Now- and Backcasting Initial Claims with High-Dimensional Daily Internet Search-Volume Data

Daniel Borup[1] David Rapach[2,3] Erik C. Montes Schütte[1]

[1]Aarhus University [2]Washington University in St Louis [3]Saint Louis University

2020 Banca d'Italia & Federal Reserve Board
Joint Conference on Nontraditional Data & Statistical Learning

November 11, 2020

- COVID-19 $\Rightarrow$ economic upheaval in US
  - March 13, 2020 $\Rightarrow$ national emergency $\Rightarrow$ closures of non-essential retail establishments in many parts of US
- Unemployment insurance initial claims
  - Spiked in late March $\Rightarrow$ 6.9M for week ending March 28
    - Subsequently declined, but remains at elevated levels
  - Perhaps most closely watched economic variable during COVID-19 crisis
    - High-frequency (weekly) labor market indicator
    - US weekly economic indicator (Lewis, Mertens & Stock 2020)

- We use rich trove of daily internet search-volume data from Google Trends to predict US weekly UI
  - Seek to improve prediction during COVID-19 crisis
- High-dimensional set of GT terms related to *unemployment*
  - Daily data for 103 GT terms for most recent 7 days
- Sequence of now-/backcasts $\Rightarrow$ Week$_t$ UI
  - Week$_t$ UI $\Rightarrow$ Sat$_t$ to Sun$_t$
  - In anticipation of data release by Dept of Labor on Thu$_{t+1}$

- Predictive models
  - UI $= f$(first/second lag of UI, 7 days of 103 GT terms)
  - High-dimensional setting $\Rightarrow$ machine learning
    - # predictors/inputs $= 7 \times 103 + 1 = 722$
- Linear specification $\Rightarrow$ penalized regression
  - LASSO (Tibshirani 1996)
  - ENet (Zou & Hastie 2005)
- Artificial neural nets
  - Allow for complex, nonlinear predictive relationships

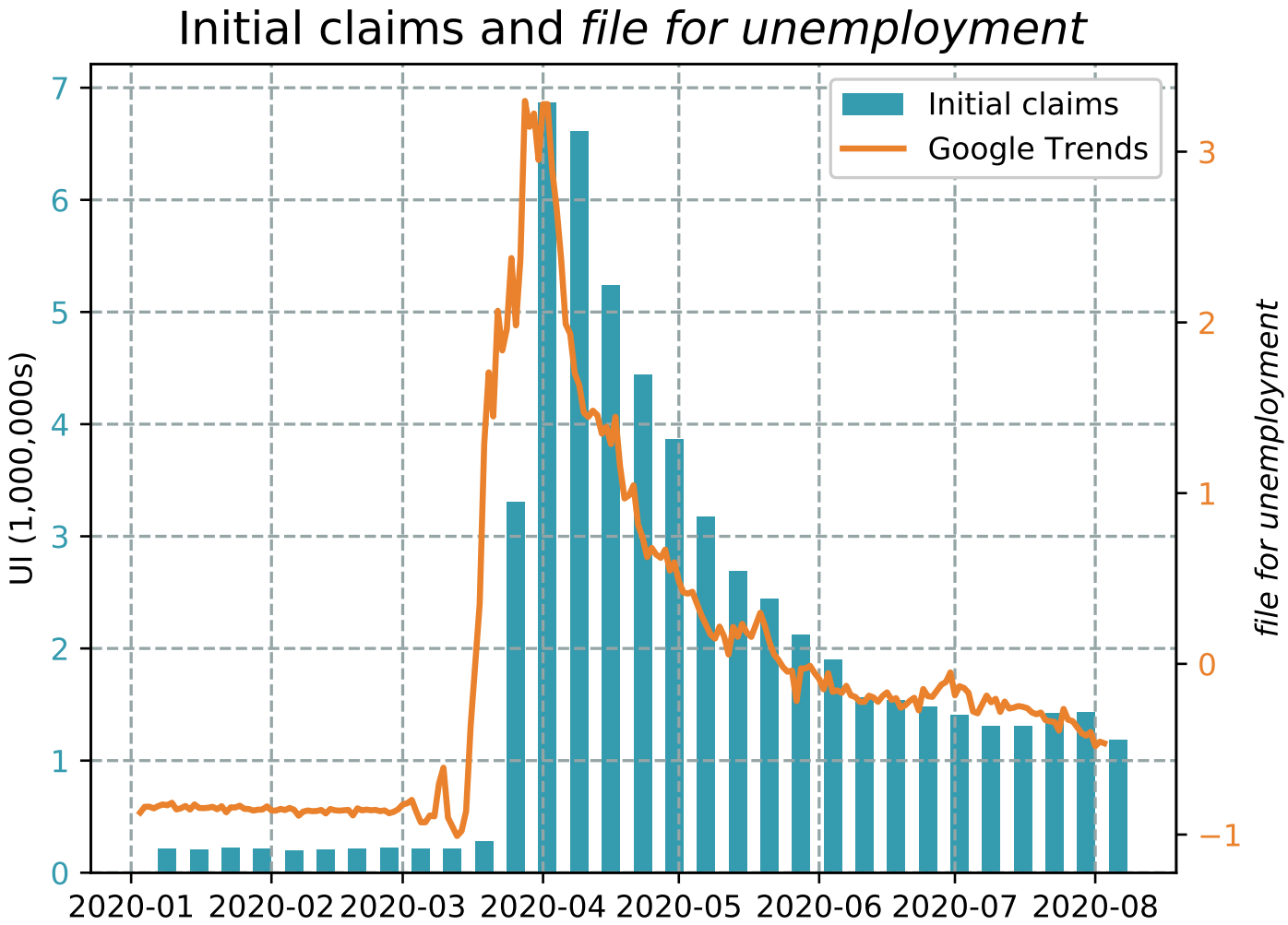
- Mixed-frequency data ⇒ weekly/daily
  - U-MIDAS (Foroni, Marcellino & Schumacher 2015)
  - Analyze how information flow affects accuracy of sequence of now-/backcasts
    - Successive elements include more recent GT data
- Combine 2 branches of economic forecasting literature
  - Machine learning (eg, Diebold & Shin 2019; Kotchoni, Leroux & Stevanovic 2019; Medeiros et al forthcoming)
  - Mixed-frequency data (eg, Clements & Galvao 2008; Foroni & Marcellino 2014; Brave, Butters & Justiniano 2019)

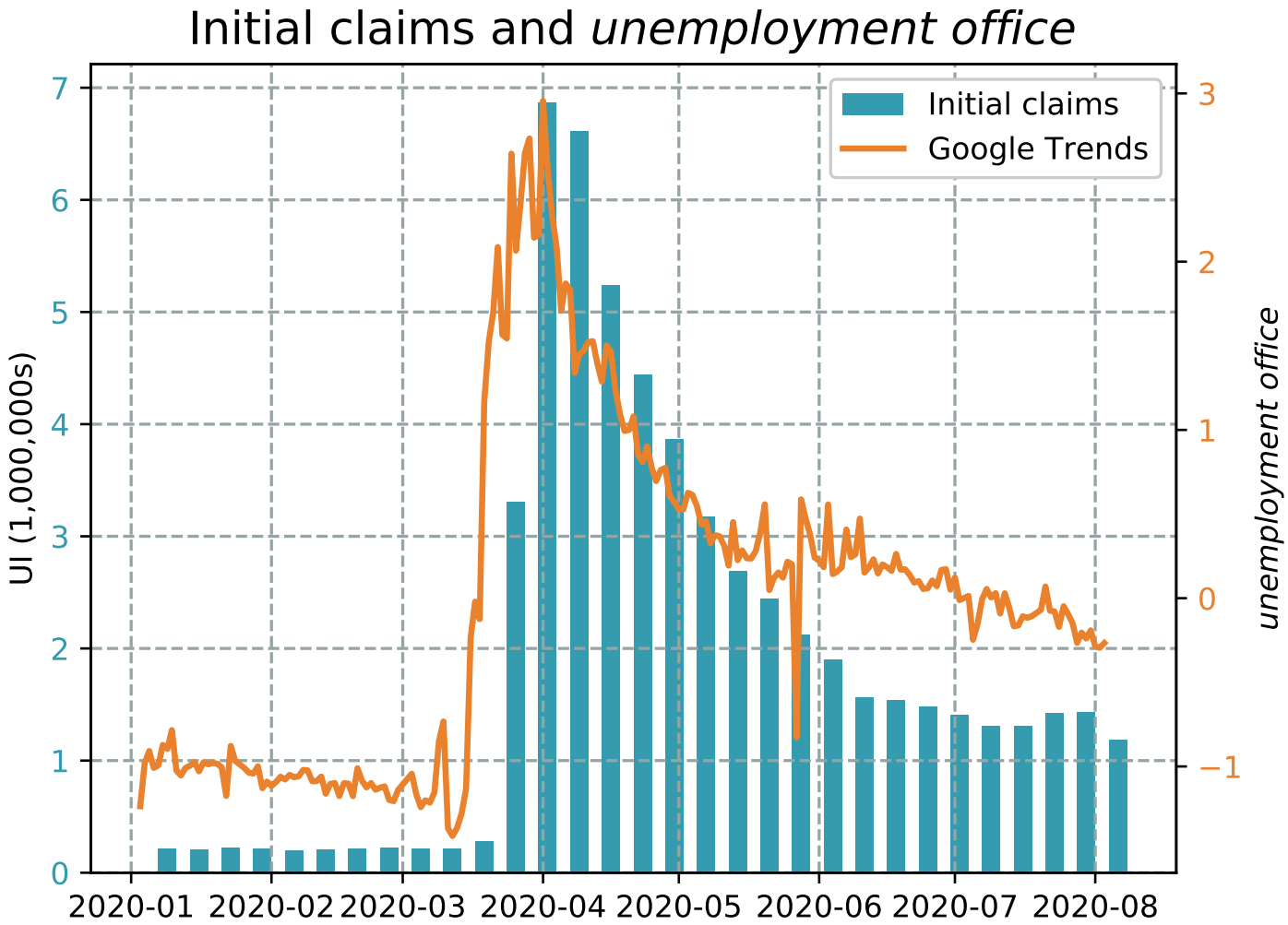
- Emerging literature ⇒ internet search-volume data to predict labor market variables
  - Choi & Varian (2012) ⇒ small # of GT terms to predict US UI (through mid 2011)
  - D'Amuri & Marcucci (2017) ⇒ GT terms including *jobs* improve US unemployment rate prediction
  - Niesert et al (2020) ⇒ array of GT terms useful for predicting unemployment rates in collection of developed countries
  - Borup & Schütte (forthcoming) ⇒ large # of GT terms & machine-learning tools improve US employment growth prediction

