Supervised machine learning

A recap and overview

Agenda

- Why machine learning
- When machine learning
- How machine learning

Motivation

What has machine learning revolutionized

Machine learning is useful for making very precise models and used widely:

- Decide which product to show in an app / webpage. Facebook also have algorithms controlling who to connect with, which stories to show you.
- Recognizing handwriting
- Making self driving cars or bots to play computer games.
- Creating chat bots that can speak with humans.
- To detect fraud, spam etc.

Why social scientists care (1)

We are often interested in making a model with **good** fit.

- When we are interested in predicting the future (stock markets, demographics etc.).
- Our linear models do not account for non-linear effects.
- If we do not have certain we can impute the missing data.
    - What is the socioeconomic composition of neighorhoods from Google Street Maps images.

Why social scientists care (2)

Implement a policy where identification of certain people is important.

- Identify poor people in a program for poverty alleviation.
- Identify people who are likely to survive a surgery before beginning one.

Estimation procedures in causal identification (econometrics)

- Some models rely on estimation
  - Instrument variables in the first stage of 2SLS
  - Propensity score matching
- Estimation procedures are often not systematic about estimating heterogeneous effects

Whether to use ML

When to use a model

When should I model data?

Look at your research question. You should use a model when:

1) you try to explain or predict a certain variable, and;

2) a conclusion cannot be made without a model.

- Example: the work on inequalit from 1900-now by Thommas Piketty

When to use machine learning

Machine learning is powerful, when should I use it?

What is the goal?.

- To make a formal test about model parameters.
  - Use econometrics/OLS.
- You want a model with good performance.
  - Use machine learning.
  - But can I still investigate partial effects of a variable? We can visualize it:
    - individual conditional expectation (ICE).
    - partial dependence plots (PDP).

Determine model framework

Step A: What problem

What kind of problem am I working on? My target is,

- Continuous:
  - We want to use a regression model
  - We aim for a model with the least mean squared error (MSE).
- Categorical / finite integers:
  - We want to use a classification model
  - We aim for a model with highest accuracy (ACC).

Step B: Which model

Depending on the problem we pick a specific model. If regression, pick a model from:

- A linear model (least squares, lasso, ridge)
- Random forest

If classification, choose from:

- Random forest
- Support vector machine
- Logistic regression with regularization

Note: You are welcome to try out other more complicated models but be sure to explain how the models works in your project.

Step C: Determine hyperparameters

What hyperparameters exist for the model I have chosen?

- Ridge/Lasso: λ;
- Elastic net: λ_1, λ_2;
- Random forest: number of trees, max depth, max number of features, etc..
- SVM ..

Note if we include unsupervised learning before the supervised learning, e.g. Principal Component Analysis or K-means, then the number of either components or clusters is also a hyperparameter.

Applying supervised machine learning

Step 1: data split

Split into test and development (train) data.

- A normal split is 30 pct. for test and 70 pct. for train if you have ~ observations.
- If you have more than 10,000 observations then use 20 pct. for test, 80 pct. for train.

Polynomial transformation of features:

- This step is optional - only makes sense for linear and logistic models (e.g. lasso)

Step 2: model pipeline

Construct a model building pipeline.

- Preprocessing phase
  - Unsupervised learning, e.g. principal components (optional)
  - Variable scaling (optional, not relevant for random forest)
- Supervised model (classification or regression)

Note: It is optional on whether to use `make_pipeline`. We recommend using it as there will be fewer mistakes and contain less code.

Step 3: model selection (1)

Main idea:

- We want to select the optimal model.
- We measure model performance with out-of-sample prediction on validation data.

Pick the model which performed the best on the validation data during cross validation.

Step 3: model selection (2)

Cross validation (CV):

- We only use the training/development data.
- We use 10 fold CV and split this data into 10 even sized validation bins.
- For each validation bin:.
  - We fit our model on the data outside the validation bin, i.e. in one of the remaining 9 bins..
  - We transform and predict the target in the validation bin using our model.
  - Note: we must perform our whole model building process and transformation in each fold.

Finally we compute the mean across the 10 validation bins for each hyperparameter combination we are testing and pick the one that maximize out-of-sample performance.

Step 4: check list

- Check that we NEVER fit our model building on validation data.
- Check that test data has NEVER been used.
- Ensure that model has converged (do not suppress warning)
- Am I using a static model for time series?

Step 5: final model training and evaluation

We train the model with the optimal hyperparameters on ALL the training/development data.

Evaluate the model out-of-sample on the test set.