Computational Data Analysis

Sparse Regression

Line H. Clemmensen
Lars Arvastson

February 14, 2018

Todays Lecture

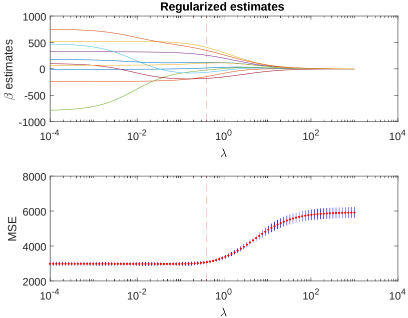
- Recap
- Curse of dimensionality
- Regularization
- Multiple hypothesis testing

Recap lecture 2

- What model selection methods did we use?
  - Regression
  - Classification
- What models did we use?

Recap lecture 2

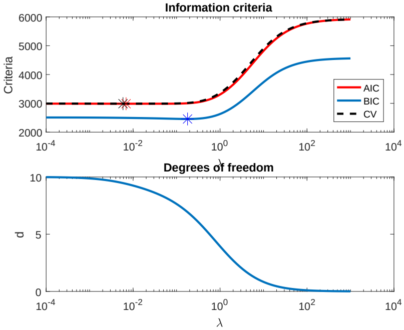
Cross validation
Exercise 1a, b, c




```
I = (mod(1:N,K)+1);
I = I(randperm(N));
for i=1:K
  Xtrain = X(I~=i,:);
  Ytrain = y(I~=i,:);
  Xtest  = X(I==i,:);
  Ytest  = y(I==i,:);
  for j=1:100
    Beta=(Xtrain'*Xtrain+lambda(j)*eye(10))\Xtrain'*Ytrain;
    SSE(i,j)= sum((Ytest-Xtest*Beta).^2);
  end
end
MSE = sum(SSE,1)/N;
```

Recap lecture 2

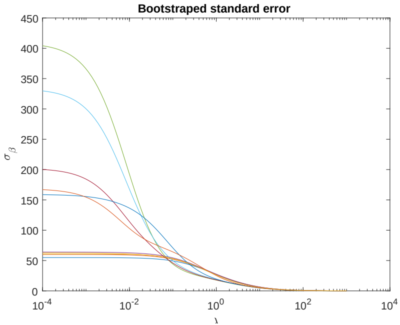
Information criteria
Exercise 1d




```
Beta    = X \ y;
e       = y-X*Beta;
% Low bias model std
s       = std(e);
for j=1:100
  Beta    = (X'*X+lambda(j)*eye(10)) \ X'*y;
  d       = trace(X * inv(X'*X+lambda(j)*eye(10))* X');
  e       = y-X*Beta;
  err     = sum(e.^2)/N;
  AIC(j) = err + 2 * d / N * s^2;
  BIC(j) = N / s^2 * (err + log(N)* d / N * s^2);
  D(j)   = d;
end
```

Recap lecture 2

Bootstrap
Exercise 1e

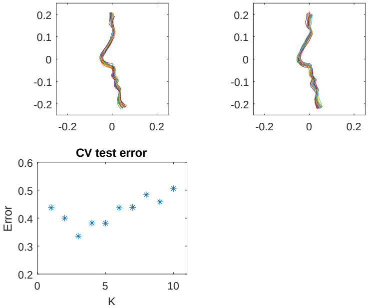



```
for i=1:Nboot
  I = randi(N,N,1);
  Xboot = X(I,:);
  Yboot = y(I,:);
  for j=1:100
    Beta(:,j,i) = ...
      (Xboot'*Xboot+lambda(j)*eye(10))\Xboot'*Yboot;
  end
end
BetaStd = std(Beta,[],3);
```

Recap lecture 2

Model selection and KNN classification
Exercise 2

```
% Leave-one-out CV
K      = length(Y);
Error = zeros(K,10);
I = (mod(1:N,K)+1);
I = I(randperm(N));
for i=1:K
  Xtrain = Xa(I~=i,:);
  Ytrain = Y(I~=i,:);
  Xtest  = Xa(I==i,:);
  Ytest  = Y(I==i,:);
  for Nknn=1:10
    Error(i,Nknn) = knn(Xtrain,Ytrain,Xtest,Ytest,Nknn);
  end
end
CVTestError=mean(Error,1);
```