- Recent studies ⇒ GT terms to predict UI during COVID-19 crisis
  - Aaronson et al (2020), Goldsmith-Pinkham & Sojourner (2020), Larson & Sinclair (2020)
- Key elements of our approach
  - Consider large # of GT terms
  - Mixed-frequency framework
  - Machine learning

- Target $\Rightarrow$ weekly UI (Sat to Sun)
  - Thursday 8:30a EST $\Rightarrow$ Dept of Labor releases UI figure for previous week
- Seasonally adjusted or non-seasonally adjusted UI?
  - Subject of debate during COVID-19 crisis (Rinz 2020)
  - Dept of Labor $\Rightarrow$ recent Δ to additive SA (Davidson 2020)
    - New procedure only used going forward
  - We analyze both SA & NSA UI

- Daily search-volume data from GT
  - Index ⇒ proportion of queries for specific search term within geographical area
  - Index released with 36-hour delay
    - Google filters irregular search activity (automated, spam)
- Start with source term *unemployment*
  - Google Keyword Planner ⇒ obtain top 15 primitive terms associated with *unemployment*
  - Expand primitive terms with 25 related terms (*top* category)
  - Remove low-volume queries ⇒ 103 unique terms


Initial claims and *file for unemployment*
Initial claims
Google Trends
UI (1,000,000s)
file for unemployment
0
1
2
3
4
5
6
7
−1
0
1
2
3
2020-01
2020-02
2020-03
2020-04
2020-05
2020-06
2020-07
2020-08


Initial claims and *unemployment office*
Initial claims
Google Trends
UI (1,000,000s)
unemployment office
0
1
2
3
4
5
6
7
−1
0
1
2
3
2020-01
2020-02
2020-03
2020-04
2020-05
2020-06
2020-07
2020-08

- Also consider 10 daily macro-financial predictors
  - INF $\Rightarrow$ expected Inflation (rate at which T-bill & TIPS achieve same yield)
  - GLD $\Rightarrow$ gold return (London afternoon fixing price)
  - VIX $\Rightarrow$ CBOE market volatility index
  - TED $\Rightarrow$ 3-mo LIBOR yield minus 3-mo T-bill yield
  - TERM $\Rightarrow$ 10-yr T-bond yield minus 3-mo T-bill yield
  - REITS $\Rightarrow$ Wilshire US REIT total market index return

- 10 daily macro-financial predictors (cont'd)
  - DEF $\Rightarrow$ Baa-rated corp bond yield minus Aaa-rated corp bond yield
  - SP500 $\Rightarrow$ S&P 500 index return
  - EPU $\Rightarrow$ economic policy uncertainty index (Baker, Bloom & Davis 2016)
  - DIS $\Rightarrow$ news-based infectious disease equity market volatility tracker (http://www.policyuncertainty.com/)

Prediction formation	Backcast/ nowcast	GT used for prediction	Overlap (days)	Available UI release
Mon_t	Nowcast	Sun_{t-1} to Sat_{t-1}	0	Week_{t-2}
Tue_t	Nowcast	Mon_{t-1} to Sun_t	1	Week_{t-2}
Wed_t	Nowcast	Tue_{t-1} to Mon_t	2	Week_{t-2}
Thu_t	Nowcast	Wed_{t-1} to Tue_t	3	Week_{t-1}
Fri_t	Nowcast	Thu_{t-1} to Wed_t	4	Week_{t-1}
Sat_t	Nowcast	Fri_{t-1} to Thu_t	5	Week_{t-1}
Sun_{t+1}	Backcast	Sat_{t-1} to Fri_t	6	Week_{t-1}
Mon_{t+1}	Backcast	Sun_t to Sat_t	7	Week_{t-1}
Tue_{t+1}	Backcast	Mon_t to Sun_{t+1}	6	Week_{t-1}
Wed_{t+1}	Backcast	Tue_t to Mon_{t+1}	5	Week_{t-1}

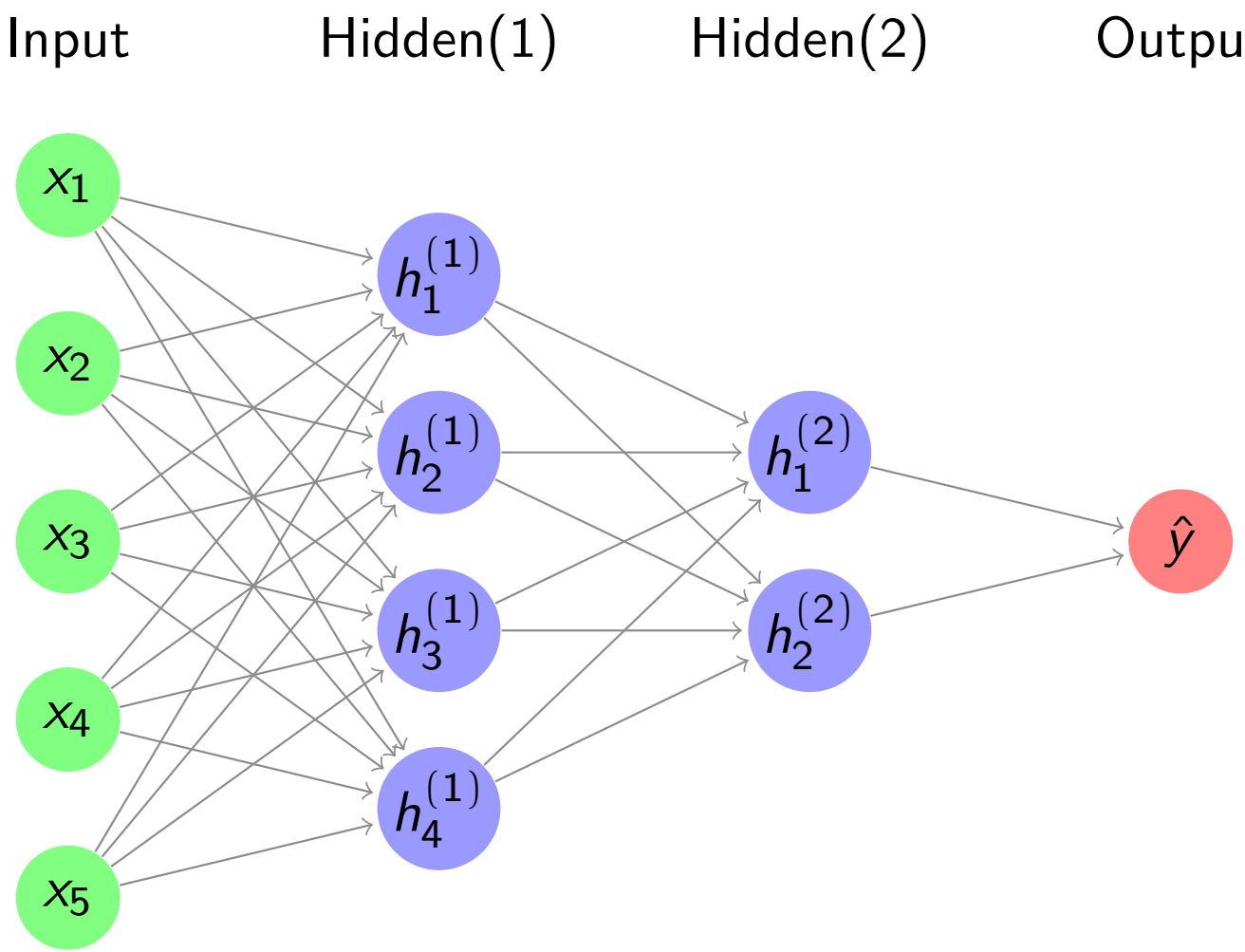
- $\text{UI}_t = f^{(j)}\left(\text{UI}_{t-1(2)}, \boldsymbol{g}_t^{(j)}; \theta^{(j)}\right)$
  - $\underbrace{\boldsymbol{g}_t^{(j)}}_{7K\times 1} = \left[\begin{array}{cccc} \boldsymbol{g}'_{t-j/7} & \boldsymbol{g}'_{t-(j+1)/7} & \cdots & \boldsymbol{g}'_{t-(j+6)/7} \end{array}\right]'$
  - $\boldsymbol{g}_{t-i/7} \Rightarrow K$-vector of GT terms for $(7-i)$th day of Week_t
    - Sunday, $\boldsymbol{g}_{t-6/7}$; Monday, $\boldsymbol{g}_{t-5/7}$; Tuesday, $\boldsymbol{g}_{t-4/7}$; Wednesday, $\boldsymbol{g}_{t-3/7}$; Thursday, $\boldsymbol{g}_{5-2/7}$; Friday, $\boldsymbol{g}_{t-1/7}$; Saturday, $\boldsymbol{g}_t$
  - $K = 103$
- $\#$ of predictors $\Rightarrow 7 \times 103 + 1 = 722$
  - $7 \times 113 + 1 = 792$ with macro-financial predictors

