System 2 Applied Mathematics: Machine Learning for Power Market Clearing

Misha Chertkov Applied Math @ UArizona

May 23, 2022 – DANniversary, MIP2022, Rutgers

MIP 2022 MIXED INTEGER PROGRAMMING WORKSHOP feat. DANniversary

- "Uncertainty"2010 (theory+experiment)

SIAM REVIEW
Vol. 56, No. 3, pp. 461–495

© 2014 Society for Industrial and Applied Mathematics

Chance-Constrained Optimal Power Flow: Risk-Aware Network Control under Uncertainty*

Daniel Bienstock†
Michael Chertkov‡
Sean Harnett§

- "System 1 Learning"/2018 (experiment)

Learning from power system data stream: phasor-detective approach

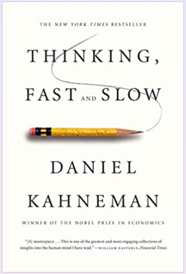
Mauro Escobar, Daniel Bienstock* & Michael Chertkov†
*Columbia University, NY, USA
†Los Alamos National Laboratory, NM, USA & Skoltech, Moscow, Russia

- "System 2 Learning"/2022 (experiment+ opportunity for theory)

ACCEPTED FOR PRESENTATION IN 11TH BULK POWER SYSTEMS DYNAMICS AND CONTROL SYMPOSIUM (IREP 2022), JULY 25-30, 2022, BANFF, CANADA

Machine Learning for Electricity Market Clearing

Laurent Pagnier *, Robert Ferrando *, Yury Dvorkin †, Daniel Bienstock ‡ and Michael Chertkov *
* Program in Applied Mathematics, University of Arizona, Tucson, AZ, USA
{laurentpagnier,rferrando,chertkov}@math.arizona.edu
† Electrical and Computer Engineering, New York University, New York, NY, USA
dvorkin@nyu.edu
† Industrial Engineering and Operations Research, Columbia University, New York, NY
dano@columbia.edu



System 1 & 2 in DL and AI

- "From System 1 Deep Learning to System 2 Deep Learning" – Yoshua Bengio, NeurIPS 2019
- "Combining Fast and Slow Thinking for Human-like and Efficient Navigation in Constrained Environments" – M. Ganappini, et al, arXiv:2201.07050

Modern ('21) Applied Mathematics as System 2 ... Harvesting 20's Applied Math + (System 1.8) Data & Model Revolution (System 1)

- System 1 – operates automatically & quickly [Deep Learning, empowered by Automatic Differentiation]
- System 2 – allocates attention to effortfull mental activities [Building **Explainable Heuristics in Quantitative Sciences**]

Applied Math '21 = Harvesting Data and Model Revolution

- Applied Math '21 = Traditional AM + Contemporary AM
- Traditional Applied Math
  - Natural Science Based (motivated by Physics, later Biology, Enviromental Sciences, etc ...)
  - Originally largely ODE, PDE, Dynamical Systems, Chaos, Turbulence, ...
- Contemporary AM
  - More applications, e.g. Engineering, Social sciences, Networks
  - AI disciplines: Statistics, Data Science, Computer Science, Machine Learning, Optimization, Control
  - Deep Learning - most prominent recent addition (automatic differentiation, very efficient large scale optimization) ... based a lot on "old" stuff (stochastic gradient descent, sensitivity analysis)

Physics Informed Machine Learning: Principal Ideas (active discussions)

- A-Priori: run a (physics-blind) ML scheme, check physics
  - Diagnostics: hierarchy of tests
- A-Posteriori: embed physics in ML
  - Loss Function
  - Graphical Model (structure=explainable) Learning
  - What we know (structure) vs what we do NOT know (NN)
- Model Reduction
  - Check Hypotheses, Phenomenologies (e.g. forgotten)
- Derive New/Old Physical Laws

Outline

Physics Informed Machine Learning for Power Systems

Machine Learning (e.g. Neural Network, Graph Models, etc)

- will make Power System Computations
  - faster (efficient)
  - possible even when data/measurements incomplete
- requires ground-truth data
  - actual measurements (Phasor Measurement Units, etc)
  - power flow solvers (microscopic simulations) – reliable, possibly heavy
- can be power-system "informed" (System 2) vs "agnostic" (System 1)
  - What is System 1 today may become System 2 tomorrow (with proper theory & enough of experiments)
- methods/options are many
  - should be gauged to available data, level of uncertainty, etc

Incomplete Review: Brief, Recent, Biased

AI/ML in Power Systems (System 1, System 2 & juxtaposition)

