Introduction to Machine Learning for Economists: Lecture 1

Sonan Memon
Institute of Business Administration,
Karachi

7th December 2020

Course Information

i. See course outline for all information.

ii. The first part (Lec 1-2) of course will be conducted on Dec 7 and 8th, 2 to 4 pm.

iii. Total of four lectures, each lasting for two hours.

iv. We will cover shrinkage estimators in Lecture 1, Bandit Problems in Lecture 2, Deep Neural Networks in Lecture 3 and Computational Linguistics or Text Analysis in Lecture 4

v. Focus is on concepts, not computational implementation.

Textbooks, Readings and Resources

i. See course outline for full list of readings. Some major books are:

ii. Hastie, T., Tibshirani, R., & Friedman, J. (2017). The elements of statistical learning.

iii. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: Springer.

iv. Course GitHub Page contains course outline, lecture slides and all other resources https://github.com/sonanmemon/Introduction-to-ML-For-Economists

Readings for Lecture 1

i. Alpaydin, Ethem (2010), Introduction to Machine Learning, 2nd edition, MIT Press, Chapter 1.

ii. Robert, C. (2007). The Bayesian choice: from decision-theoretic foundations to computational implementation. Springer Verlag, chapter 2.

iii. Hastie, T., Tibshirani, R., & Friedman, J. (2017). The elements of statistical learning, Chapter 3.4.

iv. Stachurski, John (2016), A Primer in Econometric Theory, MIT Press (Chapter 14).

Programming Tips

i. R :) :) (cool for econometrics, statistics, ggplot is cool)

ii. Julia :) :) (as fast as C++, as easy to use as MATLAB, good for complicated structural models)

iii. Python: I would say combination of R and Julia makes Python redundant for econometrics but widely used in ML and data science; good general programming language.

iv. Dynare :), MATLAB :— (Dynare is cool for solving DSGE models, MATLAB is becoming outdated)

v. Stata :((good for regression monkeys but boring, outdated and dominated by R)

vi. Maple, Mathematica: Whatever (Not used much in Econ)

vii. LaTeX :) :) (absolutely essential)

Outline of Lecture 1

i. Introduction to Machine Learning
ii. Decision Theory: Loss Functions, Risk, Bayesian approaches: Integrated Risk, Posterior Expected Loss, Bayes' Risk.
iii. Cross Validation and Variance/Bias Trade off
iv. Shrinkage Estimators such as SURE, Ridge Regression and LASSO
v. Brief Discussion of Economic Application of LASSO if time permits

What is Machine Learning (ML)?

i. ML refers to the set of tools, computer programs, algorithms or technologies which enable computers to learn patterns from training data without being explicitly programmed to do so.

ii. ML use training data to learn patterns, estimate a mathematical model and make predictions in hold out sample/out of sample based on new input data.

iii. Capacity to find complex, flexible and crucially *generalizable* structure in training data. ML algorithms can be thought of as complex function approximation techniques, outperforming traditional methods.

iv. $ML \subset AI$. $AI \neq ML$.

History of Machine Learning

i. In 1950, Alan Turing created the world-famous Turing Test.

ii. 1952 saw the first computer program which could learn as it ran, made for playing checkers, created by Arthur Samuel, who also coined the term ML.

iii. In 1957, Frank Rosenblatt – at Cornell created the "perceptron" which had been constructed for image recognition.

iv. In 1967, the nearest neighbor algorithm was conceived, which was beginning of basic pattern recognition.

History of Machine Learning

i. In the 1960s, it was discovered that multi-layers in the perceptron offered significantly more processing power than when using one layer..

ii. Backpropagation, developed in the 1970s, allows a network to adjust its hidden layers of neurons to adapt to new situations. This is used to train "deep" neural networks.

iii. DeepFace: A Deep Neural Network created by FB, which claimed in 2014 that it could recognize faces as well as humans do

iv. DeepMind's AlphaGo program defeated world Go champion in 2016.

v. Amazon Machine Learning Platform (2015)

Types of ML Approaches with Applications

i. Supervised learning involves training data on inputs X *and* output Y to learn the mapping $f(X) = Y$. If Y is discrete, we are dealing with a classification problem and if Y is continuous, we are dealing with regression problem.

ii. Classification problem examples are spam filtering and face recognition, meanwhile regression example is estimating probability of disease given patient characteristics for example or estimating $P(Y|X)$.

iii. Unsupervised learning does not try to learn $f(X) = Y$ but digs out patterns and regularities in input space X, no data on Y/no supervision. Clustering algorithms and latent dirichlet allocation in topical analysis of text data are examples.

