developing trade consultants

policy • research • capacity building

Lecture 1: Introduction to Machine Learning

Ben Shepherd, Principal.

Ben@Developing-Trade.Com

Key Takeaways

1. Some elements of machine learning can be seen as related to familiar concepts from econometrics, though the terminology often differs.

2. Econometrics tends to focus on inference; ML tends to focus on prediction (or classification).

3. Lasso and related techniques provide a convenient entry point into machine learning, because they are easily recognizable in terms of regression models.

4. Lasso, Ridge, and Elastic Net are all shrinkage estimators: they penalize OLS estimates to "shrink" some parameter estimates towards zero.

5. ML workflow requires discipline and focus:
   1. Training/testing split.
   2. K-Fold cross validation.
   3. Prediction, and assessment of accuracy.
   4. Be careful to avoid too much pre-testing, as the testing data will bleed into the training data.
   5. Beware overfitting!

6. Simple ML applications are straightforward in R with GLMNet, though considerable data work is often required first.

Outline

1. What is Machine Learning?
2. The Lasso and Related Approaches
3. Workflow, Tips, and Traps
4. Demonstration in R: The Logistics Performance Index

1. What is Machine Learning?

- ML or "algorithms" are everywhere, we constantly hear about them:
  - When Netflix suggests a movie we might like, based on past choices.
  - Automatic translation of text into other languages.
  - Mining of sentiment databases, like tweets.
  - Predictive text in Gmail (scarily good).

- Where does ML fit into economics, and specifically into policy-relevant economics related to international trade?

- How does ML relate to what we already know as "econometrics"?

1. What is Machine Learning?

Inference Problem

- What is the elasticity of bilateral trade flows with respect to trade facilitation performance?
  - Data on trade flows → Gravity model.
  - Variable of interest + controls.
  - Fixed effects to account for panel structure.
  - Appropriate econometric estimator (PPML) to deal with known issues with OLS.
  - Test with diagnostics.

Prediction/Classification Problem

- Which countries are the most likely to experience "explosive" export growth in the next five years?
  - Data on trade growth in the past.
  - Data on country characteristics.
  - Let the data decide which characteristics matter the most.
  - Predictive algorithm, not econometric estimator.
  - Test with predictive accuracy.

1. What is Machine Learning?

Inference Problem

- Y = XB + e
- We're interested in estimates of B that:
  - Satisfy desirable large sample properties (consistency, bias, efficiency).
  - Are informative as to an economic mechanism underlying the problem.
  - The mechanism is of primary interest.
- Econometric methods make assumptions about the data generating process to produce estimates of B with desirable properties.
- Pay little attention to predictions of Y.

Prediction/Classification Problem

- Y = XB + e
- We're interested in predictions of Y, not estimates of B.
- ML pays (relatively) little attention to estimates of B.
- ML makes no assumptions about the data generating process.
- Typically little attention to large sample properties; question is simply "how well does the model predict Y?"

1. What is Machine Learning?

Econometrics	Machine Learning
Estimation	Training
Estimation sample	Training sample
Out-of-sample	Prediction sample
Explanatory variables	Features
Estimated parameters	Weights
Statistical model	Regularization / Algorithm
Goodness of fit	Predictive accuracy

1. What is Machine Learning?

- Econometrics provides a useful entry point into ML, and some concepts "translate" relatively well.

- But the point of view is usually quite different: the end use of the model is different.
  - Though there are cases where the relationship is much closer (e.g., matrix completion methods for causal inference).

