Demystifying out-of-sample discrete choice prediction: What can we learn from machine learning?

Identification of techniques and best practices from machine learning to improve discrete choice models

CMC Leeds seminar series
17 November 2020

Melvin Wong
Post-doctoral researcher
Transport and Mobility Laboratory, École polytechnique fédérale de Lausanne (EPFL)

Motivation

The Black Swan Theory

- Events that are highly unlikely to happen, but would have impacting consequences if they happened
- The “unknown unknowns”
- Positive and negative events

Model risk

- One cannot predict the behaviour of such Black Swan events
- Try to rationalize how our models will perform on unseen data in other ways

Discrete Choice and Machine Learning take a different approach

- Is one method better than the other?
- How can we learn from each other?

Motivation

Estimate a choice model

Model X

Model transfer

Model X

Misspecified model?

Predicting market share

Predicting market share

Could we use out of sample prediction error % as an indicator of model reliability?

Current research gap

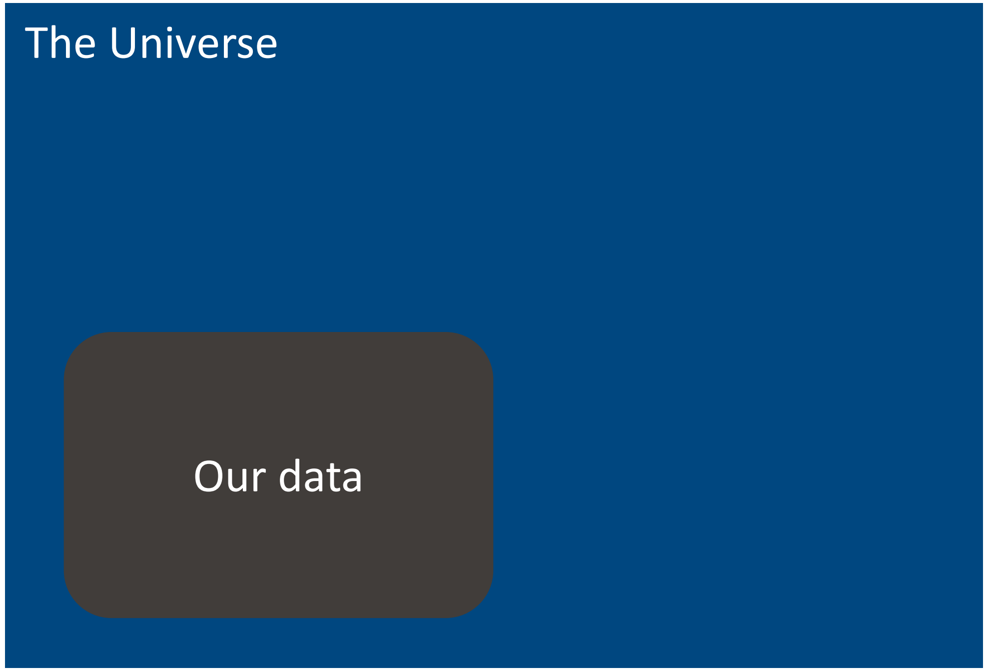
- Bridge the gap between Discrete Choice and Machine Learning
  - Common testing practices are transferrable (both ways)
- Measuring out-of-sample prediction performance in literature
  - Pros & cons of DCM & ML
  - Incorporating ML techniques into discrete choice?
- Evaluating statistical significance of Machine Learning models
  - Use out-of-sample performance (+ statistical tests) to validate our model
  - In addition to economic indicators

Overview

1. Introduction: What is out-of-sample data and out-of-sample prediction performance?
2. Discrete Choice vs Machine Learning: Improving out-of-sample prediction
    - Data
    - Testing
    - Models
3. Optimizing our models on out-of-sample performance
    - Example using residual neural networks
4. Conclusions and future work

Introduction

- Out-of-sample data
  - Unseen data: absent data, major disruption event, etc.
  - Not in our data sample/collection
- Out-of-sample prediction
  - We want our model to perform equally well for any problem we throw at it
  - Generalization
  - As an indicator for model specification reliability



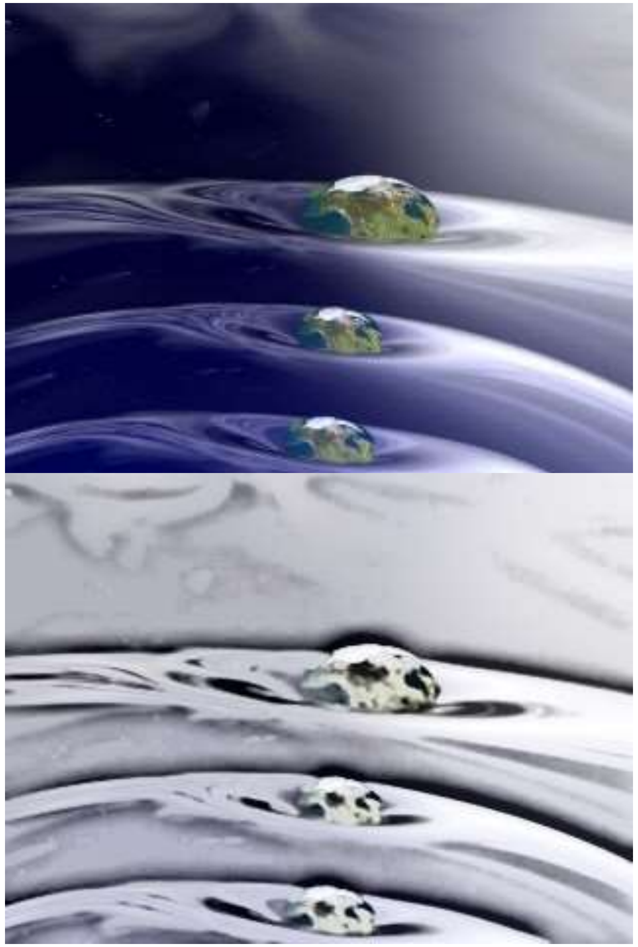
Data magic

We cannot possibly test our models on unseen data!