Other ML Approaches and Algorithms

i. Reinforcement Learning: dynamic interaction with environment and learning dynamic policy rules. Applications: DeepMind.

ii. Neural Networks. Application: Image recognition but also many more things.

iii. Regression Trees and Random Forests

iv. Shrinkage: LASSO, Ridge Regression, PCA. Application: Dimensionality Reduction.

v. Ensemble approaches.

More Applications

i. Fraud detection, speech recognition, translation and analysis of text, recommendation systems.

ii. Image compression, medical diagnosis, DNA sequencing, automation.

iii. Sales prediction for supermarkets, customer segmentation and market research, stock market prediction, house price prediction.

Economic Applications with Big Data

i. Training neural nets on satellite data to predict local economic outcomes in African countries Jean et al (2016).

ii. Cellphone usage data has been used to measure wealth, quantifying poverty in Rwanda at the individual level Blumenstock et al (2016).

iii. Text as data: Financial board messages were classified as bullish, bearish or neutral to help predict stock market volatility Antweiler and Frank (2004).

iv. Heterogeneous treatment effects and their use in assigning treatments in experiments such as those involving price targeting: Misra and Dubé (2016).

Economic Applications with Big Data

i. Heterogeneous treatment effects and their use for assigning treatments in experiments such as those involving price targeting: Misra and Dubé (2016).

ii. Preprocessing using traditional data sets by matching individuals such as fathers and sons across historical records for instance Feigenbaum (2015a, b) and predicting unobserved variables by training ML on small scale experiments Bernheim, Bjorkegren, Naecker, and Rangel (2013).

iii. Prediction policy problems such as bail problem, refugee assignment to jobs and hiring decisions such as for teachers Chalfin et al. (2016).

Overfitting Problem and Regularization

i. Overfitting problem exists due to complexity and flexibility of ML functional forms.

ii. Regularization solves the trade off between complexity and out of sample fit or variance bias trade off.

iii. k fold cross-validation on validation samples or splits of training data but firewall principle means hold out sample must not be used in this process.

iv. $\min_{f \in \mathcal{F}} \sum_{i=1}^{n} L(f(x_i), y_i)$ subject to $R(f) \leq c$ where c measures complexity such as depth of regression tree or tuning parameter in LASSO.

Relationship Between ML and Econometrics

i. ML is excellent at solving prediction problems but does not allow clear inference for coefficients due to non-existent standard errors, instability of coefficients across splits and inconsistent model selection.

ii. Regularization penalizes complexity, selecting wrong model in order to avoid overfitting which makes coefficients biased.

iii. Related issue is that variables observed but excluded by LASSO for example may be correlated with included variables, leading to a version of omitted variable bias.

Relationship Between ML and Econometrics

i. The coefficients in ML models do not have causal and policy, relevant structural interpretations unless we impose very strong assumptions on the data generating process.

ii. The black box is still elusive and we do not yet fully understand what is going on behind the scenes.

iii. Lesson: Look for $\widehat{y}$ problems not $\widehat{\beta}$ problems. Read Mullainathan and Spiess (2014).

Statistical Decision Theory

i. Decision theory provides a general framework to evaluate desirable properties of various ML estimators.

ii. It also provides theoretical justification for Bayesian estimators and in what sense they are best or most "reasonable" or "un-dominated".