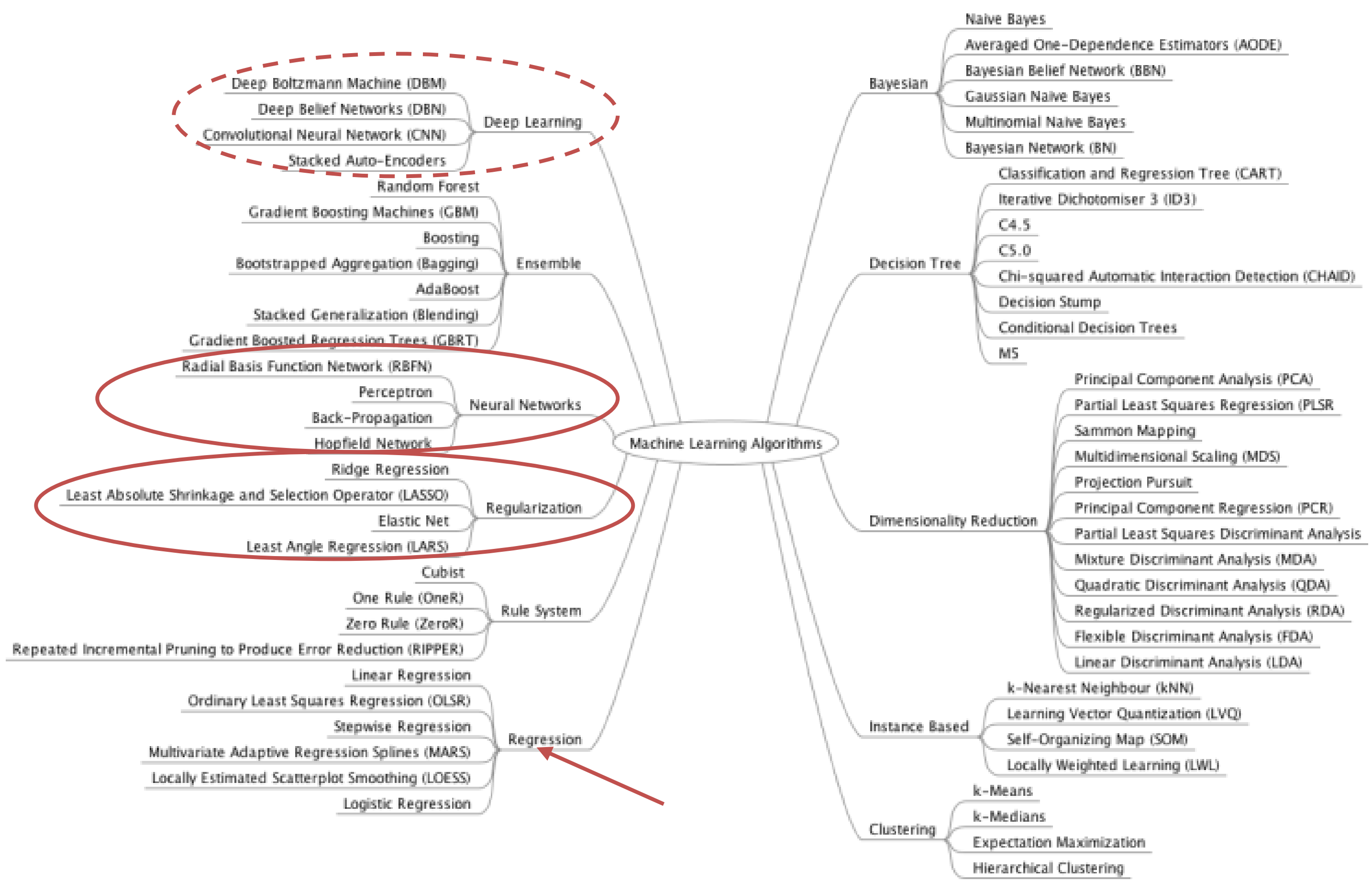
- Also ML is a broad family of algorithms and approaches; we will only look at a small sample.

- There is a lot of statistics behind ML, but it is more accessible than traditional econometrics for practical learning.
  - Mobilize basic concepts from econometrics.
  - Focus initially on techniques that are closely linked to econometric models.
  - Focus on an intuitive understanding of ML algorithms, not the mechanics of how they work.
  - Develop a workflow/process designed to match problems to algorithms, and avoid typical pitfalls.

- Still very few applications of ML in the international trade literature, and some existing applications are a little eccentric.

- Lots of scope to add to the policy literature!

1. What is Machine Learning?



Source: https://antontarasenko.com/2015/12/28/machine-learning-for-economists-an-introduction/

2. The Lasso and Related Approaches

- The simplest Least Absolute Shrinkage and Selection Operator (Lasso) solves the following problem:
  - $\hat{B} = argmin(Y - XB)'(Y - XB) + \lambda \sum_{j=1}^{J} |B_j|$

    OLS → $(Y - XB)'(Y - XB)$

    Penalty Factor → $\lambda \sum_{j=1}^{J} |B_j|$

  - Solution by numerical methods.
  - The second terms penalizes (shrinks) weights (parameter estimates, a total of J parameters), so that some are zero.

