

Valid Inference after Model Selection and the selectiveInference R Package

Joshua Loftus - @joftius



NYU

Based on work with my co-authors (and others)

Jonathan Taylor



Stats@Stanford

Rob Tibshirani



Stats@Stanford

Ryan Tibshirani



Stats@CMU

Xiaoying Tian



Farallon

And my current student @ NYU Stern, Weichi Yao

Artificial Intelligence in the 19 century & inference in the 20th

Galton: “regression towards mediocrity”

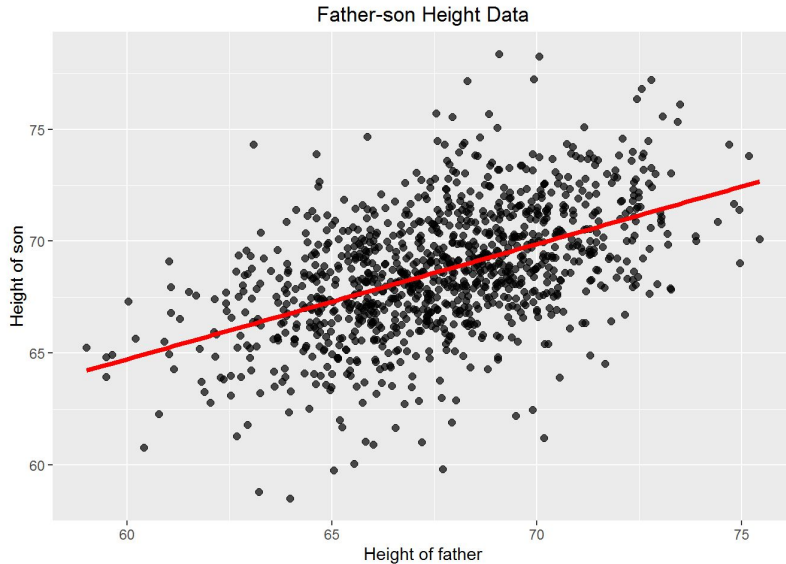


Image credit: Faiyaz Hasan

Inference: Gosset 1908 to Fisher 1922

THE widespread desire to introduce into statistical methods some degree of critical exactitude has led to the employment, now general in careful work, of the two types of quantity which characterize modern statistics, namely, the “probable error” and the test of “goodness of fit.” The test of goodness of fit was devised by Pearson, to whose labours principally we now owe it, that the test may readily be applied to a great variety of questions of frequency distribution. It is an essential means of justifying *a posteriori* the methods which have been employed in the reduction of any body of data. Slutsky and Pearson have extended the test to apply also to the fitness of regression formulæ, Pearson’s correlation ratio having also been employed for this purpose.

It has been shown in a previous communication [2 Fisher, 1922] that the χ^2 test of goodness of fit can be accurately applied only if allowance is made for the number of constants fitted in reconstructing the theoretical population. This correction is particularly important in contingency tables, but is necessary in all cases; and the fact that it has not been recognized has led to the adoption of erroneous values in almost all the cases in which tests of goodness of fit have been employed. The values of P have been exaggerated, and it is to be feared that in many cases wrong conclusions have been drawn from the values of P obtained.

It has, therefore, been necessary to extend the examination to the tests of goodness of fit of regression lines. The errors due to neglecting the number of constants fitted are here very pronounced;

One slide hypothesis test review

Sophisticated, high-dimensional AI:
multiple linear regression

Goodness of fit: testing the whole model,
do assumptions fail?

Testing individual regression coefficients

Tests should control type 1 error rate

p-values: how often a null test statistic
would be as extreme as observed

(Bayesians: sorry this talk mostly doesn't fit with your philosophy but also you
should care about optional stopping and selection bias and HARKing and so on, so
hopefully you can still take something away from this)

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \epsilon_i$$

$$H_{0,j} : \beta_j = 0$$

$$P_{H_{0,j}}(\text{reject } H_{0,j}) \leq \alpha$$

$$P(|T| > t_{\text{obs}}) \text{ for } T \sim t_{n-1}$$

```
> summary(lm(y ~ x, data = data.frame(x = rnorm(60), y = rnorm(60))))
```

Call:

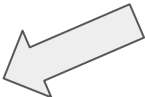
```
lm(formula = y ~ x, data = data.frame(x = rnorm(60), y = rnorm(60)))
```

Residuals:

Min	1Q	Median	3Q	Max
-2.61407	-0.79391	0.02925	0.63878	2.41739

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.04443	0.13654	-0.325	0.746
x	-0.07350	0.15269	-0.481	0.632



Residual standard error: 1.003 on 58 degrees of freedom

Multiple R-squared: 0.00398, Adjusted R-squared: -0.01319

F-statistic: 0.2317 on 1 and 58 DF, p-value: 0.632

Synthetic data: **predictor and response have no relationship**

p-value for test of predictor coefficient: 0.632

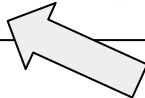
Frequentism: repeat for many samples...