Examples of Decision Problems

i. Testing the hypothesis that whether graduates with masters degrees earn more in labor market, relative to those without masters or not.

ii. Forecasting recessions by determining recession probabilities, conditional on observable information.

iii. Estimate the causal impact of inflation volatility on income inequality [selfish example, free publicity of my paper :')]

Components of General Statistical Decision Problem

i. Observed data $X \in \mathcal{X}$.

ii. Statistical decision function $a = d(X)$, $d(.) \in \mathcal{D}$, and state of world $\omega \in \Omega$.

iii. Loss function $L(a, \omega)$.

iv. Statistical model $f(X|\omega)$ giving conditional probability of data, given the state of world.

Goal of Statistical Decision Theory

i. First, nature draws a state $\omega \in \Omega$, which determines conditional distribution of data X: $f(X|\omega)$.

ii. Decision $a = d(X)$ is taken as function of observable data but *goal is to minimize* $L(a, \omega)$, which depends on what you do and the hidden state.

iii. X is informative about ω through f and hence, informative about L, which is why observing X helps in solving decision problem.

Example of Decision Problem

i. We want to determine probability of recession tomorrow.

ii. ω is joint distribution of relevant business cycle variables over time.

iii. X includes past values of business cycle variables.

iv. $a = d(X)$ is expected probability of recession in next quarter.

v. $L(a, \omega)$ is the expected, economic loss from recession in next quarter in the form of unemployment, GDP loss etc. Why is this expected loss from recession dependent on a?

Defining Loss functions for Estimation

i. We want to do choose $a = d(X)$ to minimize loss $L(a, \omega)$. How to define L?

ii. Common approach is to determine a moment or any other function $\mu(\omega)$, which depends on unobservable state or distribution and minimize distance between a and μ.

iii. For example, $\mu(\omega) = \mathbb{E}[X|\omega] = \int_{X \in \mathcal{X}} X f(X|\omega) dX$ is conditional mean which we want to target.

iv. Distance can be squared error loss as also in OLS or absolute deviations, i.e $L(a, \omega) = (a - \mu(\omega))^2$ or $L(a, \omega) = |a - \mu(\omega))|$.

Risk Function

i. $R(d,\omega) = \mathbb{E}[L(d(X),\omega)|\omega] = \int_{X\in\mathcal{X}}[L(d(X),\omega)]f(X|\omega)dX$ is the expected loss, conditional on the true state of nature, for a decision function d.

ii. Decision functions d which produce lower risk are better.

iii. However, one usually does not have uniform dominance and risk is lower for some ω and higher for other $\bar{\omega} \in \Omega$. Decision theory deals with this key problem which leads to variance-bias trade off.

iv. Dominated estimators d have property that $\exists \bar{d} \in \mathcal{D},\ R(\bar{d},\omega) \leq R(d,\omega), \forall\omega$ and $R(\bar{d},\omega) < R(d,\omega)$ for at least one $\omega \in \Omega$. Admissible estimators are not dominated.

Variance Bias Trade Off

i. For loss function, $L(a, \omega) = (a - \mu(\omega))^2$,
$R(d, \omega) = \mathbb{E}_\omega[(d(X) - \mu(\omega))^2] =$

ii. Add and substract $\mathbb{E}_w\{d(X)\}$ in this expression to get
$R = \mathbb{E}_\omega[(d(X) + \mathbb{E}_w\{d(X)\} - \mathbb{E}_w\{d(X)\} - \mu(\omega))^2] =$
$\mathbb{E}_\omega[(d(X) - \mathbb{E}_w\{d(X)\})^2] + (\mathbb{E}_w\{d(X)\} - \mu(\omega))^2$
$+2(\mathbb{E}_w\{d(X)\} - \mu(\omega))\mathbb{E}_\omega\{d(X) - \mathbb{E}_\omega\{d(X)\}\}$
$\mathbb{E}_\omega\{\mathbb{E}_\omega\{d(X)\}\} = \mathbb{E}_\omega\{d(X)\} \implies$ blue term is zero.

iii. Hence $R(d, \omega) = Var_\omega(d(X)) + (Bias_\omega(d(X)))^2$,
illustrating variance bias trade off.

