Outage Detection from PMU Data

Stephen Wright

University of Wisconsin-Madison

3/28/17

Overview

Line outages cause distinctive "signatures" in the phasor data that can be recognized via machine learning classification techniques:

- multiclass logistic regression
- neural network (deep learning).

Regularized versions of these techniques help to identify PMU locations that yields most effective identification.

Most computation is moved offline (classifier training).

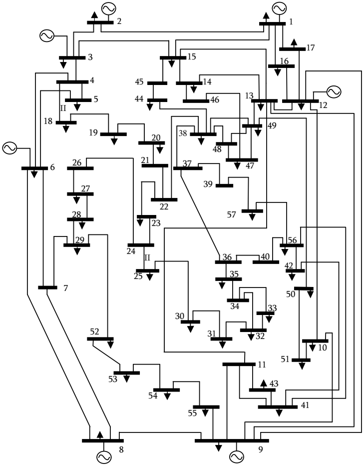
Collaborators (UW-Madison):

- Taedong Kim
- Ching-pei Lee

Optimization in Power Systems

- Optimal Power Flow
- Economic dispatch / market design
- Modeling renewable energy sources
- Grid expansion / planning
- Environmental effects e.g. air pollution control

AC Power Flow Model



IEEE 57-Bus System

- Set of nodes ("buses") $\mathcal{N}$
  - $\mathcal{G} \in \mathcal{N}$ (generators).
  - $\mathcal{D} \in \mathcal{N}$ (demand/load buses).
- Set of transmission lines: $\mathcal{L} \subset \mathcal{N} \times \mathcal{N}$.
- For node $i \in \mathcal{N}$;
  - Complex voltage: $V_i e^{\jmath \theta_i}$.
  - Complex power: $P_i + jQ_i$
- For transmission line $(i,k) \in \mathcal{L}$:
  - Complex admittances: $Y_{ik} = G_{ik} + jB_{ik}$

Power Balance Equations for each bus i:

$$P_i = V_i \sum_{k \in \mathcal{N}} V_k (G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k)),$$

$$Q_i = V_i \sum_{k \in \mathcal{N}} V_k (G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k)).$$

P_i, V_i are given at $\mathcal{G}$ buses, P_i, Q_i given at $\mathcal{D}$

AC Power Flow Problem

Power mismatch at each node i:

$$F_i^P(V,\theta) := V_i \sum_{k\in\mathcal{N}} V_k(G_{ik}\cos(\theta_i-\theta_k) + B_{ik}\sin(\theta_i-\theta_k)) - P_i,$$

$$F_i^Q(V,\theta) := V_i \sum_{k\in\mathcal{N}} V_k(G_{ik}\sin(\theta_i-\theta_k) - B_{ik}\cos(\theta_i-\theta_k)) - Q_i.$$

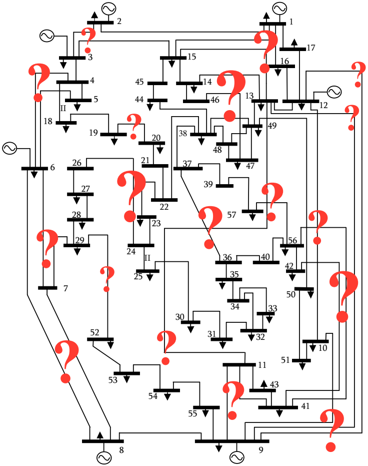
All F_i^P and F_i^Q should be zero at a solution, an operation point of the grid.

AC power flow *unknowns are* V_i *for* $i \in \mathcal{D}$, *and* θ_i *for* $i \in \mathcal{G}\cup\mathcal{D}$:

$$F(V,\theta) := \begin{bmatrix} F_{\mathcal{G}}^P(V,\theta) \\ F_{\mathcal{D}}^P(V,\theta) \\ F_{\mathcal{D}}^Q(V,\theta) \end{bmatrix} \quad \text{(square nonlinear system).}$$

- Can be solved using Newton's method, or enhancements. *(Much recent interest in SDP relaxations.)*
- $F(V,\theta)=0$ may have no solution, or multiple solutions. (Often have at most one operationally desirable solution, where $V_i \approx 1$.)
- When voltages formulated in "rectangular coordinates" $(U_i + jV_i)$, the AC equations are a system of quadratic equations. Can all solutions be found?

Line Outage Identification



Goal: Fast and accurate identification of power line outages. Important for preventing further faults, routine monitoring, control.

