MGMTMFE 431:

Data Analytics and Machine Learning

Topic 4: Model Selection and Shrinkage

Spring 2025

Professor Lars A. Lochstoer

UCLA Anderson
SCHOOL of MANAGEMENT

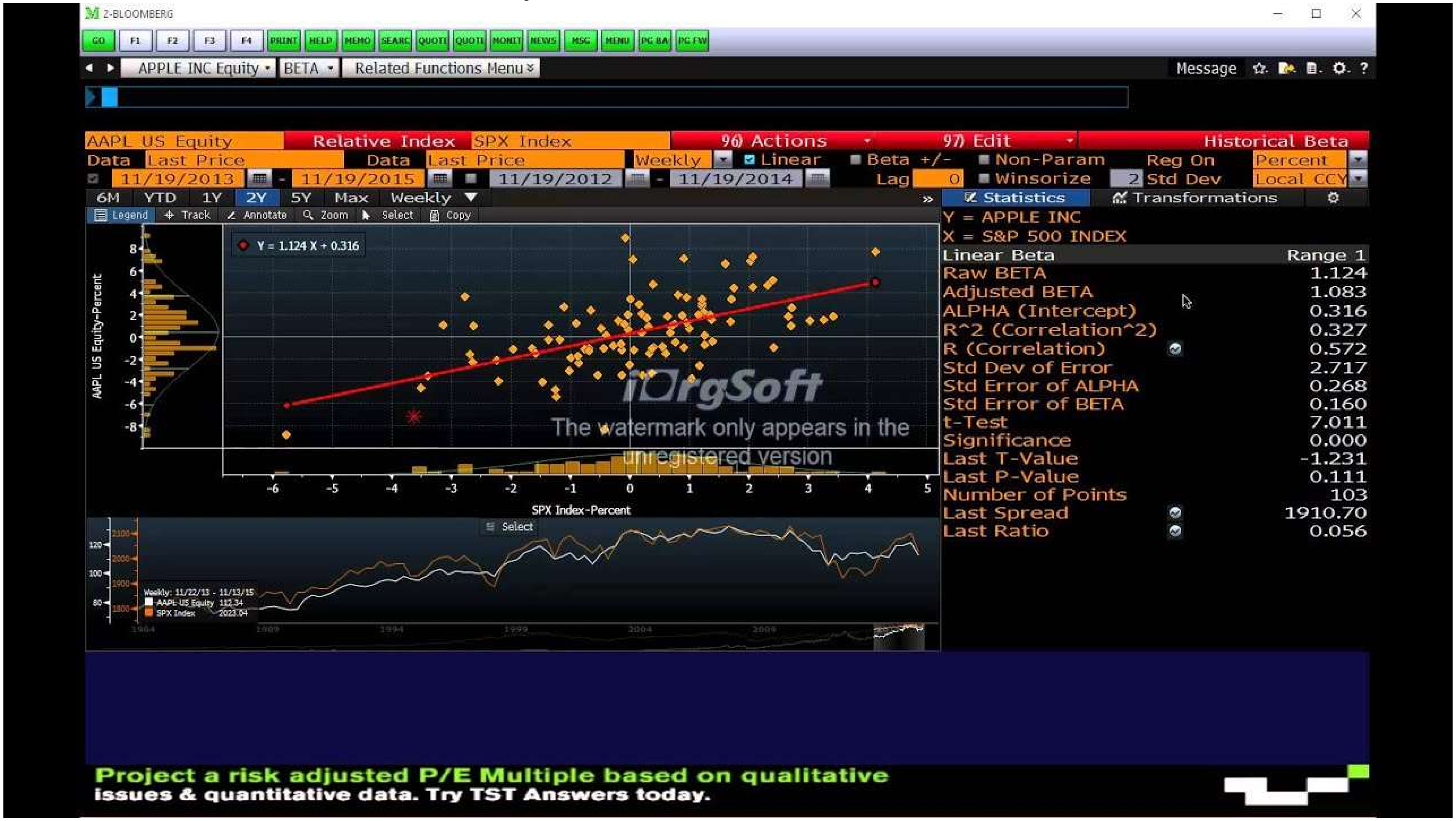
Model Selection and Shrinkage

a. The “Bloomberg Beta” – bias vs. estimation error
b. Shrinkage
c. Ridge Regression
d. Cross-Validation and Model Selection
e. Sparsity: The Lasso
f. Bayesian Regression
g. An automated “Machine Learning”-based dynamic trading strategy

UCLAAnderson
SCHOOL of MANAGEMENT

a. The Bloomberg Beta

- Notice raw beta vs. adjusted beta



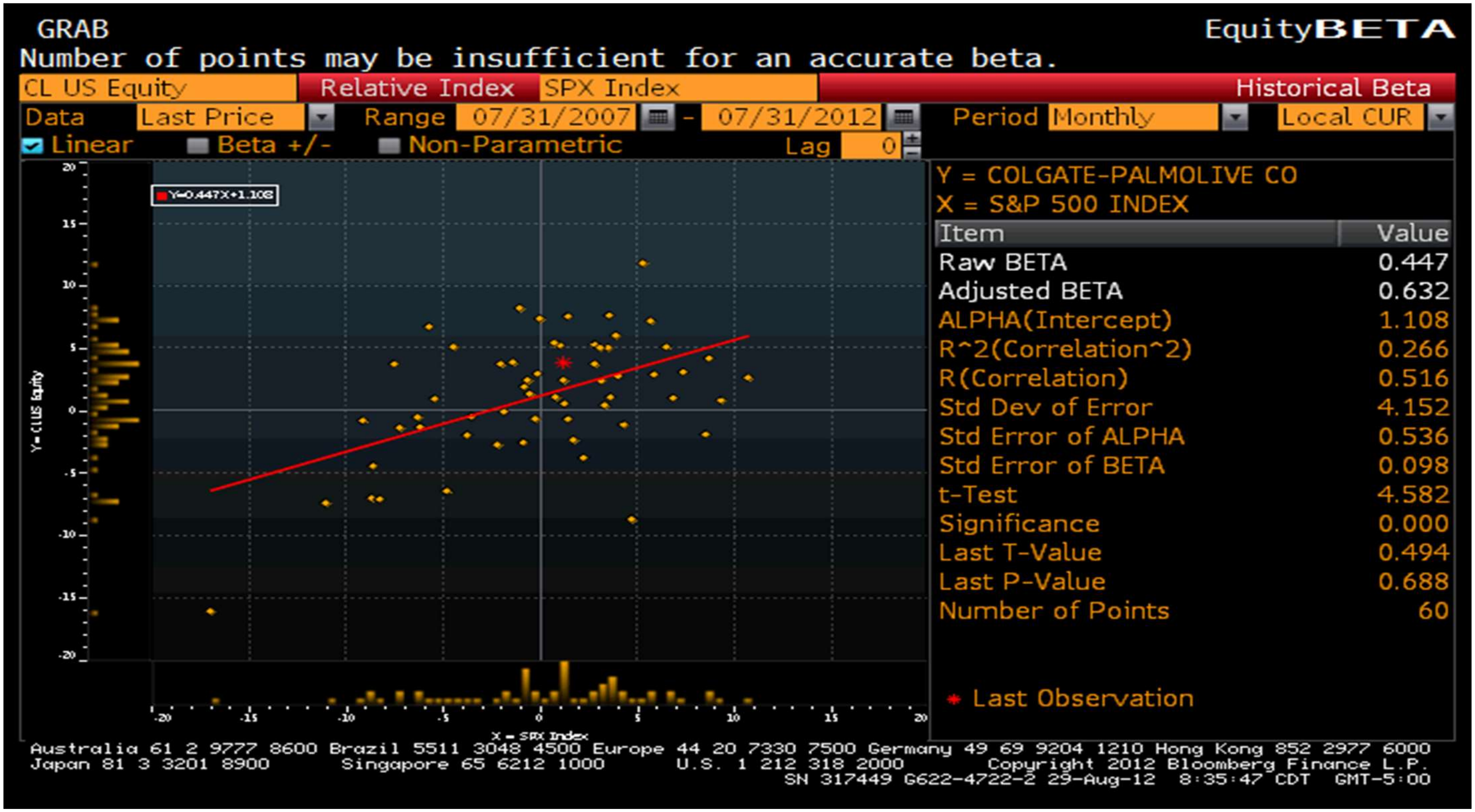
Y = APPLE INC
X = S&P 500 INDEX

Linear Beta	Range 1
Raw BETA	1.124
Adjusted BETA	1.083
ALPHA (Intercept)	0.316
R^2 (Correlation^2)	0.327
R (Correlation)	0.572
Std Dev of Error	2.717
Std Error of ALPHA	0.268
Std Error of BETA	0.160
t-Test	7.011
Significance	0.000
Last T-Value	-1.231
Last P-Value	0.111
Number of Points	103
Last Spread	1910.70
Last Ratio	0.056