- AR benchmark $\Rightarrow UI_t = \alpha_0 + \alpha_1 UI_{t-1(2)} + \varepsilon_t$
- Linear $\Rightarrow UI_t = \beta_0^{(j)} + \beta_{AR}^{(j)} UI_{t-1(2)} + \boldsymbol{\beta}_g^{(j)\prime} \boldsymbol{g}_t^{(j)} + \varepsilon_t^{(j)}$
  - $\boldsymbol{\beta}_g^{(j)} \Rightarrow 7K$-vector of slope coefficients for daily GT terms
  - $\theta^{(j)} = \begin{bmatrix} \beta_0^{(j)} & \beta_{AR}^{(j)} & \boldsymbol{\beta}_g^{(j)\prime} \end{bmatrix}'$
- U-MIDAS model
  - Higher-frequency predictors in $\boldsymbol{g}_t^{(j)}$ have own coefficients
  - Machine-learning methods $\Rightarrow$ flexibly estimate weights (rather than impose lag-polynomial structure) while guarding against overfitting

- LASSO $\Rightarrow$ penalized regression
  - $$\underset{\boldsymbol{\theta}^{(j)} \in \mathbb{R}^{7K+2}}{\arg\min} \; \frac{1}{2\mathcal{T}} \left\{ \sum_{t=1}^{\mathcal{T}} \left[\mathrm{UI}_t - \left(\beta_0^{(j)} + \beta_{\mathrm{AR}}^{(j)} \mathrm{UI}_{t-1(2)} + \boldsymbol{\beta}_g^{(j)\prime} \boldsymbol{g}_t^{(j)} \right) \right] \right\}^2 + \lambda \| \boldsymbol{\beta}^{(j)} \|_1$$
    - $\boldsymbol{\beta}^{(j)} = \begin{bmatrix} \beta_{\mathrm{AR}}^{(j)} & \boldsymbol{\beta}_g^{(j)\prime} \end{bmatrix}'$
    - $\lambda \geqslant 0 \Rightarrow$ regularization parameter
    - Select λ via ERIC (Hui, Warton & Foster 2015)
  - ℓ_1 penalty $\Rightarrow$ permits shrinkage to zero (ie, variable selection)

- ENet $\Rightarrow$ refinement of LASSO
  - $$\underset{\boldsymbol{\theta}^{(j)} \in \mathbb{R}^{7K+2}}{\arg\min} \frac{1}{2\mathcal{T}} \left\{ \sum_{t=1}^{\mathcal{T}} \left[\mathrm{UI}_t - \left(\beta_0^{(j)} + \beta_{\mathrm{AR}}^{(j)} \mathrm{UI}_{t-1(2)} + \boldsymbol{\beta}_g^{(j)\prime} \boldsymbol{g}_t^{(j)} \right) \right] \right\}^2 + \lambda P_\alpha \left(\boldsymbol{\beta}^{(j)} \right)$$
    - $P_\alpha \left(\boldsymbol{\beta}^{(j)} \right) = 0.5(1-\alpha) \| \boldsymbol{\beta}^{(j)} \|_2^2 + \alpha \| \boldsymbol{\beta}^{(j)} \|_1$
    - $0 \leqslant \alpha \leqslant 1 \Rightarrow$ blending parameter for ℓ_1 & ℓ_2 components
    - Set $\alpha = 0.5$ (Hastie & Qian 2016)
  - Compared to LASSO, ENet tends to select highly correlated predictors as group

- Input layer (set of predictors) $\Rightarrow x_1, \ldots, x_{P_0}$
- One or more hidden layers $\Rightarrow$ take signals from neurons in previous layer to generate subsequent signals
  - $h_m^{(l)} = g\left(w_{m,0}^{(l)} + \sum_{j=1}^{P_{l-1}} w_{m,j}^{(l)} h_j^{(l-1)}\right)$
    - Hidden layers $\Rightarrow l = 1, \ldots, L$; neurons $\Rightarrow m = 1, \ldots, P_l$
    - ReLU $\Rightarrow g(x) = 0$ if $x < 0$, $g(x) = x$ otherwise
- Output layer $\Rightarrow$ translates signals from last hidden layer into prediction
  - $\hat{y} = w_0^{(L+1)} + \sum_{j=1}^{P_L} w_j^{(L+1)} h_j^{(L)}$


Input
Hidden(1)
Hidden(2)
Output
x_1
x_2
x_3
x_4
x_5
$h_1^{(1)}$
$h_2^{(1)}$
$h_3^{(1)}$
$h_4^{(1)}$
$h_1^{(2)}$
$h_2^{(2)}$
$\hat{y}$

- Specify 3 ANN architectures
  - NN1 $\Rightarrow$ 1 hidden layer with 104 neurons
  - NN2 $\Rightarrow$ 2 hidden layers with 104 & 10 neurons
  - NN3 $\Rightarrow$ 3 hidden layers with 104, 10 & 3 neurons
- Training ANN $\Rightarrow$ estimate (very many) weights
  - Minimize training-sample MSE augmented with ℓ_1 penalty
  - Adam SGD algorithm (Kingma & Ba 2015)
    - `Python` $\Rightarrow$ `keras` package

- Ensemble-Linear
  - Avg of Linear-LASSO & Linear-ENet
- Ensemble-ANN
  - Avg of NN1, NN2 & NN3
- Ensemble-All
  - Avg of Linear-LASSO, Linear-ENet, NN1, NN2 & NN3
- All models fitted using 10-yr rolling window
  - Full sample $\Rightarrow$ Jan 2005 to Aug 2020
  - Out-of-sample period $\Rightarrow$ Jan 2015 to Aug 2020