- Structure Learning, Sparse Measurements, Graphical Models, Focus on Power Distribution: Deka, et al [2016-2019]
- Learning ODE: Power Transmission, Dynamic Coefficients in Swing Equations, Deterministic and Stochastic, Lokhov, et al [2017]
- Real-time Faulted Line Localization and PMU Placement in Power Transmission through CNN: Li, et al [2018]
- Collocation Point Neural ODE for Power Systems: Misuris, et al [2018]
- Learning a Generator Model from Terminal Bus Data: many ML schemes, tradeoffs, ranking models according to regimes, Stulov et al [2019]
- Learning from power system data stream, phasor-detective approach, Escobar et al [2019]

Incomplete Review: Brief, Recent, Biased

AI/ML in Power Systems (System 1, System 2 & juxtaposition)

- **Physics-Informed** Graphical Neural Network for **Parameter & State Estimations** in Power Systems `https://arxiv.org/abs/2102.06349` (Pagnier & MC))
- **Embedding Power Flow** into Machine Learning for Parameter and State Estimation `https://arxiv.org/abs/2103.14251` (Pagnier & MC)
- **Which Neural Network to Choose** for Post-Fault Localization, Dynamic State Estimation and Optimal Measurement Placement in Power Systems? `https://arxiv.org/abs/2104.03115` (Afonin & MC))

Machine Learning (Neural Networks) Setting

NN models: General

- $\text{NN}_{\vec{\phi}}(\vec{x}) = \vec{y}$
  - Vector, $\vec{\phi}$, of Not-Interpretable Parameters
  - Input vector: $\vec{x}$
  - Output vector: $\vec{y}$

NN models: Loss Functions

- L2 norm $\|\cdots\|$
- Probabilistic (Cross Entropy or Kullback-Leibler)
- Regularizations, e.g. L1 (sparsity, physical, etc)

NN models: Architectures

- Convolutional NN (LeCun 1989 –)
- Graph NN (Scarcelli. et al 2009 –)
- Neural ODE (Chen et al 2008 –)
- Collocation Point NN (Lagaris et al 1998, Raissi et al 2019 –)
- Hamiltonian NN (Greydanus et al 2018 –)

Power Flow Equations

- grid-graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- complex-valued powers: $\forall a \in \mathcal{V}: \quad S_a \equiv p_a + \boldsymbol{i} q_a$
- complex-valued (electric) potentials, $\forall a \in \mathcal{V}: \quad V_a \equiv v_a \exp(\boldsymbol{i}\theta_a)$,
- Power Flow (PF) equations:

$$p_a = \sum_{b;\{a,b\}\in\mathcal{E}} v_a v_b \Big[g_{ab} \cos\left(\theta_a - \theta_b\right) + \beta_{ab} \sin\left(\theta_a - \theta_b\right) \Big],$$

$$q_a = \sum_{b;\{a,b\}\in\mathcal{E}} v_a v_b \Big[g_{ab} \sin\left(\theta_a - \theta_b\right) - \beta_{ab} \cos\left(\theta_a - \theta_b\right) \Big],$$

- Direct PF Map: $\boldsymbol{\Pi}_{\boldsymbol{Y}} : \boldsymbol{S} \equiv (S_a | a \in \mathcal{V}) \mapsto \boldsymbol{V} \equiv (V_a | a \in \mathcal{V})$ - implicit (need to solve eqs. - System 1 & System 2 ML may be useful `https://arxiv.org/abs/2103.14251` L. Pagnier & MC)

Task: State & Parameter Estimation

- Inverse PF Map: $\boldsymbol{S} = \boldsymbol{\Pi}_{\boldsymbol{Y}}^{-1}(\boldsymbol{V})$ – explicit (do not need to solve eqs. – System 1 and System 2 ML may be useful `https://arxiv.org/abs/2102.06349` L. Pagnier and MC)

- State Estimation
  - Full Observability: given $\mathcal{G}$ and $\boldsymbol{Y}$ to estimate injected/consumed active and reactive powers = application of the inverse PF map, $\boldsymbol{\Pi}^{-1}$
  - Limited Observability:
    - Complement Missing power injections/consumptions at the nodes where voltages and phases are measured
    - Challenging Version: to reconstruct injected/consumed powers and also voltages and phases at all nodes of the system. (super-resolution – will not discuss)
- Parameter Estimation
  - Reconstruct Graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and line characteristics, $\boldsymbol{Y}$

Task SE & PE. Reduced Modeling.

