What about Time Series? Applying polyreg and toweranNA to Time Series

Norm Matloff
University of California at Davis

Bay Area R Users Group
GRAIL, June 11, 2019

(slides will be available at
http://heather.cs.ucdavis.edu/TimeSeries.pdf)

Introduction

Introduction

Previous talks:

Introduction

Previous talks:

- BARUG/GRAIL, June 18, 2018; SaRday, UCLA, April 6
  - neural nets essentially perform polynomial regression
  - **polyreg** package

Introduction

Previous talks:

- BARUG/GRAIL, June 18, 2018; SaRday, UCLA, April 6
  - neural nets essentially perform polynomial regression
  - **polyreg** package
- BARUG, Databricks SF, November 18, 2018
  - nonimputational method for handling NAs in <u>prediction</u>
  - **toweranNA** package

Introduction

Previous talks:

- BARUG/GRAIL, June 18, 2018; SaRday, UCLA, April 6
  - neural nets essentially perform polynomial regression
  - **polyreg** package
- BARUG, Databricks SF, November 18, 2018
  - nonimputational method for handling NAs in prediction
  - **toweranNA** package

Current talk:

- Apply these methods to time series.

What about Time Series?

- How adapt **polyreg**, **toweranNA** to time series?

Norm Matloff University of California at Davis

What about Time Series?

- How adapt **polyreg**, **toweranNA** to time series?
- Say predict X_i from $X_{i-1}, X_{i-2}, ..., X_{i-m}$, lag m.
- E.g. lag 3:

What about Time Series?

- How adapt **polyreg**, **toweranNA** to time series?
- Say predict X_i from $X_{i-1}, X_{i-2}, ..., X_{i-m}$, lag m.
- E.g. lag 3:

  x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12},... now stored as data matrix

x_1	x_2	x_3	x_4
x_2	x_3	x_4	x_5
x_3	x_4	x_5	x_6
...	...	...	...

Columns 1-3 are "X", col. 4 is "Y". Run model (poly, NN, whatever) to predict Y col. from X cols.

Covariates

Norm Matloff University of California at Davis

Covariates

E.g. what if have a single covariate C, with its own time series?

x_1	x_2	x_3	c_1	c_2	c_3	x_4
x_2	x_3	x_4	c_2	c_3	c_4	x_5
...	...	...	...	...	...	...

Now cols. 1-6 are "X", col. 7 is "Y."

Does It Work?

Norm Matloff
University of California at Davis

Does It Work?

- Work in progress.
- Early results very promising.

Norm Matloff
University of California at Davis

polyreg and NNs

polyreg and NNs

Summary of earlier talk:

- Neural nets (NNs) essentially = polynomial regression (PR).
- NOT a "universal approximation theorem"; refers to actual internal operation of NNs.
- Thus, PR should (and does) give results as good as, or better than, NNs.
- Why "or better than"?

polyreg and NNs

Summary of earlier talk:

- Neural nets (NNs) essentially = polynomial regression (PR).
- NOT a "universal approximation theorem"; refers to actual internal operation of NNs.
- Thus, PR should (and does) give results as good as, or better than, NNs.
- Why "or better than"? NNs may converge to local min, wrong answer.

Norm Matloff
University of California at Davis

polyreg

polyreg

- R package.

Norm Matloff University of California at Davis

polyreg

- R package.
- Motivated by NN=PR: use PR instead of NNs.

polyreg

- R package.
- Motivated by NN=PR: use PR instead of NNs.
- Generates all possible d-degree polynomials in p variables.

polyreg

- R package.
- Motivated by NN=PR: use PR instead of NNs.
- Generates all possible *d*-degree polynomials in *p* variables. (**Not so easy.** E.g. must skip powers of dummy variables.)

polyreg

- R package.
- Motivated by NN=PR: use PR instead of NNs.
- Generates all possible d-degree polynomials in p variables. (**Not so easy.** E.g. must skip powers of dummy variables.)
- Has dimension reduction options.

polyreg

- R package.
- Motivated by NN=PR: use PR instead of NNs.
- Generates all possible d-degree polynomials in p variables. (**Not so easy.** E.g. must skip powers of dummy variables.)
- Has dimension reduction options.
- On CRAN, but latest at *github.com/matloff/polyreg*.