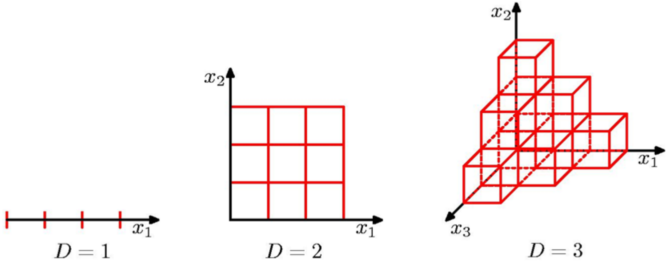
The curse of dimensionality

- Properties of high dimensional problems

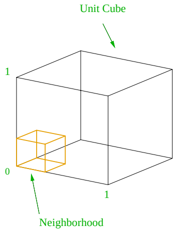
Curse of dimensionality

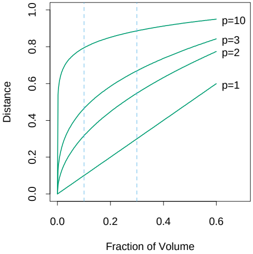
What happens when the dimension of the solution space grows, ie the number of variables grows?

- The number of regions grows exponentially with the dimensionality D



Curse of Dimensionality





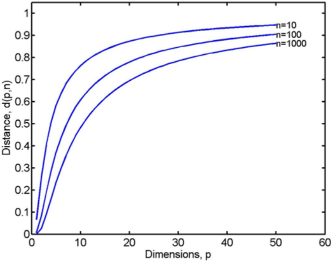
- Uniform data in a unit cube.
- Side length, e_p, needed to capture a fraction, r, of data increases with dimension, p.
- $e_p(r) = r^{1/p}$

Example With 10 features the side length has to be 80 % to cover 10 % of data.

Curse of dimensionality

For data fitted to a unit sphere the median distance from the center of the sphere to the closest point is

$$d(p,n) = \left(1 - \left(\frac{1}{2}\right)^{1/n}\right)^{1/p}$$



Interpolation becomes extrapolation in high dimensions

Blessings of dimensionality

It's not all bad...

In 2000, Donoho pinpointed **3 blessings of dimensionality.**

1. Several features will be correlated and we can average over them
2. Underlying distribution will be finite, informative data will lay on a low-dimensional manifold
3. Underlying structure in data (samples from continuous processes, images etc) will give an approximate finite dimensionality.

Donoho, D. L., August 2000. High-dimensional data analysis: The curses and blessings of dimensionality. In: Conf. Math Challenges of the 21st Century, Los Angeles.

Summing up

What considerations should we be aware of when dealing with high-dimensional data?

Dimension reduction

How to decrease the dimension and identify the most important variables, and get rid of the redundant or irrelevant variables.

Dimension reduction

- Combinatoric search, forward and backward selection
  - Previous courses - we make a recap and talk about multiple hypothesis testing
- Regularization of parameters
  - Focus of today
- Projection to lower dimensions - latent variables
  - Coming lectures, PCA, Unsupervised decomposition and Multi-way models
- Clustering of features
  - Lecture on Clustering
- Structuring parameter estimates
  - Related to regularization

Combinatoric search, forward and backward selection

Combinatoric search

Try all possible combinations of features and select the optimal one.

Pro: You will find the best combination.

Con: Number of combinations to test may be extremely large.

Forward selection

Add variables with highest information criterion one at a time.

Pro:
- Reasonable number of models to test.
- Can be used when $p > n$

Con: Might not give the best combination of features.

Backward elimination

Remove irrelevant features one at a time.

Pro: Reasonable number of models to test.

Con:
- Numerical issues when computing differences between models with many features.
- Might not give the best combination of features
  - Usually better than forward selection

Regularization

Shrinkage methods

Instead of controlling model complexity by setting a subset of coefficients to zero we can **shrink** all the coefficients some way towards zero.

Three established standard techniques

- **Ridge** regression uses quadratic shrinkage, L_2-norm
- **Lasso** regression uses absolute-value shrinkage, L_1-norm
- **Elastic net** which is a hybrid method

Norms of β

What is the definition of the L_2-norm,

$$||\beta||_2^2 =$$

What is the definition of the L_1-norm

$$||\beta||_1 =$$

Ridge regression

Ridge regression solves

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta$$

or equivalently the constrained optimization problem

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta) \text{ subject to } \sum \beta_j^2 \leq s$$

We will explore this equivalence further when we talk about Lagrange factors.