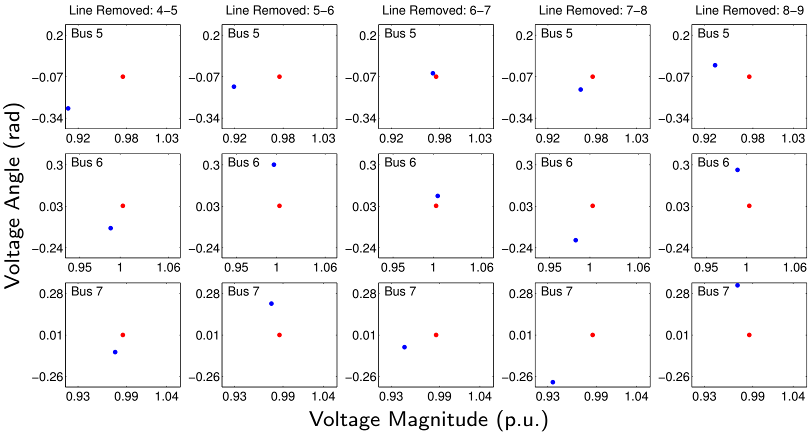
Key Observation: Each outage event has a distinctive "signature" of changes to voltage phasor measurements.

Tool: Phasor Measurement Units (PMU) can be placed on lines in the grid (typically near buses) to provide streaming data about voltage phasors and currents. Sample at 30 Hz.

Approach: Use machine learning / optimization to recognize and classify each signature, thus detecting outages rapidly.

Design: Use regularized optimization frameworks to decide where to place a limited number of PMUs in the grid, to maximize detection performance.

Example: 9-Bus System



- Red: Voltages before line outages.
- Blue: Voltages after line outages.

Distinctive Signature: $X = \begin{bmatrix} \Delta V_1 & \Delta\theta_1 & \Delta V_2 & \Delta\theta_2 & \cdots & \Delta V_n & \Delta\theta_n \end{bmatrix}^T$

where $\Delta V_i = V_i' - V_i$ and $\Delta\theta_i = \theta_i' - \theta_i$ for $i = 1, 2, \ldots, n$.

Multinomial Logistic Regression (MLR)

- K classes: $1, 2, \cdots, K$ $\Longleftarrow$ **Line Outages**
- Observation vector: X $\Longleftarrow$ **Voltage Phasor Shifts**
- Outcome given an observation X: $Y \in \{1, 2, \ldots, K\}$.

- Given X, the probability that it belongs to class $j \in \{1, 2, \ldots, K\}$ is given by

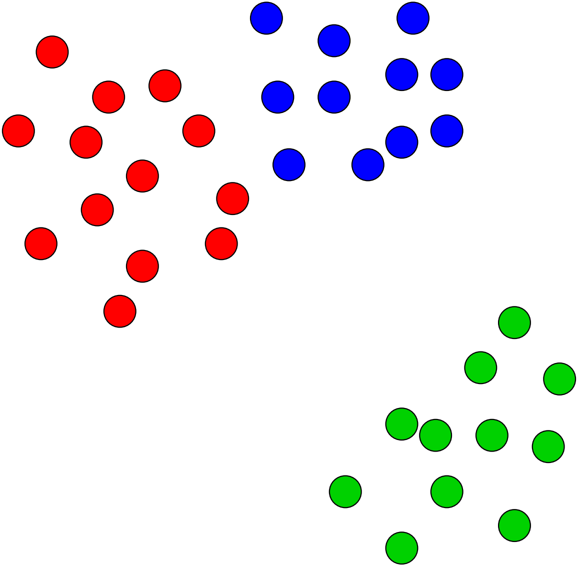
$$\Pr(Y = j|X) := \frac{e^{\langle \beta_j, X \rangle}}{\sum_{k=1}^{K} e^{\langle \beta_k, X \rangle}} \quad \text{for } j = 1, 2, \ldots, K$$

where $\beta_1, \ldots, \beta_K$ are regression coefficients.

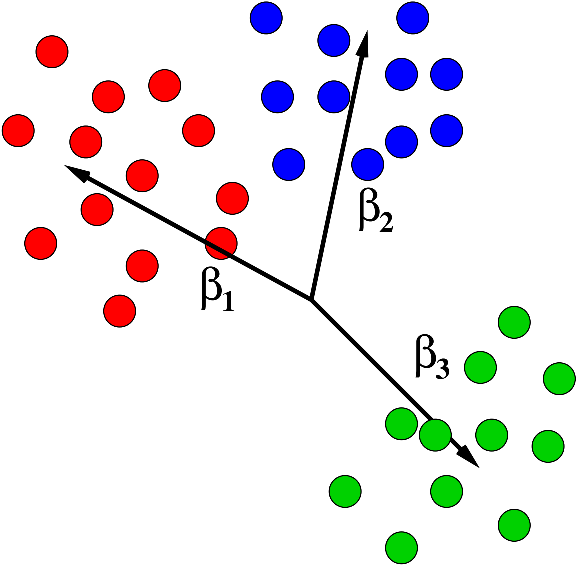
Note: $\Pr(Y = j|X) \approx 1 \iff \langle \beta_j, X \rangle \gg \langle \beta_k, X \rangle$ for all $k \neq j$.