- Surrogate testing have been developed (DCM & ML)

Idea: “Reach into an alternate universe for data”

- We have already been doing it for years
- Ensure coverage of likely and unlikely events



Big Data

- Active research in data science
  - Mainly used in Machine Learning
  - e.g. Combining mobility and epidemic dynamics (Balcan et al. 2010)
- Focuses on merging many unrelated data sources to create hypothetical scenarios
  - Difficult task to achieve, but very effective
- Not necessarily need to be “large” in size
  - Diversification is more important (rich data)

Balcan, D., Gonçalves, B., Hu, H., Ramasco, J.J., Colizza, V. and Vespignani, A., 2010. Modeling the spatial spread of infectious diseases: The GLobal Epidemic and Mobility computational model. *Journal of computational science*, *1*(3), pp.132-145.

What about discrete choice?

Stated preference (SP) surveys

- Hypothetical scenario posed to respondents
- Caveat: relies on prior knowledge of individuals
- Advantage: ability to control parameters of the data

Data synthesis

- Used in both DCM and ML
- Pop. synthesis, simulation

Data through emulated experience

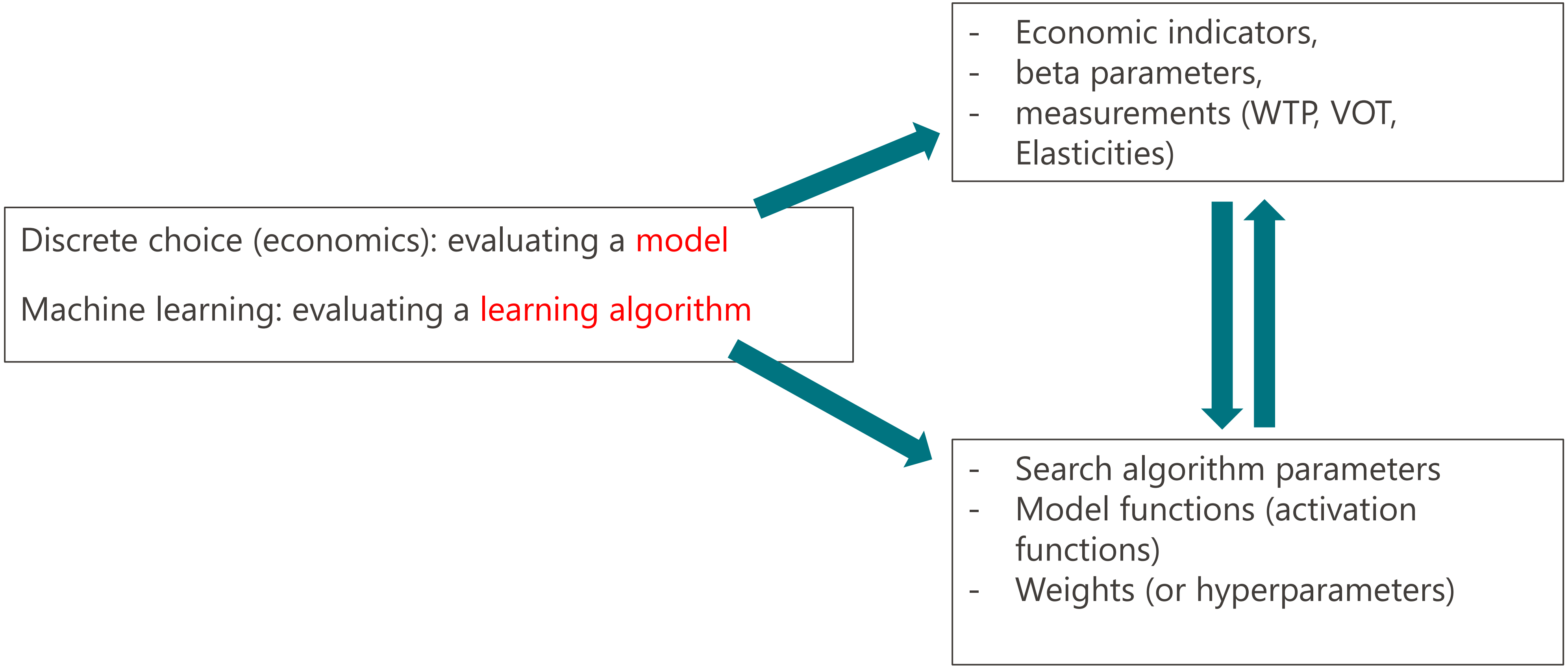
- Virtual Immersive Reality Environment (Farooq et al. 2018)

Farooq, B., Cherchi, E. and Sobhani, A., 2018. Virtual immersive reality for stated preference travel behavior experiments: A case study of autonomous vehicles on urban roads. *Transportation research record*, *2672*(50), pp.35-45.

