Machine learning won't save us: Dependencies bias cross-validation estimates of model performance

Momin M. Malik, PhD <momin_malik@cyber.harvard.edu>
Data Science Postdoctoral Fellow
Berkman Klein Center for Internet & Society at Harvard University

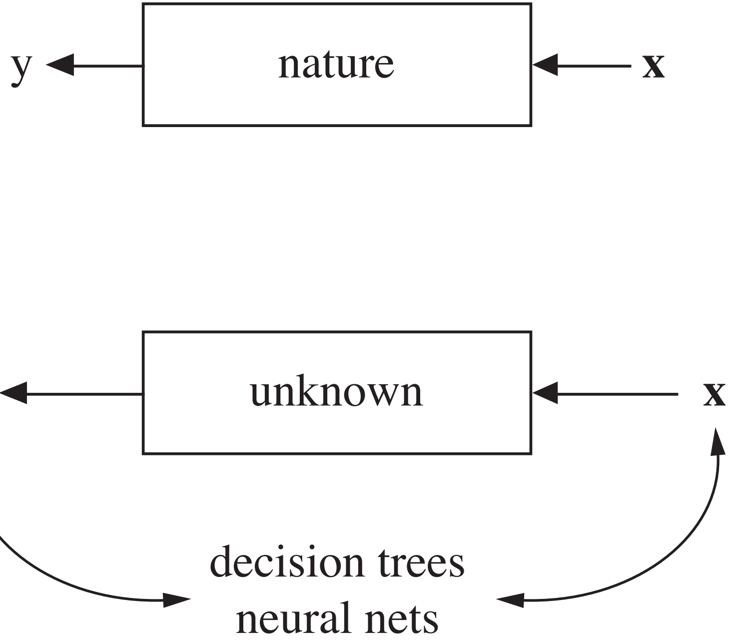
Virtual Sunbelt, 17 July 2020
Slides: https://mominmalik.com/sunbelt2020.pdf

Main take-aways

- Machine learning has ignored the problems of dependencies, but they come out in new ways!
- *Cross-validation estimates of model performance are downwardly biased* (overly "optimistic")
- Caution:
  - Analytic results, not real-world demonstration (yet)
  - General results, simulation not done specifically for a network
- Still, it's clear: machine learning can't save us!

Statistics vs. machine learning

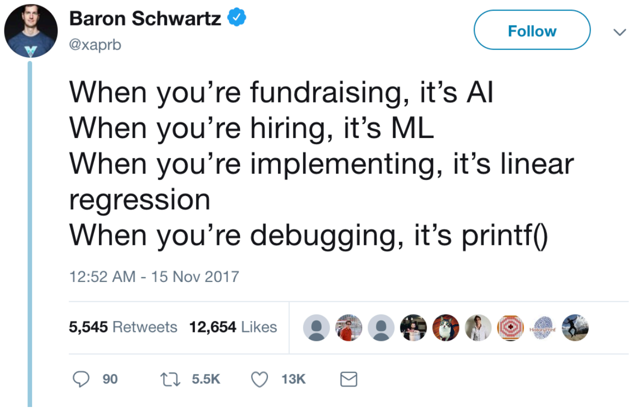
Same tools, different goals

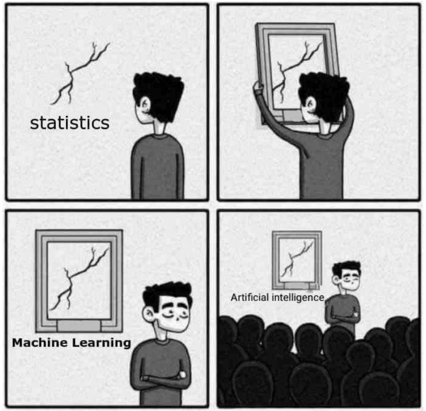


Breiman, 2001. See also Jones, 2018.