Need to learn regression coefficients $\beta_1, \cdots, \beta_K$ from observed data pairs (X_i, Y_i).

Clusters and Coefficients



Clusters and Coefficients



Learning Regression Coefficients

Given : M training pairs: $(X_1, Y_1), (X_2, Y_2), \ldots, (X_M, Y_M)$.
Require : Coefficient vectors β_k, $k = 1, 2, \ldots, K$, such that

$$Y_i = j \;\Rightarrow \langle \beta_j, X_i \rangle \gg \langle \beta_k, X_i \rangle, \;\; \text{for } k \neq j.$$

- Joint probability of observing $(Y_1, Y_2, \ldots, Y_M)$, given $(X_1, X_2, \ldots, X_M)$:

$$\prod_{i=1}^{M} \Pr(Y = Y_i | X_i) = \prod_{i=1}^{M} \frac{e^{\langle \beta_{Y_i}, X_i \rangle}}{\sum_{k=1}^{K} e^{\langle \beta_k, X_i \rangle}}$$

- Log-likelihood function $f(\beta)$ while $\beta := \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_K \end{bmatrix}$:

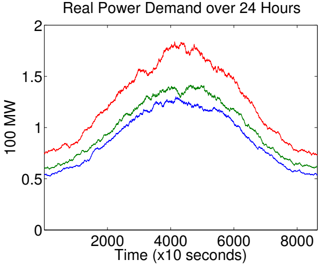
$$f(\beta) := \sum_{i=1}^{M} \left(\langle \beta_{Y_i}, X_i \rangle - \log \sum_{k=1}^{K} e^{\langle \beta_k, X_i \rangle} \right) \quad \Leftarrow \text{Concave function!}$$

- Maximize f to estimate coefficient $\beta = [\beta_1, \beta_2, \ldots, \beta_K]$.
- This is a smooth convex optimization problem. Can use standard optimization tools, e.g. stochastic gradient (for large data sets) or L-BFGS.
- Solving this problem can be expensive! But it's done offline. Once β_j are known, online computation is just a $K \times M$ matrix-vector multiplication.

Experiment Setup: Training Sets and Test Sets

Synthetically generated 24-hour demand data (Ornstein-Uhlenbeck process).

The feature vectors X corresponding to a particular outage j will change with demand conditions, thus need to sample a variety of demands. (Hopefully they are distinctive enough from other outages that we can still do effective classification.)



Use L-BFGS to learn regression coefficients

- Training Set: 5 equally-spaced samples from the first half of the 24-hour period, for each line outage.

  $\Rightarrow$ From historical data.

- Test Set: 50 random samples from the second half of the 24-hour period, for each line outage.

System	# of Outage Scenarios	# of Samples	
		Training Set	Test Set
14-Bus	18	90	900
30-Bus	37	185	1850
57-Bus	67	335	3350
118-Bus	170	850	8500

Experiment Setup: Procedure

- For each outage Y_i, $i = 1, 2, \ldots, M$, sample a set of demands at the $\mathcal{D}$ nodes from the random distribution above.
- For each of these outage / demand pairs, solve the AC power flow equations to obtain voltage magnitude V_k and angle θ_k for all nodes $k = 1, 2, \ldots, n$.
- Compare with the reference (no-outage) solution of the AC power flow equations for this set of demands, to obtain the feature vector X, which consists of ΔV_k and $\Delta \theta_k$, $k = 1, 2, \ldots, n$.
- Solve the MLR problem over all these training vectors to obtain regression coefficients $\beta_1, \ldots, \beta_K$.
- Form test feature vectors, in the same manner as the training vectors.
- For each test vector X, form $\langle \beta_j, X \rangle$, $j = 1, 2, \ldots, K$ to see how well it predicts the outage Y that was used to generate X.
- Gather statistics on test data, including misclassification error rates.

Results: PMUs Measurements from All Buses

System	Probability			Ranking			DC Model (NAD)
	$\geq 90\%$	$\geq 70\%$	$\geq 50\%$	1	≤ 2	≤ 3	
IEEE 14-Bus	0%	0%	0%	0%	0%	0%	0%
IEEE 30-Bus	0%	0%	0%	0%	0%	0%	1.3%
IEEE 57-Bus	0.5%	0.3%	0.2%	0.2%	0%	0%	4.4%
IEEE 118-Bus	0.2%	0.2%	0.2%	0.2%	0.1%	0%	3.1%

Line Outage Identification Error Rates on Test Set with PMUs on All Buses.