- Increased λ will make the estimated β's smaller but not exactly zero.
- We typically do not penalize the intercept β_0

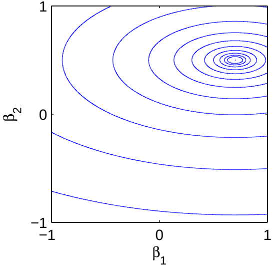
Ridge regression optima

Optimization of a weighted sum

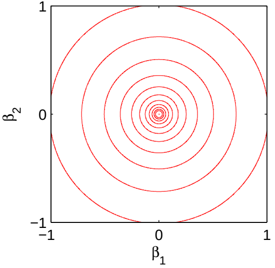
$$\beta_{Ridge} = \arg\min_{\beta} ||Y - X\beta||_2^2 + \lambda||\beta||_2^2$$

Contour plots of

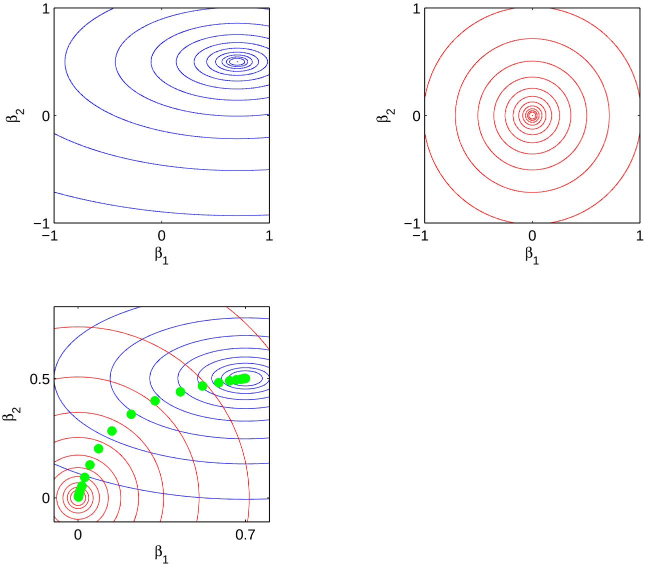
$||Y - X\beta||_2^2$



$||\beta||_2^2$



Regularization path



The Lasso

The Lasso regression solves

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta) + \lambda|\beta|$$

or equivalently the constrained optimization problem (known as basis pursuit)

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta) \text{ subject to } \sum |\beta| \leq s$$

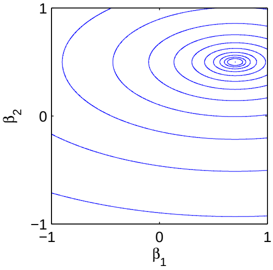
- Notice that the L_2-penalty is replaced by a L_1-penalty.
- This makes the solution nonlinear in Y and a quadratic programming algorithm is used to compute it.
- For large enough λ some of the β will be set to **exactly zero**.
- The effective numbers of parameters, *df*, equals the number of coefficients different from zero.

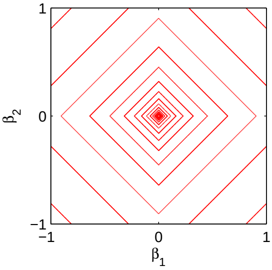
Lasso regularization

- Lasso regularization will gear parameters towards zero.

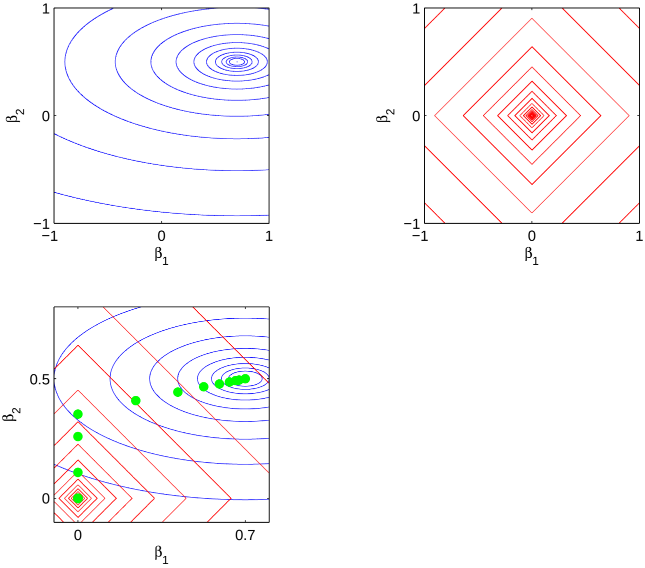
$$\hat{\beta}_{Lasso} = \arg\min_{\beta} ||Y - X\beta||_2^2 + \lambda ||\beta||_1$$
$$= \arg\min_{\beta} ||Y - X\beta||_2^2 + \lambda \sum_i |\beta_i|$$

- Non-trivial optimization problem...