Key polyreg functions

Key polyreg functions

```
polyFit(function (xy, deg, maxInteractDeg = deg,
    use = "lm", pcaMethod = NULL, pcaLocation =
    "front", pcaPortion = 0.9, glmMethod = "one",
    return_xy = FALSE, returnPoly = FALSE)

predict.polyFit(object, newdata, ...)
```

E.g. if choose dimension reduction by PCA in **polyFit()**, **predict()** will automatically take care of it.
Various other dim. reduction, helper functions.

NN=PR

NN=PR

- Consider toy example:

NN=PR

- Consider toy example:
- Activation function $a(t) = t^2$.

NN=PR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.

NN=PR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.

NN=PR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.
- Output of Layer 2 is all quartic ($d = 4$) functions of u, v.

NN=PR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.
- Output of Layer 2 is all quartic ($d = 4$) functions of u, v.
- Etc.

NN=PR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.
- Output of Layer 2 is all quartic ($d = 4$) functions of u, v.
- Etc.
- Polynomial regression!

NN=PR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.
- Output of Layer 2 is all quartic ($d = 4$) functions of u, v.
- Etc.
- Polynomial regression!
- **Important note:** The degree of the fitted polynomial in NN grows with each layer.

Norm Matloff
University of California at Davis

NN=PR: General Activation Functions

NN=PR: General Activation Functions

- Clearly this analysis for $a(t) = t^2$ extends to any polynomial activation function.

NN=PR: General Activation Functions

- Clearly this analysis for $a(t) = t^2$ extends to any polynomial activation function.
- What about transcendental $a()$?

Norm Matloff
University of California at Davis

NN=PR: General Activation Functions

- Clearly this analysis for $a(t) = t^2$ extends to any polynomial activation function.
- What about transcendental $a()$? Computer implementatations often use a Taylor series rep., i.e. a polynomial!

NN=PR: General Activation Functions

- Clearly this analysis for $a(t) = t^2$ extends to any polynomial activation function.
- What about transcendental $a()$? Computer implementatations often use a Taylor series rep., i.e. a polynomial!
- What about reLU? Same analysis, but now have piecewise polynomials, so NN=PPR.

NN=PR: General Activation Functions

- Clearly this analysis for $a(t) = t^2$ extends to any polynomial activation function.
- What about transcendental $a()$? Computer implementatations often use a Taylor series rep., i.e. a polynomial!
- What about reLU? Same analysis, but now have piecewise polynomials, so NN=PPR.
- Even without Taylor series etc.] any reasonable activation function is "close" to a polynomial.

NN=PR: General Activation Functions

- Clearly this analysis for $a(t) = t^2$ extends to any polynomial activation function.
- What about transcendental $a()$? Computer implementatations often use a Taylor series rep., i.e. a polynomial!
- What about reLU? Same analysis, but now have piecewise polynomials, so NN=PPR.
- Even without Taylor series etc.] any reasonable activation function is "close" to a polynomial.
- Hence NN=PR.

Implications of NN=PR

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!
    - No dealing with numerous hyperparameters.

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!
  - No dealing with numerous hyperparameters.
  - No convergence issues.

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!
    - No dealing with numerous hyperparameters.
    - No convergence issues.
    - No "fake minima" (NN iteration settles on a local min).

Norm Matloff University of California at Davis

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!
  - No dealing with numerous hyperparameters.
  - No convergence issues.
  - No "fake minima" (NN iteration settles on a local min).

Possible drawbacks/remedies of PR:

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!
  - No dealing with numerous hyperparameters.
  - No convergence issues.
  - No "fake minima" (NN iteration settles on a local min).

Possible drawbacks/remedies of PR:

- Large memory requirement.

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!
    - No dealing with numerous hyperparameters.
    - No convergence issues.
    - No "fake minima" (NN iteration settles on a local min).

