Statistical Learning and Sparsity

with applications to biomedicine

Rob Tibshirani
Departments of Biomedical Data Science & Statistics
Stanford University

AIStats 2019

Outline

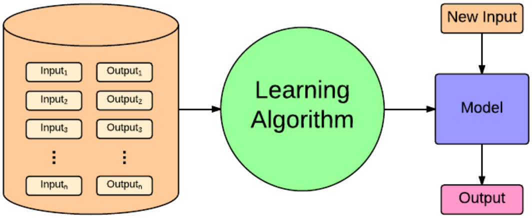
1. Some general comments about supervised learning, statistical approaches and deep learning
2. **Example**: Predicting platelet usage at Stanford Hospital
3. Two recent advances:
    - **Principal components lasso** [Combines PC regression and sparsity]
    - **Pliable lasso** [enables lasso model to vary across the feature space]

For Statisticians: 15 minutes of fame

- 2009: “ I keep saying the sexy job in the next ten years will be statisticians.” Hal Varian, Chief Economist Google
- 2012 “Data Scientist: The Sexiest Job of the 21st Century” Harvard Business Review

Sexiest man alive?

The Supervising Learning Paradigm



Training Data Fitting Prediction

Traditional statistics: domain experts work for 10 years to learn good features; they bring the statistician a small clean dataset

Today's approach: we start with a large dataset with many features, and use a machine learning algorithm to find the good ones. A huge change.

This talk is about supervised learning: building models from data for predicting an outcome using a collection of input features.

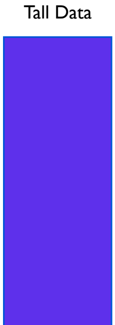
Big data vary in *shape*. These call for different approaches.

Thousands / Millions of Variables

Hundreds of Samples

Lasso & Elastic Net

We have too many variables; prone to overfitting.
Lasso fits linear models to the data that are *sparse* in the variables.
Does automatic variable selection.



Tens / Hundreds of Variables

Thousands / Tens of Thousands of Samples

Random Forests & Gradient Boosting

Sometimes simple models (linear) don't suffice.
We have enough samples to fit nonlinear models with many interactions, and not too many variables.
A Random Forest is an automatic and powerful way to do this.

The Elephant in the Room: DEEP LEARNING

Will it eat the lasso and other statistical models?

The Lasso

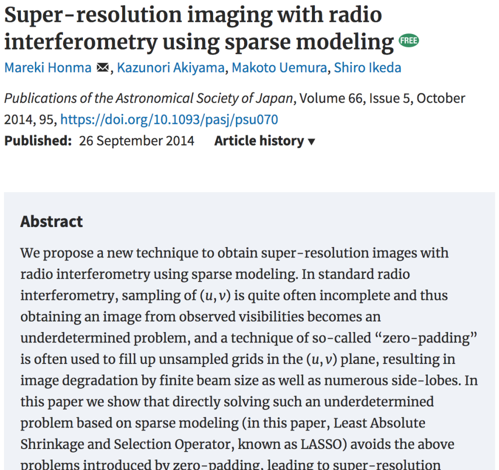
The **Lasso** is an estimator defined by the following optimization problem:

$$\underset{\beta_0,\beta}{\text{minimize}} \frac{1}{2}\sum_i (y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2 \qquad \text{subject to} \quad \sum |\beta_j| \leq s$$

- Penalty $\implies$ sparsity (feature selection)
- Convex problem (good for computation and theory)
- Our lab has written a open-source R language package called **glmnet** for fitting lasso models (Friedman, Hastie, Simon, Tibs). Available on CRAN.
  More than one million downloads!