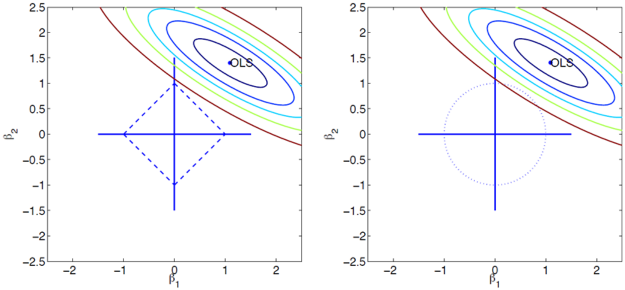


Regularization path



Geometry of solutions with L_1 and L_2 penalties

Visual solution to the constrained optimization problems for lasso and ridge,



Example with the diabetes data set

Name	OLS β	Ridge β, λ=1000	Lasso β, *8-nonzero*	Lasso β, *4-nonzero*
Age	-10.0122	0.3027	0	0
Sex	-239.8191	0.0685	-226.1337	0
BMI	519.8398	0.9468	526.8855	505.6596
BP	324.3904	0.7125	314.3893	191.2699
S1	-792.1842	0.3412	-195.1058	0
S2	476.7458	0.2797	0	0
S3	101.0446	-0.6369	-152.4773	-114.1010
S4	177.0642	0.6939	106.3428	0
S5	751.2793	0.9132	529.9160	439.6649
S6	67.62540	0.6168	64.4874	0

Algorithms for Lasso

There exist several implementations to solve the Lasso problem, examples

- Least angle regression selection (LARS)
- Cyclical coordinate descent

Least angle regression selection (LARS)

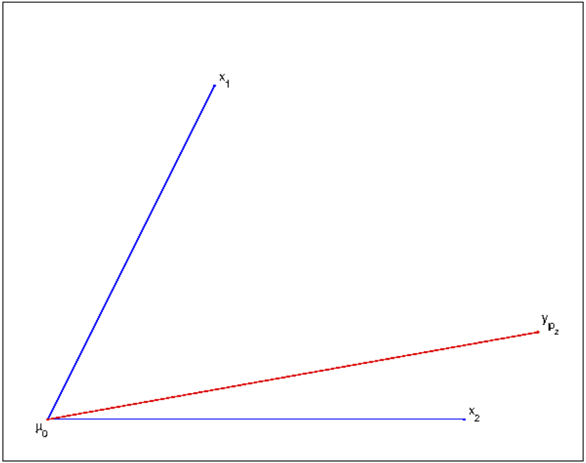
- Fast - calculates the entire path (all λ values) in the speed of one OLS fit.
- Easy to implement, intuitive.
- C_p-like statistic for choosing the number of steps.

$$C_p = \frac{1}{\hat{\sigma}^2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 - n + 2k$$

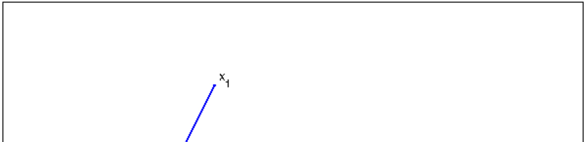
where k is the number of steps.

Hesterberg et al., 2008, Least angle and L1 penalized regression: A review, Statistics Surveys, Vol. 2, p. 61-93.

Least angle regression selection (LARS) - Algorithm



$$\mu_0 = \mathbf{0}$$
$$\mu_1 = \mu_0 + \gamma_1 \mathbf{x}_2$$
$$\mu_2 = \mu_1 + \gamma_2 \mathbf{x}_1$$



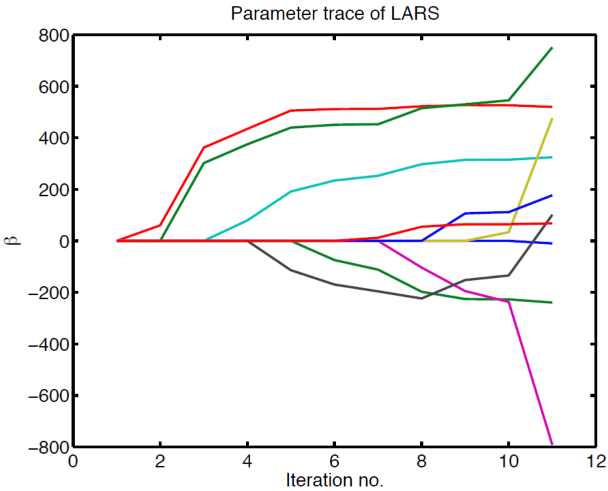
Least angle regression selection (LARS) - Algorithm

Assumptions: Data is centered and normalized (each variable has length one). This means that: $X^T X \approx Corr(X)$.

