Introduction to Machine Learning: Training and Testing Algorithms

PADP 9200

Class 2

For today…

- Continue our work introducing R.
- Dive deeper into some of the concepts from machine learning.
- Build a machine learning algorithm (conceptually) to predict whether an H1-B visa will be **certified** or **not certified** based on data from the application.

Concepts from machine learning theory

1. Training, testing and cross validation.
2. Assessing model performance.
3. Overfitting.
4. The bias-variance tradeoff.

Training, Testing and Cross Validation

Motivation: *US Citizenship and Immigration Services (USCIS) is strapped for cash and does not have enough agents to review applications. They would like to move to an automated system (**AUTOAGENT)** that will allow them to process visa applications quickly using machine learning.*

The system should:

- Be able to use visa data as inputs (features).
- Use these features to predict whether an application is likely to be **certified** or **not certified**.
- Accomplish this in a way that replicates human performance.

Setting Up the AUTOAGENT

- **Data**: H1 B Visa petitions
  https://www.kaggle.com/nsharan/h-1b-visa

- **Target**: Whether an application is certified or not.

- **Features?**

Training and Testing: Setup

- Goal is to build an algorithm that will predict whether an application will be certified or not.
- Choose an algorithm which maps the features onto P(Certification)
- Logistic regression might be a good choice.

Training the model

- Randomly divide the data into a “training” set, which will be used for estimating the parameters of the model and a “testing” set which will be used to measure the final performance of the model.

- Typical training/testing splits
  - 80/20
  - 70/30
  - 60/40

- Depends on how much data you have and other considerations.

Estimate a model on the training data…

If N = 1000, split by N = 800 for training, N= 200 for testing

Logit(Certification|X) = f(X) Using the training data (N = 800)

Apply the trained model to the test data…

- Using the estimated model from the training data, feed the X's from the test data into the model to generate the predictions.

f(X(test)) = P(Certification)

Make Classification Decisions on the Basis of the Predicted Probabilities

- If P(Certification) > 0.5 → Classify as “certified”
- If P(Certification) < 0.5 → Classify as “not certified”

Compare the classifications produced by the model with those of the bureaucrats

		BUREAUCRAT	
	MODEL	**Certified**	**Not Certified**
	Certified	60	20
	Not Certified	20	40

Compute Performance Statistics

- **Accuracy**: % Correctly identified by the algorithm.
- **Precision**: True Positive/(True Positive + False Positive)
- **Recall:** True Positive/(True Positive + False Negative)
- **F1:** 2 x Precision * Recall/(Precision + Recall)

Fitting the model

- We want to make our model as complicated as possible to maximize performance but…
- If we make the model TOO complicated we risk overfitting.
- This is the result of the bias-variance tradeoff which (explained on the board).