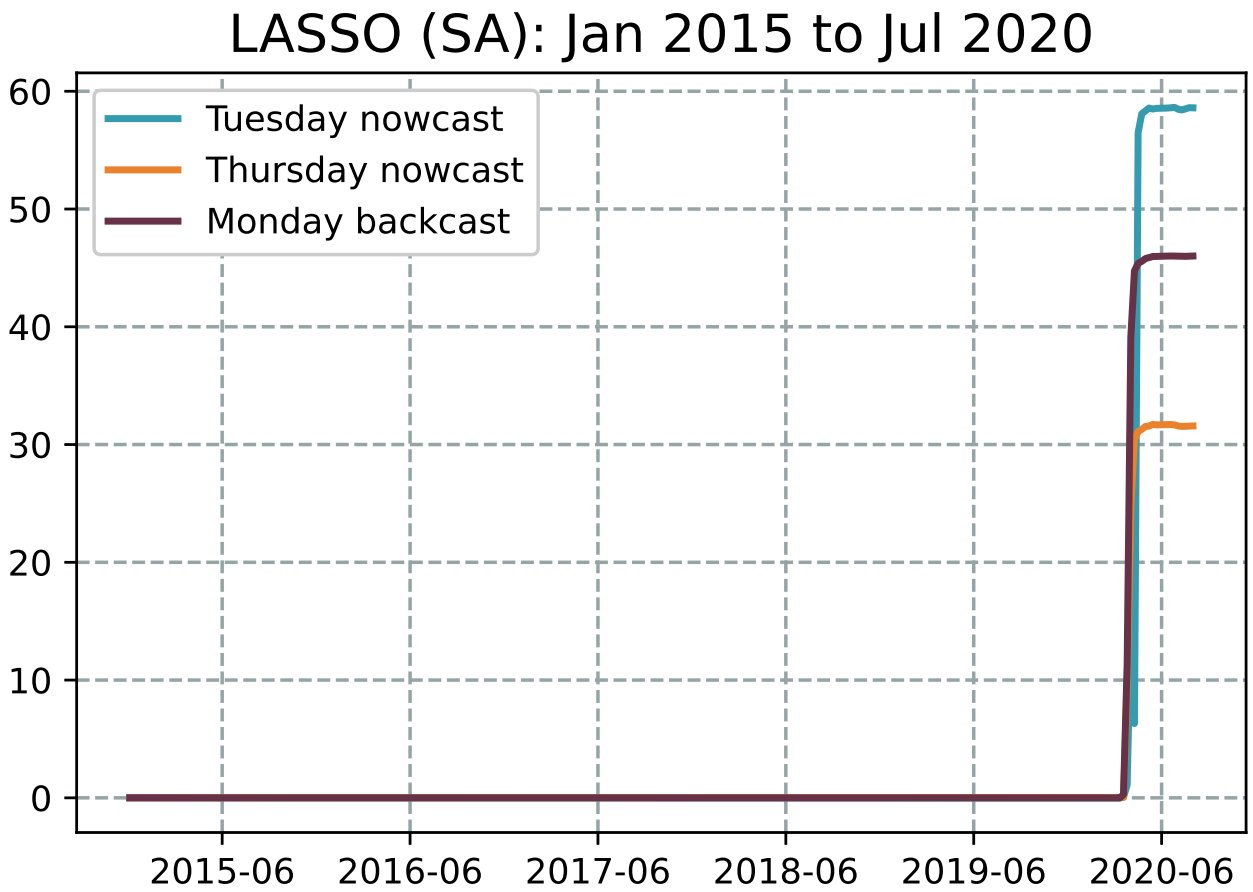
RMSE ratios, seasonally adjusted UI

Prediction	AR RMSE	LASSO	ENet	NN1	NN2	NN3
Mon_t	642,032	0.8291	**0.8111**	0.9736	0.9123	0.8130
Tue_t	642,032	**0.7152**	0.7181	0.8658	0.9225	0.7477
Wed_t	642,032	0.6769	**0.6698**	0.7475	0.7418	0.8340
Thu_t	445,565	0.6748	**0.6598**	0.9753	0.6890	0.7610
Fri_t	445,565	**0.6031**	0.6167	0.7583	0.6301	0.6435
Sat_t	445,565	**0.4925**	0.5130	0.5998	0.5761	0.7204
Sun_{t+1}	445,565	**0.4651**	0.5269	0.5489	0.5591	0.5971
Mon_{t+1}	445,565	**0.4543**	0.5171	0.5355	0.5367	0.6049
Tue_{t+1}	445,565	**0.4643**	0.4743	0.4969	0.5345	0.6017
Wed_{t+1}	445,565	0.4601	0.4947	**0.4413**	0.5384	0.6857

RMSE ratios, seasonally adjusted UI

Prediction	AR RMSE	Ens-Linear	Ens-ANN	Ens-All
Mon_t	642,032	**0.8197**	0.8862	0.8422
Tue_t	642,032	**0.7145**	0.8194	0.7566
Wed_t	642,032	**0.6723**	0.7345	0.6898
Thu_t	445,565	**0.6653**	0.7739	0.7124
Fri_t	445,565	**0.6059**	0.6381	0.6175
Sat_t	445,565	**0.5008**	0.5892	0.5321
Sun_{t+1}	445,565	**0.4913**	0.5259	0.5044
Mon_{t+1}	445,565	**0.4828**	0.5281	0.5018
Tue_{t+1}	445,565	**0.4607**	0.5167	0.4845
Wed_{t+1}	445,565	**0.4743**	0.4963	0.4827

Cumulative differences in squared errors



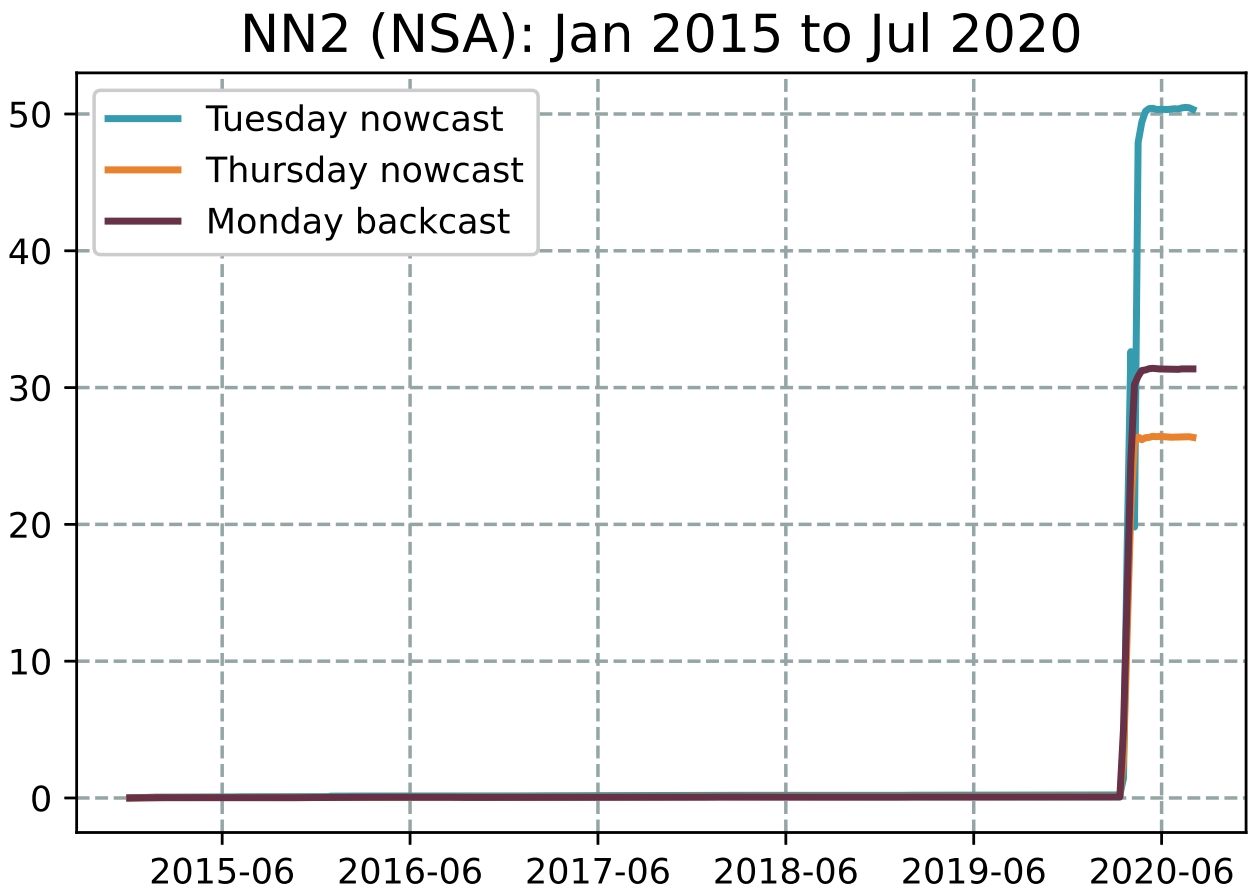
RMSE ratios, non-seasonally adjusted UI

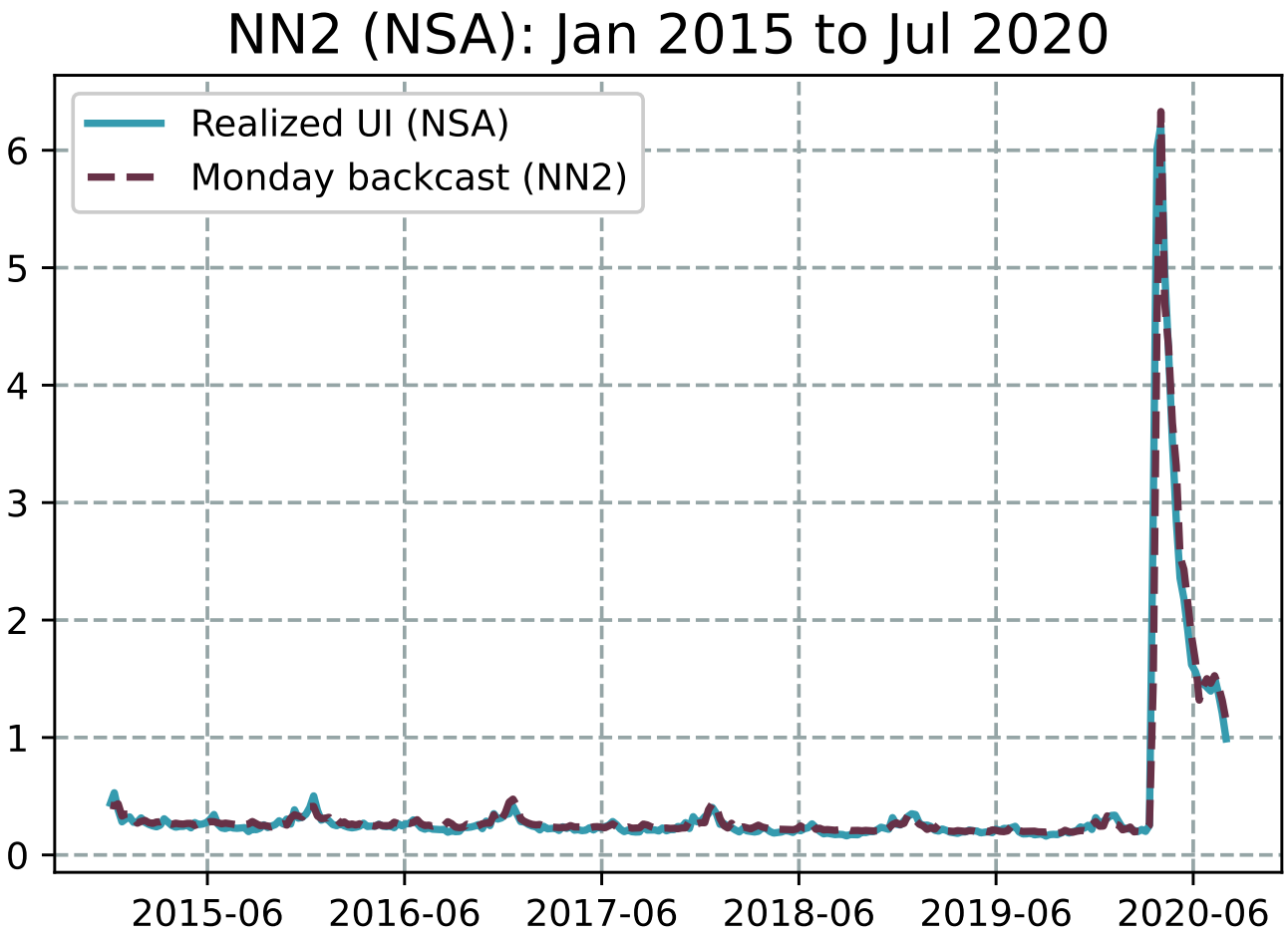
Prediction	AR RMSE	LASSO	ENet	NN1	NN2	NN3
Mon_t	569,418	0.7448	0.7553	**0.6975**	0.8412	0.7629
Tue_t	569,418	0.6821	0.6336	**0.6291**	0.6830	0.7291
Wed_t	569,418	0.6262	0.6189	0.6583	0.6285	**0.5826**
Thu_t	352,507	0.6478	0.6424	0.7046	**0.5230**	0.6274
Fri_t	352,507	0.6014	**0.5734**	0.6766	0.6071	0.5967
Sat_t	352,507	0.5525	0.5434	0.6770	**0.4330**	0.6773
Sun_{t+1}	352,507	0.5802	0.5867	**0.4357**	0.4753	0.4505
Mon_{t+1}	352,507	0.4970	0.4805	0.3830	**0.3682**	0.5540
Tue_{t+1}	352,507	0.4958	0.4909	**0.3925**	0.4632	0.6192
Wed_{t+1}	352,507	0.5334	0.5404	**0.4473**	0.5544	0.6067