State of research

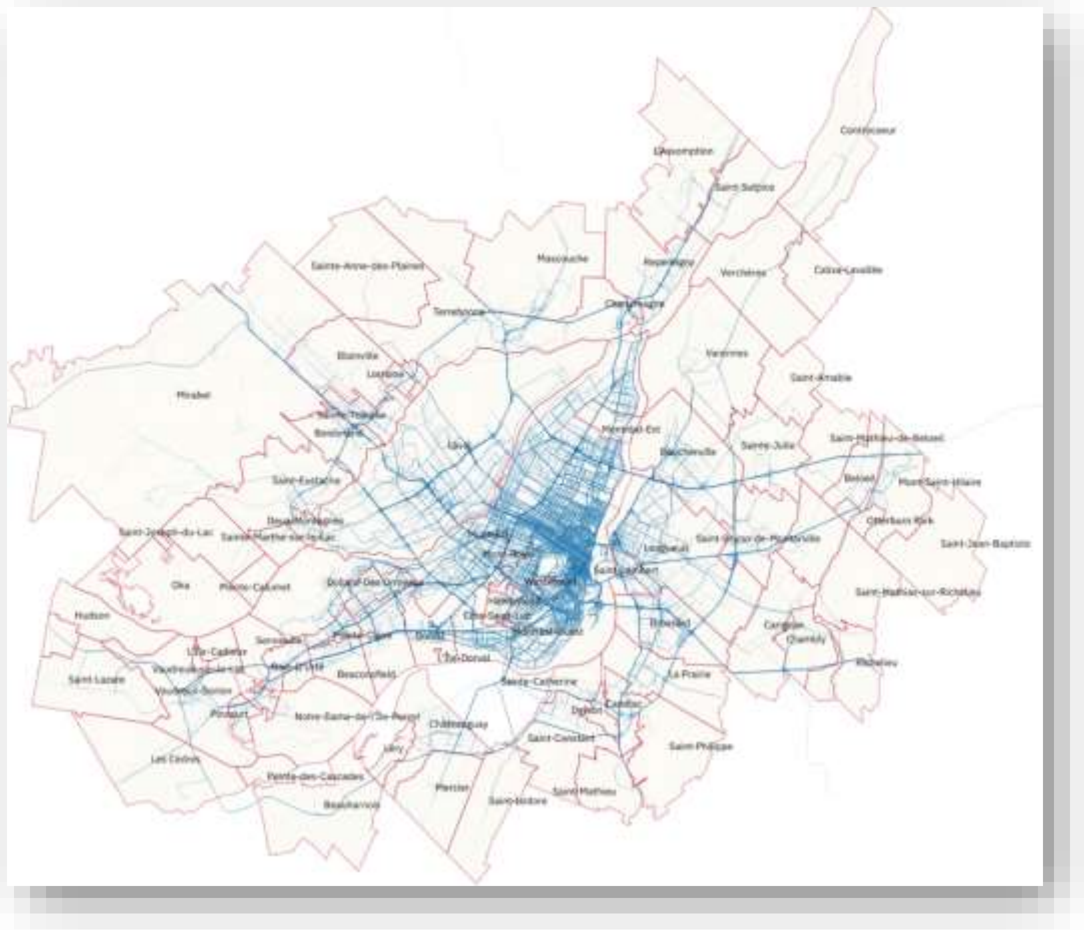
	Discrete Choice	Machine Learning
Data	Stated Preference (SP) survey Data synthesis	“Big Data” – combining data Data synthesis
Testing		
Model		

Testing


Discrete choice (economics): evaluating a model
Machine learning: evaluating a learning algorithm
- Economic indicators,
- beta parameters,
- measurements (WTP, VOT, Elasticities)
- Search algorithm parameters
- Model functions (activation functions)
- Weights (or hyperparameters)

Out-of-sample validation

- Data represents the entire known space
- How do we "test" unseen data using only seen data?
- Bootstrapping
  - In-sample data → Universe
- Holdout method
  - Split the data into training/validation set
- Cross Validation



Holdout validation

- Segmentation of data into training/validation set
  - Simple test of out-of-sample performance
  - Alternative: k-fold validation
- Ratio of split can affect results
  - Usually 70:30 (train:valid) used in literature
- Method to test in-sample prediction
  - Assuming no significant behaviour difference between the two datasets
- How large of a sample to use? (Alwosheel et al. 2018)

Training set

Validation set

Alwosheel, A., van Cranenburgh, S. and Chorus, C.G., 2018. Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of choice modelling*, *28*, pp.167-182.