- Lasso makes it possible to select features (variables) with non-zero weights (parameters), then use them to predict Y.

- A neat trick is that because of the nonlinearity, Lasso can have MORE features than observations in the dataset!
  - So we can start from a potentially huge dataset, and narrow it down to the variables that really matter for predictive purposes.

2. The Lasso and Related Approaches

- A close relative is Ridge regularization:
  - $\hat{B} = argmin(Y - XB)'(Y - XB) + \lambda \sum_{j=1}^{J} B_j^2$

    OLS　　　　Penalty Factor

- Same principle as Lasso, but the penalty works on the square of the weight rather than its absolute value.

- Elastic Net regularization combines these two approaches:
  - $\hat{B} = argmin(Y - XB)'(Y - XB) + \lambda \sum_{j=1}^{J} (\frac{1-\alpha}{2} B_j^2 + \alpha |B_j|)$
  - So for alpha = 0, EN = Ridge. For alpha = 1, EN = Lasso. For other alphas, EN is a blend of the two approaches, with the total penalty governed by lambda.

2. The Lasso and Related Approaches

- The simplest applications of shrinkage regularization are linear (like OLS).
  - But can also be used with nonlinear models like Poisson, Logit, etc.
  - Choice depends on the nature of the problem, as well as empirical performance.

- Shrinkage regularization is an easy entry point into the ML literature, because it is essentially a different way of looking at a regression problem.
  - Before the days of widespread ML (~2000), I learned about "ridge regression" as a way of dealing with collinearity in regression models.

- The key difference in applying shrinkage regularization as an ML algorithm really lies in:
  - Type of problem.
  - Presentation of results.
  - Workflow and model comparison.

3. Workflow, Tips, and Traps

- How do we implement Lasso in an ML context?

- Recall the problem we're solving:
  - $\hat{B} = argmin(Y - XB)'(Y - XB) + \lambda \sum_{j=1}^{J} |B_j|$

- The key choice is the penalty parameter λ.

- In an ML context, we want to choose λ so that the model has the "best possible" predictive performance, as measured by some criterion such as mean squared error.

3. Workflow, Tips, and Traps

- A typical ML approach to model selection is cross-validation:
  - Split the data into training and test samples.
  - Estimate a model using the training data only, then use it to make predictions for the test sample.
  - Compute a prediction accuracy measure.
  - Repeat for all the candidate models.
  - Select the model with the highest prediction accuracy measure.

3. Workflow, Tips, and Traps

- The gold standard in many ML applications (including Lasso as an example) is k-fold CV

1. Randomly split the data into k subsamples.
2. Hold back one of the k subsamples as a testing sample, then estimate a model using the remainder of the data as a training sample for a given value of λ.
3. Use the model to make predictions for the testing sample, and calculate MSE.
4. Repeat steps 1-3 for the other k subsamples, and calculate average MSE.
5. Repeat steps 1-4 for alternative values of λ by moving over a grid.

- K=10 is typical, use 5 for quick exploratory work.
- Don't worry: the computer automates k-fold cross-validation!

3. Workflow, Tips, and Traps

- If we search over a grid for λ, we can select the model with lowest average MSE for the testing sub-samples.

- It represents the "best possible" predictive performance.

- The selected model will imply a certain number of zero weights, so the non-zero weights represent features that have been "selected" by the model on the basis of its predictive performance.

- Final step: obtain predictions using the full sample.

3. Workflow, Tips, and Traps

- K-fold CV helps minimize the risk of over-fitting the data, but we need to be rigorous and disciplined in exploratory work.
  - Given enough features, we can always come up with a model that will fit arbitrarily well in-sample.
  - CV focuses on out-of-sample predictions, but if we do it too much in pre-testing, we are "cheating" by effectively giving the model the full sample.
  - So beware of effectively using the full sample to overfit a model—performance will look very good, but when you use it with new data, it will do much worse.
  - Familiar problem from forecasting applications in econometrics.

- First, split the sample into training and testing subsamples.

- Then, use k-fold CV on the training subsample.

- Assess model performance based on the testing subsample.

- Avoid repeating this process over and over: the information from the testing subsample effectively "bleeds" into the training subsample!

