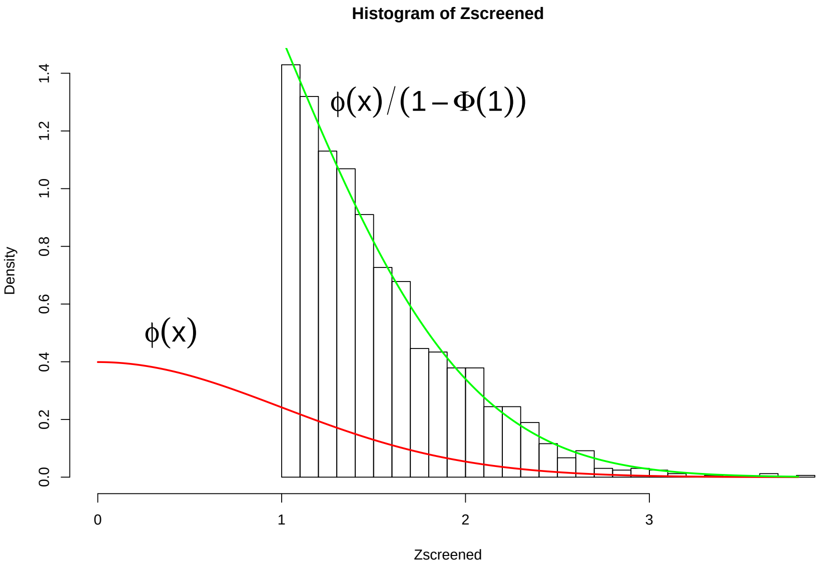
Observe many (independent) means, select the effects which might be large, say > 1

```r
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)

Previous work: affine model selection

- Model selection map $M : \mathbb{R}^n \to \mathcal{M}$, 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

Truncated χ significance test

Suppose $y \sim N(\mu, \sigma^2 I)$ with σ^2 known, $H_0(m) : P_m\mu = 0$, P_m is constant on $\{M(y) = m\}$, $r := \mathrm{Tr}(P_m)$, $R := P_m y$, $u := R/\|R\|_2$, $z := y - R$, $D_m := \{t \geq 0 : M(ut\sigma + z) = m\}$, and the observed statistic $T = \|R\|_2/\sigma$

Post-selection $T\chi$ distribution

$$T|(m, z, u) \sim \chi_r|_{D_m} \tag{1}$$

where the vertical bar denotes truncation. Hence, with f_r the pdf of a central χ_r random variable

$$T\chi := \frac{\int_{D_m \cap [T,\infty]} f_r(t)dt}{\int_{D_m} f_r(t)dt} \sim U[0, 1] \tag{2}$$

is a p-value controlling selective type 1 error.

Geometry problem: intersection of quadratic regions

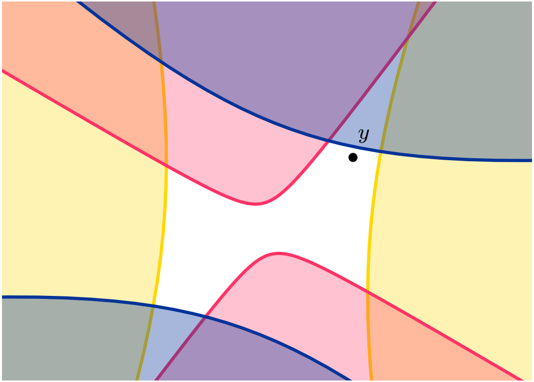


Figure 1: The *complement* of each quadratic is shaded with a different color. The unshaded, white region is E_m.

Geometry problem: intersection of quadratic regions

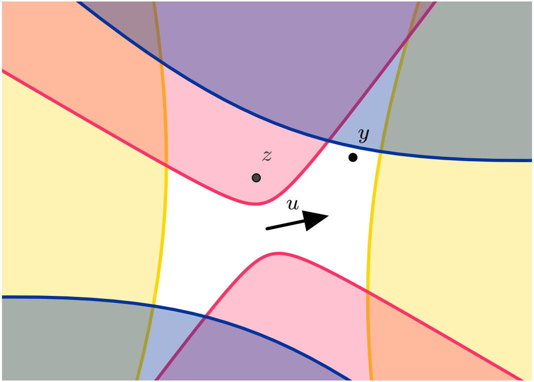


Figure 1: The *complement* of each quadratic is shaded with a different color. The unshaded, white region is E_m.

Geometry problem: intersection of quadratic regions

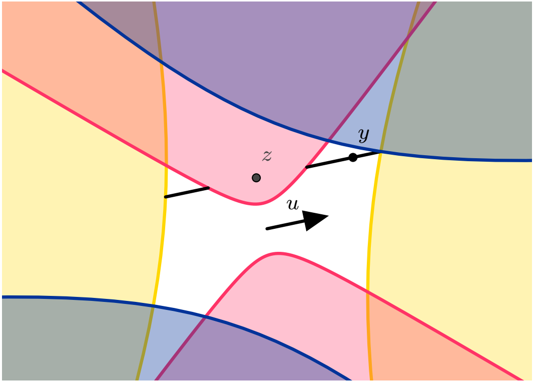


Figure 1: The *complement* of each quadratic is shaded with a different color. The unshaded, white region is E_m.