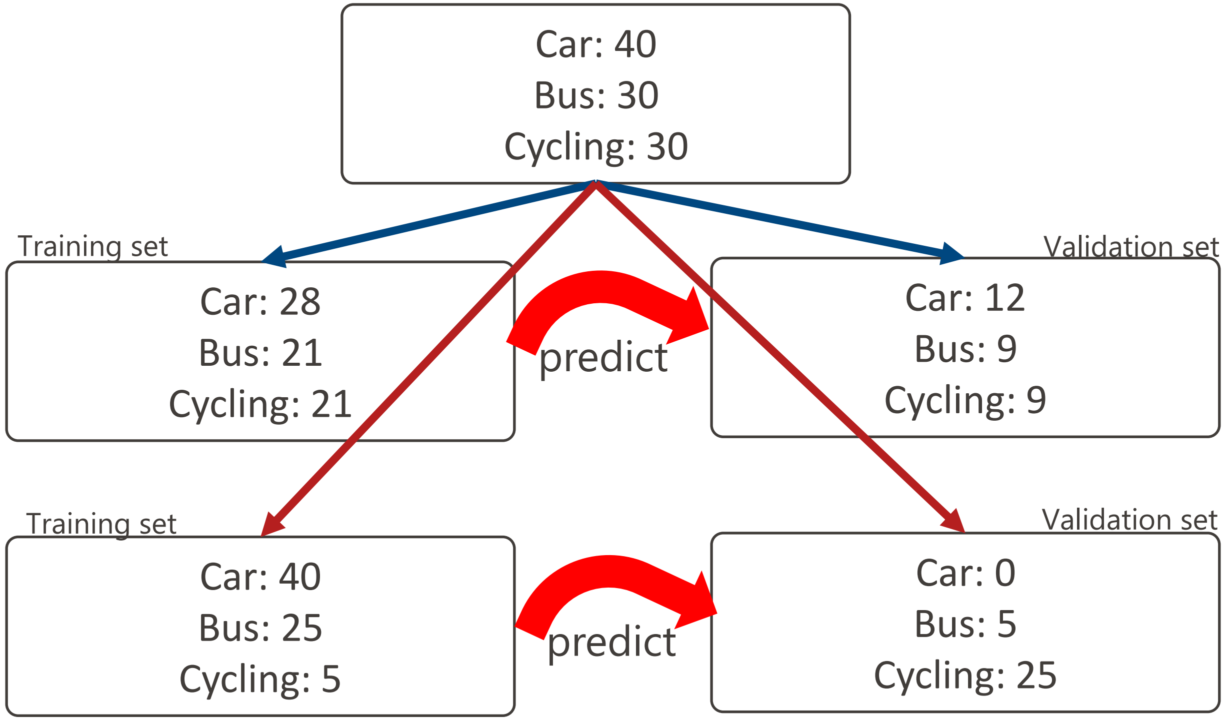
Training/validation split

Scenario 1:

- Balanced 70:30 split
  - Performance within similar trend/behaviour
  - Not so informative on generalization

Scenario 2:

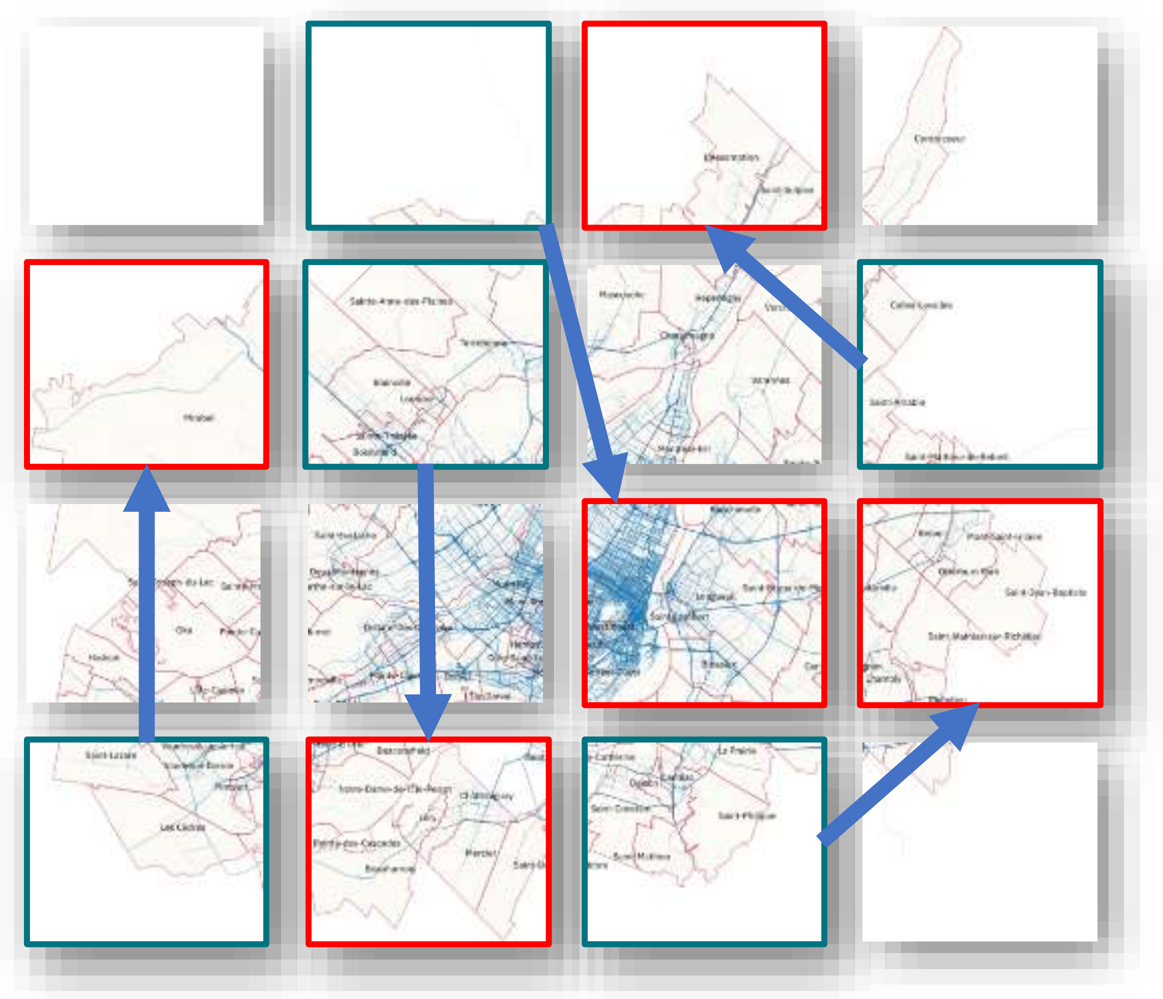
- Unbalanced 70:30 split
  - Simulating hypothetical scenarios
  - More informative
  - On an aggregate level only



Cross-validation

Approach

- Use just a part of our data to predict another part for every possible combination of train/validation data
- The average error would be representative of out-of-sample data (even though we don't have any)
    - Smaller the variation in error across pairs, better the generalization across problems



Generalization error: how well our algorithm would perform if we use real world data to predict unseen data (Nadeau & Bengio, 2000)

Nadeau, C. and Bengio, Y., 2000. Inference for the generalization error. In *Advances in neural information processing systems* (pp. 307-313).