Possible drawbacks/remedies of PR:

- Large memory requirement. Maybe use R's **bigmemory** package (with backing store).

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!
  - No dealing with numerous hyperparameters.
  - No convergence issues.
  - No "fake minima" (NN iteration settles on a local min).

Possible drawbacks/remedies of PR:

- Large memory requirement. Maybe use R's **bigmemory** package (with backing store).
- Run time (worse than NN?).

Implications of NN=PR

- Use our understanding of PR to gain insights into NNs.
- E.g. NN = PR predicts that in NNs, multicollinearity will increase from layer to layer; we've confirmed empirically.
- Heed the "advice" of NN=PR, and use PR instead of NNs!
  - No dealing with numerous hyperparameters.
  - No convergence issues.
  - No "fake minima" (NN iteration settles on a local min).

Possible drawbacks/remedies of PR:

- Large memory requirement. Maybe use R's **bigmemory** package (with backing store).
- Run time (worse than NN?). Remedy with C code, and/or GPU.

Norm Matloff
University of California at Davis

Non-Time Series Experimental Results

Non-Time Series Experimental Results

- *https://arxiv.org/abs/1806.06850*

Non-Time Series Experimental Results

- *https://arxiv.org/abs/1806.06850*
- Compared PR vs. NNs on a wide variety of datasets.

Non-Time Series Experimental Results

- *https://arxiv.org/abs/1806.06850*
- Compared PR vs. NNs on a wide variety of datasets.
- In every single dataset, **PR matched or exceeded the accuracy of NNs**.

Non-Time Series Experimental Results

- *https://arxiv.org/abs/1806.06850*
- Compared PR vs. NNs on a wide variety of datasets.
- In every single dataset, **PR matched or exceeded the accuracy of NNs**.
- Note:
  - Some attempt at optim, but certainly not exhaustive.
  - **Warning:** Beware of "p-hacking" effects. Don't take timings rankings overly seriously.

Times Series Example I

Norm Matloff University of California at Davis

Times Series Example I

- *https://github.com/jbrownlee/Datasets*

Times Series Example I

- *https://github.com/jbrownlee/Datasets*
- J. Brownlee, "Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras," *machinelearningmastery.com*

Times Series Example I

- *https://github.com/jbrownlee/Datasets*
- J. Brownlee, "Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras," *machinelearningmastery.com*

setting	RMSE
JB NN, lag 1	58.88
PR lag 1, deg 2	33.99
PR lag 5, deg 3	26.86

Time Series Example II

Norm Matloff
University of California at Davis

Time Series Example II

- Electric power demand.

Time Series Example II

- Electric power demand.
- “Using Deep Convolutional Neural Networks (DCNNs) for Time Series Forecasting Using Tensorflow - Parts 1-3,” *blog.avenuecode.com*

Time Series Example II

- Electric power demand.
- "Using Deep Convolutional Neural Networks (DCNNs) for Time Series Forecasting Using Tensorflow - Parts 1-3," *blog.avenuecode.com*
- Predict whether demand will be higher than in the last period.

Time Series Example II

- Electric power demand.
- “Using Deep Convolutional Neural Networks (DCNNs) for Time Series Forecasting Using Tensorflow - Parts 1-3,” *blog.avenuecode.com*
- Predict whether demand will be higher than in the last period.

setting	accuracy
DCNN, lag 1	0.88
PR lag 10, deg 3	0.88
Keras, ts matrix	“Broken Clock”

Missing Values

- A perennial headache.
- Vast, VAST literature.
- Major R packages, e.g. **mice** and **Amelia**.
- New CRAN Task View, already quite extensive.

Estimation vs. Prediction

- Almost all (all?) of the MV literature is on *estimation*, e.g. estimation of treatment effects.

Estimation vs. Prediction

- Almost all (all?) of the MV literature is on *estimation*, e.g. estimation of treatment effects.
- Almost all of those methods are based on *imputation*.

Estimation vs. Prediction

- Almost all (all?) of the MV literature is on *estimation*, e.g. estimation of treatment effects.
- Almost all of those methods are based on *imputation*. Requires extra assumptions beyond usual MAR, e.g. multvar. normal.