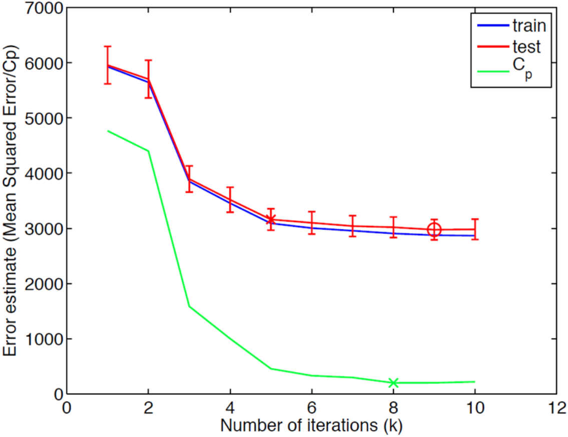
Lasso modification: If the parameter estimate of an active variable crosses zero, set it to zero and re-compute the direction.

- Gives a piecewise linear path to obtain lasso solutions for all relevant values of lambda.

Parameter trace for Diabetes example



C_p in LARS for Diabetes example



Cyclical coordinate descent

Solve

$$\min_{\beta} \frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i \beta)^2 + \lambda |\beta|$$

iteratively by cyclic updating one coordinate β_k at a time, while holding the others fixed.

Compute residual $r_i = y_i - \tilde{y}_i^{(k)}$ for $\tilde{\beta}$ excluding parameter $\tilde{\beta}_k$,

$$r_i = y_i - \sum_{i \neq j}^{p} x_{ij} \tilde{\beta}_j(\lambda)$$

Calculate the OLS solution to $r_i = x_{ik} \tilde{\beta}_k$. This is

$$\tilde{\beta}_k^{OLS} = \frac{1}{n} \sum_{i=1}^{n} x_{ik} r_i$$

(Assume standardization $\sum_i x_{ij} = 0$ and $\frac{1}{n} \sum_i x_{ij}^2 = 1$, $j = 1, ..., p$)

Cyclical coordinate descent, cont'd

Obtain the new lasso coordinate $\tilde{\beta}_k$ by shrinking the OLS estimate and set it to zero if it is close to zero,

$$\tilde{\beta}_k(\lambda) = sign(\tilde{\beta}_k^{OLS})(|\tilde{\beta}_k^{OLS}| - \lambda)_+$$

this is called **soft thresholding**.

Cycle through $k = 1, ..., p$ repeatedly until convergence.

The elastic net

By combining the L_1 and the L_2-norm we obtain sparsity and shrinkage

$$\min_{\beta} \frac{1}{2n}||Y - X\beta||_2^2 + \lambda\left(\frac{1}{2}(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1\right)$$

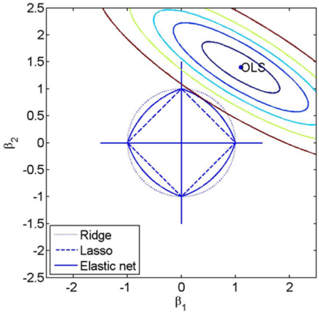
or equivalently

$$\min_{\beta} \frac{1}{2n}||Y - X\beta||_2^2 \quad \text{such that} \quad \frac{1}{2}(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1 \leq t$$

for some t.

Advantage: Combines the shrinkage of ridge and parameter selection of the lasso to obtain a robust sparse estimate.

Contour plot



Contour plot of OLS criteria,

$$||Y - X\beta||_2^2$$

and the elastic net restriction,

$$\frac{1}{2}(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1$$

In figure $\alpha = 0.5$.

Augmented problem

We can change an elastic net problem into a Lasso problem,

$$\min_{\beta} ||Y - X\beta||_2^2 + \lambda_2 ||\beta||_2^2 + \lambda_1 ||\beta||_1$$

by extending data,

$$X^* = (1+\lambda_2)^{-1/2} \begin{bmatrix} X \\ \sqrt{\lambda_2} I_p \end{bmatrix} \text{ and } y = \begin{bmatrix} y \\ 0_p \end{bmatrix}$$

Yields the OLS solution

$$\frac{1}{\sqrt{1+\lambda_2}}(X^t X + \lambda_2 I_p^T I_p)\beta^* = X^T y$$

We see that $1/\sqrt{1+\lambda_2}\beta^*$ is a scaled ridge solution.