Lasso and black holes

Apparently, sparse modelling and Lasso played an important role in the recent reconstruction of black hole image. And the work was done in part by **Japanese scientists.**



Super-resolution imaging with radio interferometry using sparse modeling FREE

Mareki Honma ✉, Kazunori Akiyama, Makoto Uemura, Shiro Ikeda

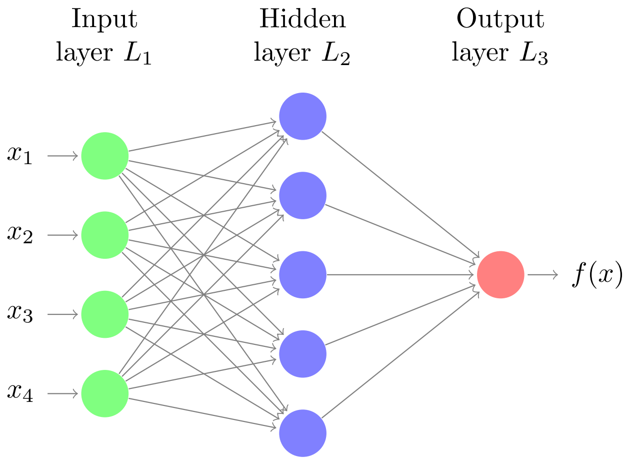
Publications of the Astronomical Society of Japan, Volume 66, Issue 5, October 2014, 95, https://doi.org/10.1093/pasj/psu070

Published: 26 September 2014 **Article history** ▾

Abstract

We propose a new technique to obtain super-resolution images with radio interferometry using sparse modeling. In standard radio interferometry, sampling of (u, v) is quite often incomplete and thus obtaining an image from observed visibilities becomes an underdetermined problem, and a technique of so-called "zero-padding" is often used to fill up unsampled grids in the (u, v) plane, resulting in image degradation by finite beam size as well as numerous side-lobes. In this paper we show that directly solving such an underdetermined problem based on sparse modeling (in this paper, Least Absolute Shrinkage and Selection Operator, known as LASSO) avoids the above problems introduced by zero-padding, leading to super-resolution

Deep Nets/Deep Learning



Neural network diagram with a single hidden layer. The hidden layer derives transformations of the inputs — nonlinear transformations of linear combinations — which are then used to model the output

Back to the Elephant

What makes Deep Nets so powerful:
(and challenging to analyze!)

It's not one "mathematical model" but a **customizable framework**– a set of **engineering tools** that can exploit the special aspects of the problem (weight-sharing, convolution, feedback, recurrence ...)

Will Deep Nets eat the lasso and other statistical models?

Not in cases where

- we have moderate #obs or wide data (#obs $<$ #features),
- SNR is low, or
- interpretability is important

In Praise of Simplicity

'Simplicity is the ultimate sophistication" — Leornardo Da Vinci

- Many times I have been asked to review a data analysis by a biology postdoc or a company employee. Almost every time, they are unnecessarily complicated. Multiple steps, each one poorly justified.
- Why? I think we all like to justify– internally and externally— our advanced degrees. And then there's the "hit everything with deep learning" problem
- Suggestion: Always try simple methods first. Move on to more complex methods, only if necessary

How many units of platelets will the Stanford Hospital need tomorrow?

WE WANT YOUR GOLD.

The stuff in your blood, not your bank.

Allison Zemek

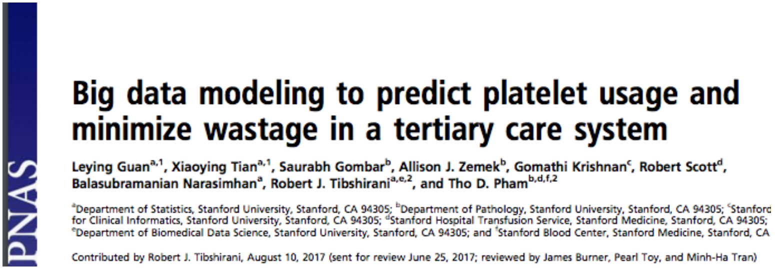
Tho Pham

Saurabh Gombar

Leying Guan

Xiaoying Tian

Balasubramanian Narasimhan



PNAS

Big data modeling to predict platelet usage and minimize wastage in a tertiary care system

Leying Guan[a,1], Xiaoying Tian[a,1], Saurabh Gombar[b], Allison J. Zemek[b], Gomathi Krishnan[c], Robert Scott[d], Balasubramanian Narasimhan[a], Robert J. Tibshirani[a,e,2], and Tho D. Pham[b,d,f,2]