Project a risk adjusted P/E Multiple based on qualitative issues & quantitative data. Try TST Answers today.

a. The Bloomberg Beta

- Notice again raw beta vs. adjusted beta (this time raw beta < 1)



GRAB EquityBETA

Number of points may be insufficient for an accurate beta.

CL US Equity | Relative Index | SPX Index | Historical Beta

Data: Last Price | Range: 07/31/2007 - 07/31/2012 | Period: Monthly | Local CUR

✓ Linear | Beta +/- | Non-Parametric | Lag 0

Y = COLGATE-PALMOLIVE CO
X = S&P 500 INDEX

Item	Value
Raw BETA	0.447
Adjusted BETA	0.632
ALPHA(Intercept)	1.108
R^2(Correlation^2)	0.266
R(Correlation)	0.516
Std Dev of Error	4.152
Std Error of ALPHA	0.536
Std Error of BETA	0.098
t-Test	4.582
Significance	0.000
Last T-Value	0.494
Last P-Value	0.688
Number of Points	60

```
Australia 61 2 9777 8600 Brazil 5511 3048 4500 Europe 44 20 7330 7500 Germany 49 69 9204 1210 Hong Kong 852 2977 6000
Japan 81 3 3201 8900       Singapore 65 6212 1000       U.S. 1 212 318 2000       Copyright 2012 Bloomberg Finance L.P.
                                                       SN 317449 G622-4722-2 29-Aug-12  8:35:47 CDT   GMT-5:00
```

a. The Bloomberg Beta

What's Raw Beta, what's Adjusted Beta?

Raw Beta is standard OLS Market Beta (2-years weekly data)

- Properties of OLS beta: Unbiased, consistent

Adjusted Beta: $\beta_{adj} = 0.33 \times 1 + 0.67 \times \beta_{raw}$

- Properties: ***Not*** unbiased, ***not*** consistent
- BUT: Smaller mean-squared error out of sample

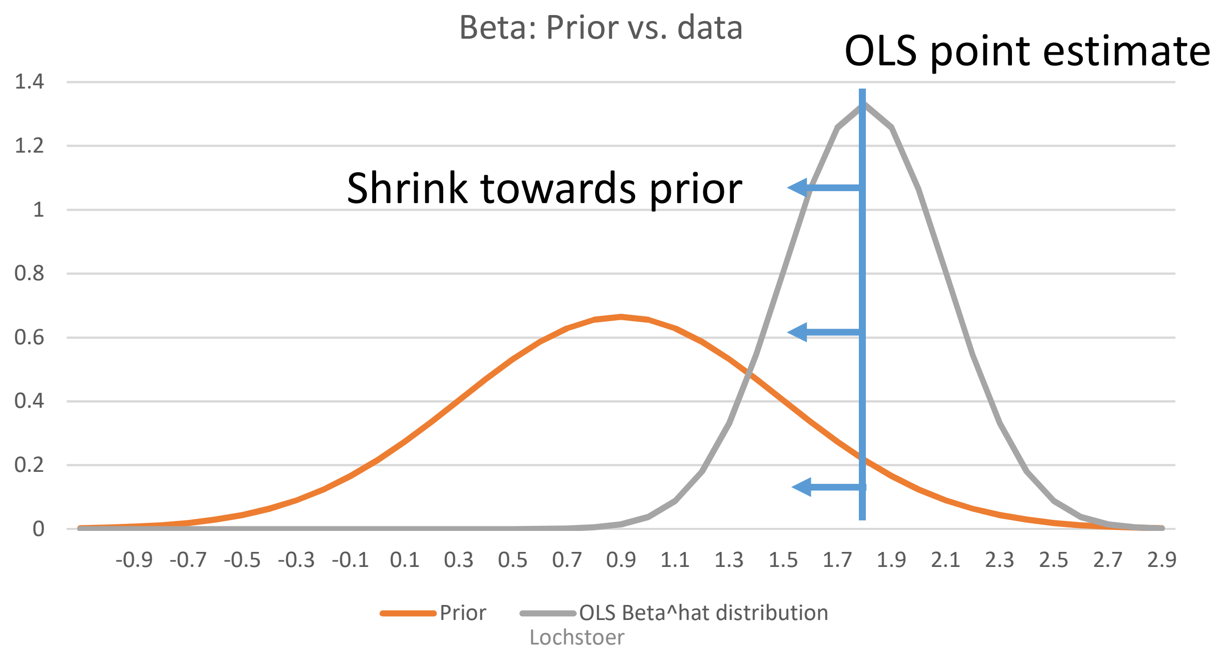
a. Bias vs. estimation error

Prior knowledge:

- Betas across firms are distributed with mean 1 and st.dev. around 0.5-0.7
- Approximately Normal distribution

From data (a particular regression):

- OLS estimated beta is 1.8, standard error 0.3, (approx.) Normal



UCLA Anderson
SCHOOL of MANAGEMENT

b. Shrinkage

Shrinkage is a well-known concept from ***Bayesian statistics***

- Similar in practice to regularization
- Improves out-of-sample mean-squared error

Next, we go through the Bayesian version for the simplest case:

- Estimating an unconditional average
- Example: ***estimate average return*** for a stock

$$\widehat{\mu}_i = \frac{1}{T}\sum_{t=1}^{T} R_{i,t}^{e}$$

UCLA Anderson SCHOOL of MANAGEMENT

b. Shrinkage

Some of the expected return estimates will be ***unreasonable***

- Expected excess returns of, say, 32% or -16% seem too extreme
- Why?

Perhaps you have some underlying model of investor preferences and beliefs in mind, for instance the CAPM

- Long-run market betas are typically between 0 and 2 (95% confidence interval)
- So, expected excess returns should be between 0% and 16%, if market risk premium is 8%
- That implies a standard deviation in the cross-section of true expected returns of 4%.

How can we incorporate such 'prior' beliefs?

UCLA Anderson SCHOOL of MANAGEMENT

b. Shrinkage (cont'd)

Solution: 'shrink' your estimates towards the prior distribution

$$\mu_i^{shrunk} = w_i \widehat{\mu_i} + (1 - w_i)\mu_{prior}$$

where μ_{prior} is the unconditional mean of your prior distribution

- Intuitively, the weight should be a function of how informative your prior is relative to the standard error of your estimate
- For instance, if your estimate has zero standard error, it is truth and we should have $w_i = 1$.
- Conversely, if the estimate has infinite standard error it is useless and we should have $w_i = 0$.

Bayesian inference formalizes this logic and tells you what w_i is

UCLAAnderson
SCHOOL of MANAGEMENT

b. Shrinkage and Bayesian Inference

Bayesian inference starts from the proposition that we can use probability statements to assess our views about a model parameter like a regression coefficient.