% of rejections at 5% level: 6%

```
> reject <- function(fitted_lm) summary(fitted_lm)$coefficients[2,4] < 0.05
```

```
> mean(replicate(1000, reject(lm(y ~ x, data = data.frame(x = rnorm(60), y = rnorm(60)))))
```

```
[1] 0.06
```



Hypothesis tests designed to control type 1 error rate

(Inference after) Model selection

Choose from a set of many candidate models

Subset selection: choose subset of predictors

Dimension reduction, sparse/parsimonious model, interpretability

Necessity: more predictors than observations, e.g. PGS from GWAS

“Found” data, don’t know which predictors might be useful--if any.

Forward stepwise: greedy algorithm adding one predictor at a time, supervised orthogonalization

Lasso (Tibshirani, 1996)

$$\arg \min \sum_{i=1}^n (y_i - \beta^T x_i)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

Like forward stepwise but less greedy. Shrinks coefficients toward 0, more so for larger lambda

Both can find sparse models

```
candy <- fivethirtyeight::candy_rankings
head(candy[, c(1:2, 11:13)])
```

```
## # A tibble: 6 x 5
##   competitorname chocolate sugarpercent pricepercent winpercent
##   <chr>                <lgl>          <dbl>          <dbl>          <dbl>
## 1 100 Grand             T              0.732          0.860          67.0
## 2 3 Musketeers         T              0.604          0.511          67.6
## 3 One dime            F              0.0110         0.116          32.3
## 4 One quarter         F              0.0110         0.511          46.1
## 5 Air Heads           F              0.906          0.511          52.3
## 6 Almond Joy          T              0.465          0.767          50.3
```

chocolate, fruity, caramel,
peanutyalmondy, nougat,
crispedricewafer, hard, bar,
pluribus, sugarpercent,
pricepercent

Candy data: which attributes predict popularity?

```
# Forward stepwise with AIC
model <- step(lm(winpercent ~ . - competitorname, candy),
              k = 2, trace = 0)
# Significance tests for selected model
print(summary(model)$coefficients, digits = 2)
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	76	6.3	12.1	9.6e-20
## chocolateTRUE	-18	7.7	-2.3	2.2e-02
## hardTRUE	-16	8.1	-2.0	5.3e-02
## barTRUE	12	8.8	1.4	1.7e-01
## pricepercent	-27	12.4	-2.2	3.0e-02

Stepwise chooses 4 predictors. Which are significant?

FACT CHECK!



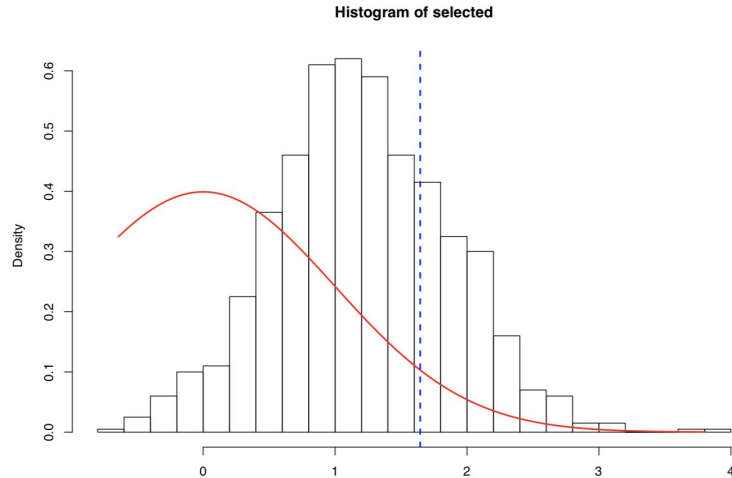
```
candy$winpercent <- 100 * runif(nrow(candy))
```



Replaced outcome variable with pure noise before running model selection!

Still got “significant” results?!

Top 5 predictors example



Type 1 error: about 26% instead of 5%...

Largest out of 5 null effects