3. Workflow, Tips, and Traps

- We've already noted that a neat feature of Lasso (and many other ML procedures) is that the number of features can be large relative to the sample.
  - Typically a major problem for econometric models, both because inference is difficult due to correlations among variables, but also due to mechanical limits.

- A linear Lasso, like OLS, assumes a linear model for the relationship between features/weights and the prediction variable.

- But if our only limit on the number of features is computing time, we can include:
  - Nonlinear terms (powers).
  - Interactions.

- Not uncommon to start with thousands of features, and use Lasso selection to identify a small number with strong predictive value.

- Since we're only secondarily interested in inference, we don't necessarily need a behavioral model to support nonlinearities or interactions.

- Again, beware of overfitting!

4. Demonstration in R: The Logistics Performance Index

- The World Bank's Logistics Performance Index (LPI) summarizes performance on six dimensions using a survey.
- Data are available for a range of countries (not all; ~150) for 2007, then 2010-2018 at two-year intervals.
- Although widely used in policy settings, the LPI methodology will be fundamentally changed in the near future, meaning that new observations will not be comparable with old ones.
- Wouldn't it be nice to:
  - Fill in LPI values for countries and years not covered?
  - Continue to produce LPI estimates that are compatible with the "old" methodology?
- From an ML perspective, this is a classic prediction problem: we can't run our own surveys, but can we use observations on existing data series to make "good" predictions of the LPI?
  - Extending the index is then just a question of using observations of those series for other countries and years.

4. Demonstration in R: The Logistics Performance Index

- We can come at the problem from two complementary angles:
  - Prediction: We want to use ML to predict LPI scores based on other data.
  - Classification: We want to use ML to put countries into their LPI quintiles based on other data.

- The WB produces scores, but often talks about countries in the five performance groups (quintiles) as sharing similar characteristics.

- Prediction can use a linear model. Classification will use a multinomial model (5 categories). All can be run using the standard workflow and approaches including Lasso, Ridge, and Elastic Net.

4. Demonstration in R: The Logistics Performance Index

- How do we do this in R?
- The answer is what it nearly always is: "There's a package for that!".
- GLMNet: Elastic net based on the GLM family, so covers linear, Poisson, Logit, Multinomial, etc.
- GLMNet is less fancy than many R packages:
  - It doesn't support missing values.
  - Its native format takes data in matrix form; similar for other ML approaches, so we will do that, even though formula wrappers are available.
  - So there is some work required to manipulate the data, both inputs and outputs.
- What data can we use to predict the LPI? Let's just try the whole World Development Indicators database, 2000-2019.
  - Lots of missing values, so we need to clean.
  - Take levels and interactions.

4. Demonstration in R: The Logistics Performance Index

- Here’s the strategy, starting with prediction:
  - Clean up the data.
  - Set up matrices for GLMNet, with the full set of explanatory variables.
  - Split into training and testing subsamples.
  - Run Lasso, Ridge, and 50-50 Elastic Net on the training subsample using 10-fold CV to choose the penalty parameter.
  - Construct predictions for the testing subsample, check accuracy using RMSE.
  - Choose a model, and use it to predict out of sample.
  - Repeat the above steps for classification.

- Now for the code…

Key Takeaways

1. Some elements of machine learning can be seen as related to familiar concepts from econometrics, though the terminology often differs.

2. Econometrics tends to focus on inference; ML tends to focus on prediction (or classification).

3. Lasso and related techniques provide a convenient entry point into machine learning, because they are easily recognizable in terms of regression models.

4. Lasso, Ridge, and Elastic Net are all shrinkage estimator: they penalize OLS estimates to "shrink" some parameter estimates towards zero.

5. ML workflow requires discipline and focus:
    1. Training/testing split.
    2. K-Fold cross validation.
    3. Prediction, and assessment of accuracy.
    4. Be careful to avoid too much pre-testing, as the testing data will bleed into the training data.
    5. Beware overfitting!

6. Simple ML applications are straightforward in R with GLMNet, though considerable data work is often required first.

Additional Resources

- Two nice overview papers by economists:
  - Athey and Imbens (2019): https://arxiv.org/pdf/1903.10075.pdf.
  - Mullainathan and Spies (2017): https://pubs.aeaweb.org/doi/pdfplus/10.1257/jep.31.2.87.

- Quick start tutorial for GLMNet: https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html.

- Links to resources: https://antontarasenko.com/2015/12/28/machine-learning-for-economists-an-introduction/.