- "Probability" indicates statistics for the probability assigned by the MLR classifier to the actual outage event.
- "Ranking" indicates whether the actual event was ranked in the top 1, 2, or 3 of probable outage events by the MLR classifier.
- Normalized angle distances (NAD) [6] are obtained by solving DC power flow models for test instances using MATPOWER. (Only phase angles.)

However can't assume that PMUs are deployed at all buses! (If so, we could detect outages directly, rather than inferring them by changes to voltage phasors.)

Given only a limited number of PMUs, can we place them optimally?

PMU Placement Problem

Goal: Find a subset of buses at which to locate PMUs, to maximize outage detection performance.

Approach: sparse optimization: regularization and greedy heuristics.

Recall that the feature vector is

$$X = [\underbrace{\Delta V_1 \quad \Delta\theta_1}_{\text{Bus 1}} \quad \underbrace{\Delta V_2 \quad \Delta\theta_2}_{\text{Bus 2}} \quad \cdots \quad \underbrace{\Delta V_n \quad \Delta\theta_n}_{\text{Bus } n}]^T,$$

If bus i is not instrumented, we cannot observe $(\Delta V_i, \Delta\theta_i)$, so need to set the two corresponding coefficients of β_j (elements $2i-1$ and $2i$) to zero for all $j = 1, 2, \ldots, K$.

Add regularization functions to the max-likelihood objective, to suppress the entries $2i-1$ and $2i$ (for each $i = 1, 2, \ldots, n$) unless they are essential to distinguishing between different outages. Solve for $\tau > 0$:

$$\beta^* = \arg\max_{\beta} \; f(\beta) - \tau \sum_{i=1}^{n} q_i(\beta)$$

where $q_i(\beta) := \left\|\beta_{\{2i-1,2i\},\cdot}\right\|_F$ for $i = 1, \cdots, n$ (Frobenius norm of a $2 \times K$ submatrix).

Greedy Group Sparse Heuristic

- In the greedy variation, only one bus is selected at each "outer" iteration.
- Then solve GroupLASSO to select the next bus — but remove regularization terms (and hence bias in β) for the buses already selected.

Let $\mathcal{S} = \{1, 2, \cdots, n\}$, the set of all buses.

GroupLASSO

1: $\beta^* = \arg\max_{\beta} f(\beta) - \tau \sum_{i \in \mathcal{S}} q_i(\beta)$

2: $\mathcal{R}^r = \arg\max_{\mathcal{R}:|\mathcal{R}|=r, \mathcal{R} \subseteq \mathcal{S}} \sum_{j \in \mathcal{R}} q_j(\beta^*)$

Greedy Variant

1: $\mathcal{R}^0 = \emptyset$

2: **for** $l = 1, 2, \cdots, r$ **do**

3: $\beta^* = \arg\max_{\beta} f(\beta) - \tau \sum_{j \notin \mathcal{S} \setminus \mathcal{R}^{l-1}} q_j(\beta)$

4: $s^l \leftarrow \arg\max_{j \notin \mathcal{R}^{l-1}} q_j(\beta^*)$

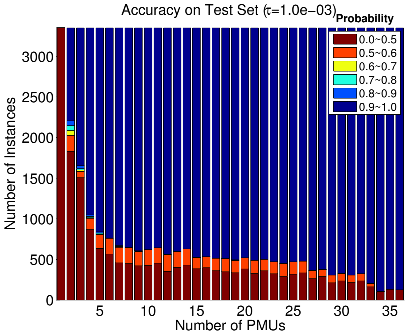
5: $\mathcal{R}^l \leftarrow \mathcal{R}^{l-1} \cup \{s^l\}$

6: **end for**

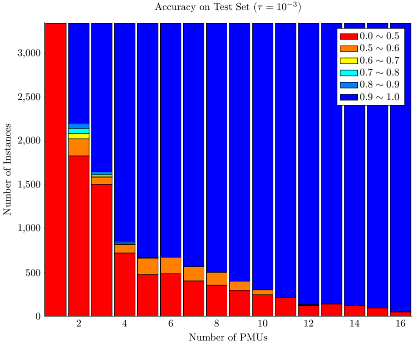
Greedy Advantage: Selection of redundant PMU locations is suppressed by already-selected, non-penalized locations.

Results: IEEE 57-Bus System

We enhanced the basic **GroupLASSO** heuristic with a **Greedy Group Sparse** heuristic that selects one PMU location at a time.