```
> maxz <- function(n) return(max(rnorm(n)))  
> selected <- replicate(1000, maxz(5))
```

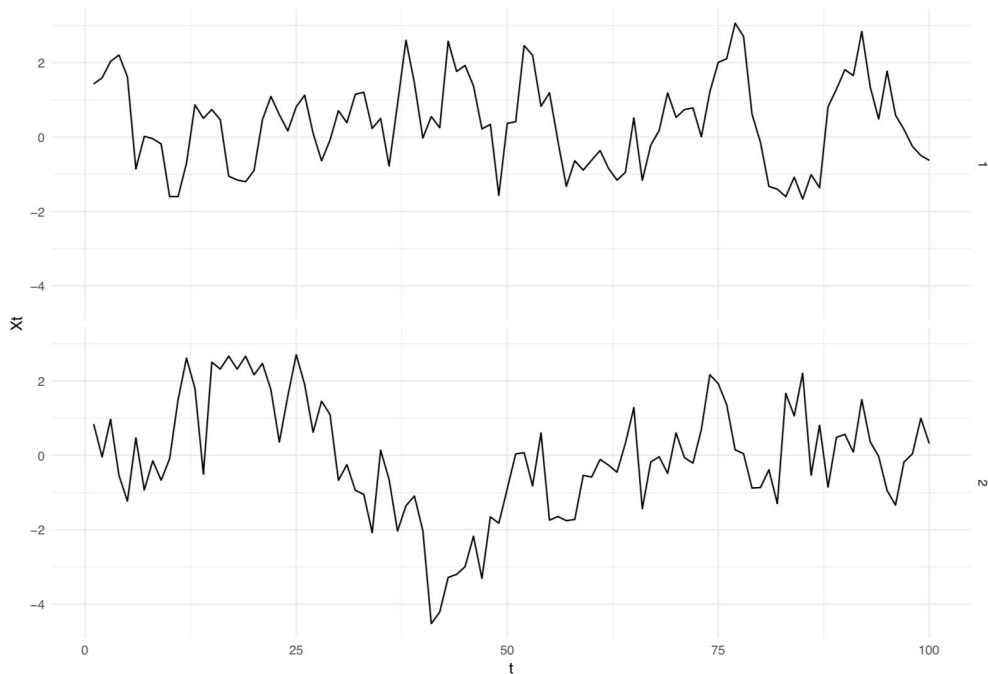
Various names / related concepts:

Winner's curse

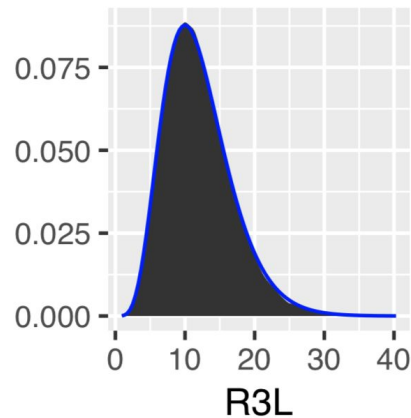
Overfitting

Selection bias

AR(p) selection & goodness of fit

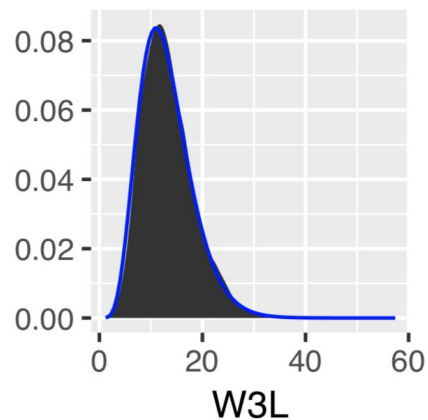


Select p with AICc, test fit with Ljung-Box test



Test distribution when AICc selects...

correct order



wrong order

Blue line: null distribution. No power!

Anti-conservative significance tests

High type 1 error, many false discoveries

Conservative goodness of fit tests

High type 2 error, conditional on selecting
wrong model *we can't tell* if it's wrong

How much does this really matter?

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).

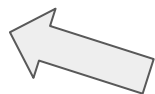
Machine learning solution: data splitting

Data: 240 lymphoma patients, 7399 genes

Lasso penalized coxph model with glmnet:

```
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
```



15 out of 7399 genes selected to predict survival time

Inference from an independent set of test/validation data

```
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
```

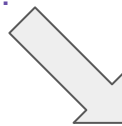
```
## Call:
```

```
## 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
```

Valid!



Data splitting...

Pros

Usually straightforward to apply

Usually doesn't require assumptions

Works almost automatically in many settings

Cons

Irreproducibility: can try many random splits

Inefficiency: doesn't use all the available data

Infeasibility: data structure (dependence), sample size bottlenecks (rare observations), etc

no free lunch

A SIGNIFICANCE TEST FOR THE LASSO¹

BY RICHARD LOCKHART², JONATHAN TAYLOR³,
RYAN J. TIBSHIRANI⁴ AND ROBERT TIBSHIRANI⁵

*Simon Fraser University, Stanford University, Carnegie Mellon University
and Stanford University*

EXACT POST-SELECTION INFERENCE, WITH APPLICATION TO THE LASSO

BY JASON D. LEE^{*,1}, DENNIS L. SUN^{+,2}, YUEKAI SUN^{*,3}
AND JONATHAN E. TAYLOR^{‡,4}

University of California, Berkeley, California Polytechnic State University⁺
and Stanford University[‡]*

Conditional approach

Motivated by selection bias rather than overfitting

INFERENCE IN ADAPTIVE REGRESSION VIA THE KAC–RICE FORMULA

BY JONATHAN E. TAYLOR^{*,1}, JOSHUA R. LOFTUS^{*,2} AND
RYAN J. TIBSHIRANI^{†,3}

Stanford University and Carnegie Mellon University[†]*

Optimal Inference After Model Selection

William Fithian^{*1}, Dennis L. Sun², and Jonathan Taylor³

¹Department of Statistics, University of California Berkeley
²Department of Statistics, California Polytechnic State University
³Department of Statistics, Stanford University