Geometry problem: intersection of quadratic regions

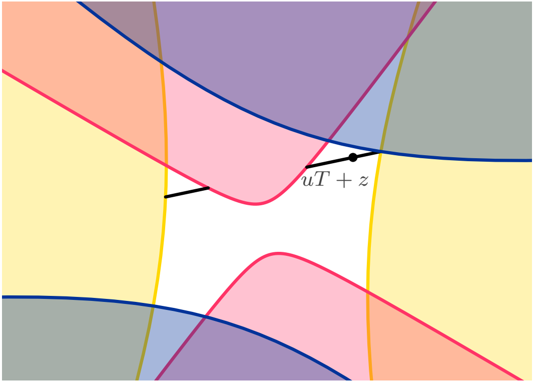


Figure 1: The *complement* of each quadratic is shaded with a different color. The unshaded, white region is E_m.

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

Adaptive model selection with cross-validation

- For K-fold cv, data partitioned (randomly) into $D_1, \dots, D_K$. For each $k = 1, \dots, K$, hold out D_k as a test set while training a model on the other $K - 1$ folds. Form estimate RSS_k of out-of-sample prediction error. Average these estimates over test folds.
- Use to choose model complexity: evaluate $RSS_{k,s}$ for various sparsity choices s. Pick s minimizing the cv-RSS estimate.
- Run forward stepwise with maxsteps S. For $s = 1, \dots, S$ evaluate the test error $RSS_{k,s}$. Average to get RSS_s. Pick s^* minimizing this. Run forward stepwise on the whole data for s^* steps.

Can we do selective inference for the final models chosen this way?

Notation for cross-validation

- Let f, g index CV test folds.
- On fold f, model m_f at step s, and $-f$ denoting the training set for test fold f (complement of f).
- Define $P_{f,s} := X^{f}_{m_f,s}(X^{-f}_{m_f,s})^{\dagger}$ (not a projection)
- $s = \text{argmin}_s \sum_{f=1}^{K} \|y^f - P_{f,s}y^{-f}\|_2^2$
- Sums of squares… maybe it's a quadratic form?

Blockwise quadratic form of cv-RSS

Key result of Loftus (2015).

Define $Q^s_{ff} := \sum_{g \neq f} (P_{g,s})^T_f (P_{g,s})_f$ and

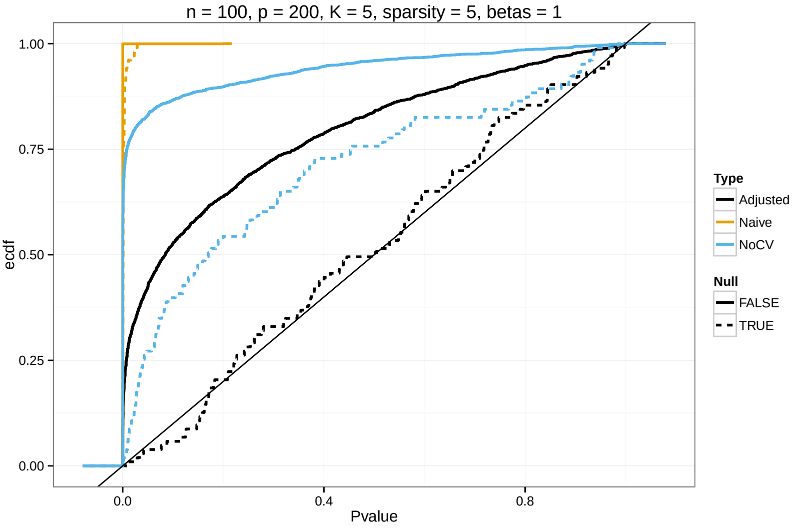
$$Q^s_{fg} := -(P_{f,s})_g - (P_{g,s})^T_f + \sum_{\substack{h=1 \\ h \notin \{f,g\}}}^{K} (P_{h,s})^T_f (P_{h,s})^T_g$$

Then with y_K denoting the observations ordered by CV-folds,

$$\text{cv-RSS}(s) = y_K^T Q^s y_K$$

This quadratic form allows us to conduct inference conditional on models selected by cross-validation

Simulation: empirical CDF



Remarks

Technical details in the papers, a few notes:

- Tests not independent (with one notable exception)
- Some examples are computationally expensive (cross-validation)
- May be low powered against some alternatives
- Can also do σ^2 unknown case
- Most usual limitations of model selection still apply

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. . .

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?

jloftus@turing.ac.uk or visit me at the Alan Turing Institute.

More references

Azaïs, Jean-Marc, Yohann de Castro, and Stéphane Mourareau. 2015. "Power of the Kac-Rice Detection Test." *ArXiv Preprint ArXiv:1503.05093*.

Barber, Rina Foygel, and Emmanuel J. Candès. 2015. "Controlling the False Discovery Rate via Knockoffs." *Ann. Statist.* 43 (5). The Institute of Mathematical Statistics: 2055–85. doi:10.1214/15-AOS1337.

Belloni, Alexandre, Victor Chernozhukov, and Christian Hansen. 2014. "Inference on Treatment Effects After Selection Among High-Dimensional Controls." *The Review of Economic Studies* 81 (2). Oxford University Press: 608–50.

Berk, Richard, Lawrence Brown, Andreas Buja, Kai Zhang, and Linda Zhao. 2013. "Valid Post-Selection Inference." *The Annals of Statistics* 41 (2). Institute of Mathematical Statistics: 802–37.

Blier, Léonard, Joshua R. Loftus, and Jonathan E. Taylor. 2016. "Inference on the Number of Clusters in k-Means Clustering." *In*