The idea is very appealing:

> ***Given our information (data and, e.g., economic theory), where do we think it "likely" that the parameters are?***

To translate "likely" into something concrete, we say we would like to make probability statements about a parameter.

For example, suppose we are trying to make inferences regarding expected returns. We would like to say, e.g., "the probability is .95 that the expected (annual) excess return is between 0% and 15%."

UCLAAnderson
SCHOOL of MANAGEMENT

b. Bayesian Inference and Bayes Theorem

We need to start with some basic ideas from probability. Let's start with the joint distribution of two random variables, (x,y). This is represented by the joint or bivariate distribution. For continuous random variables, we use a joint or bivariate density function.

$$p_{X,Y}\left(x,y\right)$$

This is a surface over the (x,y) plane and provides the probability rate per unit of volume. The joint distribution of two random variables can always be written as (conditional times marginal).

$$p_{X,Y}\left(x,y\right)=p_{Y|X}\left(y\,|\,x\right)p_X\left(x\right)$$

b. Bayesian Inference and Bayes Theorem

Here the marginal distribution of X is obtained by integrating out Y from the joint or averaging all the conditional distributions:

$$p_X(x) = \int p_{X,Y}(x,y)\,dy$$

Let's consider a simple case. Suppose X,Y are bivariate normal with a simple covariance matrix

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N\left(\begin{matrix} \mu_X \\ \mu_Y \end{matrix}, \Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \right)$$

$$X \sim N(\mu_X, 1)$$
$$Y \mid X = x \sim N\left(\mu_Y + \rho(x - \mu_X), 1 - \rho^2\right)$$

UCLA Anderson SCHOOL of MANAGEMENT

b. Shrinkage and Bayesian Inference

Bayes theorem tells us how to do this.

We start with a prior distribution over the model parameters

$$p(\theta)$$

And a model (i.e. joint distribution for our data given the theta).

$$p(\text{data} \mid \theta)$$

In our case, the parameters are the true mean excess return (μ_i) and stock return variance for each stock (σ_i^2). This density is the likelihood function of the stock returns, viewed as a function of these parameters.

b. Shrinkage and Bayesian Inference

Example: Let's assume stock returns are *i.i.d.* normally distributed and consider a particular stock's return series with *T* observations (skip *i*'s). The likelihood of the "data," conditional on the parameters, is then:

$$\theta = \left(\mu, \sigma^2\right)$$

$$p(R_1^e, R_2^e, \ldots, R_T^e | \mu, \sigma^2) = \prod_{t=1}^{T} p(R_t^e | \mu, \sigma^2)$$

$$= (2\pi\sigma^2)^{-T/2} exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{T}(R_t^e - \mu)^2\right)$$

UCLA Anderson SCHOOL of MANAGEMENT

b. Bayesian Inference and Bayes Theorem

Bayes theorem.

Let D denote our "data."

$$p(\theta|D) = \frac{p(D,\theta)}{p(D)} = \frac{p(D|\theta)p(\theta)}{p(D)}$$

$$p(\theta|D) \propto p(D|\theta)p(\theta)$$

$$\text{Posterior} \propto \text{"Likelihood"} \times \text{Prior}$$

The distribution, $p(\theta|D)$, is called the posterior distribution (posterior for "after"). Bayes Theorem tells us how to update our Prior views after we see a set of data!

UCLA Anderson
SCHOOL of MANAGEMENT

b. The posterior

So, let's find our posterior

Let the prior distribution be μ ~ $N\left(\bar{\mu}, \sigma^2_{prior}\right)$ and assume σ^2 is known.

Let, σ^2_{Post}, denote the variance of the posterior distribution (work this out).

Then the posterior for μ is normal (since the likelihood and the prior are both normal; prove this yourself through a complete-the-squares argument):

$$p(\mu | R_1^e, R_2^e, \ldots, R_T^e, \sigma^2)$$

$$\propto exp\left(-\frac{1}{2\sigma^2_{Post}}\left(\mu - \left(w\frac{1}{T}\sum_{t=1}^{T} R_t^e + (1-w)\bar{\mu}\right)\right)^2\right)$$

$$where\ \ w = \frac{T\sigma^{-2}}{T\sigma^{-2} + \sigma^{-2}_{Prior}}$$

UCLA Anderson SCHOOL of MANAGEMENT

b. The Shrinkage Estimator

In sum, the best estimates of the means, given the data and your prior beliefs, are given by a weighted average of the sample mean and the prior mean where the weights depend on the dispersion in the prior and the standard error of the sample average:

$$\mu_i^{shrunk} = w_i \frac{1}{T}\sum_{t=1}^{T} R_{i,t}^{e} + (1 - w_i)\mu_{prior}$$

$$where \ \ w_i = \frac{T\sigma_i^{-2}}{T\sigma_i^{-2} + \sigma_{Prior}^{-2}}$$

Incorporating some notion of ***shrinkage*** or ***regularization*** typically will improve out-of-sample fit

- Also, provides an estimate even for stocks that have absolutely no return history (e.g., new listing)

UCLA Anderson SCHOOL of MANAGEMENT

b. The Shrinkage Estimator

In our example, $\sigma_i \; = 40\%$ and $\sigma_{Prior} = 4\%$.

Thus: $w_i = \frac{25\times 0.4^{-2}}{25\times 0.4^{-2}+0.04^{-2}} = 0.2.$

With $\frac{1}{T}\sum_{t=1}^{T} R_{i,t}^{e} = -16\%$ we get

$$\mu_i^{shrunk} = 0.2 \times (-16\%) + 0.8 \times 8\% = 3.2\%$$

Very strong shrinkage due to the fact that even 25 years of past returns provides a very noisy signal of the mean return when return volatility is high (40% p.a.)

UCLAAnderson
SCHOOL of MANAGEMENT

b. Where do priors come from?

Great question!

- Bayes rule does not provide guidance on this issue…

Three approaches:

1. ***Uninformative priors and historical data***
   - Start with a flat prior at the beginning of earliest historical data
   - Get posterior at end of historical sample
   - Use this posterior as your prior for data going forward

2. ***Economic theory / introspection***
   - Prior means: Average market beta is 1, average HML beta is 0
   - Prior dispersion: Very few firms counter-cyclical (i.e., market beta < 0), so st.dev of prior on market beta perhaps in the 0.5-0.75 range…

3. ***Cross-validation and regularization (discussed next)***

c. Ridge Regressions

Let’s carry this idea over to regressions.

In particular, estimation error and overfitting means regression coefficients (betas) are away from their true value.

Let’s ***penalize*** the objective function when coefficients get too far away from an hypothesized true value

In the Ridge regression, this hypothesized true value is zero for all betas

UCLA Anderson
SCHOOL of MANAGEMENT

c. Recall: OLS objective function

Consider the OLS regression:

$$y_i = \beta_0 + \sum_{k=1}^{K} \beta_k X_{ki} + \epsilon_i$$

The objective function is

$$\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{k=1}^{K} \beta_k X_{ki} \right)^2$$

Also, recall that the fit (R2) is not affected if the right hand side variables (X) are demeaned and standardized to have standard deviation or norm of 1.

In addition, we for convenience assume the left hand side variable (y) will be demeaned. Thus, the intercept can be dropped as it will equal zero.

c. Ridge regression objective function

The objective function of the ridge regression (using variables normalized as described on the previous slide) is

$$\min_{\beta} \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \sum\nolimits_{k=1}^{K} \beta_k X_{ki} \right)^2 \quad s.t.\ \sum\nolimits_{k=1}^{K} \beta_k^2 \leq B$$

Here, *B* is the "coefficient budget" the regression is given.

It is the nature of the constraint that makes the standardization of the signals (X) useful; each signal is given equal importance in the objective function.