Bayes Optimality for Getting Global Rankings Across Estimators

i. Integrated Risk $R(d, \pi) = \int R(d, \omega)\pi(\omega)$ where π are prior weights across the states of nature. Bayes' risk is $\inf_d R(d, \pi) = R(\pi)$.

ii. Bayesian decision functions d^* minimize integrated risk i.e $d^* = \underset{d}{argmin}\, R(d, \pi)$.

iii. Posterior expected loss $R(d, \pi|X) = \int L(d(X), \mu)\pi(\mu|X)$

iv. Bayesian estimators minimize posterior expected loss.

Why Should You be a Bayesian?

i. Under mild conditions, bayesian decision functions are always admissible.

ii. Under some additional conditions, all admissible estimators are bayesian decision functions for some prior distribution π. This is called complete class theorem.

iii. $R(d) \leq \bar{R} := \inf_{d} \sup_{\omega} R(d, \omega)$, where latter is minimax risk or worst case scenario.

Regularization and Shrinkage Estimators

i. Regularization is about solving variance bias trade off and this is what shrinkage estimators do.

ii. Shrinkage estimators perform a form of continuous subset selection which avoids the problem of high variance of subset selection.

iii. Shrinkage Estimator 1: Ridge Regression shrinks the coefficients by adding penalty in criterion function on their size.

iv. Shrinkage Estimator 2: LASSO first selects and then shrinks toward zero.

Ridge Regression

i. Ridge Regression solves the following problem: $\hat{\beta}_{ridge} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{i=1}^{p} x_{ij}\beta_j)^2 + \lambda \left(\sum_{j=1}^{p} \beta_j^2 \right) \right\}$.

ii. The above problem can equivalently be stated as $\hat{\beta}_{ridge} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{i=1}^{p} x_{ij}\beta_j)^2 \right\}$ s.t $\sum_j^p \beta_j^2 \leq t$.

iii. In matrix form, $RSS(\lambda) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda\beta^T\beta$.

iv. The solution to above problem is $\hat{\beta}_{ridge} = \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda I \right)^{-1} \boldsymbol{X}^T\boldsymbol{y}$ If $\lambda = 0$, we get our old friend OLS. I is $p \times p$ identity matrix.

Ridge Regression

i. Ridge regression projects y onto the principal components of X and shrinks components with lower variance more than those with high variance.

ii. Directions in which X has higher variance also likely to be directions in which Y varies more, helping minimize prediction error.