April 19, 2017

Motivation: screening/thresholding selection rule

From many independent effects, select those that lie above some threshold

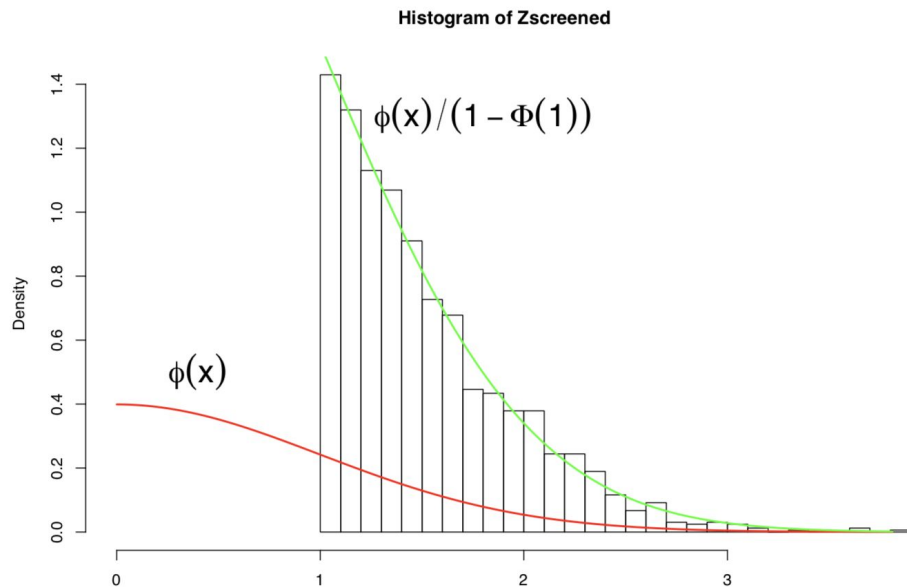
```
Zs <- rnorm(10000)
screen <- Zs > 1
Zscreened <- Zs[screen]
```

If the (global) null is true, which probability law would describe the **selected effects**?

An effect “surprises” us once to be selected, but must surprise us again to be declared significant conditional on (after) selection

Null distribution **truncated** at the threshold

In general: null distribution **conditional on selection**



Selective type 1 error


Conduct tests that control conditional type 1 error criterion:

$$P_{m, H_0}(\text{reject } H_0 \mid \hat{M} = m) \leq \alpha$$

where \hat{M} is the selected model

and H_0 is a null hypothesis about m

Reduces to classical type 1 error definition if the model is chosen *a priori*

Conditional control  marginal control

Data splitting controls this by using independent data subsets to select the model and test hypotheses

In general, need to work out how null distribution of test statistic is affected by conditioning

Typically results in truncated distributions

EXACT POST-SELECTION INFERENCE, WITH APPLICATION TO THE LASSO

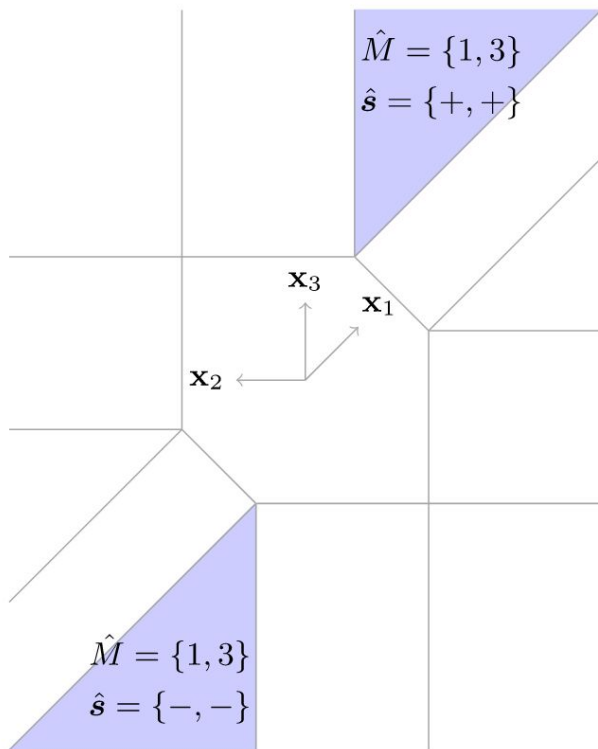
Lasso geometry

The event (set of outcomes) where lasso selects a given subset of variables is affine, a union of polytopes

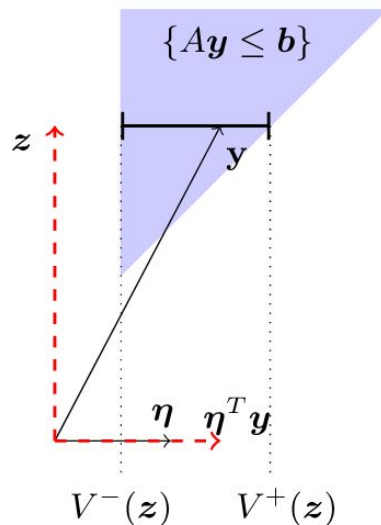
Reduce to one polytope by conditioning on the signs of selected variables

For significance tests, statistics are linear contrasts of the outcome

Reduce to one dimension by conditioning on orthogonal component



Model selection event



Test statistic truncation region