One can un-demean all variables (y and X) by setting $\widehat{\beta_0} = \bar{y} - \sum_{k=1}^{K} \widehat{\beta_k} \overline{X_k}$

c. Ridge regression objective function

The objective function in ***penalty form***:

$$\min_{\beta} \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \sum_{k=1}^{K} \beta_k X_{ki} \right)^2 + \lambda \sum_{k=1}^{K} \beta_k^2$$

There exists a one-to-one mapping between *B* and $\lambda > 0$.

Clearly, for a small enough *B*, the coefficients will be ***regularized***, effectively shrunk towards zero. Note that the coefficients will then be biased.

The coefficients can be found by simply solving the Lagrangian problem, which is nicely behaved since the problem is convex, continuous and differentiable everywhere.

Let's consider some logistic Ridge regressions using the default model from the last topic notes

UCLAAnderson
SCHOOL of MANAGEMENT

c. The scikit-learn package

Scikit-learn (www.scikit-learn.org) is a Machine Learning package for Python

- sklearn.metrics and sklearn.linear_models are sub-packages we import
- First, assess Lecnding Club model using only dummies based on “grade”

```
LC_RawData  = pd.read_stata('LendingClub_LoanStats3a_v12.dta')
LC_Baseline = LC_RawData[(LC_RawData.loan_status=='Fully Paid') | (LC_RawData.loan_status=='Charged Off')  ]
LC_Baseline['Default'] = LC_Baseline['loan_status'] == "Charged Off"
LC_Baseline['Default'] = LC_Baseline['Default'].astype('int') # bool to int

# Define a dependent variable
y = LC_Baseline.Default

grade_dummy = pd.get_dummies(LC_Baseline.grade)

# Define regressors
X = grade_dummy.drop(['A'], axis = 1)

# Define the range of lambda (the degree of penalty)
alphas = 10**np.linspace(10,-1,50)*0.02

# Logistic Ridge regression, set some parameters
logistic_ridge = LogisticRegression(penalty='l2', solver='liblinear', max_iter = 500, fit_intercept = True)

# Define function to compute residual deviance
def deviance(X, y, model):
    return 2*metrics.log_loss(y, model.predict_proba(X), normalize=True)

# Run regression
for i, a in enumerate(alphas):
    logistic_ridge.C = 1/a   # set the severity of the constraint
    logistic_ridge.fit(X, y)
    coefs.iloc[i] = logistic_ridge.coef_
    devi.iloc[i] = deviance(X, y, logistic_ridge)
```

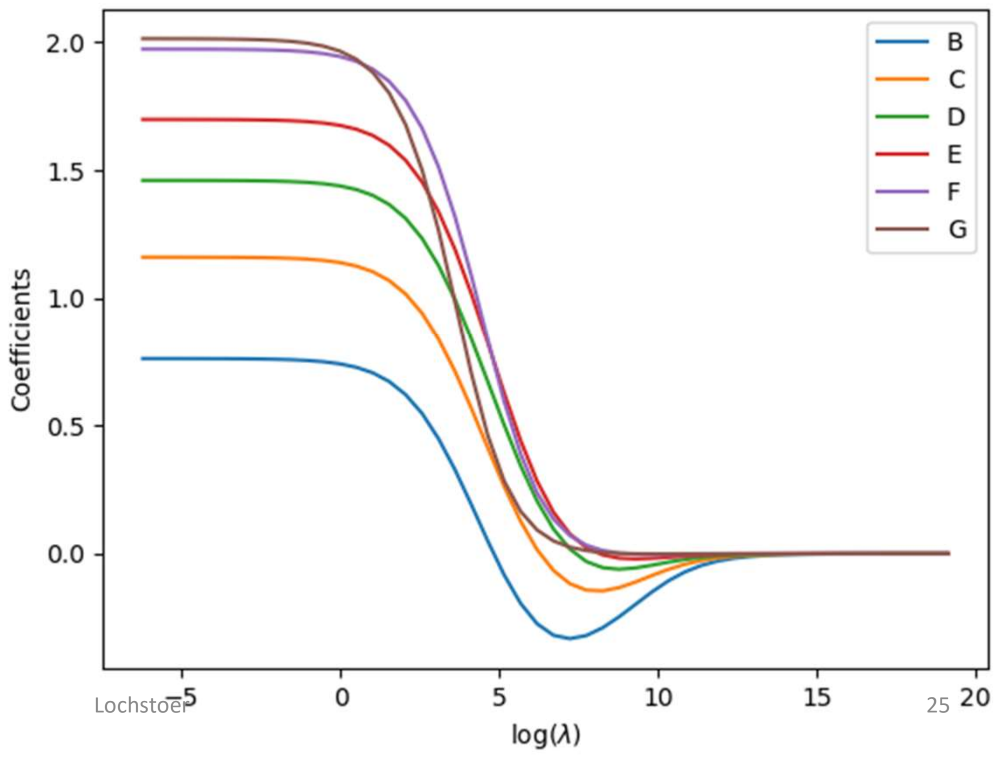
c. Ridge regression: Simple default model

Plot the log of lambda (1/alpha, in code) versus the estimated coefficients

- A high value of *lambda* means the constraint is tight, a low value means it is not.
- On left hand side, constraint does not bind at all and we have OLS coefficients

```
log_alphas = np.log(alphas)
ax.plot(log_alphas, coefs)
```

- Grade A was dropped as a dummy as it is the baseline case against which all others are measured.

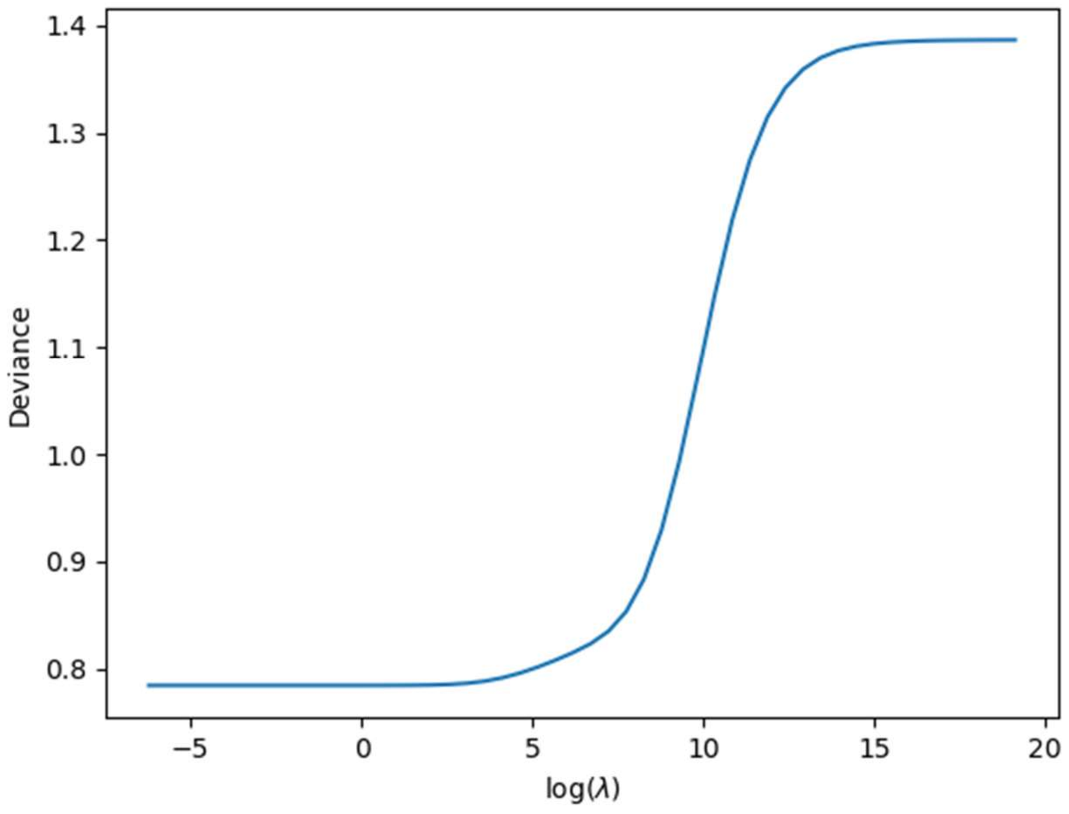


c. Ridge regression: Simple default model

Next, show the residual deviance of each of these models

- As expected, increasing the severity of the constraint increases deviance as we explain less of the data *in sample*

```
# Plot residual deviance over log lambda
plt.clf()
ax = plt.gca()
ax.plot(log_alphas, devi)
plt.axis('tight')
plt.xlabel('$\log(\lambda)$')
plt.ylabel('Deviance')
```