- Goal of statistics: model underlying process and relationships
- Goal of ML: automatically, reliably replicate input/output relationships

Benefits of machine learning vs. stats





xkcd.com/1838

- (Lots of hype, I'll spare you the rhetoric…)
- Automatically finding the strongest correlation often gets better model fit than using domain knowledge
- "Flexible," automatic fits (including nonparametrics) involve fewer assumptions: so there is less to go wrong

The statistical problem of dependencies

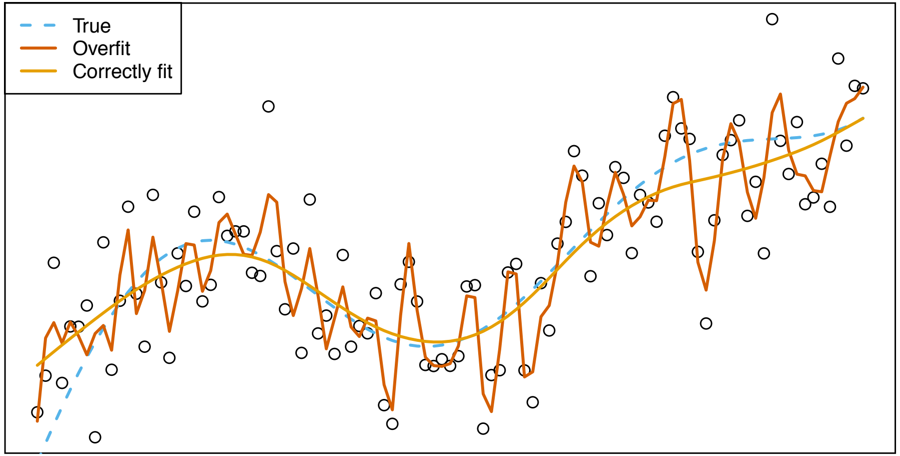
- Statistics: all about central tendencies, which need multiple observations
- Need to make independence assumptions for a network to not be $n = 1$
- Dependencies: "merge" observations
- E.g.: duplicated data. No bias, but decreases effective sample size ("deflates" standard errors), can lead to wrong inferences
- More complex dependencies (e.g., transitivity, reciprocity) lead to [omitted variable] bias *and* wrong inferences

Can machine learning help?

- Can't have deflated standard errors if you don't estimate standard errors
- Doesn't matter if you have omitted variable bias if you don't care about bias
- Fewer assumptions means fewer places for things to go wrong
- Correlation-only: good for *high-dimensional data* (networks: can think of as a subspace of $\mathbb{R}^{\binom{n}{2}}$)

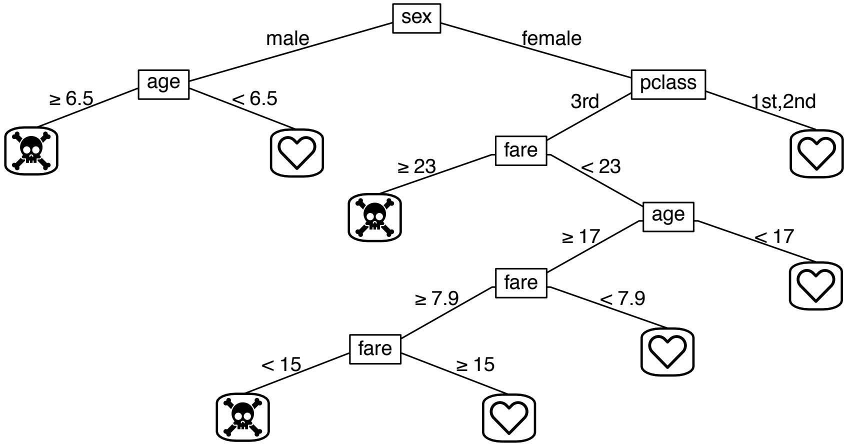
Overfitting and cross validation

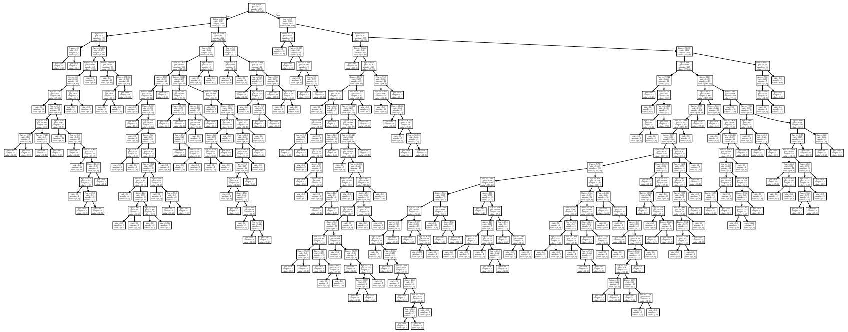
Central problem: overfitting (fit to noise)