Going further than cross-validation

- Other useful methods for testing disaggregate level
  - Adding noise
  - Synthetic population + distribution noise

- Learning from behaviour theory
  - DCM practices offer better ways of generating noise via expert knowledge than ML

State of research

	Discrete Choice	Machine Learning
Data	Stated Preference (SP) survey Data synthesis	“Big Data” – combining data Data synthesis
Testing	Goodness-of-fit, t-test, bootstrap Random parameter tests[1]	Cross-validation Noise/data corruption Un-balancing data
Model		

1 Fosgerau and Bierlaire, 2007. A practical test for the choice of mixing distribution in discrete choice models. Trans. Res. Part B: Methodological, 41 (7), pp.784-794.

Model choice

Information in unobserved factors

- Problems with misspecification:
  - From SP design w/ RP data (Guevara, C. A., & Hess, S., 2019)
  - Ommited attributes (Petrin, A. & Train K., 2003, 2010)
  - Latent Variables (Walker, Ben-Akiva, 2002)

Statistical theory assumes that a model is correctly specified

- Model misspecification from endogeneity problem

ML: There is endogeneity problem → unable to “learn” → poor performance: model misspecified

Discrete Choice approach

Control function (CF) (Petrin, A. & Train K., 2003; Guevara, C. A., & Hess, S., 2019)

$$U_{nj} = V\left(x_{nj}, \beta_{nj}\right) + \boxed{CF\left(\mu_n, \lambda\right)} + \varepsilon_{nj}$$

- Utility corrected for demand error in attributes
- “two-stage residual inclusion estimation” (Terza, 2018)

Mother logit (Timmermans et al., 1991; McFadden, Train & Tye, 1977)

$$U_{nj} = V\left(x_{nj}, \beta_{nj}\right) + \boxed{z\left(V_{1,\ldots,J}\right)} + \varepsilon_{nj}$$

- Utility depend on attributes from *all* alternatives
- Cross-effects representing correction to the *utility*
- Generalized function to account for IIA property violations

Machine Learning approach

Neural networks are learning algorithms developed for maximizing out-of-sample predictive performance

- Selects the hyperparameters that minimizes out-of-sample error
  - Structure, activation function, learning rate, gradient descent method, regularization, etc.
  - Independent from model parameters

If we have no knowledge about the unobserved information, we can optimize a generic neural network to "capture" this bias or error.

- Solution: Cast the model correction as an neural network optimization problem → residuals

$$U_{nj} = V(x_{nj}, \beta_{nj}) + \boxed{\text{Neural net}} + \varepsilon_{nj}$$

Goal

- Statistical test:
  - Null hypothesis: neural network is equal to zero → model correctly specified

$$U_{nj} = V\left(x_{nj}, \beta_{nj}\right) + \cancel{\boxed{\text{Neural net}}} + \varepsilon_{nj}$$

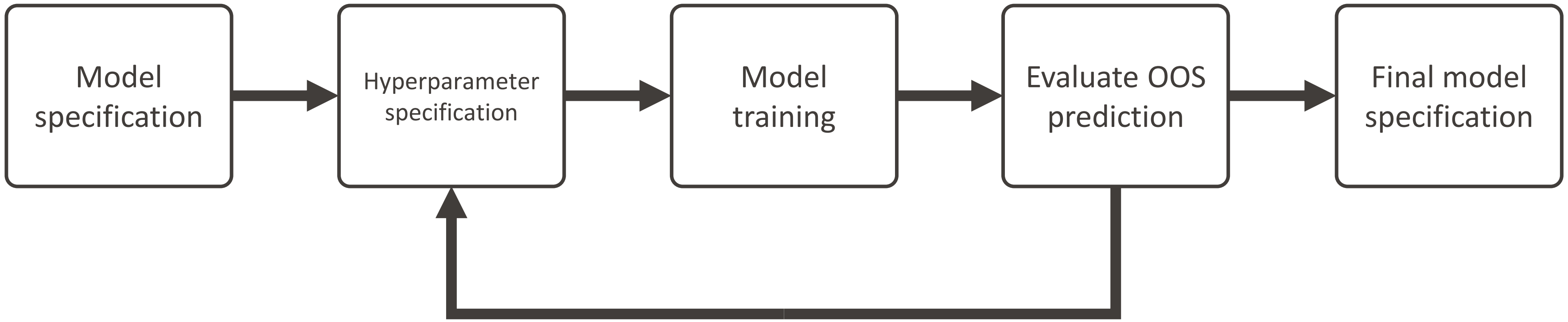
- Optimization approach: minimizing out-of-sample error
  - Compare performance with and without residuals → whether misspecification is present or not
  - **Statistical significance**: We have OOS error mean, variance and # of tests! → we can compute conf. interval

- Generalization to unseen data

Meta-model optimization

- For optimizing performance on unseen data, the goal is to minimize:
  - out-of-sample prediction error, AND
  - out-of-sample variance

- We choose the hyperparameters, model type and data generating process
- Once we have the algorithm → obtain our final set of model parameters as our final specification



State of research

	Discrete Choice	Machine Learning
Data	Stated Preference (SP) survey Data synthesis	"Big Data" – combining data Data synthesis
Testing	Goodness-of-fit, t-test, bootstrap Random parameter tests[1]	Cross-validation Noise/data corruption Un-balancing data
Model	Mixture models Random parameters, Control functions Dynamics	Deep nets (CNN, ResNet) Dynamics (LSTM, RNN) Regularization techniques

1 Fosgerau and Bierlaire, 2007. A practical test for the choice of mixing distribution in discrete choice models. Trans. Res. Part B: Methodological, 41 (7), pp.784-794.