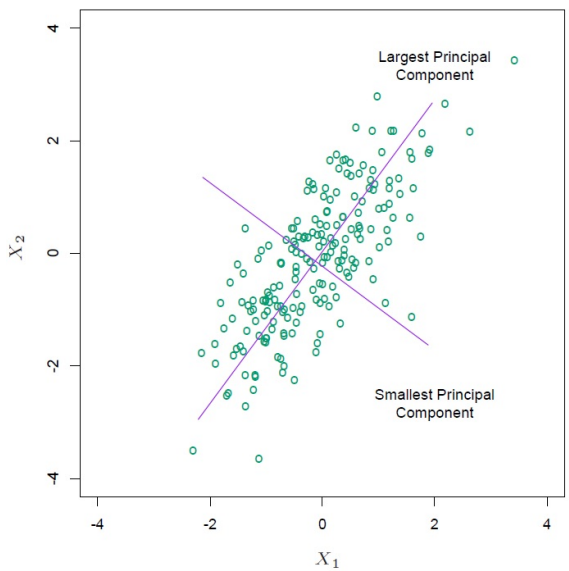


Figure: PCA and Ridge Regression

LASSO

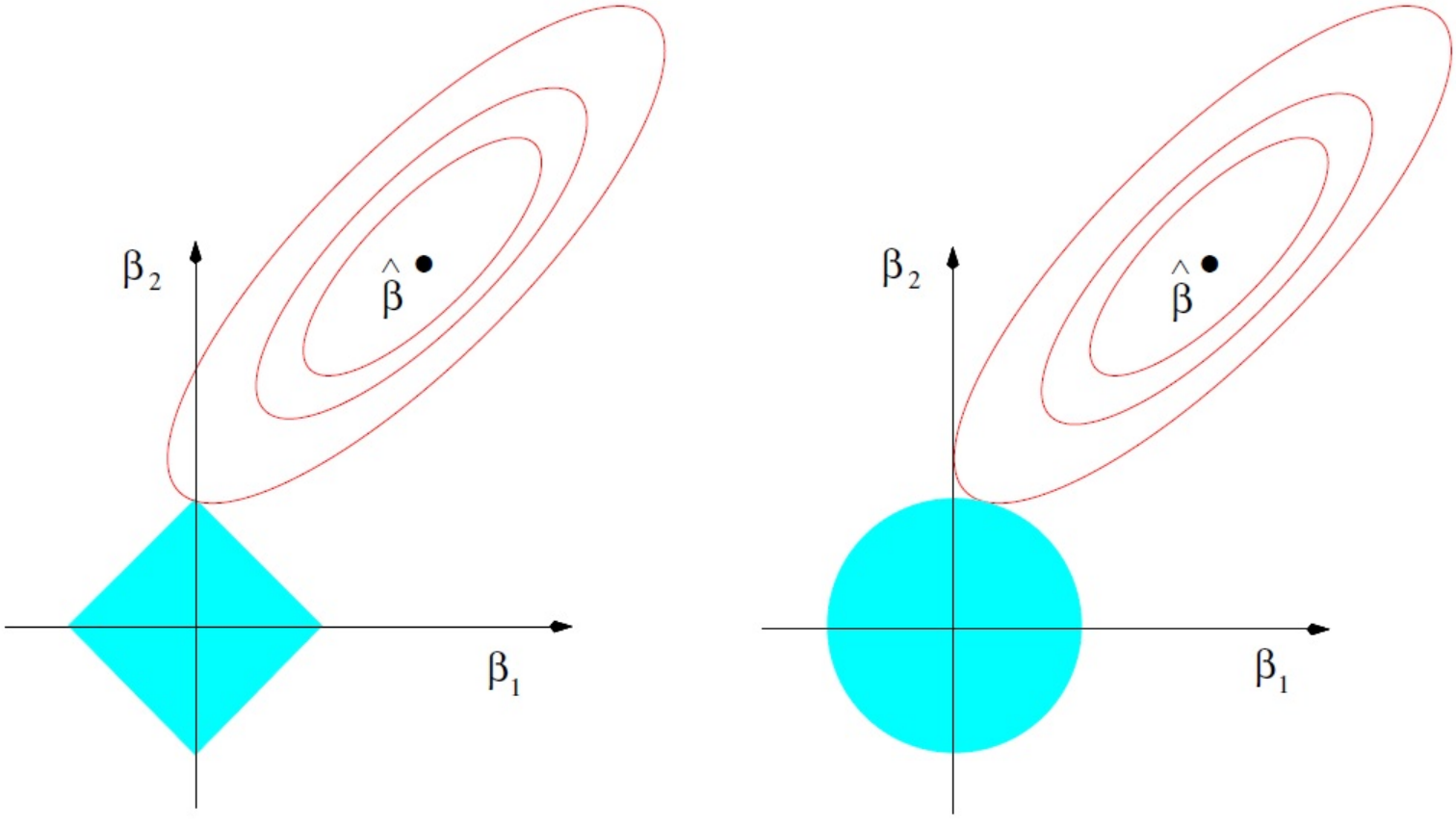
i. LASSO uses L_1 penalty instead of L_2 penalty.

ii. $\hat{\beta_{lasso}} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{i=1}^{p} x_{ij}\beta_j)^2 \right\}$ s.t $\sum_{j}^{p} |\beta_j| \leq t$.

iii. No closed form solution but numerical routines can solve the above quadratic optimization problem.

iv. Making t sufficiently small with cause some of the coefficients to be exactly zero.