- Setting of Partial Observability
- Find Equivalent (Reduced) Model of Power System
- "Inspired" by Kron Reduction
  - $\boldsymbol{I}^{(o)} = \boldsymbol{Y}^{(r)}\boldsymbol{V}^{(o)}$
  - "o" - observed; "r" - reduced
  - $\mathcal{G}^{(r)} \equiv (\mathcal{V}^{(o)}, \mathcal{E}^{(r)})$
  - $\boldsymbol{Y}^{(r)} \doteq (\{a,b\}|Y_{ab}^{(r)} \neq 0)$ – associated with the effective (not necessarily real) power lines, $\{a,b\} \in \mathcal{E}^{(r)}$. $\boldsymbol{Y}^{(r)}$
- Reduced Model
  - $\boldsymbol{S}^{(o)} = \Pi^{-1}_{\boldsymbol{Y}^{(r)}}(\boldsymbol{V}^{(o)})$
  - Learn it !?

Task: SE & PE. PIML of Reduced Model

- Power Graphical NN (System 2):

$$\min_{\varphi, \boldsymbol{Y}^{(r)}} L_{\text{Power-GNN}}\left(\varphi, \boldsymbol{Y}^{(r)}\right),$$

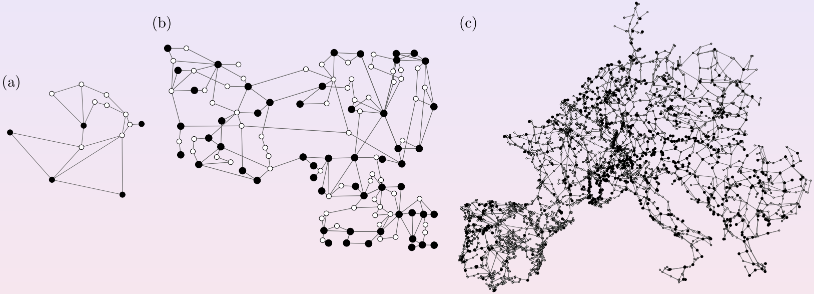
$$L_{\text{Power-GNN}}\left(\varphi, \boldsymbol{Y}^{(r)}\right) \equiv \frac{1}{N|\mathcal{V}^{(o)}|}\sum_{n=1}^{N}\Bigg\| \boldsymbol{S}_n^{(o)} - \underbrace{\Pi^{-1}_{\boldsymbol{Y}^{(r)}}\left(\boldsymbol{V}_n^{(o)}\right)}_{\text{physics = interpretable}} - \underbrace{\Sigma_{\varphi}\left(\mathcal{V}_n^{(o)}, S_n^{(o)}\right)}_{\text{NN = "sub-scale"}}\Bigg\|^2 + \underbrace{\mathcal{R}(\varphi)}_{\text{regularization}}$$

  - SIMULTANEOUSLY physics-informed and physics-blind parts
  - Compare with Vanilla-NN (System 1)

$$L_{\text{NN}} \doteq \frac{1}{N|\mathcal{V}^{(\mathbf{o})}|}\sum_{n=1}^{N}\left\|\boldsymbol{S}_n^{(o)} - \text{NN}_{\varphi}(\boldsymbol{V}_n^{(o)})\right\|^2$$

Task: SE & PE. Power GNN vs Vanilla NN. Experiments.

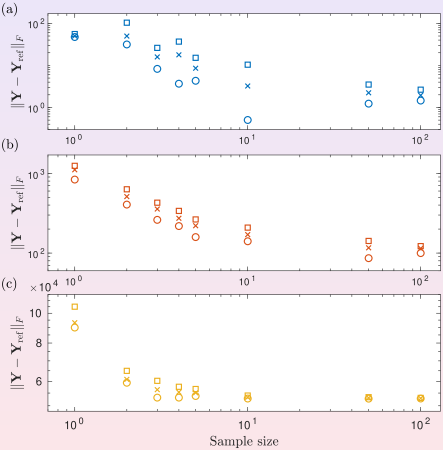
IEEE 14-bus [panel (a)], IEEE 118-bus [panel (b)] and PanTaGruEl [panel (c)] models



State Estimation Test: Six set of samples were generated for each network. Average mismatch of predicted power injections (on the training set in parentesis)

	case #1	case #2	case #3	case #4	case #5	case #6
Vanilla NN	4.9E-6	7.2E-5	6.3E-3	5.2E-2	6.3E-2	1.4E0
	(4.2E-6)	(6.6E-5)	(5.0E-5)	(3.7E-5)	(1.2E-4)	(4.2E-6)
Power-GNN	3.0E-6	5.8E-7	6.9E-7	1.3E-6	2.9E-7	3.0E-6

Task: SE & PE. Power GNN vs Vanilla NN. Experiments.



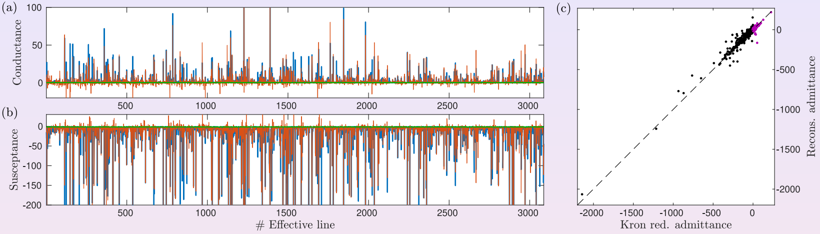
Full Observability. Parameter Estimation.