Optimization on out-of-sample data

Case study on applied Machine Learning methods in Discrete Choice Models

Case study: Residual Logit model

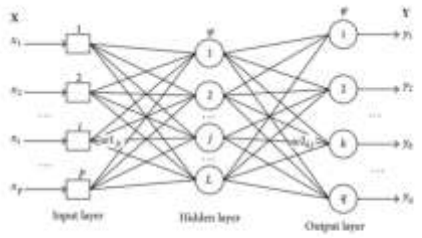
Multinomial logit model

$$U_{nj} = V(x_{nj}, \beta_{nj}) + \varepsilon_{nj}$$

$$\boxed{P_n(j) = \frac{e^{V(x_{nj},\beta_{nj})}}{\sum_{k \in J} e^{V(x_{nk},\beta_{nk})}}}$$

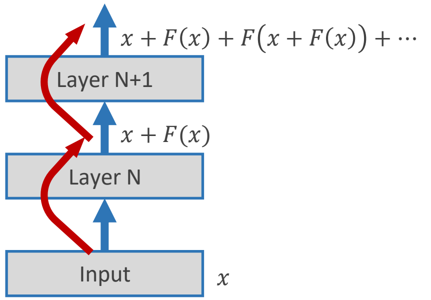
Multi-layer perceptron (MLP) neural net

$$U_{nj} = \boxed{\text{Neural net}}$$



Residual Logit (ResLogit)

$$U_{nj} = V(x_{nj}, \beta_{nj}) + \boxed{\text{Neural net}} + \varepsilon_{nj}$$



Choice of residual function:

$$U_{nj} = V(x_{nj}, \beta_{nj}) + \underbrace{f(h_{T-1}; \omega_T) + f(h_{T-2}; \omega_{T-1}) + \cdots + f(V; \omega_1)}_{} + \varepsilon_{nj}$$

Enabled by the **shortcut** connection

Paper: Wong, M. and Farooq, B., 2019. ResLogit: A residual neural network logit model. arXiv preprint arXiv:1912.10058.

Case study: Residual Logit model

Derived from the Residual Neural Network (ResNet) model

Intuition:

- Deeper neural network should perform better than a shallow network

In practice:

- **Increasing** # of neural net layers leads to **worse** performance (He, 2015)
- Problem occurs due to vanishing/exploding gradient problem

Solution:

- Focus on optimizing a residual function instead
- Reusing inputs from the previous layer

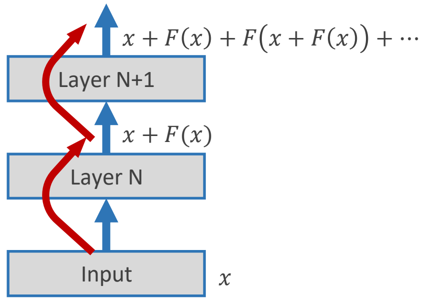
$$F(x) \coloneqq H(x) - x$$

$$H(x) = x + F(x)$$

Similar problem identified in Machine Learning and Discrete Choice!

Residual Logit (ResLogit)

$$U_{nj} = V(x_{nj}, \beta_{nj}) + \boxed{\text{Neural net}} + \varepsilon_{nj}$$



DCM Explanation:
Information propagation through layers
→ Endogeneity is a problem for learning algorithms too!

Residual function

Probability function:

$$P_n(j) = \frac{\exp(V_{nj} + g_{jn})}{\sum_{j' \in \{1 \ldots, J\}} \exp(V_{nj} + g_{j'n})} \forall j \in \{1, \ldots, J\}$$

Residual function g_{jn}:

$$g_{jn} = -\sum_{m=1}^{M} \ln(1 + \exp(\boldsymbol{\theta}^{(m)} \boldsymbol{h}_n^{(m-1)}))$$

$$-\ln(1 + \exp(\boldsymbol{\theta}^{(m)} \boldsymbol{h}_n^{(m-1)})) \quad (1)$$

$$= \int \frac{1}{1+\exp\left(\boldsymbol{\theta}^{(m)} \boldsymbol{h}_n^{(m-1)}\right)} \quad (2)$$

$$= \sum_{d=1,\ldots,\infty} \frac{1}{1+\exp\left(\boldsymbol{\theta}^{(m)} \mathbf{h}_n^{(m-1)} - d\right)} \quad (3)$$

"sum of logits"

Residual weights $\theta^{(m)}; m = 1, \ldots, M$ is a $J \times J$ matrix.