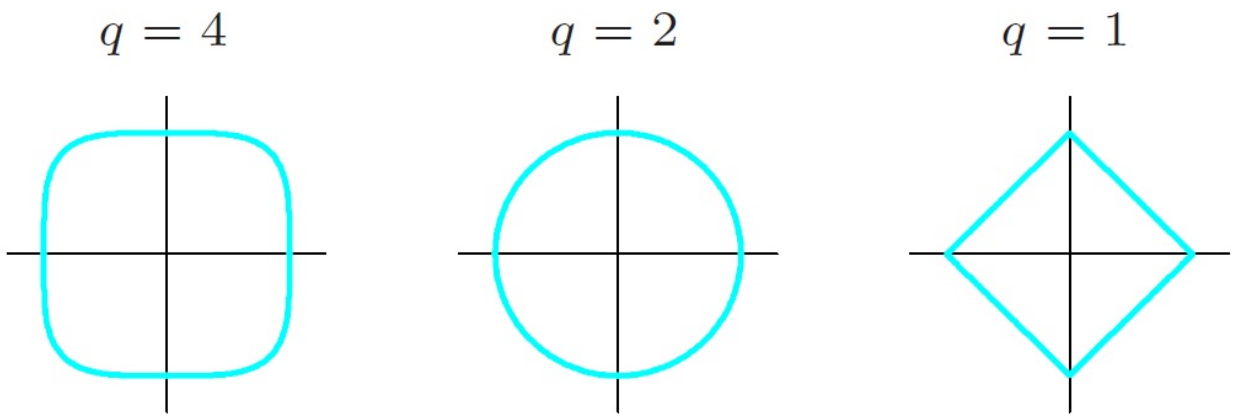
LASSO Verus OLS ($\hat{\beta}$) Versus Ridge Regression



Constraint Sets for General Criterion Function

i. $\tilde{\beta} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{i=1}^{p} x_{ij}\beta_j)^2 + \lambda \left(\sum_{j=1}^{p} |\beta_j|^q \right) \right\}$.

ii. For various q, the constraint sets vary in form. $q = 2$ is ridge, $q = 1$ is LASSO. Both these estimators are Bayesian estimators for different prior distributions.



Tuning Shrinkage Parameter for LASSO

i. Tuning λ can be done through k fold cross validation to optimize prediction performance.

ii. When K is too small, the variance of estimators is high but the bias is low and when K is too high, the variance is low and λ does not depend much on the split of data made but estimate is likely to be upward biased.

iii. $K = 5$ or $K = 10$ is advised for many applications.