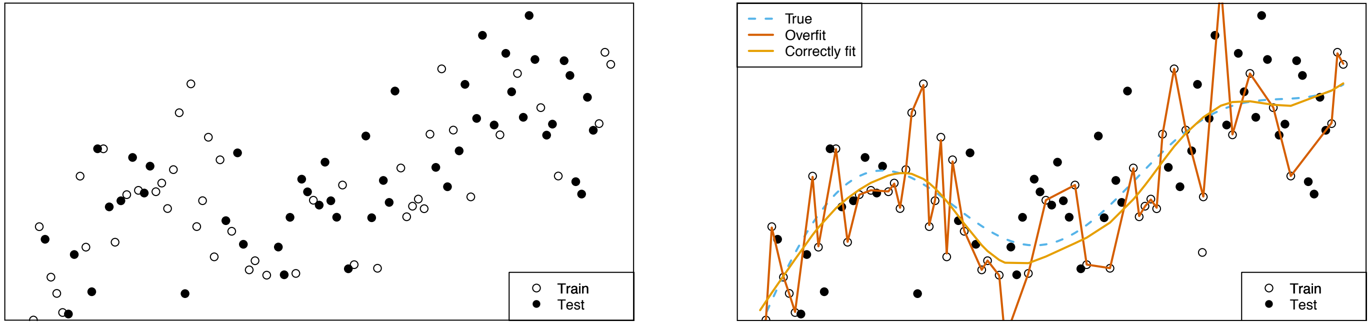
- If we are no longer guided by theory, and use automatic methods, we risk *overfitting*: fitting to the the noise, not the signal (“memorize the data”)
- Even if we don’t care about recovering the “true” function, overfitted models also *generalize* poorly

(Overfitting, discrete case: *Titanic* deaths)





Data splitting: Catch overfitting



- Idea: if we split data into two parts, the signal should be the same but the noise would be different
- *Cross validation:* Fit on one part of data, then choose smoother bandwidth, tree depth, etc., by what minimizes loss on *held-out* data
- Also used for model *testing*

Classic argument for CV for testing

$$
\begin{aligned}
\mathrm{Err}(\hat{\mu}) &= \frac{1}{n}\mathbb{E}_f \| Y^* - \widehat{Y} \|_2^2 \\
&= \frac{1}{n}\left[\mathbb{E}_f\|Y^*\|_2^2 + \mathbb{E}_f\|\widehat{Y}\|_2^2 - 2\mathbb{E}_f({Y^*}^T\widehat{Y})\right] \\
&= \frac{1}{n}\left[\mathbb{E}_f\|Y^*\|_2^2 + \mathbb{E}_f\|\widehat{Y}\|_2^2 - 2\,\mathrm{tr}\,\mathbb{E}_f(Y^*\widehat{Y}^T)\right] \\
&\quad + \frac{1}{n}\left[\ \mu^T\mu + \mathbb{E}_f(\widehat{Y})^T\mathbb{E}_f(\widehat{Y}) + 2\,\mathrm{tr}\,\mu\mathbb{E}_f(\widehat{Y})^T\right] \\
&\quad + \frac{1}{n}\left[-\mu^T\mu - \mathbb{E}_f(\widehat{Y})\mathbb{E}_f(\widehat{Y})^T - 2\mu^T\mathbb{E}_f(\widehat{Y})\right] \\
&= \frac{1}{n}\left[\mathrm{tr}\,\Sigma + \|\mu - \mathbb{E}(\widehat{Y})\|_2^2 + \mathrm{tr}\,\mathrm{Var}_f(\widehat{Y}) - 2\,\mathrm{tr}\,\mathrm{Cov}_f(Y^*, \widehat{Y})\right] \\
&= \ \text{irreducible error} + \text{bias}^2 + \text{variance} - \text{optimism}
\end{aligned}
$$