```

> n <- 100; p <- 200; sparsity <- 5; beta <- rep(0, p); beta[1:sparsity] <- 1
> x <- matrix(rnorm(n*p), nrow=n); y <- x %*% beta + rnorm(n)
> fit <- lar(x, y, maxsteps = 20)
> larInf(fit, sigma = estimateSigma(x, y)$sigmahat, type = "aic", ntimes = 1)

```

Call:

```

larInf(obj = fit, sigma = estimateSigma(x, y)$sigmahat, type = "aic",
      ntimes = 1)

```

Standard deviation of noise (specified or estimated) sigma = 0.969

Testing results at step = 6, with alpha = 0.100

Var	Coef	Z-score	P-value	LowConfPt	UpConfPt	LowTailArea	UpTailArea
5	0.926	8.858	0.123	-0.400	1.042	0.05	0.049
1	1.140	10.865	0.166	-1.950	10.063	0.05	0.050
2	0.931	9.112	0.008	0.639	8.521	0.05	0.050
4	0.832	8.405	0.570	-Inf	7.185	0.00	0.050
3	0.873	8.650	0.247	-2.206	5.898	0.05	0.050
188	-0.263	-2.528	0.459	-Inf	Inf	0.00	0.000

Estimated stopping point from AIC rule = 6

Necessary reduction in power to control conditional type 1 error

R: selectiveInference

True model: coefficients 1-5 out of $p = 200$, sample size $n = 100$

`lar()` algorithm fits the lasso path

AIC chooses model complexity

`larInf()` computes conditional inference, p-values and intervals

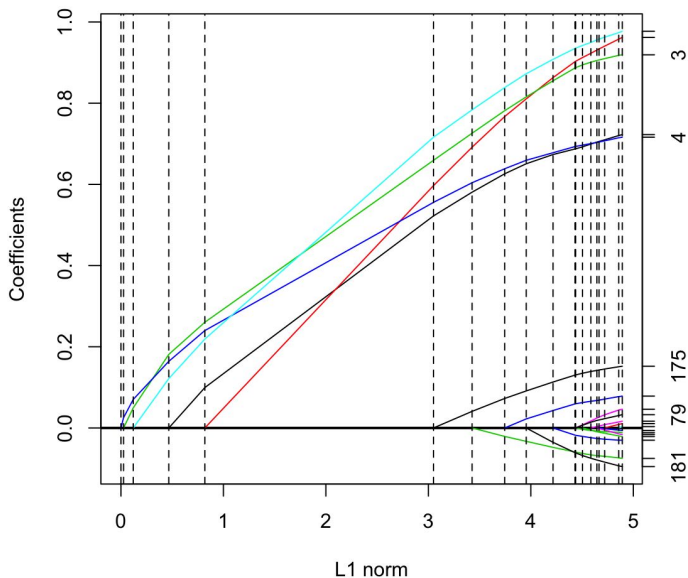
`estimateSigma()` uses cross-validated lasso

(Some numerical instability with intervals)

“Fixed lambda” lasso

Instead of AIC/CV

Least angle regression path



```
> sigma <- estimateSigma(x, y)$sigmahat
> lambda <- 3*sqrt(log(p)) * sigma
> beta = coef(fit, s=lambda, mode="lambda")
> fixedLassoInf(x, y, beta, lambda, sigma = sigma)
```

$$\lambda = 3\hat{\sigma} \sqrt{\log(p)}$$

Call:

```
fixedLassoInf(x = x, y = y, beta = beta, lambda = lambda, sigma = sigma)
```

Standard deviation of noise (specified or estimated) sigma = 0.983

Testing results at lambda = 6.785, with alpha = 0.100

Var	Coef	Z-score	P-value	LowConfPt	UpConfPt	LowTailArea	UpTailArea
1	0.938	9.587	0	0.776	1.099	0.049	0.050
2	1.184	10.557	0	0.999	1.370	0.049	0.049
3	1.051	9.567	0	0.869	1.232	0.048	0.050
4	0.866	8.773	0	0.702	1.031	0.048	0.048
5	1.205	12.344	0	1.043	1.368	0.048	0.048

Note: coefficients shown are partial regression coefficients

Warning message:

In fixedLassoInf(x, y, beta, lambda, sigma = sigma) :
Solution beta does not satisfy the KKT conditions (to within specified tolerances)

Target: projection of population mean onto $X_{\hat{M}}$

More powerful post-selection inference, with application to the Lasso

Keli Liu^{*1}, Jelena Markovic^{†1}, and Robert Tibshirani^{‡1,2}

Improving power

Conditioning on more (signs, component of y orthogonal to test contrast) reduces computation but also reduces power

One strategy: condition on

$j \in \hat{M}$ instead of $\hat{M} = m$

when testing $\beta_j = 0$

- Different target
- More computation
- More power

```
> ROSI(x, y, beta, lambda, dispersion = sigma)
```

Call:

```
ROSI(X = x, y = y, soln = beta, lambda = lambda, dispersion = sigma)
```

Dispersion taken to be dispersion = 1.108

Testing results at lambda = 7.654, with level = 0.90

	Var	Coef	Z-score	P-value	LowConfPt	UpConfPt
	0.01260	0.845	7.512	0	0.655	1.030
	0.00988	0.868	8.733	0	0.702	1.031
	0.01390	0.825	7.004	0	0.625	1.019
	0.01180	0.841	7.737	0	0.655	1.019
	0.01060	0.846	8.211	0	0.670	1.015

Note: coefficients shown are full regression coefficients.

Target: projection of population mean onto X

Randomized model selection

Low power and computational instability
observed when the outcome variable is near the
boundary of the truncated region

Another strategy: solve randomized model
selection problems, selection a given model no
longer implies hard constraints on the outcome
variable