Input is a vector of utility from all alternatives:

$$\boldsymbol{h}_n^{(0)} = [V_{n1}, V_{n2}, \ldots, V_{nJ}]$$

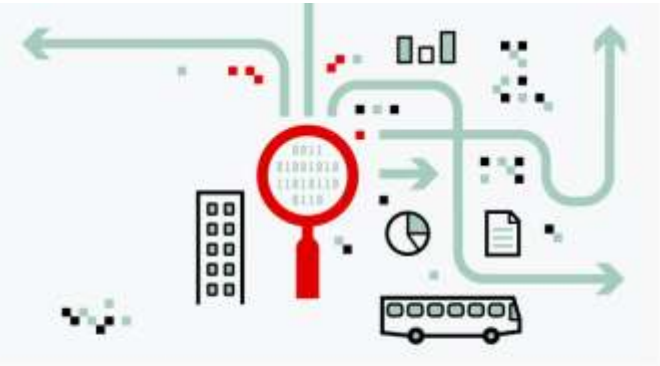
Other studies

Papers that work on similar principles:

- ResLogit (Wong and Farooq, 2019)
- TasteNet-MNL (Han et al., 2020)
- Learning-MNL (Sifringer et al., 2020)
- Assisted specification (Ortelli et al, 2020)

Correcting for endogeneity problem using data-driven machine learning

Case study: Data & Experiment



Data

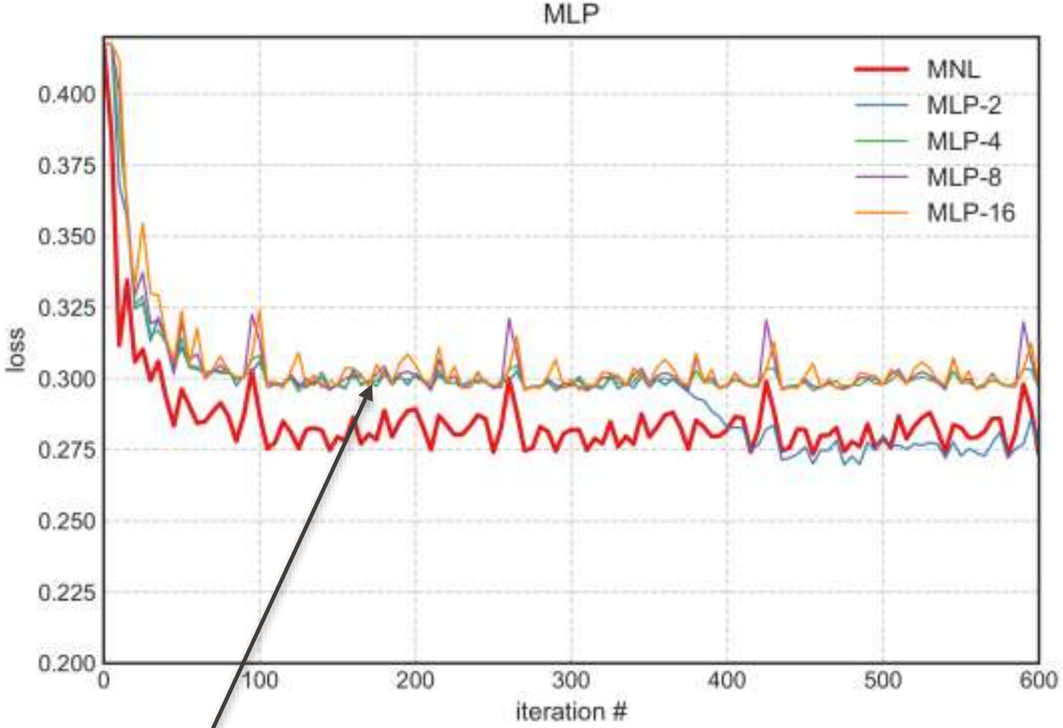
- Travel dataset from Montreal (open data, 2016 ed.)
- GPS traces (60,365 unique trips)
- Holdout validation 70:30 split
- Mode choice prediction