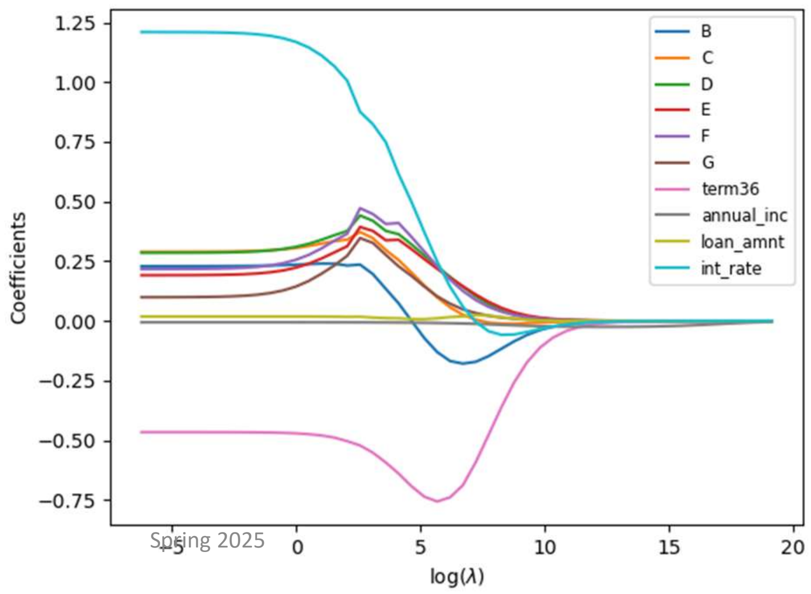
UCLAAnderson
SCHOOL of MANAGEMENT

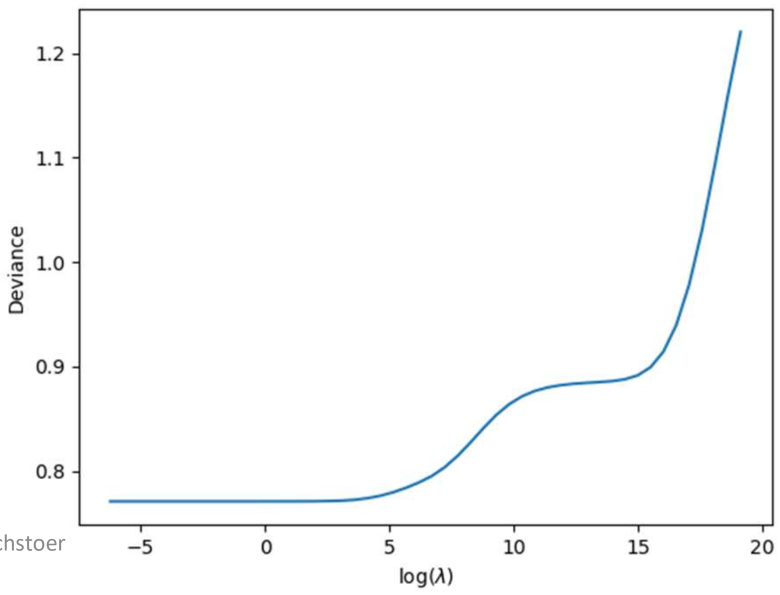
c. An extended default model

- In addition to Lending Club's "grade", we now add loan amount, annual income, term of loan (two dummies for up to 36 month, and then to 60 months), and the interest rate on the loan

```
# Run regression with more inputs
X2 = pd.concat([X, term_dummy.term36, LC_Baseline.annual_inc/1000, \
                LC_Baseline.loan_amnt/10000, LC_Baseline.int_rate*10],\
               axis = 1)
for i, a in enumerate(alphas):
    logistic_ridge.C = 1/a
    logistic_ridge.fit(X2, y)
    coefs2.iloc[i] = logistic_ridge.coef_
    devi2.iloc[i] = deviance(X2, y, logistic_ridge)
```

Note nonlinearities: interactions between regressors matter, and deviance is lower (as we'd expect)





UCLAAnderson
SCHOOL *of* MANAGEMENT

d. Ridge regression: Which *B* (or *lambda*)?

Two questions are immediate:

1. How do you choose the level of the constraint *B* (or "*lambda*")?
2. How do you compute standard errors of the coefficients?

We will get back to 2. It is a little subtle as there is bias in the coefficients, so the standard error can severely understate the total uncertainty

Question 1. is handled through ***Cross-Validation***

- Cross-validation is an ***out-of-sample model selection criterion***
- Note that for each different B (lambda), we have a different model

d. Cross-validation

Basic idea

- Split up the sample in random training and test sets and estimate performance on test sets.
- Choose the level of the constraint that gives best prediction

K-folds

- Split up the sample in K equal-sized groups (typically, 5 or 10)
- For each of the K folds, estimate the model on the other K-1 folds and test on the K'th fold (so K 'out-of-sample' tests)
- The prediction error (typically, it's mean squared error, averaged across folds) is the basis for the choosing the best model

LogisticRegressionCV implements this!

- That is, it finds the value of the "C" of the code that minimizes K-fold MSE

d. Optimal constraint, and prediction

```
# Cross-validation, choosing 10 folds, L2 refers to Ridge penalty
logistic_ridge_CV = LogisticRegressionCV(penalty='l2', \
                                         solver='liblinear', \
                                         Cs = 1/alphas, \
                                         cv = 10, \
                                         max_iter = 500, \
                                         fit_intercept = True)

logistic_ridge_CV.fit(X2, y)

# Compute best \lambda
best_alpha = "%0.4f" %  np.log(1/logistic_ridge_CV.C_)
print("The best $\log(\lambda)$ is", best_alpha)
```

```
The best $\log(\lambda)$ is 0.5052
```

```
# predict default probability given a set of feature values
feature = [1, 0, 0, 0, 0, 0, 1, 200000/1000, 50000/10000, 0.05*10]
feature = pd.DataFrame([feature])
prob_default = "%0.4f" % logistic_ridge_CV.predict_proba(feature)[0,1]
print("The probability of default is", prob_default)
```

```
The probability of default is 0.0277
```

UCLA Anderson SCHOOL of MANAGEMENT

e. The LASSO

LASSO: Least Absolute Selection and Shrinkage Operator

Same idea as Ridge regression, but penalty function is not sum of squared coefficients, but some of absolute values. Mathematically, the LASSO has an L_1 constraint, while Ridge has an L_2 constraint:

$$\min_{\beta} \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \sum_{k=1}^{K} \beta_k X_{ki} \right)^2 \quad s.t. \; \sum_{k=1}^{K} |\beta_k| \leq B$$

Here, *B* is the "coefficient budget" the regression is given.

It is the nature of the constraint that makes the standardization of the signals (X) useful; each signal is given equal importance in the objective function.