```
> X = scale(x, TRUE, TRUE) / sqrt(n-1)
> rfit <- randomizedLasso(X, y, lam = lambda)
> out <- randomizedLassoInf(rfit)
> out$pvalues
1 2 3 4 5
0 0 0 0 0
```

R package version not quite user friendly yet...

The Annals of Statistics
2018, Vol. 46, No. 2, 679–710
<https://doi.org/10.1214/17-AOS1564>
© Institute of Mathematical Statistics, 2018

SELECTIVE INFERENCE WITH A RANDOMIZED RESPONSE

BY XIAOYING TIAN AND JONATHAN TAYLOR¹

MAGIC: a general, powerful and tractable method for selective inference

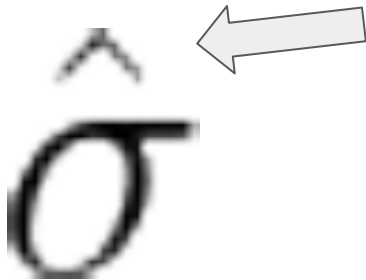
Xiaoying Tian, Nan Bi and Jonathan Taylor*

Selective sampling after solving a convex problem

Xiaoying Tian Harris, Snigdha Panigrahi, Jelena Markovic, Nan Bi,
Jonathan Taylor

$$\lambda = 3\hat{\sigma} \sqrt{\log(p)}$$

Not really an affine
selection event...



`estimateSigma()` uses cross-validation



The good news

Biometrika (2018), **105**, 4, pp. 755–768
Printed in Great Britain

doi: 10.1093/biomet/asy045
Advance Access publication 20 September 2018

Selective inference with unknown variance via the square-root lasso

BY XIAOYING TIAN

Farallon Capital Management LLC, One Maritime Plaza, 21st Floor, San Francisco, California 94115, U.S.A.

xtian@faralloncapital.com

JOSHUA R. LOFTUS

Department of Information, Operations, and Management Sciences, New York University, 44 West Fourth Street, New York, New York 10012, U.S.A.

loftus@nyu.edu

AND JONATHAN E. TAYLOR

Department of Statistics, Stanford University, Sequoia Hall, 390 Serra Mall, Stanford, California 94305, U.S.A.

jonathan.taylor@stanford.edu

$$\arg \min \left\| y - X\beta \right\|_2 + \lambda \left\| \beta \right\|_1$$

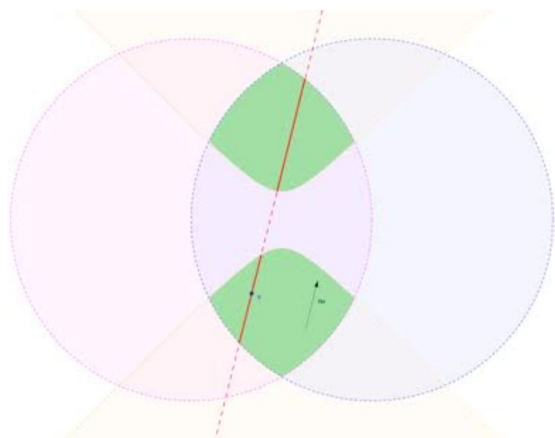
Can pick lambda without using outcome variable

The bad news

It's not in the R package...

More good news

Can handle quadratic model selection events!
(my dissertation work)



Selective inference in regression models with groups of variables

Joshua Loftus^{*,†} and Jonathan Taylor[†]

Selective inference after cross-validation

Joshua Loftus

More bad news

Conditioning on cross-validation selected models
is both computationally expensive and has low
power

Cross-validation not in the R package...

But! `groupfs()` and `groupfsInf()` functions allow
model selection respecting variable groupings,
e.g. levels of a categorical predictor

Conclusions

A few other approaches / R packages

SSLASSO - Spike and slab prior Bayesian approach

stabs - Stability selection, [re/sub]sampling and many cross-validation lasso paths, stable set

hdi - Stability selection and debiasing methods

EAinference - bootstrap inference for debiased estimators

PoSI - simultaneous inference guarantee over all possible submodels

Coming soon(?) to **selectiveInference**: goodness of fit tests. See also **RPtests** package for alternative.

Using data to decide which inferences to conduct results in **selection bias**

- Prediction error optimism (overfitting)
- Predictor significance (anti-conservative)
- Goodness of fit (conservative)

Variety of new statistical tools accounting for such bias

Selective inference: probability model is conditioned on selection, classical test statistics can then be compared to correspondingly truncated null distributions

Try out the **selectiveInference** R package and let us know what you think!

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