GroupLASSO

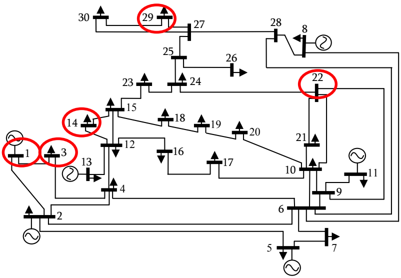


Greedy Heuristic

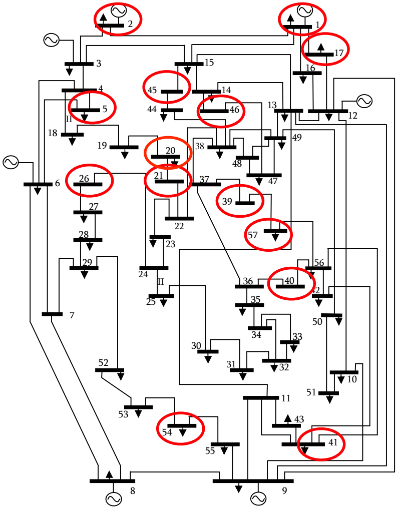
Greedy heuristic gives **Highly reliable detection obtained with just 15 PMUs.**

For our test networks, classification performance is as good with 25% PMUs as with PMUs everywhere.

PMU Locations on 30-Bus and 57-Bus Systems



IEEE 30-Bus system (5 PMUs)



IEEE 57-Bus system (14 PMUs)

100% Accuracy on Test Set

98.5% Accuracy on Test Set

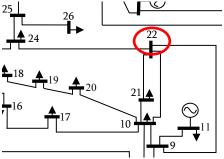
Greedy Heuristic Placement Results on IEEE Test Cases

Buses	τ	# of PMUs	Probability ≥ 0.9	Ranking 1	Ranking ≤ 2	Ranking ≤ 3	DC Model (NAD)
14	.05	3	0.4%	0.2%	0%	0%	30.9%
	.005	3	0%	0%	0%	0%	43.0%
30	.05	4	0.4%	0.4%	0%	0%	57.9%
	.005	5	0%	0%	0%	0%	45.9%
57	.05	12	2.9%	2.9%	0.2%	0.2%	28.0%
	.005	14	1.5%	1.5%	0.1%	0.1%	25.0%
118	.05	15	5.8%	5.8%	3.8%	3.8%	30.3%
	.005	21	0.7%	0.4%	0.1%	0.1%	28.2%

The competition (based on the simpler DC power flow model) does poorly with limited observations.

Extension: Explicit Line Outage Information

So far, we have identified outages using only the "indirect" evidence of voltage phasor changes from PMUs placed at certain buses.



In fact, a PMU also measures current on its line.

$\Rightarrow$ Can detect outage on this line **directly**.

We extend the MLR framework to incorporate direct knowledge of outages.

Construct a $2n \times K$ matrix L using parameter $\eta > 0$.

- Column L_k is associated with transmission line k.
- Rows $2k-1$ and $2k$ are associated with the buses that can detect the outage of transmission line k. (The two end points of the line.)

The extended observation vector $\overline{X}$ for a line-k outage:

$$\overline{X} = \begin{bmatrix} X \\ L_{\cdot k} \end{bmatrix}$$

$$L := \begin{bmatrix} \eta & 0 & \cdots & 0 \\ \eta & 0 & \cdots & 0 \\ 0 & \eta & \cdots & 0 \\ 0 & \eta & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \eta \\ 0 & 0 & \cdots & \eta \end{bmatrix}$$

Experiments Setup

We compare the following 2 strategies for the outage identification performances.

- **Indirect**: Direct observations are ignored. Use original features X.
- **Combined**: Direct observations are incorporated as above; feature vectors $\overline{X}$.

Use PMU placement for Indirect and Combined strategies.

Results: 57-Bus System with $\eta = 10^{-2}$

Strategy	1* 5 20 21 26 39 40 43 54 57 (10 PMUs)					
	Probability			Ranking		
	≥ 0.9	≥ 0.7	≥ 0.5	1	≤ 2	≤ 3
Indirect	5.9%	5.9%	5.9%	5.9%	0.9%	0.9%
Combined	4.6%	4.6%	4.6%	4.6%	1.2%	1.2%

Selected Bus	5	20	21	26	39	40	43	54	57	Total
# of Lines Touching the Bus	2	2	2	2	2	2	2	2	2	18

PMU Placement Based on Indirect Observations

Strategy	1* 5 9 21 26 39 45 46 49 56 (10 PMUs)					
	Probability			Ranking		
	≥ 0.9	≥ 0.7	≥ 0.5	1	≤ 2	≤ 3
Indirect	6.7%	6.7%	5.3%	5.3%	0.6%	0.1%
Combined	2.6%	2.6%	2.6%	2.6%	0.1%	0.1%

Selected Bus	5	9	21	26	39	45	46	49	56	Total
# of Lines Touching the Bus	2	6	2	2	2	2	2	4	4	26

PMU Placement Based on Combined (Direct + Indirect) Observations

Wider Range of Demands

Training is for a typical single-day profile of demands. A referee asked: "What about seasonal variations?," that is, a wider range of demand variations.