Cross Validation

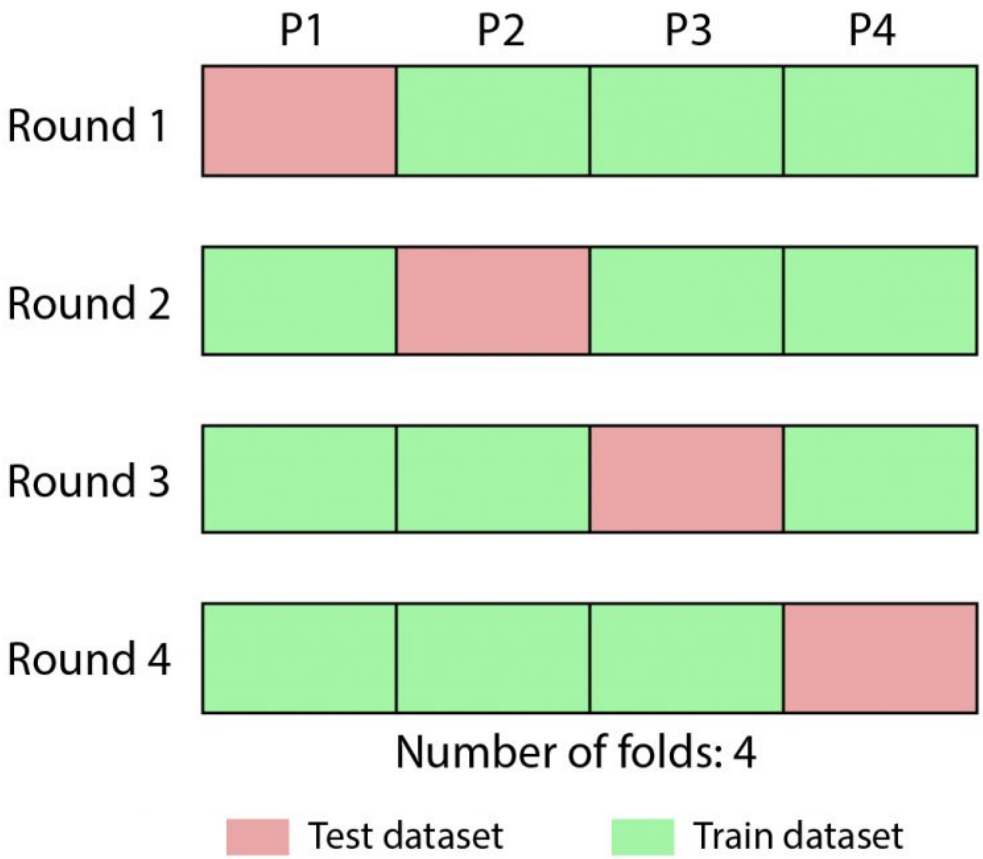


Figure: Cross Validation

Economic Applications of LASSO

i. LASSO is helpful for dimensionality reduction and improved prediction in the context of big data applications.

ii. It is now being frequently used in macroeconomic forecasting as well as big scanner data for supermarkets and data in neuroeconomics.

iii. For example, in a Stock and Watson type data set complied by Ludvigson and Ng (2009), there are 131 macroeconomic time series with 8 main sub categories.

iv. LASSO throws away many of these variables in prediction problems and retains a small subset which it also shrinks toward zero, favoring sparsity.

v. Remember LASSO *selects* and *shrinks*, unlike Ridge which only shrinks.

High Dimensional VARS Based on Callot and Kock (2014)

i. They consider stationary $VAR'S$ of following form: $y_t = \sum_{i=1}^{p} B_i y_{t-i} + \epsilon_t$.

ii. In high dimensional settings the matrix B is likely to sparse due to p being larger than true number of lags, irrelevance of certain lags for some variables in predicting outcome variables and/or Granger non-causality.

iii. The lasso as well as adaptive lasso (we won't cover this) have the *oracle* property in the sense that all truly zero parameters will be classified as such asymptotically.

iv. Also, estimators of the non-zero parameters have the same asymptotic distribution as if least squares had been used to estimate the model only including the relevant variables.

High Dimensional VARS Based on Callot and Kock (2014)

i. Data from Ludvigson and Ng (2009) which has 131 macroeconomic time series with 8 main sub categories for period Jan 1964 to Dec 2007.

ii. The main groups are: Output and Income (17), Labor Market (32), Housing (10), Consumption, Orders and Inventory (14), Money and Credit (11), Bonds and Exchange Rates (22), Prices (21) and Stock Market (4).

iii. Their training sample uses data from Jan 1964 to Dec 1999 and they compute forecasts of y_t at $t = 1999 : 12 + h$ for $h = 1, 3, 12$. Rolling window estimation until out of sample forecast is generated for $2007 : 12$.

High Dimensional VARS Based on Callot and Kock (2014)

i. They compare forecasts of various versions of LASSO such as plain lasso, adaptive lasso and group lasso with a standard $VAR(1)$ as well as Bayesian VARS, STAR (Smooth Transition Auto Regression) and Factor Models.

ii. $VAR(1)$ has the following form: $y_{t+h} = \sum_{l=1}^{p} B_l^h y_{t-l+1} + \epsilon_{t+h}^h$, , $t \in [p, .., T-h]$ for each forecast horizon.

iii. Plain lasso is the best predictor with lowest MSE for most groups as well as overall. Forecasts from common factor models were relatively precise whereas non-linear LSTAR was much more volatile and performed poorly for some groups.