Dependencies affect machine learning, too

Some (other) problems with ML

- Automatic methods can easily pick up on non-causal correlations—sometimes okay, but can go wrong (e.g., Google Flu Trends)
- More profoundly, because of the bias-variance tradeoff, *a "true" model can predict worse than a "false" model!* (Shmueli, 2010)
  - (Relates to "Stein's paradox," see Efron & Morris 1977)
- Consequence: what "predicts" well (correlation, ML) doesn't necessarily "explain" well (causation, stats)
- Still: at least with prediction, we know we succeeded… right?

Test error on non-iid data has optimism!

- Imagine we have, for $\boldsymbol{\Sigma}_{ii} = \sigma^2$ and $\boldsymbol{\Sigma}_{ij} = \rho\sigma^2, \quad i \neq j$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{X} \\ \mathbf{X} \end{bmatrix} \boldsymbol{\beta}, \begin{bmatrix} \boldsymbol{\Sigma} & \rho\sigma^2 \mathbf{1}\mathbf{1}^T \\ \rho\sigma^2 \mathbf{1}\mathbf{1}^T & \boldsymbol{\Sigma} \end{bmatrix} \right)$$

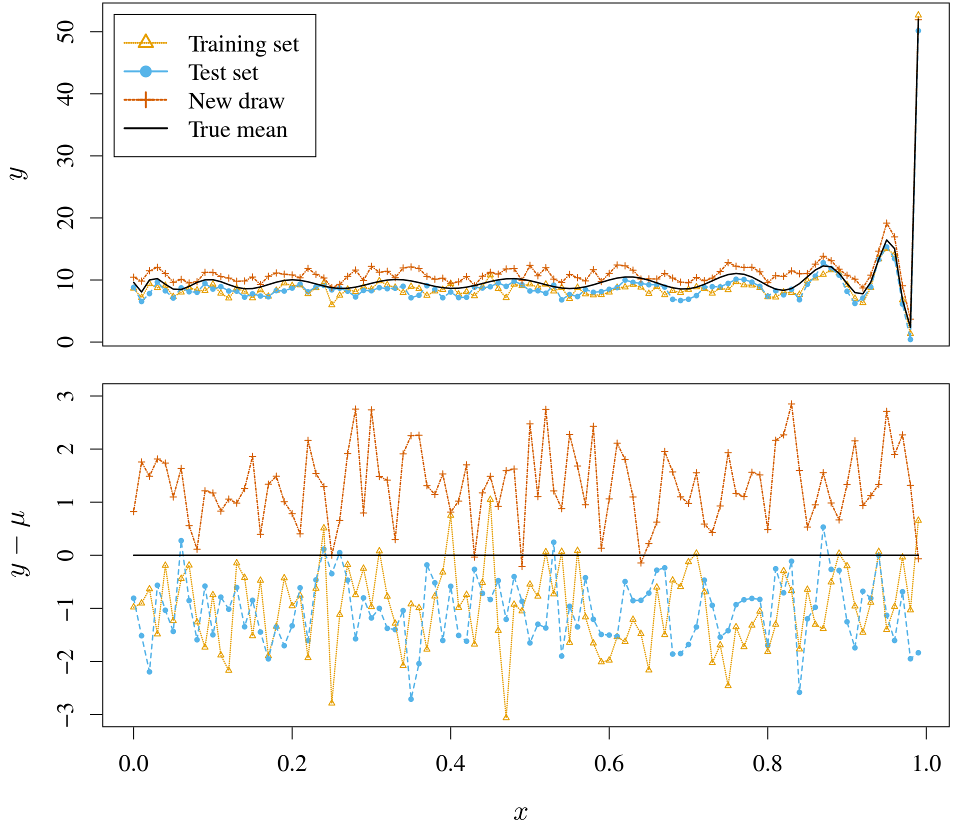
- Then, optimism in the training set is:

$$\frac{2}{n} \operatorname{tr} \operatorname{Cov}_f(Y_1, \widehat{Y}_1) = \frac{2}{n} \operatorname{tr} \operatorname{Cov}_f(Y_1, \mathbf{H} Y_1) = \frac{2}{n} \operatorname{tr} \mathbf{H} \operatorname{Var}_f(Y_1) = \frac{2}{n} \operatorname{tr} \mathbf{H}\boldsymbol{\Sigma}$$