Estimation vs. Prediction

- Almost all (all?) of the MV literature is on *estimation*, e.g. estimation of treatment effects.
- Almost all of those methods are based on *imputation*. Requires extra assumptions beyond usual MAR, e.g. multvar. normal.
- Time for a new paradigm!

Estimation vs. Prediction

- Almost all (all?) of the MV literature is on *estimation*, e.g. estimation of treatment effects.
- Almost all of those methods are based on *imputation*. Requires extra assumptions beyond usual MAR, e.g. multvar. normal.
- Time for a new paradigm!
- We're interested in *prediction*.

Estimation vs. Prediction

- Almost all (all?) of the MV literature is on *estimation*, e.g. estimation of treatment effects.
- Almost all of those methods are based on *imputation*. Requires extra assumptions beyond usual MAR, e.g. multvar. normal.
- Time for a new paradigm!
- We're interested in *prediction*.
- We'll present a novel new technique we call the Tower Method.

Estimation vs. Prediction

- Almost all (all?) of the MV literature is on *estimation*, e.g. estimation of treatment effects.
- Almost all of those methods are based on *imputation*. Requires extra assumptions beyond usual MAR, e.g. multvar. normal.
- Time for a new paradigm!
- We're interested in *prediction*.
- We'll present a novel new technique we call the Tower Method.
- Non-imputational.

Estimation vs. Prediction

- Almost all (all?) of the MV literature is on *estimation*, e.g. estimation of treatment effects.
- Almost all of those methods are based on *imputation*. Requires extra assumptions beyond usual MAR, e.g. multvar. normal.
- Time for a new paradigm!
- We're interested in *prediction*.
- We'll present a novel new technique we call the Tower Method.
- Non-imputational.
- Available at `http://github.com/matloff/toweranNA`.

Norm Matloff University of California at Davis

Theorem from Probability Theory

[Please be patient; R code and real-data examples soon. :-)]

Famous formula in probability theory:

$$EY = E[E(Y|X)] = E[g(X)]$$

Here $g()$ is regression function of Y on X.

Theoretical Background for Use in MVs

Theoretical Background for Use in MVs

- (Matloff, *Biometrika*, 1981)

Theoretical Background for Use in MVs

- (Matloff, *Biometrika*, 1981)
- My first published stat paper!

Theoretical Background for Use in MVs

- (Matloff, *Biometrika*, 1981)
- My first published stat paper!

Theoretical Background for Use in MVs

- (Matloff, *Biometrika*, 1981)
- My first published stat paper!

Theory Background (cont'd.)

Theory Background (cont'd.)

- My context: Est. E(Y).

$$\widehat{EY} = \frac{1}{n} \sum_{i=1}^{n} \widehat{g}(X_i)$$

Here $\widehat{g}$ comes from linear model, logit, nonpar.

Theory Background (cont'd.)

- My context: Est. E(Y).

$$\widehat{EY} = \frac{1}{n} \sum_{i=1}^{n} \widehat{g}(X_i)$$

  Here $\widehat{g}$ comes from linear model, logit, nonpar.
  Maybe some Y_i missing; even if not, get smaller asympt. var.

- Steady stream of theory papers since then from various authors.

Theory Background (cont'd.)

- My context: Est. E(Y).

$$\widehat{EY} = \frac{1}{n} \sum_{i=1}^{n} \widehat{g}(X_i)$$

  Here $\widehat{g}$ comes from linear model, logit, nonpar.
  Maybe some Y_i missing; even if not, get smaller asympt. var.

- Steady stream of theory papers since then from various authors.
- E.g. (U. Müller, *Annals of Stat.*, 2009).

Theory Background (cont'd.)

- My context: Est. E(Y).

$$\widehat{EY} = \frac{1}{n} \sum_{i=1}^{n} \widehat{g}(X_i)$$

  Here $\widehat{g}$ comes from linear model, logit, nonpar.
  Maybe some Y_i missing; even if not, get smaller asympt. var.