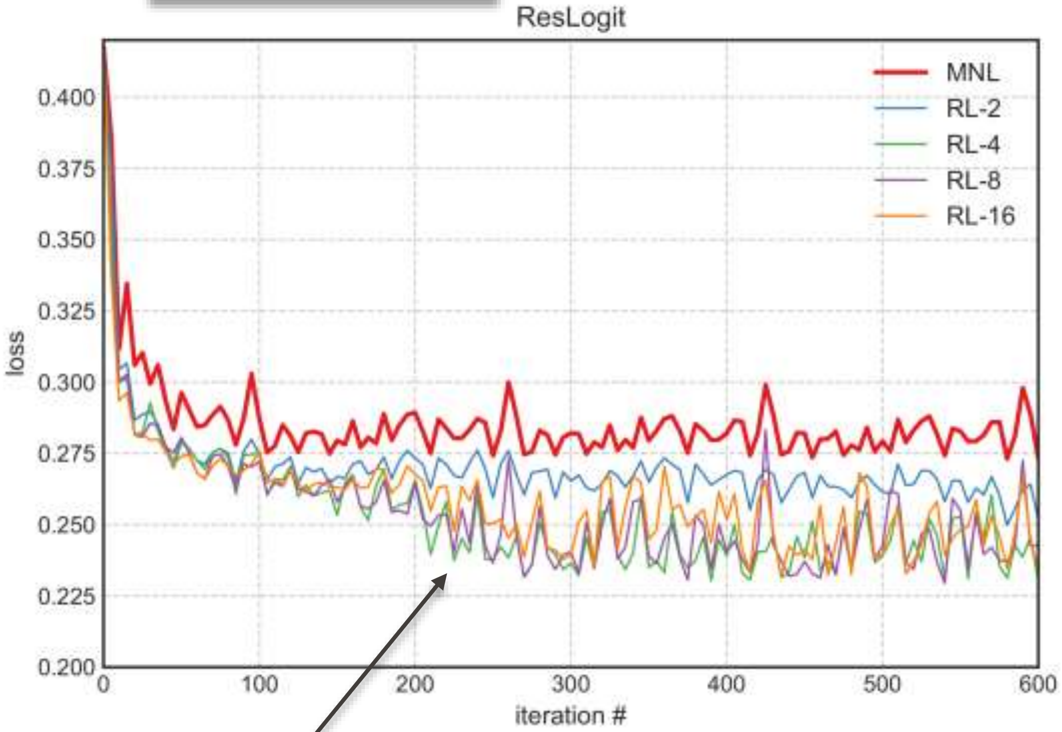
Experiment

- Flaws of using deep neural nets (DNN) → MLP does not always perform better than MNL
- Use a Residual Logit model to improve model consistency by optimizing on OOS error
- 3 model comparison: MNL (baseline), Multi-layer perceptron MLP, ResLogit
    - Hyperparameters: 2, 4, 8, and 16 layer neural net function

Results of model training

model validation loss (lower is better)





MLP: Fails to capture correlation between error terms and explanatory variables, resulting in poorer OOS performance than MNL (misspecification of model using neural net)

ResLogit: Captures correlation, reduces error → Model parameters increases generalizability. (model well specified)

Model estimates & interpretability

Parameters choices in parenthesis	**Without residuals (MNL)** Standard error in parenthesis	**With residuals (ResLogit RL-16)**
Weekend trip (1)	0.02 (0.007)	0.225 (0.006)
Trip departure time 8am-10am (4)	-0.957 (0.039)	-3.477 (0.038)
Trip departure time 5pm-7pm (1)	0.029 (0.002)	-0.836 (0.004)*
Trip distance (1)	0.409 (0.022)	-0.275 (0.001)
Trip distance (2)	0.258 (0.039)	0.133 (0.004)
Trip duration (1)	-0.653 (0.027)	0.24 (0.001)
Trip duration (4)	0.88 (0.272)	0.057 (0.005)
Home based trip (1)	-0.069 (0.246)	-0.015 (0.004)
Home based trip (3)	-1.108 (0.075)	1.357 (0.03)
Work based trip (1)	-0.016 (0.012)	-0.077 (0.004)
Work based trip (2)	-0.039 (0.002)	1.386 (0.012)*
Work based trip (5)	-1.877 (0.745)	-0.353 (0.023)
Log-likelihood	-16145	**-13121**

Choices: 1 Auto; 2 Transit; 3 Bike; 4 Walk; 5 Auto+Transit
*: Increase in standard error

More reliable estimates for generalization as it gives a better performance on OOS prediction

Recap and suggestions

Data
- Leveraging on Big Data
- Novel virtual experience data

Testing
- Optimize on out-of-sample error (+stat. tests)
- Model + learning algorithm

Model
- Neural nets to improve model reliability on error capturing
- Unknown errors from Big Data sources (social media etc._)

This discussion: What we can learn from machine learning?

1. Developing an out-of-sample data collection, testing and validation framework
   - Indicator of model reliability on forecasting extreme events
   - As an objective function for learning optimization

2. Addressing model misspecification
   - Methodologies for machine learning can be used in discrete choice
     - Our example: residual neural networks ←→ error correction function
     - Endogeneity is also an issue in deep learning
     - Prediction % performance can be informative on generalizability of our estimates

Conclusions

Neural networks are great for fitting model to the "unknown unknowns"

- Impossible to predict the future
- But neural networks (+Big Data) are getting really good at it

Similarities in Machine Learning and Discrete Choice

- What can we learn from each other?

Statistical testing of model generalization

- Leverage on out-of-sample prediction tests
- Measure model specification reliability from prediction error

References

Case Study

Wong, M. and Farooq, B., 2019. ResLogit: A residual neural network logit model. arXiv preprint arXiv:1912.10058.

Machine learning methods for DCM

Han, Y., Zegras, C., Pereira, F.C. and Ben-Akiva, M., 2020. A Neural-embedded Choice Model: TasteNet-MNL Modeling Taste Heterogeneity with Flexibility and Interpretability. arXiv preprint arXiv:2002.00922.

Sifringer, B., Lurkin, V. and Alahi, A., 2020. Enhancing discrete choice models with representation learning. Transportation Research Part B: Methodological, 140, pp.236-261.

Ortelli, N., Hillel, T., Pereira, F.C., de Lapparent, M. and Bierlaire, M., 2020. Assisted Specification of Discrete Choice Models. Tech. Report TRANSP-OR 200708, EPFL

Validation in Machine Learning

Nadeau, C. and Bengio, Y., 2000. Inference for the generalization error. In *Advances in neural information processing systems* (pp. 307-313).

Alwosheel, A., van Cranenburgh, S. and Chorus, C.G., 2018. Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of choice modelling*, *28*, pp.167-182.

Data Manipulation

Farooq, B., Cherchi, E. and Sobhani, A., 2018. Virtual immersive reality for stated preference travel behavior experiments: A case study of autonomous vehicles on urban roads. *Transportation research record*, *2672*(50), pp.35-45.

Balcan, D., Gonçalves, B., Hu, H., Ramasco, J.J., Colizza, V. and Vespignani, A., 2010. Modeling the spatial spread of infectious diseases: The GLobal Epidemic and Mobility computational model. *Journal of computational science*, *1*(3), pp.132-145.