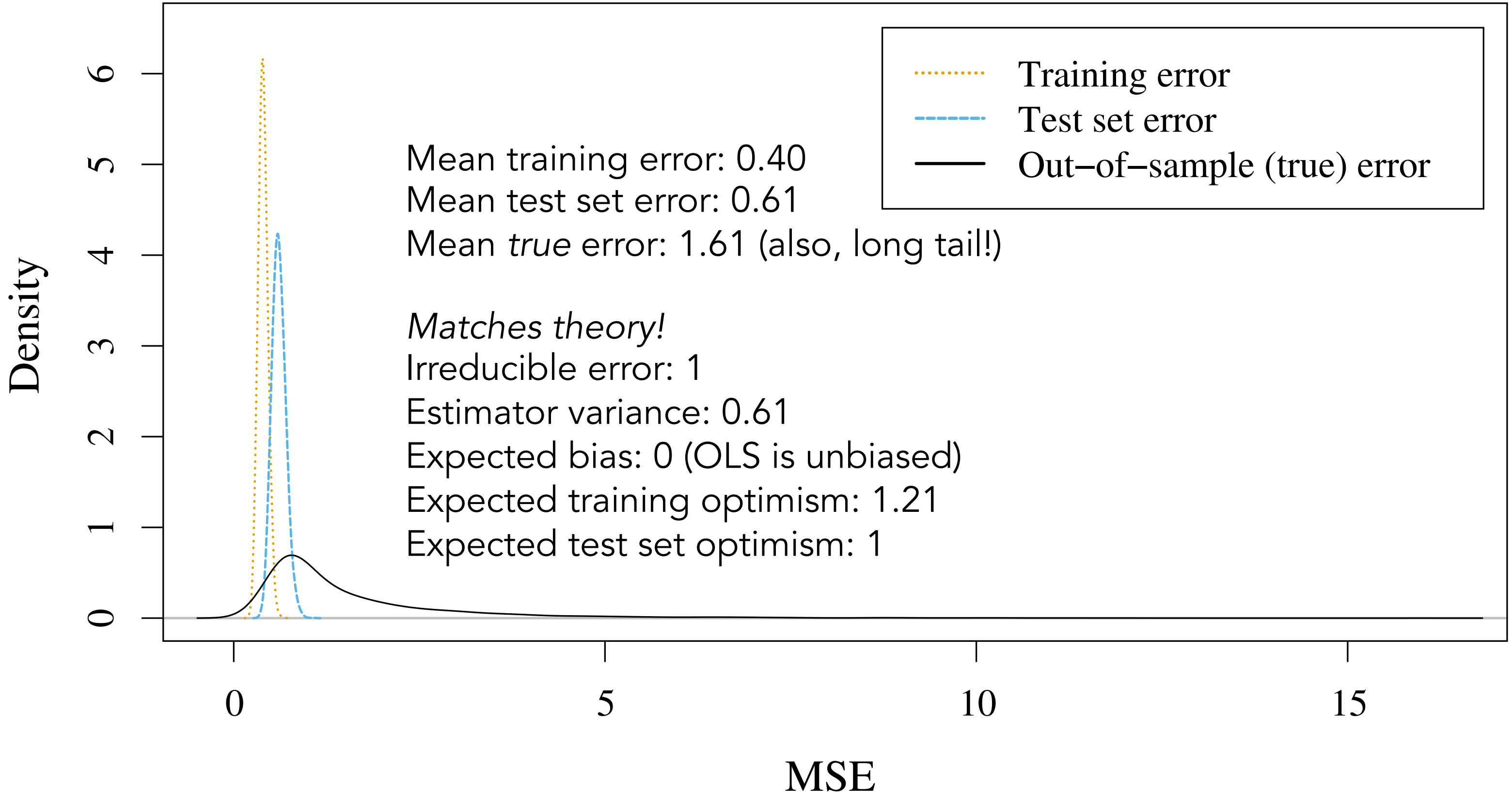
- But test set also has nonzero optimism!