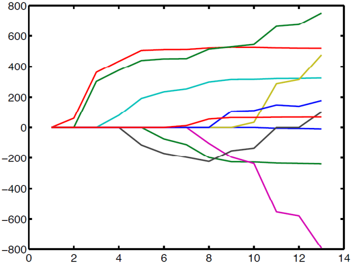
Why? Because now we can use the LARS algorithm to obtain the whole parameter trace.

The elastic net example - Diabetes

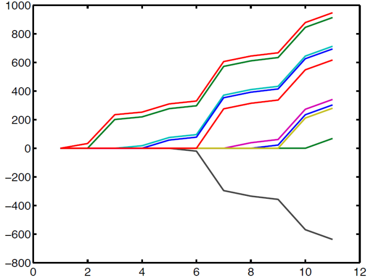
Name	OLS β	Ridge β, λ=1000	Lasso β, *4-nonzero*	EN β, λ=1000, *4-nonzero*
Age	-10.0122	0.3027	0	0
Sex	-239.8191	0.0685	0	0
BMI	519.8398	0.9468	505.6596	310.3929
BP	324.3904	0.7125	191.2699	75.6301
S1	-792.1842	0.3412	0	0
S2	476.7458	0.2797	0	0
S3	101.0446	-0.6369	-114.1010	0
S4	177.0642	0.6939	0	57.6991
S5	751.2793	0.9132	439.6649	277.0699
S6	67.62540	0.6168	0	0

Parameter traces for Diabetes example

Low ridge penalty

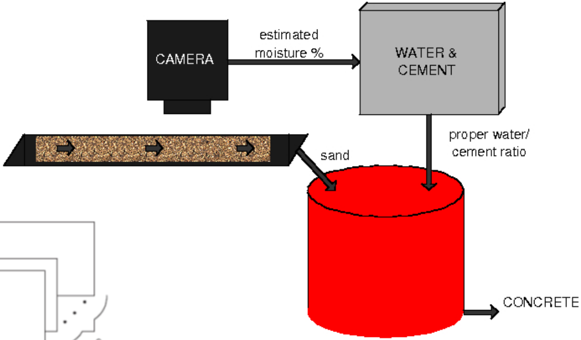


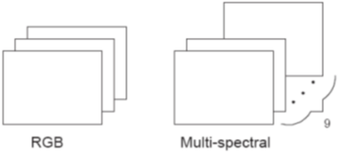
High ridge penalty



Example - Sand data set

Estimation of moisture content in sand used to make concrete.



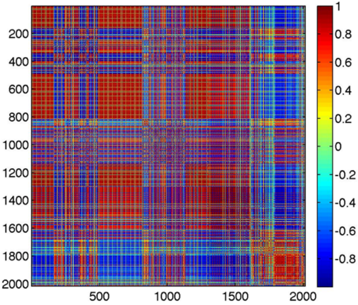


- Necessary to know in order to add the right amount of water.

The sand data set

- One sand type with 59 samples (0-8 % moisture content)
- 2016 features calculated based on multi-spectral images
- 1st order statistics of: spectral bands, differences between spectral bands, pairwise ratios of spectral bands, and scale spaces.
- High correlations exist in the covariates

The sand data set

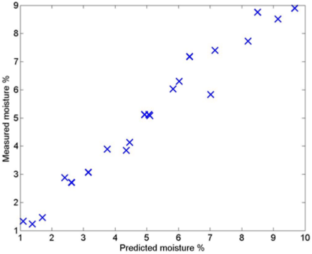


Covariance structure of the 2016 features.

Many correlated features indicating a low dimensional underlying structure.

Elastic net on sand data

- MSE = 0.2 moisture % (leave-one-out predictions)
- 109/2016 features were chosen



Elastic net and coordinate descent

Solve

$$\min_{\beta} \frac{1}{2n}||Y - X\beta||_2^2 + \lambda \left(\frac{1}{2}(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1 \right)$$

1. Calculate residuals and OLS solution as in the Lasso algorithm.
2. Update Elastic Net estimate using soft thresholding,

$$\tilde{\beta}_k(\lambda) = \frac{sign(\tilde{\beta}_k^{OLS})(|\tilde{\beta}_k^{OLS}| - \lambda\alpha)_+}{1 + \lambda(1-\alpha)}$$

3. Cycle through $k = 1, ..., p$ repeatedly until convergence.

Why use elastic net?