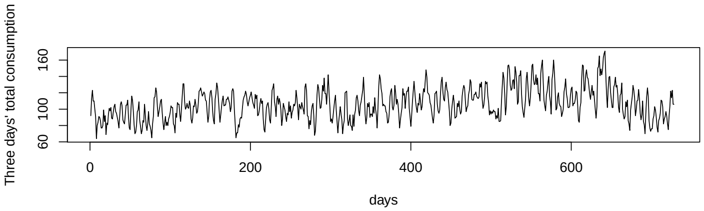
[a]Department of Statistics, Stanford University, Stanford, CA 94305; [b]Department of Pathology, Stanford University, Stanford, CA 94305; [c]Stanfor
for Clinical Informatics, Stanford University, Stanford, CA 94305; [d]Stanford Hospital Transfusion Service, Stanford Medicine, Stanford, CA 94305;
[e]Department of Biomedical Data Science, Stanford University, Stanford, CA 94305; and [f]Stanford Blood Center, Stanford Medicine, Stanford, CA

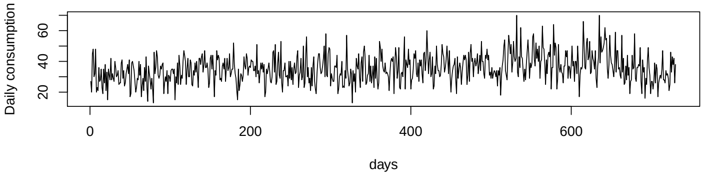
Contributed by Robert J. Tibshirani, August 10, 2017 (sent for review June 25, 2017; reviewed by James Burner, Pearl Toy, and Minh-Ha Tran)

Background

- Each day Stanford hospital orders some number of units (bags) of platelets from Stanford blood center, based on the estimated need (roughly 45 units)
- The daily needs are estimated "manually"
- Platelets have just 5 days of shelf-life; they are safety-tested for 2 days. Hence are **usable for just 3 days.**
- Currently about **1400** units (bags) are wasted each year. That's about **8%** of the total number ordered.
- There's rarely any shortage (shortage is bad but not catastrophic)
- Can we do better?

Data overview





Data description

Daily platelet use from 2/8/2013 - 2/8/2015.

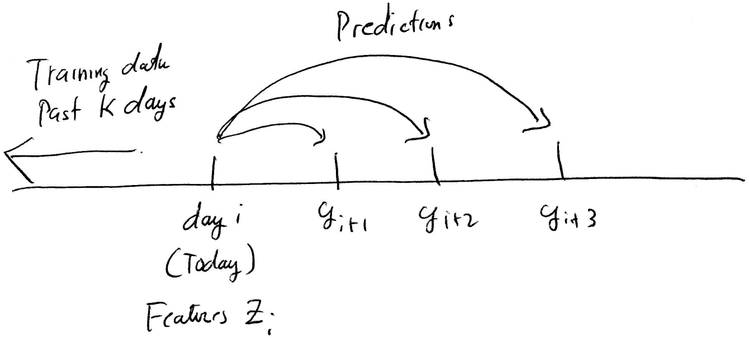
- Response: number of platelet transfusions on a given day.
- Covariates:
  1. **Complete blood count (CBC) data**: Platelet count, White blood cell count, Red blood cell count, Hemoglobin concentration, number of lymphocytes, . . .
  2. **Census data**: location of the patient, admission date, discharge date, . . .
  3. **Surgery schedule data**: scheduled surgery date, type of surgical services, . . .
  4. . . .

Notation

y_i : actual PLT usage in day i.
x_i : amount of new PLT that arrives at day i.
$r_i(k)$: remaining PLT which can be used in the following k days, $k = 1, 2$
w_i : PLT wasted in day i.
s_i : PLT shortage in day i.