$$\frac{2}{n} \operatorname{tr} \operatorname{Cov}_f(Y_2, \widehat{Y}_1) = \frac{2}{n} \operatorname{tr} \operatorname{Cov}_f(Y_2, \mathbf{H} Y_1) = \frac{2\rho\sigma^2}{n} \operatorname{tr} \mathbf{H}\mathbf{1}\mathbf{1}^T = 2\rho\sigma^2$$

Simulating the toy example



Out-of-sample MSE: *much* worse!



Implications for networks

Applying to networks

- This formulation would apply to a network autocorrelation model, where network is nuisance parameter
- But what if we are modeling the *edges*, which represent dependencies between observations?

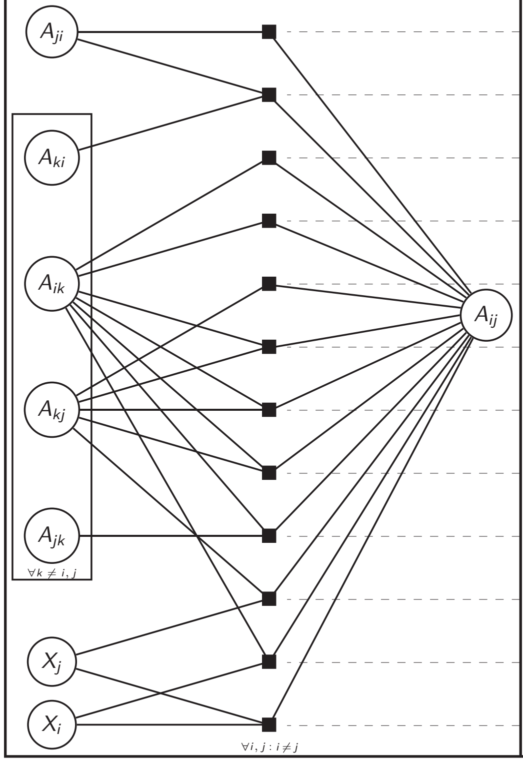
Virtual Sunbelt 2020

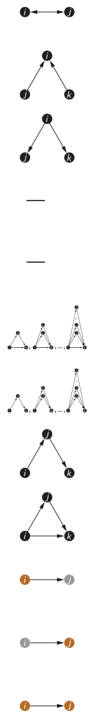
Modeling the *edges*

	Y	X_1	X_2	$\cdots$	X_d
1	y_1	x_{11}	x_{12}	$\cdots$	x_{1d}
2	y_2	x_{21}	x_{22}	$\cdots$	x_{2d}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
n	y_n	x_{n1}	x_{n2}	$\cdots$	x_{nd}

index	*from*	*to*	Y	W_1	W_2	W_3	$\cdots$
e_1	1	2	y_{12}	$\mathbf{1}(x_{11} = x_{21})$	$x_{12} - x_{22}$	x_{13}	$\cdots$
e_2	2	3	y_{23}	$\mathbf{1}(x_{11} = x_{31})$	$x_{12} - x_{32}$	x_{13}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
e_{n+1}	2	1	y_{21}	$\mathbf{1}(x_{21} = x_{11})$	$x_{22} - x_{12}$	x_{23}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
$e_{2\binom{n}{2}}$	$n-1$	n	$y_{(n-1)n}$	$\mathbf{1}(x_{(n-1)1} = x_{n1})$	$x_{(n-1)2} - x_{n2}$	$x_{(n-1)3}$	$\cdots$

But dyads are dependent too!