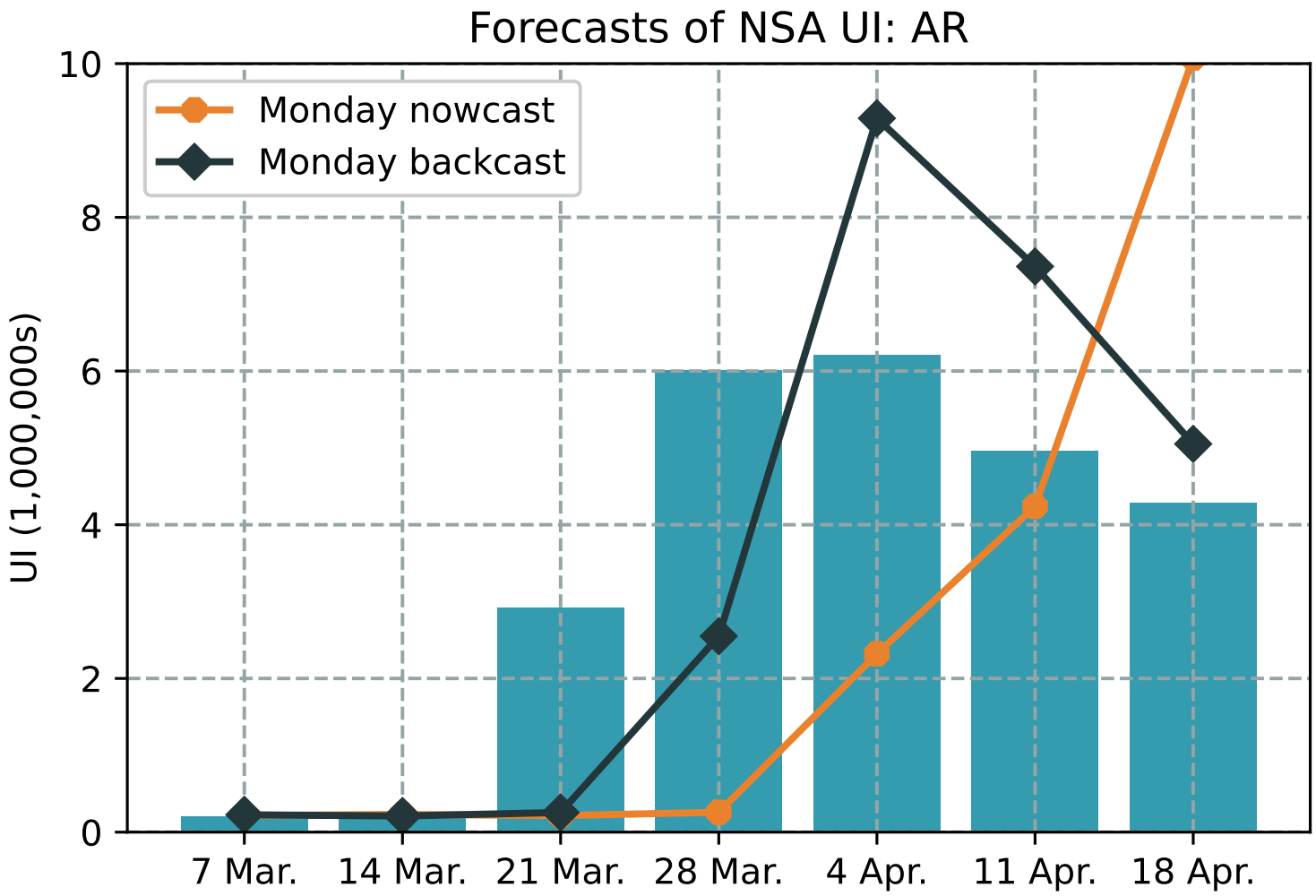
RMSE ratios, non-seasonally adjusted UI

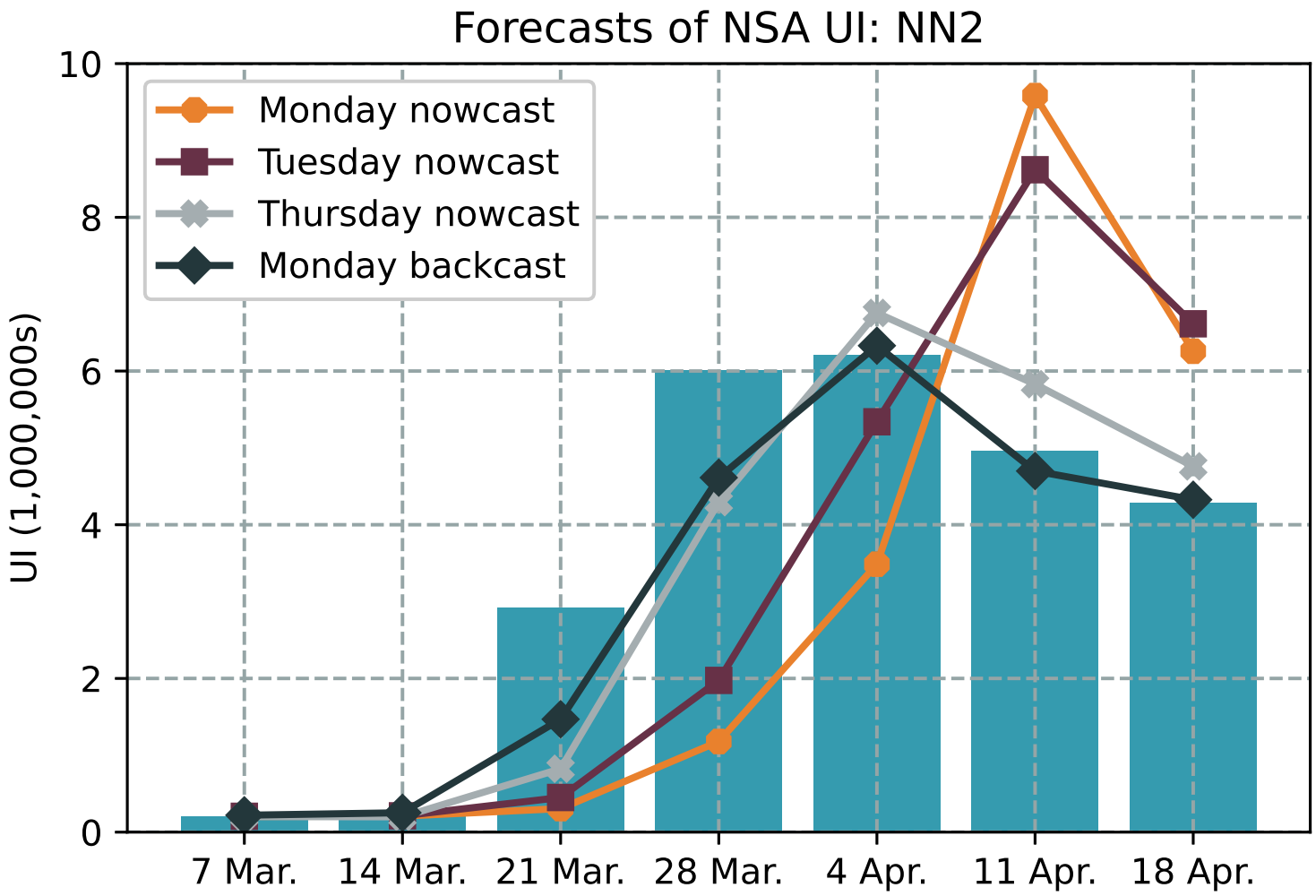
Prediction	AR RMSE	Ens-Linear	Ens-ANN	Ens-All
Mon_t	642,032	0.7491	0.7583	**0.7466**
Tue_t	642,032	0.6815	**0.6572**	0.6604
Wed_t	642,032	0.6205	**0.6025**	0.6030
Thu_t	445,565	0.6443	**0.5472**	0.5879
Fri_t	445,565	0.5865	**0.5768**	0.5773
Sat_t	445,565	0.5456	**0.4751**	0.5043
Sun_{t+1}	445,565	0.5831	**0.4208**	0.4915
Mon_{t+1}	445,565	0.4869	**0.3887**	0.4345
Tue_{t+1}	445,565	0.4927	**0.4424**	0.4631
Wed_{t+1}	445,565	0.5355	**0.4821**	0.5013