- **Overall objective**: waste as little as possible, with little or no shortage

Our first approach



Our first approach

- Build a supervised learning model (via lasso) to predict use y_i for next three days (other methods like random forests or gradient boosting didn't give better accuracy).
- Use the estimates $\hat{y}_i$ to estimate how many units x_i to order. Add a buffer to predictions to ensure there is no shortage. Do this is a "rolling manner".
- Worked quite well- reducing waste to 2.8%- - but the loss function here is not ideal

More direct approach

This approach minimizes the waste directly:

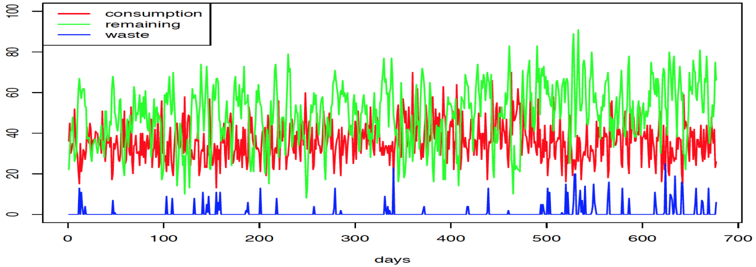
$$\begin{aligned}
J(\beta) &= \sum_{i=1}^{n} w_i + \lambda||\beta||_1 && (1)\\
&\text{where} && \\
\text{three days}' \text{ total need} \quad t_i &= z_i^T \beta, \quad \forall i = 1, 2, .., n && (2)\\
\text{number to order :} \quad x_{i+3} &= t_i - r_i(1) - r_i(2) - x_{i+1} - x_{i+2} && (3)\\
\text{waste} \quad w_i &= [r_{i-1}(1) - y_i]_+ && (4)\\
\text{actual remaining} \quad r_i(1) &= [r_{i-1}(2) + r_{i-1}(1) - y_i - w_i]_+ && (5)\\
r_i(2) &= [x_i - [y_i + w_i - r_{i-1}(2) - r_{i-1}(1)]_+]_+ && (6)\\
\text{Constraint : fresh bags remaining} \quad r_i(2) &\geq c_0 && (7)\\
& && (8)
\end{aligned}$$

This can be shown to be a convex problem (LP).

Results

Chose sensible features- previous platelet use, day of week, # patients in key wards.

Over 2 years of backtesting: no shortage, reduces waste from **1400** bags/ year (8%) to just **339** bags/year (1.9%)



Corresponds to a predicted direct savings at Stanford of $350,000/year. If implemented nationally could result in approximately $110 million in savings.

Moving forward

- System has just been deployed at the Stanford Blood center (R Shiny app).
- We are distributing the software around the world, for other centers to train and deploy
- see Platelet inventory R package `https://bnaras.github.io/pip/`

pcLasso: the lasso meets principal components regression (PCR)

Joint work with Ken Tay and Jerome Friedman

- Given a set of features, principal components regression computes the first few PCs $z_1, z_2 \ldots z_k$ and does a regression of y on these derived variables.
- PCR is a powerful way of capturing the main sources of variability, and hopefully signal, in the data. But it doesn't provide sparsity.
- How can we combine PCR and the lasso?

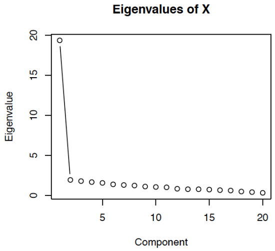
The Principal Components Lasso

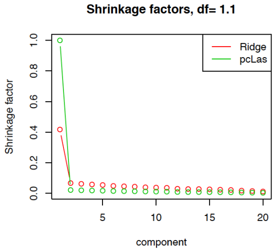
- Let $\mathbf{X} = \mathbf{UDV}^T$, the singular value decomposition of $\mathbf{X}$.
  The columns of $\mathbf{V}$ contain the PCs
  The **pcLasso** minimizes

$$J(\beta) = \frac{1}{2n}||\mathbf{y} - \mathbf{X}\beta||^2 + \lambda||\beta||_1 + \theta\frac{1}{2}\beta^T\mathbf{V}\mathbf{D}_{d_1^2-d_j^2}\mathbf{V}^T\beta \quad (9)$$