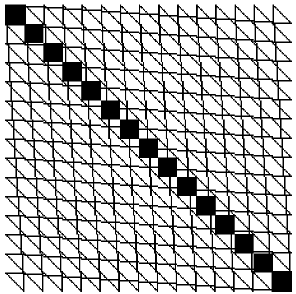




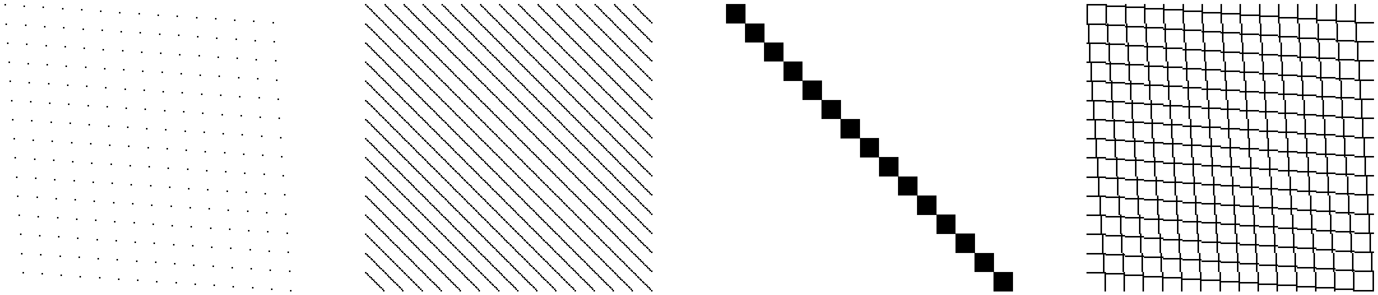
Parameter name	Network Motif	Parameterization	Matrix notation
mutual dyads		$\sum_{i<j} A_{ij}A_{ji}$	$\frac{1}{2}\,\mathrm{tr}\left(\mathbf{AA}^T\right)$
in-two-stars		$\sum_{(i,j,k)} A_{ji}A_{ki}$	$\mathrm{sum}\left(\mathbf{AA}^T\right) - \mathrm{tr}\left(\mathbf{AA}^T\right)$
out-two-stars		$\sum_{(i,j,k)} A_{ij}A_{ik}$	$\mathrm{sum}\left(\mathbf{A}^T\mathbf{A}\right) - \mathrm{tr}\left(\mathbf{A}^T\mathbf{A}\right)$
geom. weighted out-degrees	—	$\sum_i \exp\left\{-\alpha \sum_k A_{ik}\right\}$	$\mathrm{sum}\left(\exp\{-\alpha\,\mathrm{rowsum}\left(\mathbf{A}\right)\}\right)$
geom. weighted in-degrees	—	$\sum_j \exp\left\{-\alpha \sum_k A_{kj}\right\}$	$\mathrm{sum}\left(\exp\{-\alpha\,\mathrm{colsum}\left(\mathbf{A}\right)\}\right)$
alternating transitive k-triplets		$\lambda \sum_{i,j} A_{ij}\left\{1-\left(1-\frac{1}{\lambda}\right)^{\sum_{k\neq i,j} A_{ik}A_{kj}}\right\}$	$\lambda\,\mathrm{sum}\left(\mathbf{A}\odot\left(1-\left(1-\frac{1}{\lambda}\right)^{\mathbf{AA}-\mathrm{diag}(\mathbf{AA})}\right)\right)$
alternating indep. two-paths		$\lambda \sum_{i,j}\left\{1-\left(1-\frac{1}{\lambda}\right)^{\sum_{k\neq i,j} A_{ik}A_{kj}}\right\}$	$\lambda\,\mathrm{sum}\left(1-\left(1-\frac{1}{\lambda}\right)^{\mathbf{AA}-\mathrm{diag}(\mathbf{AA})}\right)$
two-paths (mixed two-stars)		$\sum_{(i,k,j)} A_{ik}A_{kj}$	$\mathrm{sum}\left(\mathbf{AA}\right) - \mathrm{tr}\left(\mathbf{AA}\right)$
transitive triads		$\sum_{(i,j,k)} A_{ij}A_{jk}A_{ik}$	$\mathrm{tr}\left(\mathbf{AAA}^T\right)$
activity effect		$\sum_i X_i \sum_j A_{ij}$	$\mathrm{sum}\left(\mathbf{X}\odot\mathrm{rowsum}\left(\mathbf{A}\right)\right)$
popularity effect		$\sum_j X_j \sum_i A_{ij}$	$\mathrm{sum}\left(\mathbf{X}\odot\mathrm{colsum}\left(\mathbf{A}\right)\right)$
similarity effect		$\sum_{i,j} A_{ij}\left(1-\frac{\lvert X_i - X_j\rvert}{\max_{k,l}\lvert X_k - X_l\rvert}\right)$	$\mathrm{sum}\left(\mathbf{A}\odot\mathbf{S}\right)$

Graphical model and matrix notations for ERGM specification terms given in: Snijders et al. 2006. Joint work with Antonis Manousis and Naji Shajarisales, 2018.