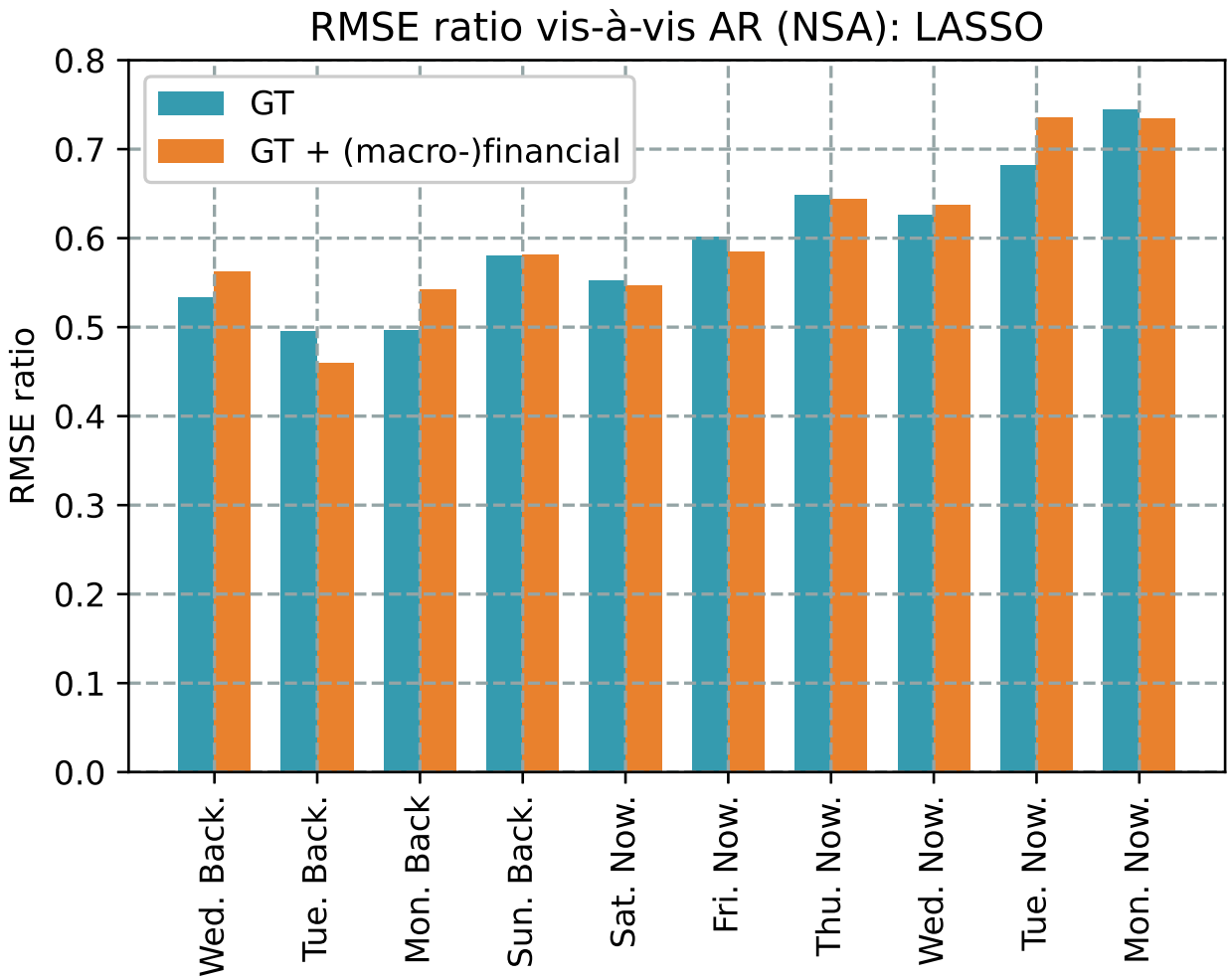
Cumulative differences in squared errors

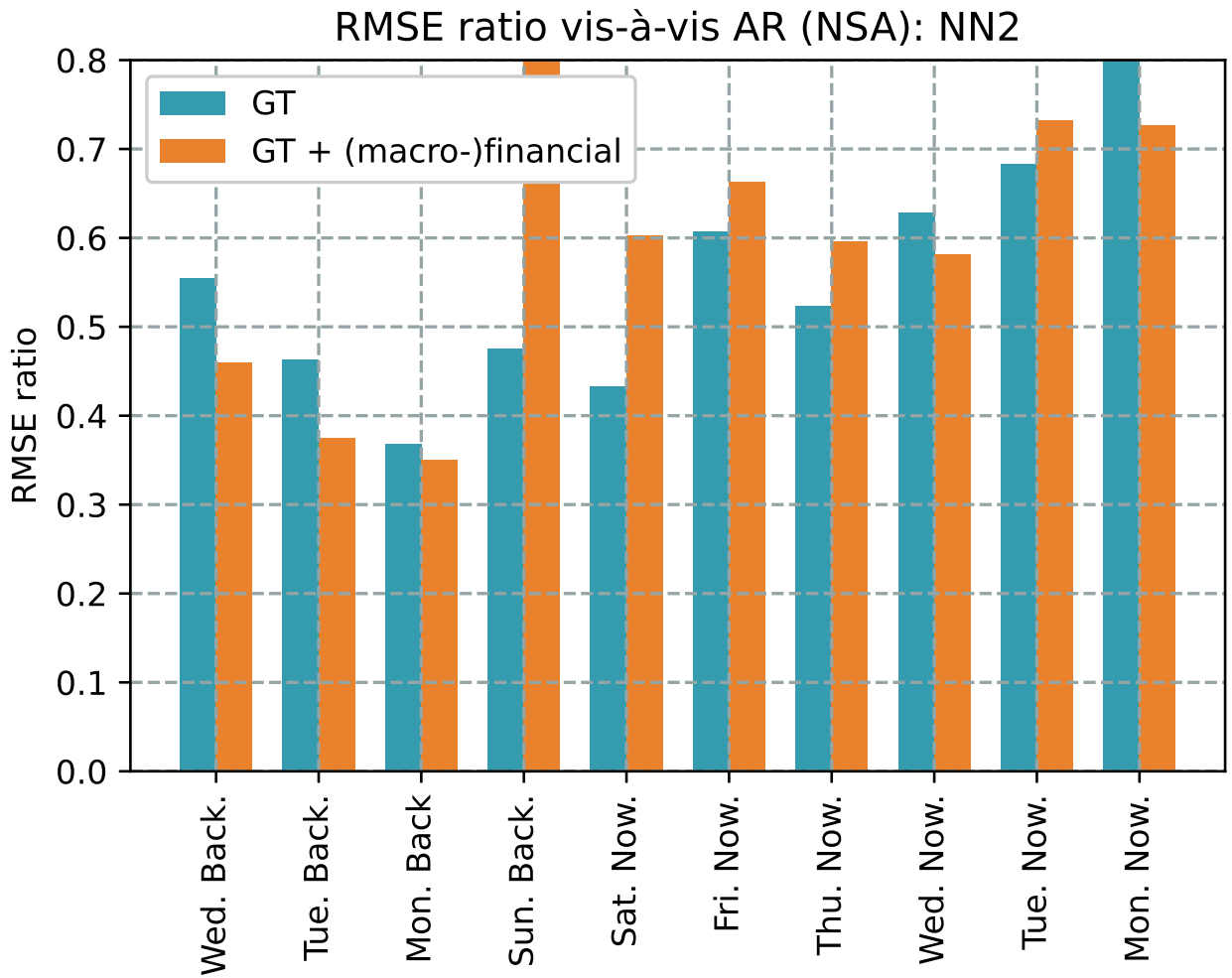



NN2 (NSA): Jan 2015 to Jul 2020
Realized UI (NSA)
Monday backcast (NN2)
0
1
2
3
4
5
6
2015-06
2016-06
2017-06
2018-06
2019-06
2020-06


Forecasts of NSA UI: AR
Monday nowcast
Monday backcast
UI (1,000,000s)
10
8
6
4
2
0
7 Mar.
14 Mar.
21 Mar.
28 Mar.
4 Apr.
11 Apr.
18 Apr.


Forecasts of NSA UI: NN2
Monday nowcast
Tuesday nowcast
Thursday nowcast
Monday backcast
UI (1,000,000s)
0
2
4
6
8
10
7 Mar.
14 Mar.
21 Mar.
28 Mar.
4 Apr.
11 Apr.
18 Apr.