- Steady stream of theory papers since then from various authors.
- E.g. (U. Müller, *Annals of Stat.*, 2009).
- But all theoretical. Not used (or even known) by MV practitioners.

Tower Property

Tower Property

More general version, known as the Tower Property:

$$E[E(Y|U,V)|U] = E(Y|U)$$

Tower Property

More general version, known as the Tower Property:

$$E[E(Y|U, V)|U] = E(Y|U)$$

Why is this relevant to us?

- Y: variable to be predicted
- U: vector of known predictor values
- V: vector of uknown predictor values

Example: Census Data

- Programmer/engineer data, Silicon Valley, 2000 (**prgeng** in pkg).

Example: Census Data

- Programmer/engineer data, Silicon Valley, 2000 (**prgeng** in pkg).
- Predict Y = wage income. In one particular case to be predicted, we might have

Example: Census Data

- Programmer/engineer data, Silicon Valley, 2000 (**prgeng** in pkg).
- Predict Y = wage income. In one particular case to be predicted, we might have
  - U = (education,occupation,weeks worked)
  - V = (age,gender)

  In another case, maybe U = (age,gender,education,weeks worked) and V = (occupation). Etc.

Example: Census Data

- Programmer/engineer data, Silicon Valley, 2000 (**prgeng** in pkg).
- Predict Y = wage income. In one particular case to be predicted, we might have
  - U = (education,occupation,weeks worked)
  - V = (age,gender)

  In another case, maybe U = (age,gender,education,weeks worked) and V = (occupation). Etc.
- Wish we had U,V, for prediction E(Y|U,V), but forced to use E(Y|U).
- But then must estimate many E(Y | U), since many different patterns for MVs (2^5 here).

Example: Census Data

- Programmer/engineer data, Silicon Valley, 2000 (**prgeng** in pkg).
- Predict Y = wage income. In one particular case to be predicted, we might have
  - U = (education,occupation,weeks worked)
  - V = (age,gender)

  In another case, maybe U = (age,gender,education,weeks worked) and V = (occupation). Etc.
- Wish we had U,V, for prediction E(Y|U,V), but forced to use E(Y|U).
- But then must estimate many E(Y | U), since many different patterns for MVs (2^5 here).
- Hard enough to fit one good model, let alone dozens or more.

Example: Census Data

- Programmer/engineer data, Silicon Valley, 2000 (**prgeng** in pkg).
- Predict Y = wage income. In one particular case to be predicted, we might have
    - U = (education,occupation,weeks worked)
    - V = (age,gender)

  In another case, maybe U = (age,gender,education,weeks worked) and V = (occupation). Etc.
- Wish we had U,V, for prediction E(Y|U,V), but forced to use E(Y|U).
- But then must estimate many E(Y | U), since many different patterns for MVs (2^5 here).
- Hard enough to fit one good model, let alone dozens or more.
- With Tower, need only one.

Tower (cont'd.)

Basic idea:

- Fit full regression model to the complete cases.
- Use Tower to get the marginal models from the full one:

$$\widehat{E}(Y \mid U = s) = \text{avg. } \underbrace{\widehat{E}(Y \mid U = s, V)}_{\text{full model}}$$

  over all complete cases with $U = s$

- In practice, use $U \approx s$ instead of $U = s$, using k nearest neighbors.

Tower (cont'd.)

Basic idea:

- Fit full regression model to the complete cases.
- Use Tower to get the marginal models from the full one:

$$\widehat{E}(Y \mid U = s) = \text{avg. } \underbrace{\widehat{E}(Y \mid U = s, V)}_{\text{full model}}$$

  over all complete cases with $U = s$

- In practice, use $U \approx s$ instead of $U = s$, using k nearest neighbors.

  In practice, $k = 1$ usually fine;

Tower (cont'd.)

Basic idea:

- Fit full regression model to the complete cases.
- Use Tower to get the marginal models from the full one:

$$\widehat{E}(Y \mid U = s) = \text{avg.} \ \underbrace{\widehat{E}(Y \mid U = s, V)}_{\text{full model}}$$

  over all complete cases with $U = s$
- In practice, use $U \approx s$ instead of $U = s$, using k nearest neighbors.