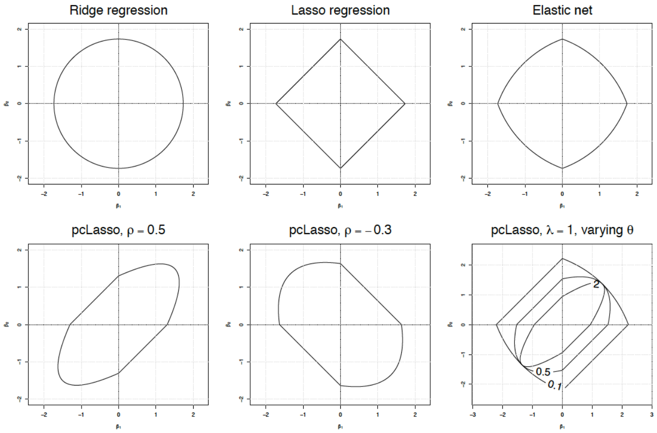
- The values d_j^2 are the eigenvalues of $\mathbf{X}$, with $d_1^2 \geq d_2^2 \cdots$
  In words: the pcLasso gives no penalty (**"a free ride"**) to the part of β that lines up with the first PC, and increasing penalties for components that line up with the second, third etc components.
- The choice $\mathbf{D} = \mathbf{I}$ results in the ridge penalty $\theta\sum\beta_j^2$ and gives the elastic net
- the parameter $\theta \geq 0$ controls the rate of increase in the penalty

Eigenvalues and shrinkage factors

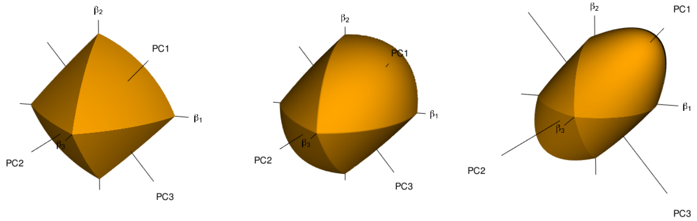




Contours of penalty functions



Three dimensional case: θ increases as we move left to right



Where it gets more interesting: grouped predictors

Suppose our features come in pre-defined groups like gene pathways, protein networks, or groups formed by clustering. Or the groups could be assay types like RNAseq, methylation, protein arrays etc

The pcLasso objective is now:

$$J(\beta) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||^2 + \lambda||\beta||_1 + \frac{\theta}{2}\sum_k \beta_k^T \Big(\mathbf{V}_k \mathbf{D}_{d_{k1}^2 - d_{kj}^2} \mathbf{V}_k^T\Big)\beta_k.$$

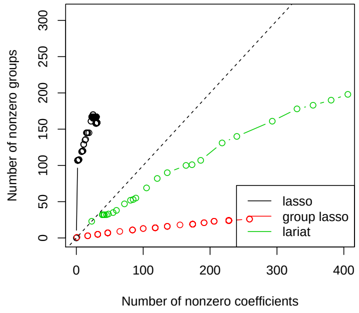
Each term in the penalty gives a *free ride* to components β_k that align with the first PC of that group

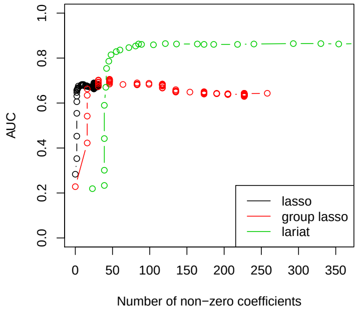
Some nice properties

- pcLasso exploits within-group similarity to boost weak signals in the individual features
- We have developed an algorithm that- -after the initial SVDs– **is as fast as glmnet**. This means it can be used to large problems (not yet GWAS size, but that's coming...)
- pcLasso automatically gives group sparsity (zeroes out some groups), if the features in a group are correlated
- Since it also has an ℓ_1 penalty, it yields feature-level sparsity too.
- In place of $\mathbf{X}^T\mathbf{X}$ one can use a pairwise similarity matrix, e.g. from a gene ontology, protein contact map etc
- We have MSE consistency results that generalize those for the lasso.