RMSE ratio vis-à-vis AR (NSA): LASSO
GT
GT + (macro-)financial
RMSE ratio
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Wed. Back.
Tue. Back.
Mon. Back
Sun. Back.
Sat. Now.
Fri. Now.
Thu. Now.
Wed. Now.
Tue. Now.
Mon. Now.


RMSE ratio vis-à-vis AR (NSA): NN2
GT
GT + (macro-)financial
RMSE ratio
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Wed. Back.
Tue. Back.
Mon. Back
Sun. Back.
Sat. Now.
Fri. Now.
Thu. Now.
Wed. Now.
Tue. Now.
Mon. Now.

- Partial dependence plot (Friedman 2001)
  - Marginal effect of $x_s^{(j)}$ on expected value of UI_t for fitted model
- Predictors $\Rightarrow \boldsymbol{x}_t^{(j)} = \left[\begin{array}{cc} \text{UI}_{t-1(2)} & \boldsymbol{g}_t^{(j)} \end{array} \right]'$
- $722(792) \times \mathcal{T}$ data matrix $\Rightarrow \boldsymbol{X}_{\mathcal{T}}^{(j)} = \left[\begin{array}{ccc} \boldsymbol{x}_1^{(j)} & \cdots & \boldsymbol{x}_{\mathcal{T}}^{(j)} \end{array} \right]'$
- $\boldsymbol{x}_{C(s)}^{(j)} = \boldsymbol{x}^{(j)} \setminus x_s^{(j)}$

- $\text{PD}\left(x_s^{(j)}\right) = \mathbb{E}_{\boldsymbol{x}_{C(s)}^{(j)}}\left[\hat{f}^{(j)}\left(x_s^{(j)}, \boldsymbol{x}_{C(s)}^{(j)}\right)\right]$

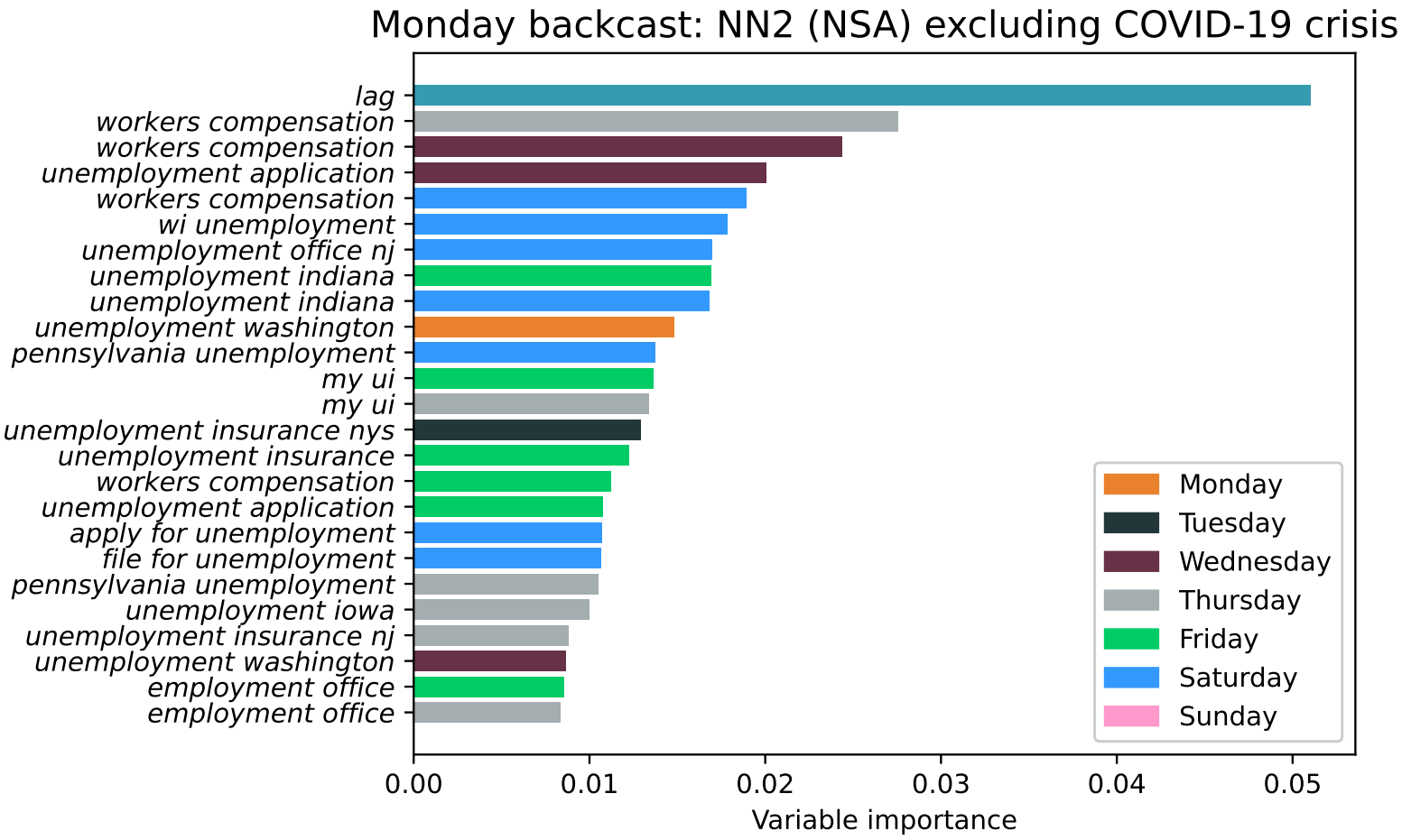
  $= \int_{\boldsymbol{x}_{C(s)}^{(j)}} \hat{f}^{(j)}\left(x_s^{(j)}, \boldsymbol{x}_{C(s)}^{(j)}\right) p_{C(s)^{(j)}}\left(\boldsymbol{x}_{C(s)}^{(j)}\right) d\boldsymbol{x}_{C(s)^{(j)}}$

  - $p_{C(s)^{(j)}}\left(\boldsymbol{x}_{C(s)}^{(j)}\right) = \int_{x_s^{(j)}} p\left(\boldsymbol{x}^{(j)}\right) dx_s^{(j)}$
  - $p\left(\boldsymbol{x}^{(j)}\right) \Rightarrow$ joint probability density for $\boldsymbol{x}^{(j)}$
  - $\hat{f}\left(\boldsymbol{x}^{(j)}\right) \Rightarrow$ prediction function for fitted model

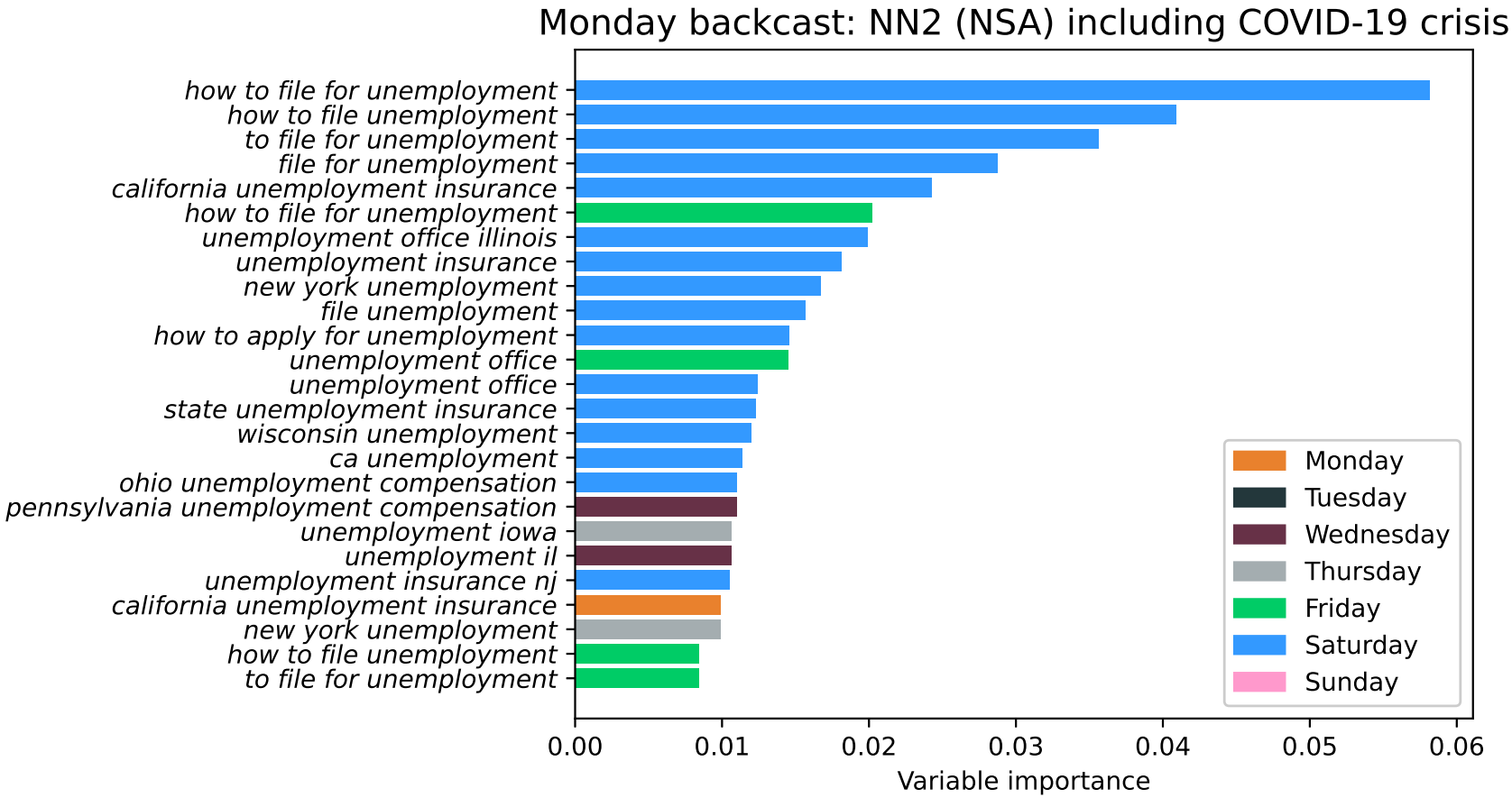
- Estimate via Monte Carlo integration using training sample
  - $\widehat{\text{PD}}\left(x_s^{(j)}\right) = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \hat{f}^{(j)}\left(x_s^{(j)}, \boldsymbol{x}_{C(s),t}^{(j)}\right)$
    - Evaluate at $x_{s,t}^{(j)}$ for $t = 1, \ldots, \mathcal{T}$ (or set of quantiles)
- PDP for linear model $\Rightarrow$ constant slope
  - Horizontal line for predictor not selected by LASSO/ENet
- Compare PDPs for fitted linear models & ANNs
  - Gauge relative importance of nonlinearities in ANNs