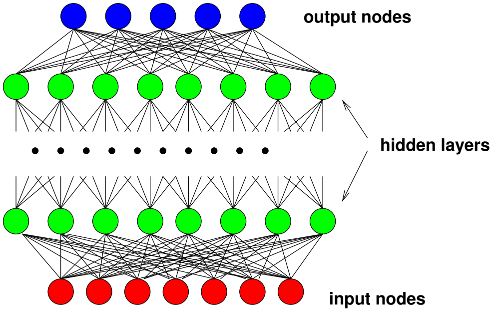
We retrained on a training set 3x as large, scaling the original demands by $.85$, 1.0, and 1.15. Different optimal PMU placements, different error rates.

Results for 118-bus case show some degradation. This gets worse when we try a wider range of demands.

# of PMUs	Load Profile (κ) for Testing	Probability	Ranking		
		≥ 0.9	1	≤ 2	≤ 3
15 ($\tau = 2.8 \times 10^{-1}$)	0.85	5.6%	5.4%	2.5%	1.8%
	1.00	6.2%	6.2%	4.8%	4.7%
	1.15	11.4%	11.0%	3.6%	2.1%
21 ($\tau = 2.25 \times 10^{-2}$)	0.85	1.9%	1.8%	0.1%	0.1%
	1.00	0.4%	0.2%	0.1%	0%
	1.15	1.1%	1.0%	0.2%	0.1%

BUT NOTE: Could improve these by training different coefficients for different conditions — no need to use "one size fits all" parameters β_j, $j = 1, 2, \ldots, K$.

Deep Neural Networks



Inputs are the vectors a_j, outputs are odds of a_j belonging to each class (as in multiclass logistic regression).

At each layer, inputs are converted to outputs by a linear transformation composed with an element-wise function:

$$a^{l+1} = \sigma(W^l a^l + h^l),$$

where a^l is node values at layer l, (W^l, h^l) are *parameters* in the network, σ is the element-wise function.

Deep Learning

The element-wise function σ makes simple transformations to each element:

- Logistic function: $t \to 1/(1+e^{-t})$;
- Hinge: $t \to \max(t, 0)$: ReLU
- Bernoulli: $t \to 1$ with probability $1/(1+e^{-t})$ and $t \to 0$ otherwise (inspired by neuron behavior).

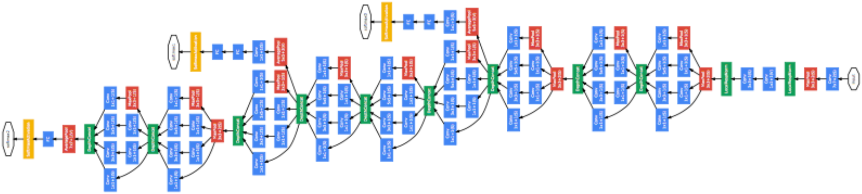
The final stage is often a multiclass-logistic-regression classifier. Lower "hidden" layers can be viewed as transforming the raw input vector to make it easier to classify e.g. tighten the clusters.

The example depicted shows a completely connected network — but more typically networks are engineered to the application (speech processing, object recognition, . . .).

- local aggregation of inputs: pooling;
- restricted connectivity + constraints on weights (elements of W^l matrices): convolutions.

GoogLeNet

Visual object recognition: Google's State of the Art network from 2014 [5].



Training Deep Learning Networks

The network contains many parameters — (W^l, h^l), $l = 1, 2, \dots, L$ in the notation above — that must be selected by training on the data (a_j, y_j), $j = 1, 2, \dots, m$.

Objective has the form:

$$\sum_{j=1}^{m} h(x; a_j, y_j)$$

where $x = (W^1, h^1, W^2, h^2, \dots)$ are the parameters in the model and h measures the mismatch between observed output y_j and the outputs produced by the model (as in multiclass logistic regression).

Nonlinear, Nonconvex. (Also random for Bernoulli activation.)

Usually trained with stochastic gradient methods, which make small adjustments to all the parameters based on the gradient $\nabla h(x; a_j, y_j)$ for a single sample $j \in \{1, 2, \dots, m\}$.

Composition of many simple functions.

WHY DOES DEEP LEARNING WORK SO WELL? is one of the great mysteries of science today. Intense investigation going on in the machine learning community.