- Reconstruction of the admittance matrix $\boldsymbol{Y}$ for IEEE 14-bus (a), IEEE 118-bus (b) and PanTaGruEl (c) models
- The min, mean and max values are displayed as circles, crosses and squares respectively (for 10 realizations.)

Notice !!

- Quality of the reconstruction by Power-GNN – especially for large network

Task: SE & PE. Power GNN vs Vanilla NN. Experiments.



Partial Observability. Parameter Estimation. PanTaGruEl model

- Initial (pre-training) values – in green.
- Trained values and their Kron-reduction counterparts – red and blue respectively.
- (c) shows alternative visualization of the reference-vs-predicted values of the line conductances (purple) and susceptances (black)

Notice !!

- Quality of the reconstruction by Power-GNN – especially for large network

Outline

Learning LMP & Dispatch

$$\min_{p_g,\theta} \quad \sum_g (q_g p_g^2 + c_g p_g)$$

s.t.

$$\forall g: \quad p_g \in \text{Range}$$

Power Flow equations

$$\forall i: \quad p_i - l_i = \sum_{j \in i} b_{ij}(\theta_i - \theta_j)$$

$$\theta_{\text{slack bus}} = 0$$

Line Constraints

$$\forall \{i,j\}: \quad b_{ij}(\theta_i - \theta_j) \in \text{Range}$$

- Linear Programming (in DC-approximation)
- Locational Marginal Prices are part of the solution

Observation:

- Very few lines are saturated

The Challenge:

- To run the dispatch for MANY l-load configurations under given network conditions
- Do it faster than your good LP (plus) solvers can do

Learning LMP & Dispatch

$$\min_{p_g,\theta} \quad \sum_g (q_g p_g^2 + c_g p_g)$$

s.t.

$$\forall g: \quad p_g \in \text{Range}$$

Power Flow equations

$$\forall i: \quad p_i - l_i = \sum_{j \in i} b_{ij}(\theta_i - \theta_j)$$

$$\theta_{\text{slack bus}} = 0$$

Line Constraints

$$\forall \{i,j\}: \quad b_{ij}(\theta_i - \theta_j) \in \text{Range}$$

- Linear Programming (in DC-approximation)
- Locational Marginal Prices are part of the solution

Observation:

- Very few lines are saturated

The Challenge:

- To run the dispatch for MANY l-load configurations under given network conditions
- Do it faster than your good LP (plus) solvers can do
- Systems 2 [Physics, i.e. Power System, Informed] Machine Learning ?

Efficient Learning of LMPs & Dispatch

Handy to get rid of PF eqs. (and θ – phase angles)

$$\min_{p_g} \quad \sum_g (q_g p_g^2 + c_g p_g)$$

s.t.

$$\forall g: \quad p_g \in \text{Range}$$

$$\sum_g p_g = \sum_i l_i$$

<u>Line Constraints</u>

$$\forall k = \text{line}: \ \sum_{i \in k} \Phi_{ki}(p_i - l_i) \in \text{Range}$$

- Power Transfer Distribution Factor (PTDF) Matrix – Φ
- Generalizable to AC-OPF (linearization around an operational point)

Efficient Learning of LMPs & Dispatch

Lagrangian Formulation & Locational Marginal Prices (LMP)

- $\mathcal{L}(p_g, \lambda, \mu_k, \nu_k) = \sum_g (q_g p_g^2 + c_g p_g) + \lambda \left(\sum_g p_g - \sum_i l_i \right) +$
  $\sum_k \mu_k \left(\sum_i \Phi_{ki}(p_i - l_i) - f_k^{max} \right) -$
  $\sum_k \nu_k \left(\sum_i \Phi_{ki}(p_i - l_i) + f_k^{min} \right)$
- $\forall i: \;\; LMP_i = \frac{\partial \mathcal{L}}{\partial l_i} = -\lambda + \sum_k \Phi_{ki}(\nu_k - \mu_k)$

Efficient Learning of LMPs & Dispatch

Financial Coherency Conditions = Consequences of KKT:

- Revenue Adequacy (RA): $\sum_g LMP_g p_g \leq \sum_i LMP_i l_i$
- Cost Recovery (CR): $q_g p_g^2 + c_g p_g \leq LMP_g p_g$
- Engineer Desiderata: Reduced Model (faster, possibly approximate evaluation of OPF) guarantees RA & CR

Efficient Learning of LMPs & Dispatch

System 2 Idea:

- Take Advantage of Strong Duality (Complementary Slackness + Dual Feasibility)
  - $$\forall k: \ \mu_k \left(\sum_i \Phi_{ki}(p_i - l_i) - f_k^{max} \right) = -\sum_k \nu_k \left(\sum_i \Phi_{ki}(p_i - l_i) + f_k^{min} \right) = 0, \quad \mu_k, \nu_k \geq 0$$

Suppose we know the saturated lines ($\mu_k^* \neq 0$ or $\nu_k^* \neq 0$)

Suppose we know the saturated lines ($\mu_k^* \neq 0$ or $\nu_k^* \neq 0$)

Then the task is much easier than solving OPF !!