- Variable importance (Greenwell, Boehmke & McCarthy 2018)
  - $\hat{\mathcal{J}}\left(x_s^{(j)}\right) = \left\{ \frac{1}{\mathcal{T}-1} \sum_{t=1}^{\mathcal{T}} \left[\widehat{\text{PD}}\left(x_{s,t}^{(j)}\right) - \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \widehat{\text{PD}}\left(x_{s,t}^{(j)}\right) \right]^2 \right\}^{0.5}$
  - Variation in PDP around average (ie, std dev)
    - Horizontal PDP $\Rightarrow$ variable importance $= 0$
- Scale to facilitate comparison across predictors
  - $\tilde{\mathcal{J}}\left(x_s^{(j)}\right) = \frac{\hat{\mathcal{J}}\left(x_s^{(j)}\right)}{\sum_{p=1}^{P} \hat{\mathcal{J}}\left(x_p^{(j)}\right)}$
    - $P \Rightarrow$ total # of predictors

Importance of GT Terms

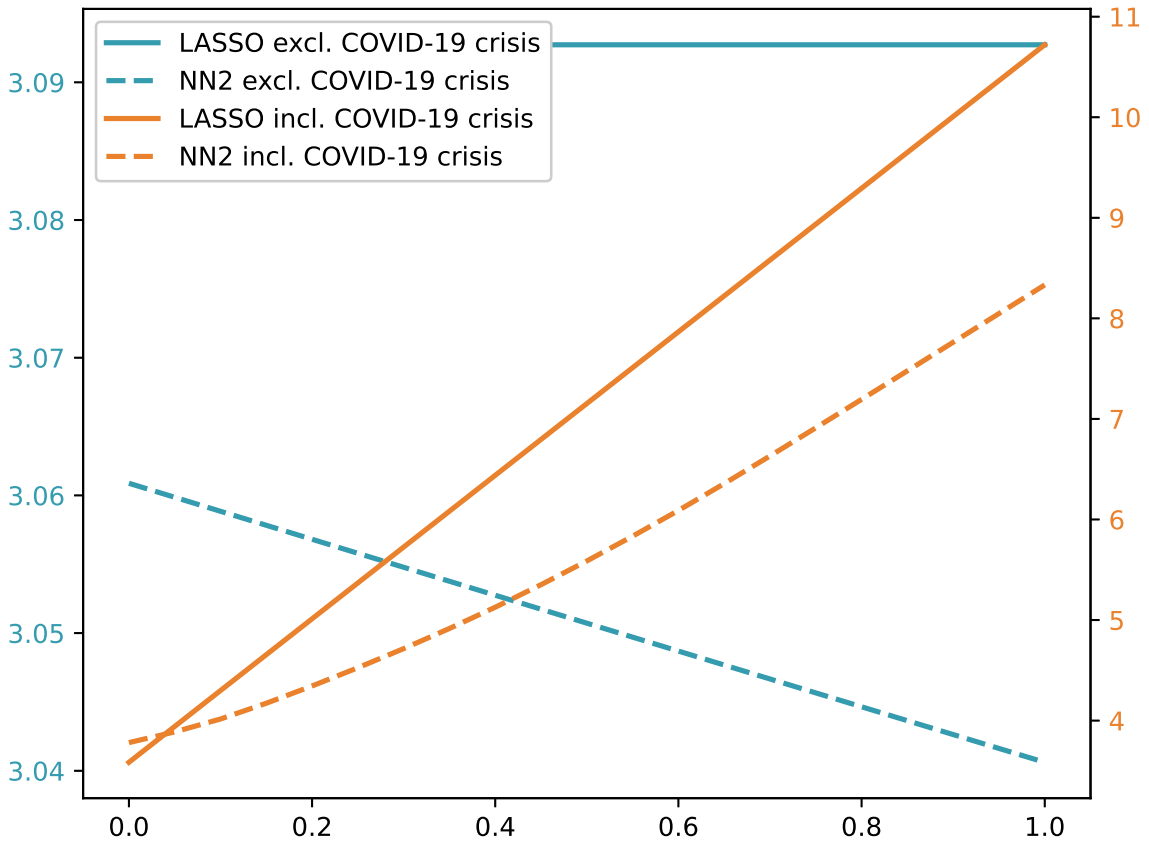


Importance of GT Terms



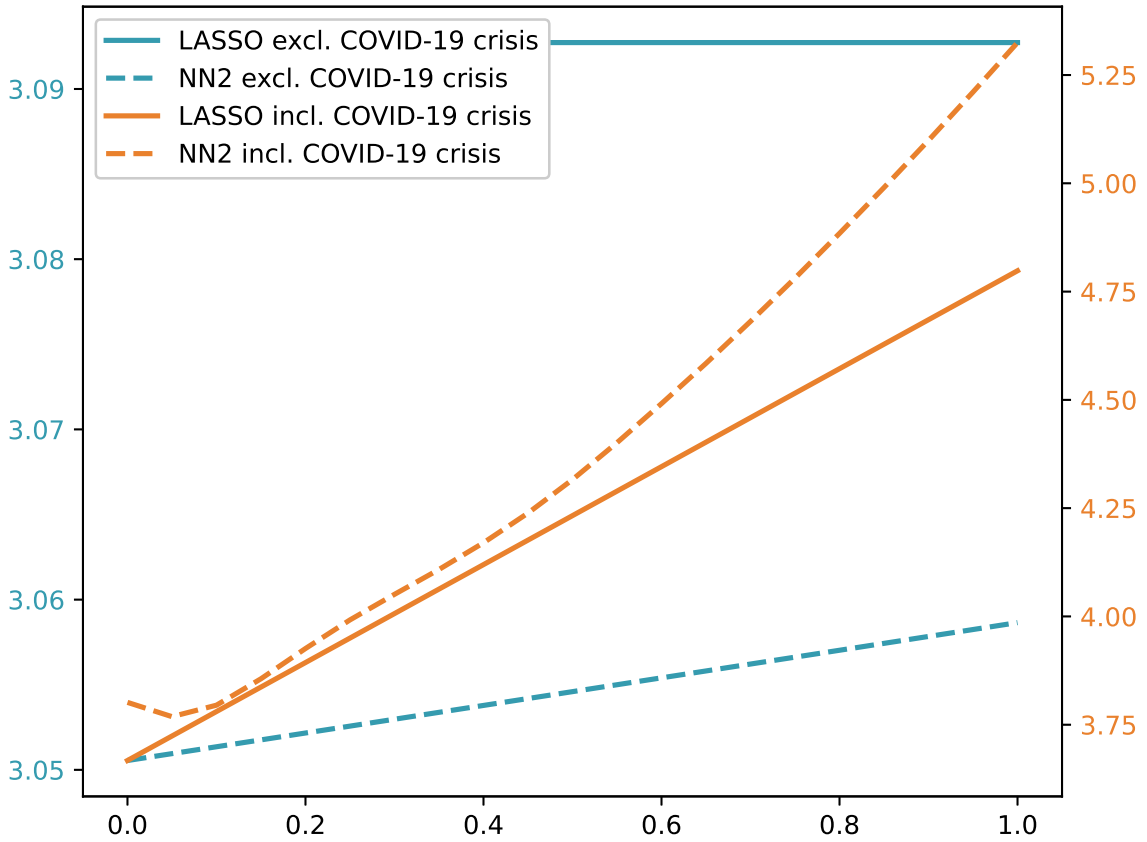
Partial dependence plots

Monday backcast: *how to file for unemployment*



Partial dependence plots

Monday backcast: *unemployment office illinois*



- Daily internet search-volume data for 103 GT terms related to *unemployment* improve now-/backcasts of weekly UI
  - Especially during advent of COVID-19 crisis
- GT terms become more important during COVID-19 crisis
- Nonlinearities more relevant for predicing NSA UI
- Combining mixed-frequency approach (via U-MIDAS) with machine learning appears promising
- New website provides daily updated now-/backcasts
  - https://www.uinowcast.org/