- Get rid of irrelevant variables/select important variables (lasso)
- When $p > n$, the number of non-zero coefficients can exceed n - unlike the lasso.
- Works well when covariates are highly correlated; allows us to "average" highly correlated features and obtain more robust estimates (grouping features).

Drawback: Issue of tuning two parameters. Use a grid search, a fine grid in λ and fewer values for α.

When do we gain from using elastic net?
Hard to know, try!

Best practice 1

Subtract mean and standardize variance on all variables before applying any regularization techniques!

Why?

Best practice 2

When you have obtained the optimal regularization parameters and evaluated performance you should build one final model on all data (using the obtained regularization parameter).

Why?

Multiple testing

Feature assessment

Assessing the significance of each of the p features.

- Traditional t-test of difference between groups.
  - Testing for differences in mean.
- Traditional F-test of parameter significance.
  - Testing if the estimated parameters are zero.

Feature assessment - the issue

If we test one hypothesis at an α-level of significance there is a chance α of falsely rejecting the hypothesis.

This is no longer the case if we do many tests!

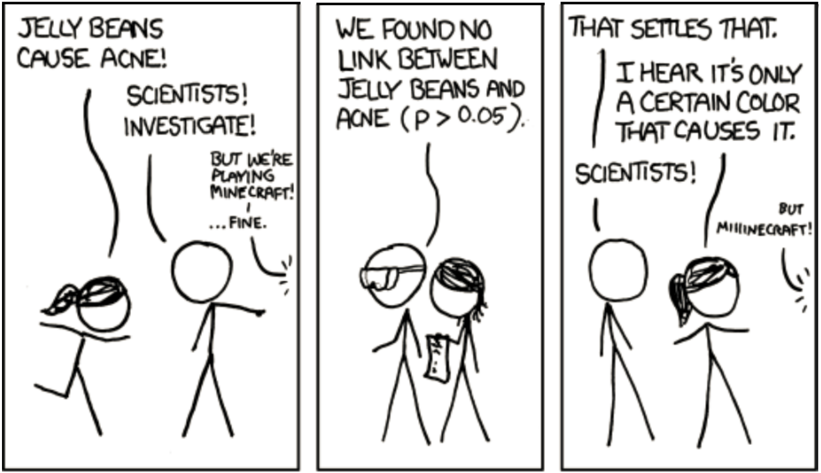
The family-wise error rate (FWER) is the probability of at least one false rejection.

If the features are independent and each tested at an α-level then $FWER >> \alpha$ for large p.

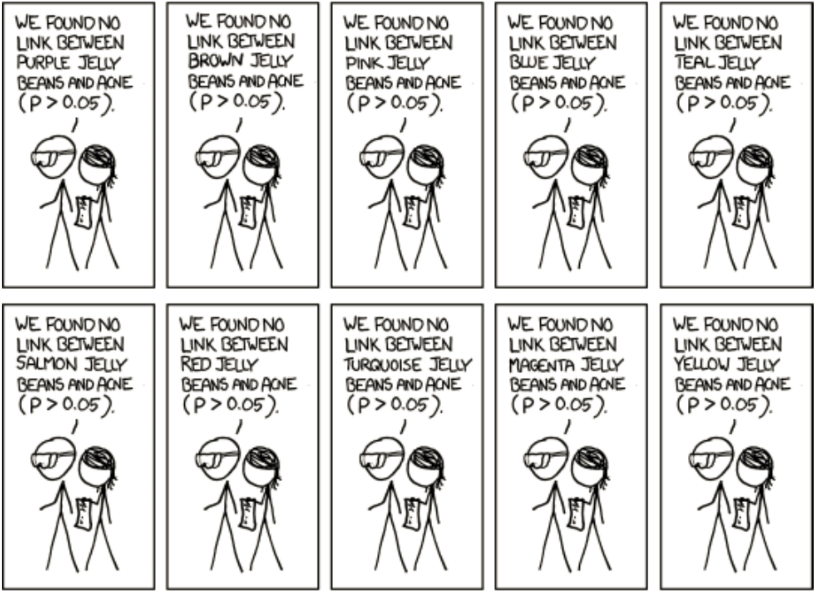
For M independent test at significance level α,