  In practice, $k = 1$ usually fine; fitted values already smoothed, don't need more smoothing.

Census Example (cont'd.)

(a) Use, say, **lm()** on the complete cases, predicting wage income from (age,gender,education,occupation,weeks worked).

(b) Save the fitted values, e.g. **fitted.values** from **lm()** output.

(c) Say need to predict case with education = MS, occupation = 102, weeks worked = 52 but with age and gender missing.

(d) Find the complete cases for which (education,occupation,weeks worked) = (MS,102,52).

(e) Predicted value for this case is average of the fitted values for the cases in (d).

toweranNA Package API

- **toweranNA(x,fittedReg,k,newx,scaleX=TRUE)**
  - **x:** Data frame of complete cases.
  - **fittedReg:** Estimated values of full regress. ftn. at those cases (from **lm()**, **glm()**, random forests, neural nets, whatever).
  - **k:** Number of nearest neighbors.
  - **newx:** Data frame of new cases to be predicted.
  - Return value: Vector of predictions.

Norm Matloff
University of California at Davis

Other Major Functions

Other Major Functions

- **towerLM(x,y,k,newx,useGLM=FALSE)**
  Wrapper for **toweranNA()**.
- **towerTS(x,lag,k)**
  Adaptation of Tower Method for time series; see below.

Structure of Examples

- 3 real datasets.
- Break into random training and test sets.
- Predict all test-set cases with at least one MV.

Example: WordBank Data

- Kids' vocabulary growth trajectories.
- About 5500 cases, 6 variables. About 29% MVs.

Mean Absolute Prediction Errors:

Amelia	Tower
102.7	96.2
122.9	119.9
89.4	88.1
115.3	107.0
111.1	102.5

- Times about 6s each.
- The **mice** package crashed.

UCI Bank Data

- About 50K cases.
- Only about 2% MVs. Not much need for MV methods, but let's make sure Tower doesn't bring harm. :-)
- Tower run 8.3s, **mice** 442.2s.
- Too long to do multiple runs. About the same accuracy, 0.92 or 0.93.
- **Amelia** crashed.

World Values Study

- World political survey.
- 48 countries, sample 500-3500 from each.
- MVs artifically added.
- Tower outperformed **mice** in 39 of 48 countries.

	Tower	*Mice*
Mean Absolute Predictive Error	1.7603	1.8270
Elapsed Time (seconds)	0.1825	14.0822

Norm Matloff
University of California at Davis

Concerning Assumptions

- Most MV methods assume MAR, Missing at Random.

Concerning Assumptions

- Most MV methods assume MAR, Missing at Random.
- Precise def. of MAR tricky (Seaman, *Stat. Sci.*, 2013).

Concerning Assumptions

- Most MV methods assume MAR, Missing at Random.
- Precise def. of MAR tricky (Seaman, *Stat. Sci.*, 2013).
- Tower assumptions similar, but assumptions matter much less in prediction than in estimation.

Norm Matloff University of California at Davis

Concerning Assumptions

- Most MV methods assume MAR, Missing at Random.
- Precise def. of MAR tricky (Seaman, *Stat. Sci.*, 2013).
- Tower assumptions similar, but assumptions matter much less in prediction than in estimation.
- **Amelia**, **mice** assume X multvar. normal, very distorting.

Norm Matloff
University of California at Davis

Time Series (cont'd.)

Time Series (cont'd.)

- A work in progress.

Norm Matloff
University of California at Davis

Time Series (cont'd.)

- A work in progress.
- Example: NH4 data in imputeTS package.

Norm Matloff University of California at Davis

Time Series (cont'd.)

- A work in progress.
- Example: NH4 data in imputeTS package.
- Mean Absolute Prediction Error:
  na.ma (based on moving avg.): 1.51
  towerTS: 1.37

Future Work

Norm Matloff
University of California at Davis

Future Work

- Most pressing issue: May have too few (or no) complete cases.