Horse Race Between Various Predictors

ADAPTIVE LASSO AND ADAPTIVE GROUP LASSO IN VARS

	Output and Income	Labor Market	Housing	Consumption Orders Inventories	Money and Credit	Bonds and Exchange Rates	Prices	Stock Market	Total
BVAR	0.561	0.600	1.262	0.645	0.640	0.772	0.714	0.588	0.723
Lasso	0.446	**0.464**	**0.466**	**0.490**	**0.585**	0.359	**0.616**	0.455	**0.485**
aLasso	0.494	0.647	3.961	0.680	0.609	**0.321**	0.732	0.479	0.990
agLasso	0.524	0.698	6.537	0.795	0.613	1.468	0.743	**0.454**	1.479
Factor model forecasts									
CF 1	0.429	0.495	0.778	0.569	0.629	0.416	0.686	0.495	0.562
CF 3	**0.417**	0.512	0.806	0.522	0.630	0.399	0.663	0.512	0.558
CF 5	0.421	0.515	0.772	0.524	0.630	0.403	0.669	0.510	0.555
CF BIC	0.436	0.508	0.829	0.578	0.643	0.443	0.683	0.499	0.577
Univariate forecasts									
lstar	0.683	0.672	0.752	0.703	1.146	0.483	11.066	0.871	2.047
ar(1)	0.959	0.862	0.764	0.932	0.774	0.687	1.185	1.216	0.922

TABLE 5.1. Mean square errors relative to the VAR(1). Average across all horizons. The last column, Total, contains the average across all series.

Forecast Combinations Example from Diebold and Shin (2019)

i. Various macroeconomic forecasts are available from several experts and forecasting agencies and there is often some disagreement regarding forecasts.

ii. There exists a forecast combinations literature which studies how to optimally combine forecast and the simple average has some desirable properties.

iii. Diebold and Sin (2019) use a lasso type procedure to combine forecasts from European Central Bank's quarterly Survey of Professional Forecasters regarding quarterly 1-year-ahead forecasts of Euro-area real GDP growth (year-on-year percentage change).

Diebold and Shin (2019)

i. 23 forecasters who answer most frequently are included for $1999Q1$ to $2016Q2$.

ii. Calculate "forecast" errors using realizations from the 2018Q1 data vintage.

iii. Rolling window of 5-years (20-quarter) is used for estimation, producing 1-year-ahead out-of-sample forecasts.

iv. $\lambda's$, ranging from a very light penalization to a very heavy penalization used, varying on grid $(0, 3269017]$ for robustness.

Diebold and Shin (2019)

i. Lasso based methods that involve selection to zero select a very small number of forecasters on average (approximately three).

ii. The average forecast does better than worst forecaster but considerably worse than best forecaster.

iii. Simple lasso is as good as average but does not beat best forecaster.

iv. "peLASSO", which is to implement lasso in first step and then simply average the forecasts of remaining selected forecasters is best.

v. "peLASSO" methods reduce the out-of-sample RMSE relative to simple average by almost ten percent and are as good as best forecaster.

Diebold and Shin (2019)

Table 1
Forecast RMSEs based on ex post optimal λs.

Regularization group	RMSE	λ*	#	DM	*p*-val
Ridge	1.51	2.66	23.00	−0.14	0.56
LASSO	1.52	0.38	2.71	−0.10	0.54
eRidge	1.50	max	23.00	−1.14	0.87
eLASSO	1.50	3.60	23.00	0.95	0.17
peLASSO (LASSO, Average)	1.40	0.21	2.95	1.06	0.15
peLASSO (LASSO, eRidge)	1.40	(0.21, max)	2.95	1.06	0.15
peLASSO (LASSO, eLASSO)	1.40	(0.21, 3.10)	2.95	1.07	0.15
Comparisons	RMSE	λ*	#	DM	*p*-val
Best	1.40	N/A	1	0.61	0.27
90%	1.44	N/A	1	0.63	0.27
Median	1.53	N/A	1	−0.57	0.72
10%	1.68	N/A	1	−1.61	0.94
Worst	1.74	N/A	1	−1.55	0.94
Average	1.50	N/A	23	N/A	N/A

Implementation of LASSO

i. `glmnet` package in `R`.
ii. Applications of LASSO in the context of causality and first stage regressions https://skranz.github.io/r/2020/09/14/LassoCausality.html
iii. Stata applications: https://voices.uchicago.edu/christianhansen/code-and-data/

Simple Hands on Application in R

i. Simple hands on application in R.

Thank you