$$FWER = 1 - (1 - \alpha)^M$$

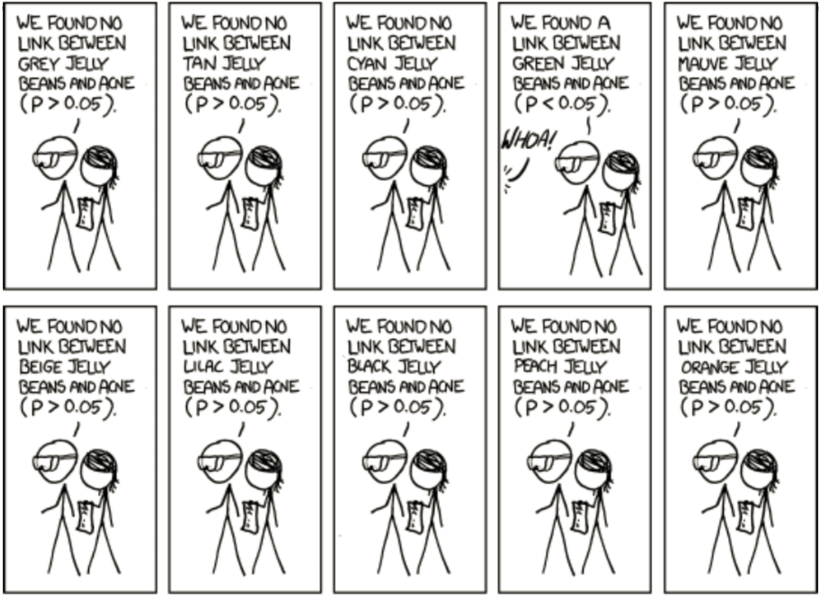
Example from www.xkcd.com



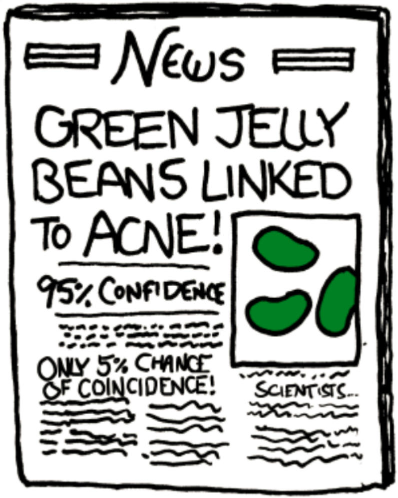
Example from www.xkcd.com



Example from www.xkcd.com



Example from www.xkcd.com



FWER for the jelly bean example

- 20 experiments conducted at a 5 % significance level
- Assume that the effect of different colors are independent, then $FWER = 1 - (1 - 0.05)^{20} \approx 0.64$.
- There is 64 % probability of at least one false rejection.

Bonferroni correction

Using the Bonferroni correction we rescale the α with the number of tests.

Reject a hypothesis if its p-value is below α/M.

1. Now we have an α-probability of making a false rejection.
   - Assuming independence
2. The resulting threshold will often result in low power.
   - We miss out on important effects

False Discovery Rate (FDR)

We can have more significant findings if we allow for a few mistakes.

The false discovery rate is a technique to control the number of falsely detected significant features.

The false discovery rate is

$$FDR = E\left(\frac{FP}{FP + TP}\right)$$

where

$$\begin{aligned} FP &= \text{False positives (false discoveries)} \\ TP &= \text{True positives (true discoveries)} \end{aligned}$$

If we accept hypotheses where $FDR < q$ then we will expect that among our findings there will be q mistakes.

FDR

Gain: We control false positives - added power.

Cost: Increased number of false negatives.

We prefer to get a few false discoveries (percentage-wise) but gain more information, than ensuring no false discoveries and loosing some information.

Benjamini-Hochbergs algorithm for FDR

THE BENJAMINI HOCHBERG PROCEDURE. Let $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$ be the ordered observed p-values. Define

$$k = \max\left\{i\colon\ p_{(i)} \leq \frac{i}{m}q\right\}, \qquad (1)$$

and reject $H^0_{(1)} \cdots H^0_{(k)}$. If no such i exists, reject no hypothesis.

1. Take your already calculated p-values and sort them from smallest to largest.
2. Walk down the sorted list and reject the hypotheses as long as $\frac{i}{m}q$ is smaller than the p-values.

q is **your choice** of acceptable fraction of mistakes. A single hypothesis is often tested at $\alpha = 0.05$ but we often accept higher values for q, say 0.1 or even 0.2.

Summary

Introduction

- The curse of dimensionality
- The blessings of dimensionality
- Dimension reduction

Regularization

- Ridge, Lasso and Elastic Net
  - Algorithms
- Shrinkage and sparsity
- Best practices

Multiple hypothesis testing

- Why it is a problem
- Bonferroni correction
- False discover rate and Benjamini-Hochberg algorithm