Example p53 gene expression data

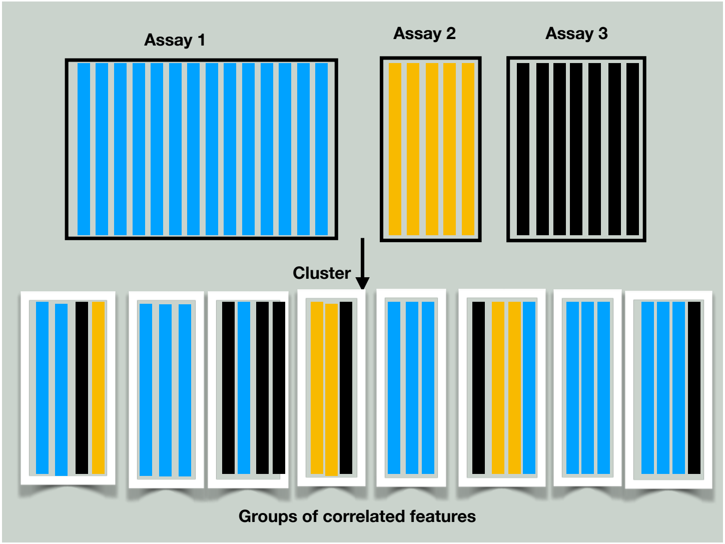
50 cell lines: 17 of which are classified as normal and 33 of which carry mutations in the p53 gene.
308 gene sets (overlapping); a total of 4301 genes



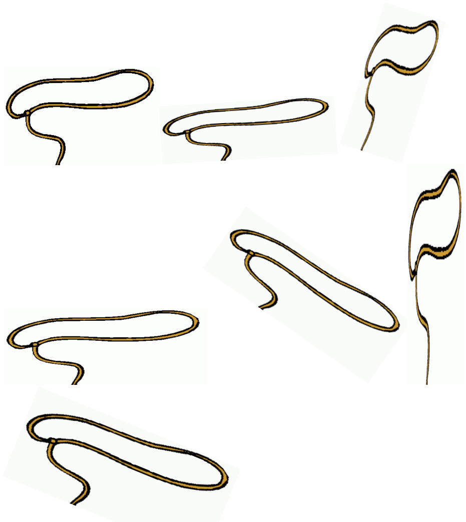


Combining data from multiple assays via pcLasso

"Data fusion"



Pliable Lasso: High level summary



The Pliable Lasso: how can we make Lasso more flexible?

- Lasso is a one size fits all model— it uses the same weights (coefficients) across the entire feature space
- Example where we might want a more flexible model: **Medical diagnosis/GWAS:** y=disease, X= (many) measurements of biomarkers; we suspect that a somewhat different set of biomarkers will be useful for males and females. Or young and old people; or ....

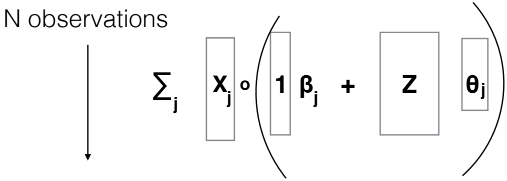
Modifying variables

We introduce a k-vector of observed modifying variables z.

Can be quantitative, categorical, or a mixture of the two; can be observed in both training and test sets, or only in training set.

The pliable lasso is defined by

$$\hat{y} \quad = \quad \beta_0\mathbf{1} + Z\theta_0 + \sum_{j=1}^{p} X_j \circ (\mathbf{1}\beta_j + Z\theta_j) \tag{10}$$



A Key Assumption

θ_j can be nonzero only if β_j is nonzero

“Weak hierarchy”

Model form and constraints maintain the sparsity and complexity control of the lasso, and lead to fast computation

Optimization

The model again:

$$\hat{y} = \beta_0 \mathbf{1} + Z\theta_0 + \sum_{j=1}^{p} X_j \circ (\mathbf{1}\beta_j + Z\theta_j) \tag{11}$$

We use the following objective function for this problem:

$$J(\beta_0, \boldsymbol{\theta}_0, \boldsymbol{\beta}, \Theta) = \frac{1}{2N}\sum (y_i - \hat{y}_i)^2 + (1-\alpha)\lambda \sum_{j=1}^{p} \Big((||(\beta_j, \boldsymbol{\theta}_j)||_2 + ||\boldsymbol{\theta}_j||_2 \Big)$$
$$+\alpha\lambda \sum_{j,k} |\theta_{jk}|_1.$$

Theta has to survive two penalties

- Overlapping group lasso penalty enforces weak hierarchy
- λ is main tuning parameter, yielding a path of solutions. We use blockwise coordinate descent.