Experiments

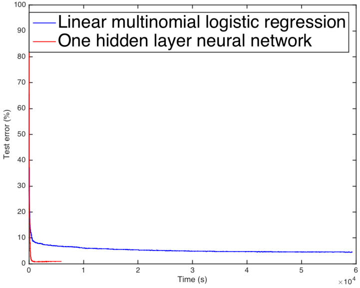
- IEEE test networks: 9, 14, 30, 57, 118.
- Use same random demand data as above, but scaled by a wider range of factors: $\{.5, .75, 1, 1.25, 1.5\}$, with same number of training data points for each factor.
- Most experiments with a single hidden layer: 100 nodes for smaller examples and 200 nodes for 57- and 118-bus cases.
- Fully connected neural network with $\tanh$ activation function.
- Generate training/validation/test sets as follows:
  - Training; For each outage Y_i, 20 points from the random demand distribution;
  - Validation: 10 points for each outage. (Used to choose parameter τ in PMU selection.)
  - Test: 50 points for each outage. (Used to measure effectiveness of classifier.)

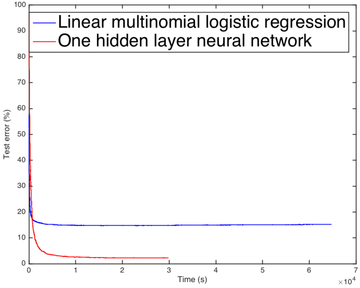
PMUs on All Buses

Get baseline rates by assuming that PMUs on all buses. Compare multiclass logistic regression (MLR) with neural network with one hidden layer (NN1).

IEEE Bus	9	14	30	57	118
MLR	0.48%	0.00%	1.76%	4.50%	15.19%
NN1	0.19%	0.43%	0.03%	0.91%	2.28%

NN1 is much better! Test error vs time for 57- and 118-bus case:





PMU Selection

For the 57-bus case, select a subset of buses for PMU instrumentation in 2 ways:

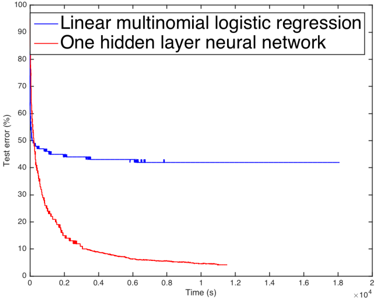
- Take the 10 optimal locations chosen by MLR for cleaner data (demands scaled by $\{.85, 1, 1.15\}$.
- Use the greedy-LASSO heuristic to choose about 11 locations.

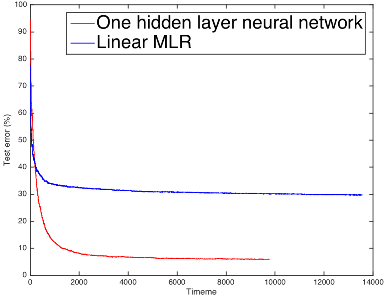
Model	Buses selected	Error rate
MLR (preselected)	[1 2 17 19 26 39 40 45 46 57]	42.01%
NN1 (preselected)	[1 2 17 19 26 39 40 45 46 57]	3.95%
MLR (greedy-LASSO)	[5 16 20 31 40 43 44 51 53 57]	29.85%
NN1 (greedy-LASSO)	[5 20 31 40 43 50 51 53 54 57]	7.11%

- MLR is not useful in this setting, where demands vary widely.
- The "preselected" buses are better!

The latter is surprising. Overfitting?

Test Err vs Time Charts





Preselected buses (left) and Greedy-LASSO (right).

Tighter Clustering in NN

The effect of the hidden layer is apparently to create clusters that are separated better than in the raw data. Measure this effect via two statistics based on the *centroid* denoted by c_i, which is the average of feature vectors in class i.

- Distance to centroid in my own class: $\|x_j - c_i\|$ where x_j belongs to class i.
- Distances between centroids: $\|c_i - c_\ell\|$ for $i \neq \ell$.

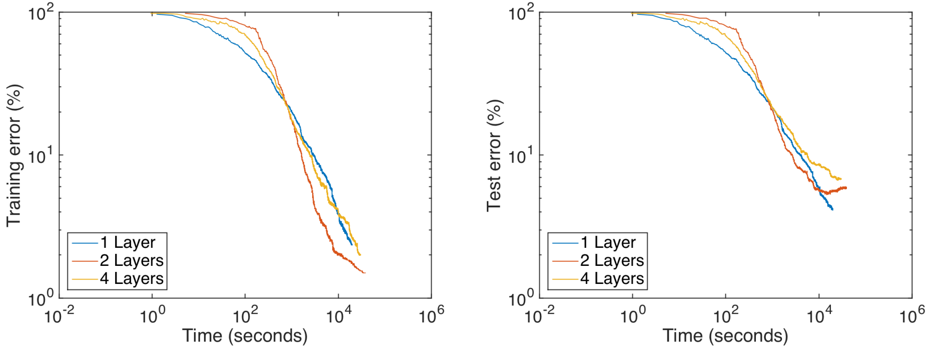
Tabulate the means and standard deviations of these quantities.