Covariance structure of *edges* (n = 15)

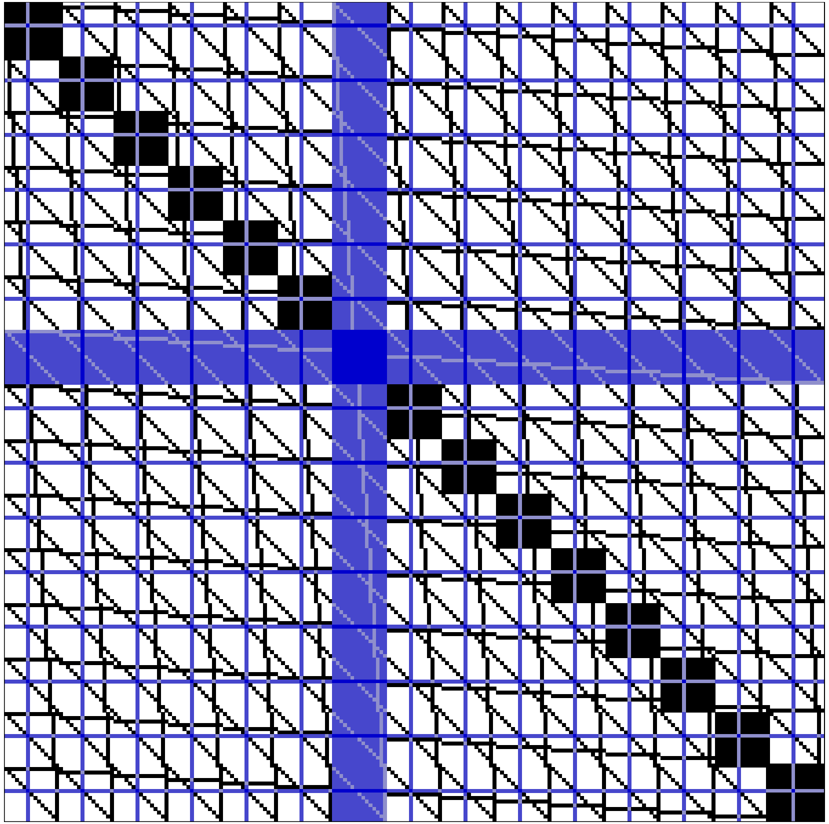


Total covariance



Mutual dyads In-2-stars Out-2-stars 2-paths

So, what to do?



- Partition nodes into training and test sets?
  - Breaks up triads; omitted edges “share” information across training and test
- Partition dyads?
  - Breaks up nodes; even worse
- Can’t *eliminate*, but can *minimize* optimism by careful data splitting

Conclusion

Never enough data

- Mean function and covariance structure jointly not identifiable (Opsomer et al., 2011)
- Means: additional data that you gather also has covariance with previous data; so without independence assumptions (or assuming the mean), can't ever estimate covariance
- Hopefully, the covariance doesn't affect cross-validation…

"But what about...?"

- Representation learning? Deep learning? Neural nets?
  - Massive successes have been on very specific, *ordered* data types (images, text, audio). Graphs not ordered
  - node2vec is based on (undirected) *random walks*; only appropriate for some tasks (Khosla et al., 2019)
- Statistical relational learning?
  - The leading textbook on this never discusses how to properly do cross-validation! Same problems

Thank you! Summary:

- With ML, we have to deal with the exact same problems of dependencies, just manifesting in different ways
- Cross-validation estimates of model performance for networks will (almost) surely be overly optimistic
- How *much* optimism depends on how strong dependencies are across training and test splits
- Can try to minimize optimism with principled cross-validation schema
- See https://arxiv.org/abs/2002.05193 for more