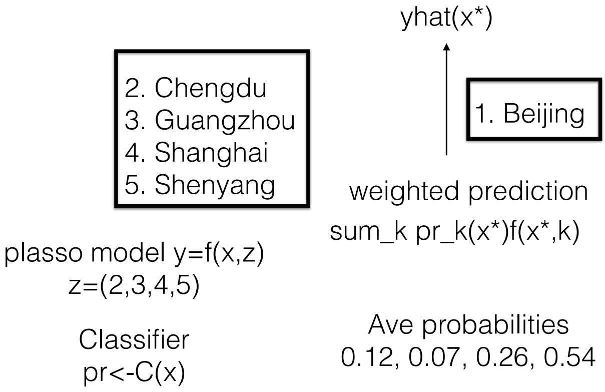
Example: Modelling pollution in five Chinese cities

From Dominik Rothenhaeusler–"**anchor regression**"'

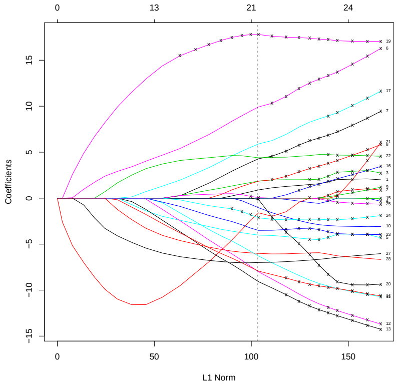
- From UCI database; Daily PM2.5 concentration measurements from 5 cities over 5 years
- Predictors: humidity, wind speed, dew point, month.... 29 in all
- Given a model built on 4 cities, predict pollution in the fifth city

Continued...

- We apply pliable lasso with $Z=$ indicator of 4 cities in each 5-fold cross-validation
- We also build a 4 city (multinomial lasso) classifier based on the features, and then use this to predict the city $\hat{z}$ in the 5th fold. This is then used to predict pollution in the 5th city

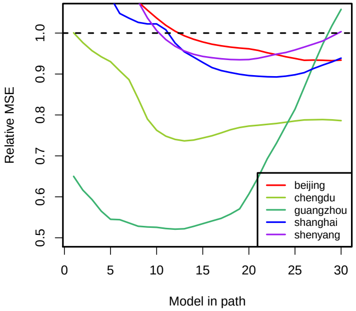

yhat(x*)
2. Chengdu
3. Guangzhou
4. Shanghai
5. Shenyang
1. Beijing
weighted prediction
sum_k pr_k(x*)f(x*,k)
plasso model y=f(x,z)
z=(2,3,4,5)
Ave probabilities
0.12, 0.07, 0.26, 0.54
Classifier
pr<-C(x)

Results



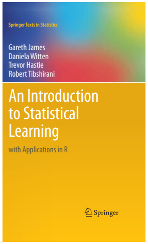
	Beijing	Chengdu	Guangzhou	Shanghai	Shenyang
humidity	$\Uparrow$			$\downarrow$	
NW wind	$\downarrow$		$\uparrow$	$\uparrow$	$\downarrow$

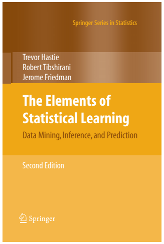
Validation set MSE relative to common linear model

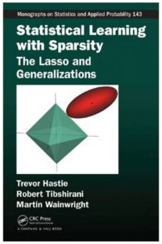


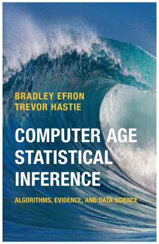
For further reading

The methods used are described in detail in our books on Statistical Learning: (last one by Efron & Hastie)


Springer Texts in Statistics
Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani
An Introduction to Statistical Learning
with Applications in R
Springer


Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer


Monographs on Statistics and Applied Probability 143
Statistical Learning with Sparsity
The Lasso and Generalizations
Trevor Hastie
Robert Tibshirani
Martin Wainwright
CRC Press


BRADLEY EFRON
TREVOR HASTIE
COMPUTER AGE STATISTICAL INFERENCE
ALGORITHMS, EVIDENCE, AND DATA SCIENCE

All available online for free

See `pcLasso` and `pliable` packages on CRAN