- Solving the System of Linear Equations (for $p_g, \mu_k, \nu_k, \lambda$):

$$\begin{aligned}
\forall \mu_k^* \neq 0: \quad & \sum_i \Phi_{ki}(p_i - l_i) = f_k^{max} \\
\forall \nu_k^* \neq 0: \quad & \sum_i \Phi_{ki}(p_i - l_i) = f_k^{min} \\
\forall g: \quad & 2q_g p_g + c_g + \lambda + \sum_k \Phi_{ki}\mu_k^* + \sum_k \Phi_{ki}\nu_k^* = 0 \\
& \sum_g p_g = \sum_i l_i
\end{aligned}$$

Suppose we know the saturated lines ($\mu_k^* \neq 0$ or $\nu_k^* \neq 0$)

Then the task is much easier than solving OPF !!

- Solving the System of Linear Equations (for $p_g, \mu_k, \nu_k, \lambda$):

$$
\begin{aligned}
\forall \mu_k^* \neq 0: \quad & \sum_i \Phi_{ki}(p_i - l_i) = f_k^{max} \\
\forall \nu_k^* \neq 0: \quad & \sum_i \Phi_{ki}(p_i - l_i) = f_k^{min} \\
\forall g: \quad & 2q_g p_g + c_g + \lambda + \sum_k \Phi_{ki}\mu_k^* + \sum_k \Phi_{ki}\nu_k^* = 0 \\
& \sum_g p_g = \sum_i l_i
\end{aligned}
$$

System 2 (PIML) learning:

- Train a model (System 1, NN) to find saturated lines
- Then solve the Linear System of Eqs.

Classification of Saturated/Binding Lines

Neural Network

- $NN_\psi : I \to \begin{pmatrix} B(\mu) \\ B(\nu) \end{pmatrix}, \quad B(x) = (1: \text{ if } x \neq 0, \text{ 0otherwise})$
- simple ... 4 CNN layers, 500 neurons/layer, last layer sigmoid activation f.

Loss Function (logistic regression)

- $L_\psi = \sum_s \sum_k \left(L_{reg}(y_k^{\mu(s)}, B(\mu_k^{(s)})) + L_{reg}(y_k^{\nu(s)}, B(\nu_k^{(s)})) \right)$
  $L_{reg}(x, y) = \begin{cases} -log(x), & \text{if } y = 1 \\ -log(1-x), & \text{if } y = 0 \end{cases}$
- Supervise Learning (classification): I -input data; y - output data

Training = Supervised Learning (classification):

- Minimize the loss function over the NN parameters, ψ
- Given: I -input data; y - output data (saturated lines)

Test: IEEE 118-bus system

	Free generators
Config. #1	$\{3, 5, 11, 12, 18, 30, 34, 40, 42, 43\}$
Config. #2	$\{2, 5, 12, 26, 30, 39\}$
Config. #3	$\{5, 12, 14, 20, 30, 37, 39\}$

- Generate many samples of Noise for given configuration of the Generator Commitments (minutes to an hour). Consider Different Unit Commitments (Configurations).
- The intra-hour re-dispatch only on "free" generators

Test: IEEE 118-bus system

	Free generators
Config. #1	$\{3, 5, 11, 12, 18, 30, 34, 40, 42, 43\}$
Config. #2	$\{2, 5, 12, 26, 30, 39\}$
Config. #3	$\{5, 12, 14, 20, 30, 37, 39\}$

- Generate many samples of Noise for given configuration of the Generator Commitments (minutes to an hour). Consider Different Unit Commitments (Configurations).
- The intra-hour re-dispatch only on "free" generators