	Within Cluster mean (std dev)	Between Clusters mean (std dev)
Full Raw Data	.30 (.14)	.17 (.14)
Preselected MLR	.27 (.12)	.08 (.06)
Preselected NN1	3.79 (1.41)	4.61 (1.99)
Greedy-LASSO MLR	.27 (.12)	.08 (.05)
Greedy-LASSO NN1	2.30 (1.01)	3.27 (1.10)

Multiple Layers

Tried experiments on 57-bus case (with 10 preselected buses) with

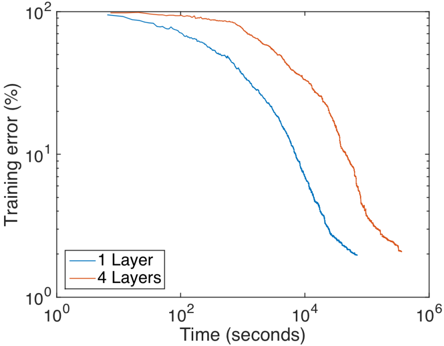
- One hidden layer of 200 nodes;
- Two hidden layers of 200 and 100 nodes;
- Four hidden layers of 50, 50, 50, 50 nodes.

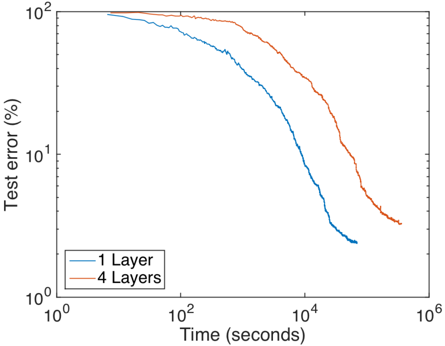


Evidence of overfitting, particularly in the 2-layer model.

Multiple Layers - Larger Data Set

Increase training set size by 5X, but generate it in the same way. Train on the 1-layer and 4-layer models.





The 4-layer model overfits less here, and may ultimately give better test error. But training time is almost 10X slower, and we did not run to completion.

Outage Detection: Other Approaches, and Context

Outages can of course be identified by other, more direct means. But detection via PMU readings provides a backup capability.

Other approaches proposed:

- [6, 7] use DC model and phase angle changes only.
- [1] use support vector machines (another machine learning technique).
- [9] use compressed sensing techniques (DC model)
- [2] use cross-entropy optimization (DC model)
- [8] use a distributed framework that avoids sharing of raw PMU data.

Recent work [3] has applied MLR to transient PMU data.

There is also recent work on PMU placement e.g. via nonlinear integer programming.

Optimization & Data Analysis in Power Systems

Electrical Power Systems provides an extremely rich source of problems that challenges all areas of optimization and data analysis.

- Optimization modeling
- Nonlinear equations
- SDP relaxations of polynomial systems
- Duality
- Equilibrium problems and games
- Nonlinear programming
- Bilevel optimization
- Classification and deep learning
- Sparse optimization
- Stochastic optimization
- Integer programming
- Online optimization and learning
- ......

References I

[1] Abdelaziz, A. Y., Mekhamer, S. F., Ezzat, M., and El-Saadany, E. F. (2012). Line outage detection using Support Vector Machine (SVM) based on the Phasor Measurement Units (PMUs) technology. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–8, San Diego, CA.

[2] Chen, J.-C., Li, W.-T., Wen, C.-K., Teng, J.-H., and Ting, P. (2014). Efficient Identification Method for Power Line Outages in the Smart Power Grid. *IEEE Transactions on Power Systems*, 29(4):1788–1800.

[3] Garcia, M., Catanach, T., Vander Wiel, S., Bent, R., and Lawrence, E. (2017). Line outage locationization using phasor measurement data in transient state. Manuscript.

[4] Kim, T. and Wright, S. J. (2016). Pmu placement for line outage identification via multinomial logistic regression. *IEEE Transactions on Smart Grid*. to appear. TSG.2016.2546339.

[5] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *CVPR*.

References II

[6] Tate, J. E. and Overbye, T. J. (2008). Line Outage Detection Using Phasor Angle Measurements. *IEEE Transactions on Power Systems*, 23(4):1644–1652.

[7] Tate, J. E. and Overbye, T. J. (2009). Double line outage detection using phasor angle measurements. In *2009 IEEE Power & Energy Society General Meeting*, pages 1–5, Calgary, AB.

[8] Zhao, L. and Song, W.-Z. (2014). Distributed power-line outage detection based on wide area measurement system. *Sensors*, 14(7):13114–13133.

[9] Zhu, H. and Giannakis, G. B. (2012). Sparse Overcomplete Representations for Efficient Identification of Power Line Outages. *IEEE Transactions on Power Systems*, 27(4):2215–2224.

Questions?

Steve Wright

swright@cs.wisc.edu