Future Work

- Most pressing issue: May have too few (or no) complete cases.
- Solution: Relax our “one size fits all” structure.

Future Work

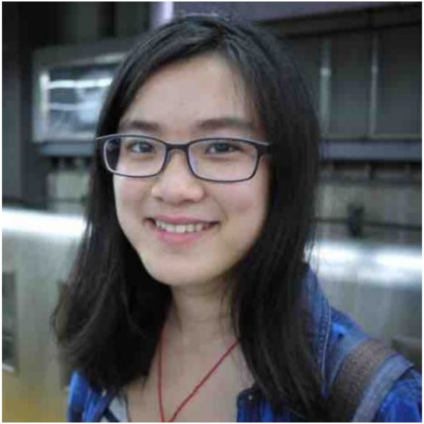
- Most pressing issue: May have too few (or no) complete cases.
- Solution: Relax our "one size fits all" structure.
- Instead of generating all marginal regression functions from one full one, have several "almost-full" ones.

Future Work

- Most pressing issue: May have too few (or no) complete cases.
- Solution: Relax our "one size fits all" structure.
- Instead of generating all marginal regression functions from one full one, have several "almost-full" ones.
- E.g. have $p = 5$ predictors. Maybe fit four 4-predictor models. Each would be based on more complete cases than the 5-predictor models.

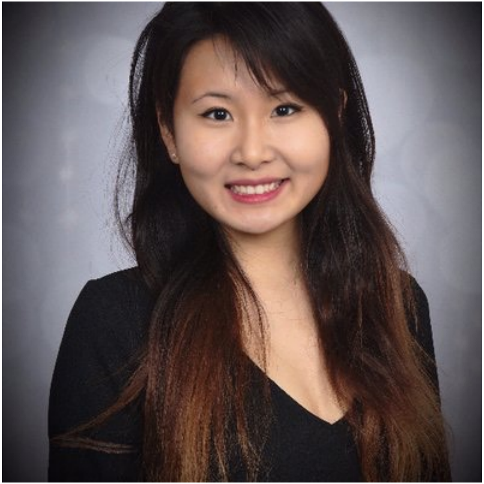
The Team!

The Team!



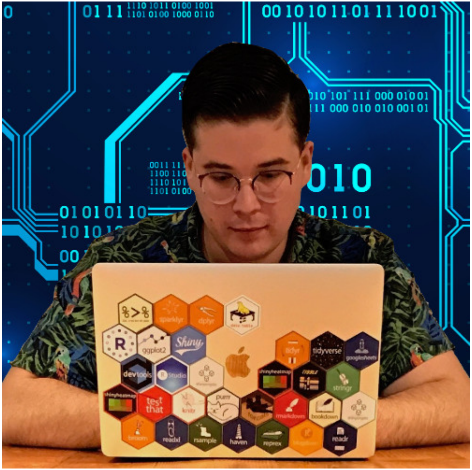
Xi Cheng

The Team! (contd.)



Tiffany Jiang

The Team! (contd.)



Bohdan Khomtchouk

The Team! (contd.)

Matt Kotila

The Team! (contd.)

Pete Mohanty

The Team! (contd.)

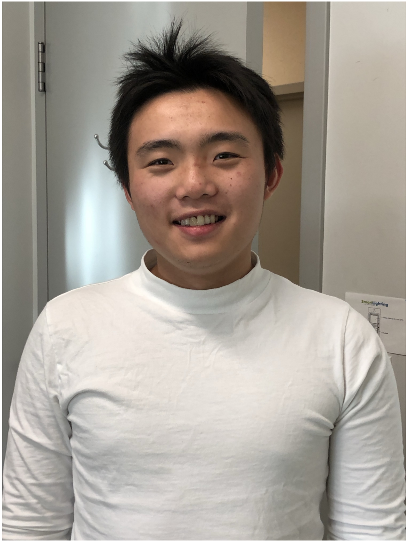
Robert Tucker

The Team! (contd.)

Robin Yancey

Norm Matloff University of California at Davis

The Team! (contd.)



Allan Zhao