		Config. #1		Config. #2		Config. #3	
σ	size	training	testing	training	testing	training	testing
1%	50	0(0.00)	0(0.00)	0(0.00)	34(0.02)	1(0.01)	66(0.05)
1%	100	0(0.00)	0(0.00)	0(0.00)	48(0.03)	0(0.00)	273(0.20)
1%	200	0(0.00)	6(0.00)	0(0.00)	104(0.06)	0(0.00)	342(0.25)
5%	50	0(0.00)	0(0.00)	0(0.00)	16(0.01)	9(0.03)	50(0.05)
5%	100	0(0.00)	0(0.00)	0(0.00)	43(0.03)	0(0.00)	165(0.14)
5%	200	0(0.00)	6(0.00)	0(0.00)	79(0.05)	0(0.00)	232(0.19)
10%	50	0(0.00)	0(0.00)	4(0.00)	19(0.02)	26(0.05)	36(0.04)
10%	100	0(0.00)	0(0.00)	0(0.00)	24(0.02)	6(0.01)	103(0.01)
10%	200	0(0.00)	2(0.00)	0(0.00)	35(0.03)	1(0.00)	133(0.15)

Table I: Number of misidentifications, i.e. $y_k^{\nu|\mu(s)} = 1$ and $\mathcal{B}(\nu|\mu_k^{(s)}) = 0$ or vice versa, over the training and testing sets. The ratio of misidentifications to number of binding line constraints is given in parenthesis.

- Quality of Training depends on Configuration and Noise (Uncertainty)
- Decays with Noise

Test: IEEE 118-bus system

σ	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
1%	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.01	0.01	0.00
5%	0.04	0.01	0.02	0.00	0.03	0.07	0.02	0.00	0.02	0.06	0.05	0.01
10%	0.05	0.03	0.04	0.02	0.06	0.08	0.06	0.02	0.06	0.09	0.05	0.05

Table II: Fraction of misidentifications over testing set for the 13 unit commitment configurations not presented in Table I.

- Testing is on different samples than training (to make sure we do not overfit)

		Config. #1		Config. #2		Config. #3	
σ	size	training	testing	training	testing	training	testing
1%	50	0(0.00)	0(0.00)	0(0.00)	34(0.02)	1(0.01)	66(0.05)
1%	100	0(0.00)	0(0.00)	0(0.00)	48(0.03)	0(0.00)	273(0.20)
1%	200	0(0.00)	6(0.00)	0(0.00)	104(0.06)	0(0.00)	342(0.25)
5%	50	0(0.00)	0(0.00)	0(0.00)	16(0.01)	9(0.03)	50(0.05)
5%	100	0(0.00)	0(0.00)	0(0.00)	43(0.03)	0(0.00)	165(0.14)
5%	200	0(0.00)	6(0.00)	0(0.00)	79(0.05)	0(0.00)	232(0.19)
10%	50	0(0.00)	0(0.00)	4(0.00)	19(0.02)	26(0.05)	36(0.04)
10%	100	0(0.00)	0(0.00)	0(0.00)	24(0.02)	6(0.01)	103(0.01)
10%	200	0(0.00)	2(0.00)	0(0.00)	35(0.03)	1(0.00)	133(0.15)

Table I: Number of misidentifications, i.e. $y_k^{\nu|\mu(s)} = 1$ and $\mathcal{B}(\nu|\mu_k^{(s)}) = 0$ or vice versa, over the training and testing sets. The ratio of misidentifications to number of binding line constraints is given in parenthesis.

- Quality of Training depends on Configuration and Noise (Uncertainty)
- Decays with Noise

Test: IEEE 118-bus system

σ	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
1%	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.01	0.01	0.00
5%	0.04	0.01	0.02	0.00	0.03	0.07	0.02	0.00	0.02	0.06	0.05	0.01
10%	0.05	0.03	0.04	0.02	0.06	0.08	0.06	0.02	0.06	0.09	0.05	0.05

Table II: Fraction of misidentifications over testing set for the 13 unit commitment configurations not presented in Table I.

σ	Revenue Adequacy	Strong Duality	Cost Recovery
1%	1.000	1.000	0.808
5%	0.999	0.997	0.352
10%	0.999	0.992	0.060

Table III: Fraction of the testing samples (over all 15 unit commitment configurations) that satisfy the key (in)equalities as stated in Eq. (16)-(18).

- Testing is on different samples than training (to make sure we do not overfit)

- Checking for other criteria than used in training

Test: IEEE 118-bus system

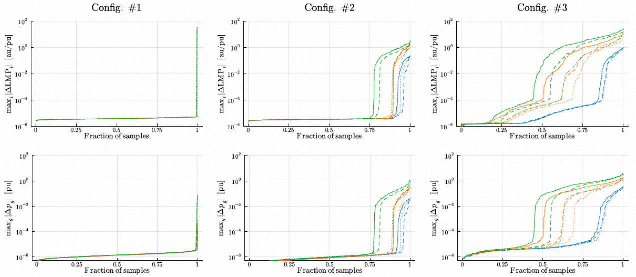


Figure 1: Mismatch between obtained LMPs (top) and generator outputs (bottom) for different load volatility standard deviations: 1% (blue), 5% (orange) and 10% (green) of their nominal values, and the training set consists of: 50 (solid), 100 (dashed) are 200 (dotted) samples. Each panel shows the fraction of the testing set that get maximal mismatches smaller than the ordinate.