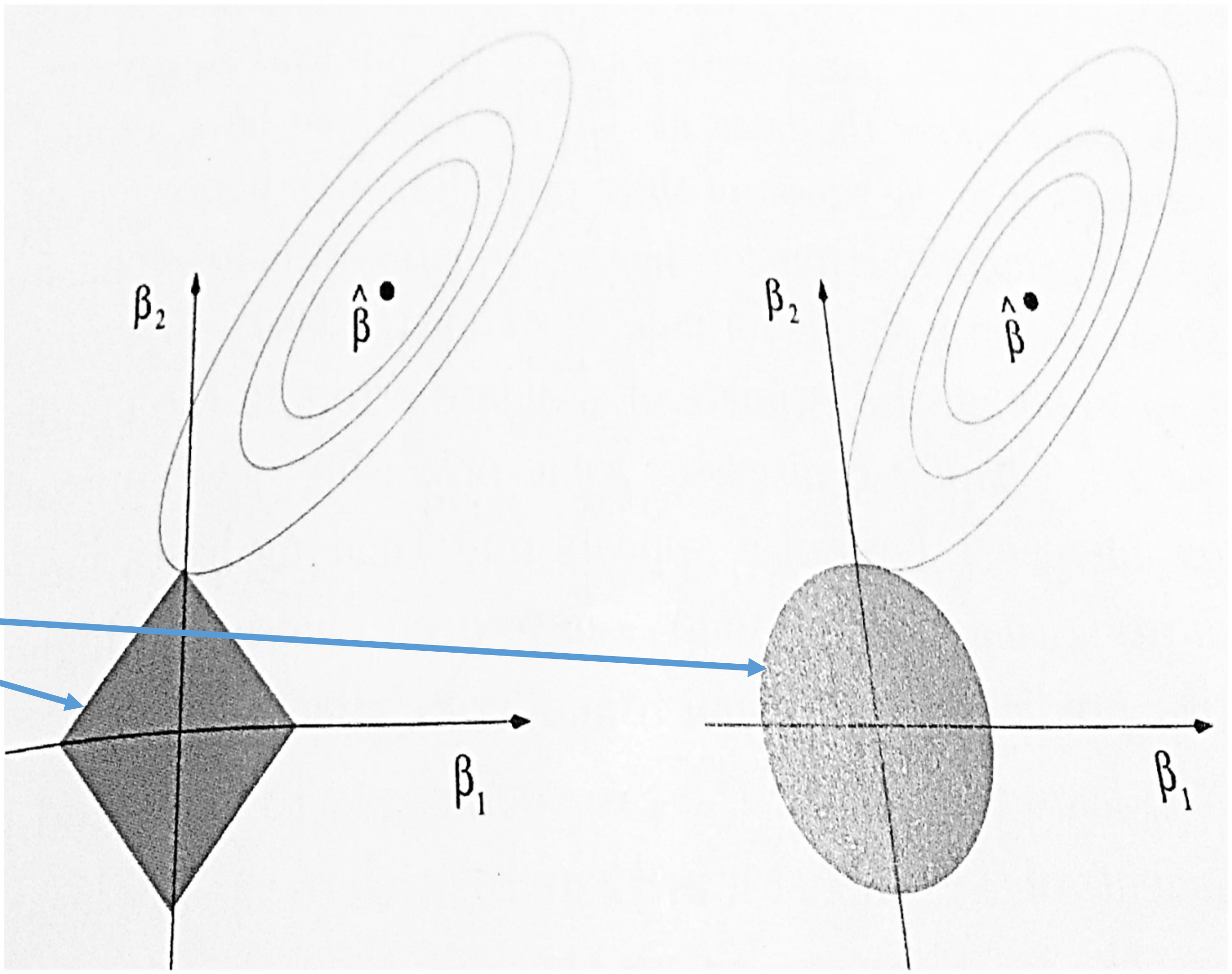
One can un-demean all variables (y and X) by setting $\widehat{\beta_0} = \bar{y} - \sum_{k=1}^{K} \widehat{\beta_k} \overline{X_k}$

e. Lasso (L1) vs. Ridge (L2) penalty

Beta^hat is OLS estimate

Ellipses are contours for SSEs

Feasible set as given by constraint



Notice that LASSO is much more likely to land on a corner solution where one of the betas equal zero

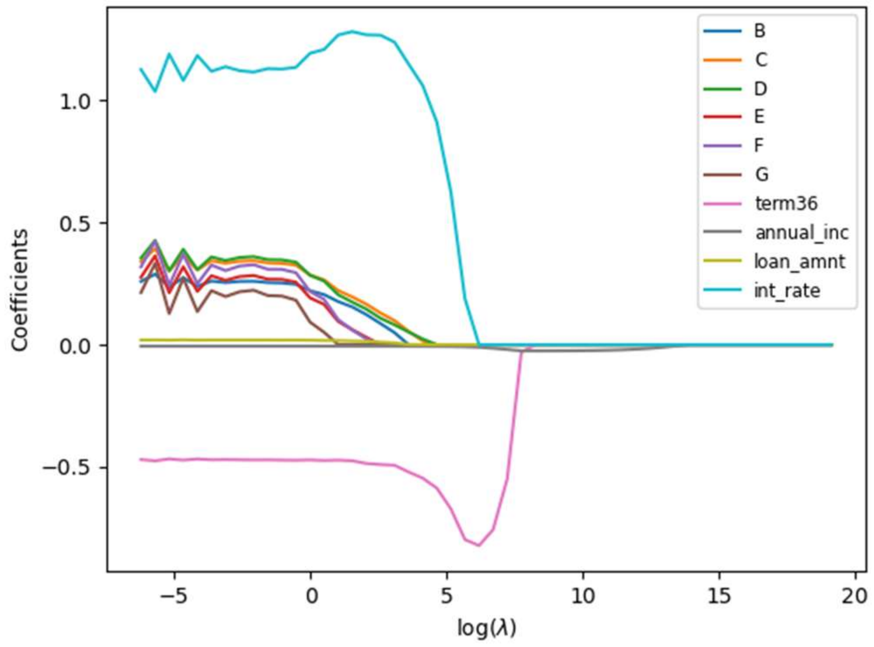
e. An extended default model

- Again, in addition to Lending Club's "grade", we now add loan amount, annual income, term of loan (two dummies for up to 36 month, and then to 60 months), and the interest rate on the loan
- We run the LASSO, by setting 'penalty' to 'L1'

```
# Define Lasso
logistic_lasso = LogisticRegression(penalty='l1', solver='liblinear', max_iter = 500,
fit_intercept = True)

# Run Lasso regression
for i, a in enumerate(alphas):
    logistic_lasso.C = 1/a
    logistic_lasso.fit(X2, y)
    coefs2.iloc[i] = logistic_lasso.coef_
    devi2.iloc[i] = deviance(X2, y,
                             logistic_lasso)
```

Notice that, different from Ridge, a lot of coefficients are set to zero early on – sparsity!



UCLAAnderson
SCHOOL of MANAGEMENT

e. LASSO: Best constraint, and prediction

```
# Cross-validation
logistic_lasso_CV = LogisticRegressionCV(penalty='l1', \
                                         solver='liblinear', \
                                         Cs = 1/alphas, \
                                         cv = 10, \
                                         max_iter = 500, \
                                         fit_intercept = True)

logistic_lasso_CV.fit(X2, y)

# Compute best \lambda
best_alpha = "%0.4f" %  np.log(1/logistic_lasso_CV.C_)
print("The best $\log(\lambda)$ is", best_alpha)

# Prediction (feature same as above)
prob_default = "%0.4f" % logistic_lasso_CV.predict_proba(feature)[0,1]
print("The probability of default is", prob_default)
```

- Predicting the default probability for a person with the above attributes (loan of $50,000, <=36 month maturity, annual income of $200,000, etc.)
- Using the lambda (.C) that gives the smallest MSE in 10-fold cross-validation exercise: *ln(lambda) = 3.6066* (refer back to previous slide to see coefficients)
- *Probability of default: 0.0243*, different from Ridge, more parsimonious model

UCLA Anderson SCHOOL of MANAGEMENT

e. Elastic net

Finally, can also set penalty to ‘elasticnet’ if using the ‘saga’ option for solver. This corresponds to the Elastic net estimator:

$$\min_{\beta}\sum_{i=1}^{N}\left(y_i-\sum\nolimits_{k=1}^{K}\beta_k X_{ki}\right)^2 \quad s.t.\ \sum\nolimits_{k=1}^{K}(1-\alpha)\beta_k^2+\alpha|\beta_k| \le B$$

The LASSO has problems with highly correlated X values. Erratic solution (switches between which coefficients are non-zero with large swings)

Adding a little of the Ridge penalty makes the problem strictly convex, which in cases with such high correlations between variables is a desirable property

UCLA Anderson
SCHOOL of MANAGEMENT

f. Bayesian Inference and Bayes Theorem

Bayesian methods are becoming increasingly popular in a wide variety of industry contexts. The reason for their popularity is that Bayesian estimators impose a form of what statisticians call "regularization" or some call "shrinkage."