- Success depends on the Unit Commitment Configuration, Level of the Noise and Number of Samples
- ⇒ Phase Transition Type of Behavior = Sharp Changes

Test: IEEE 118-bus system

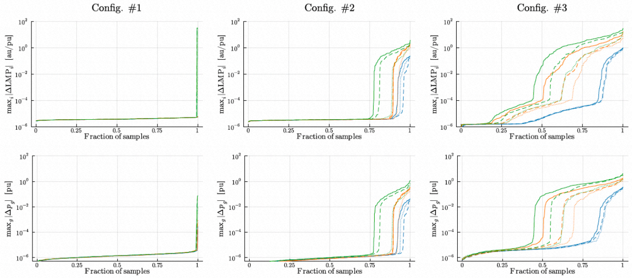


Figure 1: Mismatch between obtained LMPs (top) and generator outputs (bottom) for different load volatility standard deviations: 1% (blue), 5% (orange) and 10% (green) of their nominal values, and the training set consists of: 50 (solid), 100 (dashed) are 200 (dotted) samples. Each panel shows the fraction of the testing set that get maximal mismatches smaller than the ordinate.

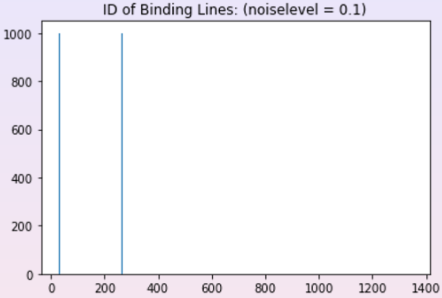
- Success depends on the Unit Commitment Configuration, Level of the Noise and Number of Samples
- $\Rightarrow$ Phase Transition Type of Behavior = Sharp Changes
- Larger System ? More realistic Noise (longer correlations)?

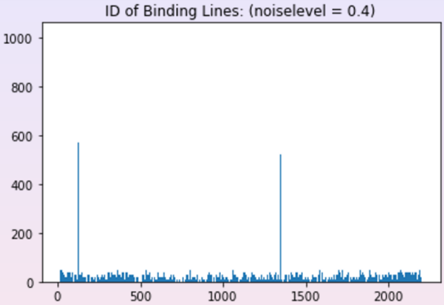
Test: New-York ISO Model (designed by Dan-The-Man)

THE MODEL (ARPA-E project led by Dan)

- Based on carefully curated real data for New York ISO
- 1814 buses, 395 generators, 2203 lines
- Time: August 28, 2018, 5 pm hour
- From Security Constrained Unit Commitment (SCUC): get unit commitment; factor in committed generators as negative load
- Noise
  - (a) Re-scaled white noise to wind farms. Ignore short time scales (10-20 min) between consecutive unit commitments
  - (b) From base load: add noise using factor stressing methodology – developed by Dan
- Major Tool — Real-Time Simulator (SCUC + Security Control Energy Dispatch/OPF) – developed by Dan

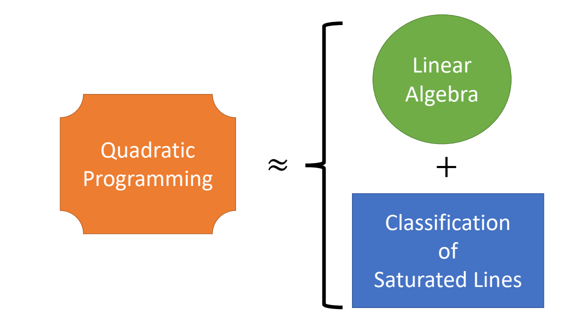
Test: New-York ISO Model (designed by Dan-The-Man)





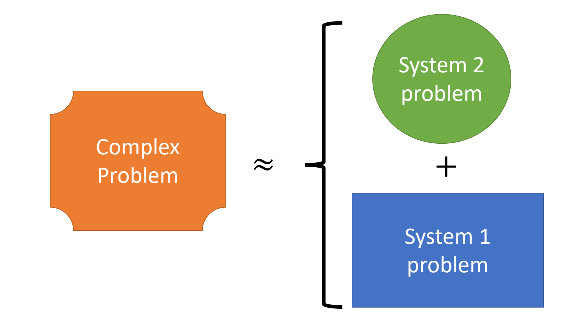
- Preliminary Results (more in two three weeks — next ARPA-E review)
- Under low noise, NN correctly identifies 90% of the line dual variables as zero or nonzero — pretty good
- ... but can be improved, perhaps with alternative ML schemes (may be even simpler than NN, e.g. Support Vector Machine)

Take Home Message



- Reduced Modeling = faster, less data, accurate (enough)

Take Home Message



- Seems Like a General Approach
- Open Challenge (for theorists):
  - Given recent NN theories (for System 1 = quality of estimations faster/less data/accurate) ⇒ How does the quality/error propagate to the overall output (of the reduced model)?

Trustworthy Scientific AI

Facets of Expert (e.g. scientist/engineer) Trust in AI

- Autonomy (distributed agents)
- Beneficence (useful for all)
- Nonmaleficence (no harm)
- Justice (fairness)

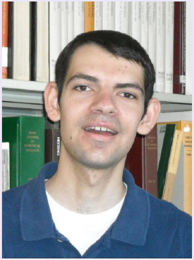
- **Explainability** — "system 2 level", i.e. in use-inspired expert (physicist, power engineer, epidemiologist) terms
- **Prepardness** (for rare but possibly devastating events)
- **Reproducability** (at least two principally different models agree)

Theory is Needed for (Scientists & Engineers) Trust

Desiderata:

- **Explain with Asymptotics:** Explain SIMPLE System 1 ML/AI system (one layer & many neurons, many layers & neurons, ReLU or whatever ..., phase transitions = asymptotic theory, finite size effects = e.g. finite system # of samples) to enable System 2
- **Explain with Structure:** Inject more structure in your theory (e.g. optimization, inference & learning) — Graphical Models – System 2 (structure) enabler
- **Prepare with Better Extrapolation:** especially of extreme events, System 2 is needed to generalize into unseen regimes
- **Reproduce with Alternatives:** Many more and complementary (System 2 and System 1, evolving) Models (and thus Theories) are NEEDED

Laurent Pagnier



Robert Ferrando

Yury Dvorkin

Dan Bienstock

Support is Appreciated !!

- **Energy Systems**: UArizona start up + DOE/ARPA-E

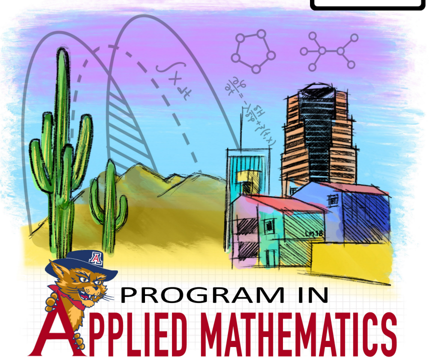
Thanks for your attention !

- Research focused, since 1976, one of the US first [dynamical systems, integrability, turbulence …]
- Interdisciplinary: 100+ professors/ 26 departments/ 8 colleges across UA campus (CoS & CoE & Optics – top 3)
- Mixing traditional @ contemporary Applied Math
- Graduate, Ph.D. focused, no terminal M.Sc.
- 60 Ph.D students (**15**/13/16/10 enrolled in **2022**/21/20/19)
- **3 Core Courses** (1st year -- Methods, Analysis, Algorithms) https://appliedmath.arizona.edu/students/new-core-courses
- Strong collaborations with Industry (e.g. Raytheon, Uber, Intel, Critical Path, etc) and National Labs (e.g. LANL, LLNL, NREL, NNSS, BNL, SNL etc)
- 5 seminar/colloquium series – recorded and posted online
- Participation in many UA & National Edu Projects

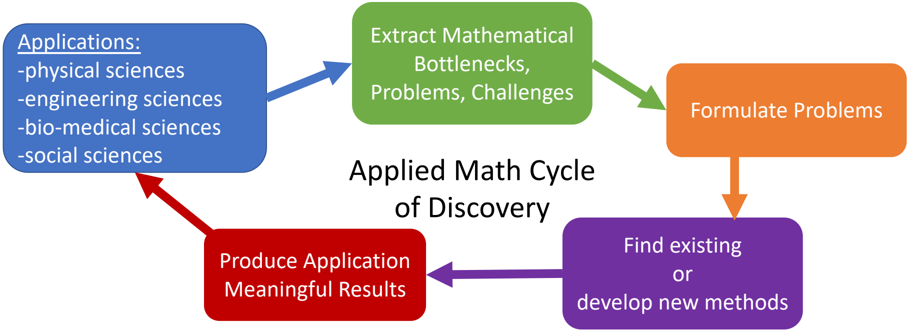
SCAN ME

http://appliedmath.arizona.edu/

chertkov@arizona.edu



How does Mathematics work with Applications @ UArizona?



Core courses provide hands on teaching of the AM-cycle methodology

- Training in **methods** (Math/APPL **581**), **theory** (Math/APPL **584**), **algorithms** (Math/APPL **589**)
- Math (quantitative) and Application-specific (qualitative) intuition