To obtain an intuition for this, consider the standard Least Squares estimator for a linear regression model.

$$b = \left(X'X\right)^{-1} X'y$$

Consider the case of one independent variable, suppose there is no variation in that variable (or very little). Then you simply can't run the regression (true also for logistic regression models).

$$X = \begin{bmatrix} \iota & x_1 \end{bmatrix}$$

f. Bayesian Inference and Bayes Theorem

In the case of a very tiny amount of x variation, you will get a huge standard error and your estimates can be all over the place. If you are using the coefficients from these models to make allocation decisions or attribution inferences, then your model will produce absurd results and you will soon lose credibility.

A “regularization” procedure will take the least squares (or MLEs) and modify them so that they are more reasonable as well as have better sampling properties. Bayesian methods provide one type of regularization called shrinkage.

Barriers to using Bayesian methods: you have to understand more about distributions and densities and you have to use simulation as your method of computing!

f. Bayesian References

Chapter 19 of Lander (very brief and with almost no explanation).

Bayesian Statistics and Marketing, Rossi, Allenby and McCulloch (detailed explanations along with accompanying R package, bayesm).

Bayesian Data Analysis, Gelman et al (good reference but weak on computing and priors).

Markov Chain Monte Carlo, Gammerman and Lopes (good more advanced treatment of Bayesian Simulation Methods).

UCLA Anderson SCHOOL of MANAGEMENT

f. Bayes Regression

Now that we have the idea of a Bayesian method, let's consider regression from a Bayesian point of view.

What is the regression model likelihood function or distribution of the observed data?

$$y \mid X \sim N\left(X\beta, \sigma^2 I_n\right)$$

$$p\left(y_1,\ldots,y_n \mid \beta,\sigma^2\right) = \left(2\pi\sigma^2\right)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - x_i'\beta\right)^2\right)$$

Now, we will need a prior on both beta and sigma-squared. To obtain the intuition, we will ignore the prior on sigma-squared for now.

UCLA Anderson SCHOOL of MANAGEMENT

f. Bayes Regression – Prior on Beta

What prior distribution should we use for beta? A particularly convenient one is

$$\beta \sim N\left(\bar{\beta}, A^{-1}\right)$$

Obviously, ***beta-bar is the prior mean***. ***A is called the prior "precision" matrix***. Variance is the opposite of precision. That is, if the elements of A are very large then we have a lot of "precision" or "tightness" in the prior or a small prior variance.

How should we pick or "assess" the prior parameters? This is hard to do. One default is to use a relatively uninformative prior:

$$\beta \sim N(0, [0.1 \times I_n]^{-1})$$

f. Bayes Regression – Prior on Beta

$$\beta \sim N(0, [0.1 \times I_n]^{-1})$$

Here the prior variance is 100 for each beta element or a prior standard deviation of 10. What we are saying is that we believe with probability .95 that each beta lies in the interval, [-20,20].

This is a very diffuse (variable) prior which appears to impart little information (note: all of the X variables are assumed to be on the same scale!).

Another suggestion for A:

- Estimate the regression betas on an earlier sample, obtain their standard errors, and use the these to form a *data-driven prior* as we did earlier when shrinking the mean return estimate

UCLA Anderson SCHOOL of MANAGEMENT

f. Bayes Regression – Bayes Estimator

The posterior corresponding to this prior and the normal regression likelihood has a non-standard form (it is not normal). But it is possible to write down a formula for the posterior mean.

$$\tilde{\beta} = \left(X'X + A\right)^{-1}\left(X'y + A\bar{\beta}\right)$$

Note that this estimator works even if there are linear dependencies in the X’X matrix (i.e. no variation in some of the X variables or perfect co-linearity). A more suggestive version is

$$\tilde{\beta} = \left(X'X + A\right)^{-1}\left(X'X\hat{\beta} + A\bar{\beta}\right)$$

The Bayes estimator is a matrix-weighted average of Least Squares and the prior mean. That means that it “shrinks” the Least Square estimator toward the prior mean.

f. Bayes Regression, Ridge Regression

$$\tilde{\beta} = \left(X'X + A\right)^{-1}\left(X'X\hat{\beta} + A\bar{\beta}\right)$$

The amount shrinkage depends on the relative size of the X’X and A matrices.

If X’X is large (large N and/or a lot of variation in our data), there will be very little shrinkage. This applies variable by variable as well. If one of the variables has only a tiny bit of variation, then its coefficient will get shrunk a lot while others will not be shrunk much.

Even if the elements of A are very small, the Bayes estimator will “keep least squares out of trouble.”

In fact, this is the Ridge regression, where the magnitude of prior uncertainty, A, scales with the Lagrangian “lambda”

f. Bayes Regression, the LASSO

If we instead of a Normal prior, specify a Laplace distributed prior, we can view the LASSO estimator as a Bayesian regression:

$$\boldsymbol{y}|\beta,\lambda,\sigma \sim N(\boldsymbol{X}\beta,\sigma^2\boldsymbol{I}_{N\times N})$$

$$\beta|\lambda,\sigma \sim \prod_{j=1}^{p}\frac{\lambda}{2\sigma}e^{-\frac{\lambda}{\sigma}|\beta_j|}$$

The negative log posterior density (minus an unimportant constant term) is then of the same form as the LASSO objective function and so the solutions are the same (try to work this out yourself!)

UCLA Anderson
SCHOOL of MANAGEMENT

f. Bayes Regression, the LASSO

The Bayesian interpretation of the Ridge and the LASSO allows for computation/simulation of the ***posterior distribution*** of the regression coefficients, which allows for ***statistical inference*** and tests

This is a major benefit, as the LASSO and Ridge themselves, with just ad hoc constraints, cannot be used in conjunction with the standard OLS standard error apparatus (including White, Rogers, Hansen-Hodrick, and Newey-West standard errors).

- With the posterior distribution of a parameter, it is straightforward to ask, e.g., "What is the probability that this parameter is less than or equal to zero?"

UCLA Anderson SCHOOL of MANAGEMENT

g. “Machine Learning”-valuation

In problem set 4, you will code up a fully automated stock valuation model based on comparables.

- We will use cross-sectional regressions to form predicted market values
- The machine learning part comes from allowing interaction and non-linearities and letting the computer find the right model
- Once we have the predicted valuations, we can compare to the actual ones. Can we identify over- or under-valued stocks?
- The math and setup for the problem set follows on the next slides

UCLA Anderson SCHOOL of MANAGEMENT

g. "Machine Learning"-valuation

Assume the log market value of firm *i* and time *t* is linear in a set of firm characteristics:

$$me_{i,t} = a_t + b_t' X_{i,t} + \varepsilon_{i,t}$$

Characteristics can include book value, profitability, profitability^2, etc.

- The regression coefficients a and b have t subscripts as we will estimate these in a cross-sectional regression at each time *t*
- Predicted time *t* market value is then

$$\widehat{me_{i,t}} = a_t + b_t' X_{i,t}$$

g. “Machine Learning”-valuation

Denote mispricing as:

$$z_{i,t} = me_{i,t} - \widehat{me_{i,t}}$$

Is this signal a useful signal for trading stocks, above and beyond our standard signals?

How would you check this?

Shrinkage Postscript

a. Ridge and Bayesian Regression Revisited

b. Standard errors vs. posterior distribution

c. Numerical posterior distributions

UCLAAnderson
SCHOOL of MANAGEMENT

a. Ridge regression objective function

Recall Ridge Regression:

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{N} (y_i - \beta x_i)^2 \quad s.t. \;\; \beta'\beta \leq B$$

Here, *B* is the scalar "coefficient budget" the regression is given. Let's express it in matrix form with an "L2" constraint:

$$\min_{\beta} \frac{1}{2} \{(Y - X\beta)'(Y - X\beta) + \lambda\beta'\beta\}$$

The *K* first-order conditions are:

$$-X'(Y - X\beta) + \lambda\beta = 0$$

Solving, we get:

$$\beta = (X'X + \lambda I_K)^{-1} X'Y$$

UCLA Anderson
SCHOOL of MANAGEMENT

a. Bayesian Regression

Standard OLS, where we assume the residual variance (σ^2) is known, says that estimated “betas” are normally distributed around the “true” value

$$\hat{\beta} \sim N(\beta_0, (X'X)^{-1}\sigma^2)$$

Assume you have a Normal prior for the true Kx1 beta-vector

$$\beta_0 \sim N\left(\bar{\beta}, \sigma^2 I_K / A\right)$$

Where A is a scalar and I_K is the $K \times K$ identity matrix.

Recall our shrinkage-math from the beginning of Topic 4, where we discussed how to get the posterior mean using both a prior about the mean and the sample mean estimate. (see next slide)

UCLAAnderson SCHOOL of MANAGEMENT

a. Shrinkage Math

Recall shrinkage-math from beginning of Topic 4

$$\mu_i^{shrunk} = w_i \frac{1}{T} \sum_{t=1}^{T} R_{i,t}^e + (1 - w_i)\mu_{prior}$$

$$where \;\; w_i = \frac{T\sigma_i^{-2}}{T\sigma_i^{-2} + \sigma_{Prior}^{-2}}$$

The multiple regression analogue is:

$$\beta^{PosteriorMean} = w\hat{\beta} + (I_K - w)\bar{\beta}$$

$$where \;\; w = \left((X'X)\sigma^{-2} + A\sigma^{-2} I_K\right)^{-1} (X'X)\sigma^{-2}$$
$$= (X'X + AI_K)^{-1}(X'X)$$

UCLAAnderson
SCHOOL of MANAGEMENT

a. Bayesian Regression Posterior

The regression analogue is:

$$\beta^{PosteriorMean} = w\hat{\beta} + (I_K - w)\bar{\beta}$$

$$= (X'X + AI_K)^{-1}(X'X)\hat{\beta} + \left(I_K - (X'X + AI_K)^{-1}(X'X)\right)\bar{\beta}$$

$$= (X'X + AI_K)^{-1}\left(X'X\hat{\beta} + AI_K\bar{\beta}\right)$$

This can be written:

$$\beta^{PosteriorMean} = (X'X + AI_K)^{-1}\left(X'Y + AI_K\bar{\beta}\right)$$

a. Bayesian Regression Posterior

Since $X'X\hat{\beta} = X'Y$, we can write:

$$\beta^{PosteriorMean} = (X'X + AI_K)^{-1}\left(X'Y + AI_K\bar{\beta}\right)$$

With a prior centered around zero for all betas ($\bar{\beta} = 0$), we have:

$$\beta^{PosteriorMean} = (X'X + AI_K)^{-1}X'Y$$

which is the same as the Ridge Regression, with A = λ

Ok, but what about confidence bounds for $\beta^{Posterior}$?

b. Standard errors vs. posterior distribution

In standard, “frequentist” statistics, the data gives an estimate that is a random variable centered around truth (truth = null hypothesis). E.g.:

$$\hat{\beta} \sim N(\beta_0, (X'X)^{-1}\sigma^2)$$

In Bayesian statistics, the parameters themselves (not the estimate) are random variables with a distribution decoded in the prior, e.g.

$$\beta_0 \sim N\left(\bar{\beta}, \sigma^2 I_K / A\right)$$

and, after observing data, the posterior. In the case where we know the residual variance, σ^2, the posterior in our case is:

$$\beta^{Posterior} \sim N\left(\beta^{PosteriorMean}, (X'X + AI_K)^{-1}\sigma^2\right)$$

b. Standard errors vs. posterior distribution

The posterior in our case is:

$$\beta^{Posterior} \sim N\left(\beta^{PosteriorMean}, (X'X + AI_K)^{-1}\sigma^2\right)$$

Note that we have the variance of the posterior, which is Normal. Thus, it is now straightforward to make probabilistic statements, such as:

- “The 95% confidence bound for the true parameter is from ‘a’ to ‘b’ ”

But, (a) where did this math come from, and (b) what do we do in the real-world case where we don’t know σ^2?

- Regarding (a), all Bayesian linear regression math is standard and from the Topic 4 slides, you can derive everything yourself.
- Regarding (b), we will get there soon.
- That said, I also posted the Wikipedia-page(!) on it to BruinLearn Week 4.

c. Numerical posterior distributions

Assume the prior and the likelihood functions both are known analytically.

- In our example, that is indeed the case. The data (regression residuals) are normally distributed given β and σ^2.
- We assumed σ^2 is known (for now) and that the prior on β is Normal.

Easy to get an approximate posterior distribution for β:

1. Choose a set of grid values for β (e.g., K = 1,000 equal spaced points between -5 and +5 standard deviations of prior). Denote the *k*'th grid value $\beta^{(k)}$.
2. Get the associated *K* values of the Normal pdf of the prior, p_k^{prior}
3. Get the associated *K* values of the Normal pdf for the data (values of the likelihood function for each beta-value on the grid), l_k^{data}
4. For each of the *K* beta-values on the grid, multiply these two numbers. Denote the *k*'th such value $p_k^{posterior} = l_k^{data} \times p_k^{prior}$
5. Define the numerical (discretized) pdf as

$$p^{posterior}\left(\beta = \beta^{(k)}\right) = p_k^{posterior} / \sum_{j=1}^{K} p_j^{posterior}$$

c. Numerical posterior distributions

If σ^2 is unknown, we need a prior over both β and σ^2.

It turns out a *hierarchical prior* is useful here. In particular, define prior over β conditional on σ^2 similar to before:

$$p(\beta|\sigma^2) = N\left(\bar{\beta}, A\sigma^2\right)$$

Next, assume the prior on σ^2 is *Inverse Gamma* distributed

$$p(\sigma^2) = IG(a, b)$$

Then, the joint prior distribution is:

$$p(\beta, \sigma^2) = p(\beta|\sigma^2)p(\sigma^2)$$

c. Numerical posterior distributions

Note that, the marginal prior distribution of β, $p(\beta)$ is not known analytically, though it is easy to simulate. (we will do this in a second)

In this case, the posterior distribution is known to be Normal-Inverse-Gamma in the same conditional way as the prior (posterior distribution of beta conditional on σ^2 is Normal, posterior of σ^2 is Inverse Gamma)

- See Wikipedia-handout for background math

How do we get confidence bound on beta? Compute marginal distribution numerically (through simulation this time)!

1. Draw a value from the posterior over $\sigma^2, p(\sigma^2)$
2. Using this value for σ^2, draw a value of β from the posterior $p(\beta|\sigma^2)$
3. Repeat lots of times
4. Compute histogram for β based on simulated data
5. Use this histogram as discretized marginal probability density function for β