

## COMPARISON OF TWO DIMENSION-REDUCTION METHODS FOR NETWORK SIMULATION MODELS

Kevin L. Mills  
James J. Filliben

Information Technology Laboratory  
National Institute of Standards & Tech.  
Gaithersburg, MD 20899, USA

### ABSTRACT

Experimenters characterize the behavior of simulation models for data communications networks by measuring multiple responses under selected parameter combinations. The resulting multivariate data may include redundant responses reflecting aspects of a smaller number of underlying behaviors. Reducing the dimension of multivariate responses can reveal the most significant model behaviors, allowing subsequent analyses to focus on one response per behavior. This paper investigates two methods for reducing dimension in multivariate data generated from simulation models. One method combines correlation analysis and clustering. The second method uses principal components analysis. We apply both methods to reduce a 22-dimensional dataset generated by a network simulator. We identify issues that an analyst must decide, and we compare the reductions suggested by the methods. We have used these methods to identify significant behaviors in simulated networks, and we suspect they may be applied to reduce the dimension of empirical data measured from real networks.

### 1 INTRODUCTION

Paxson and Floyd (1997) describe many difficult problems that impede simulation of large data communication networks, and recommend two main coping strategies: search for invariants and carefully explore the parameter space. Unfortunately, Paxson and Floyd fail to address a key, related question: What model responses should be examined? This question is key because typical network simulation models (e.g., Fall and Varadhan 2009, SSFNet 2009, Tyan et al. 2009, Riley et al. 2004, Yaun et al. 2003, Zeng et al. 1998) can measure system response through tens to hundreds of outputs, which might represent aspects of fewer significant underlying model behaviors. Usually, experimenters select a subset of model outputs to analyze because considering all available responses proves too time consuming, too costly or computationally infeasible. When choosing a subset of simulation outputs, experimenters using ad hoc selection techniques may omit responses that characterize important model behaviors. Further, experimenters may select outputs in a fashion that overemphasizes particular behaviors. These mistakes become particularly salient during careful exploration of a model's parameter space, where experimenters seek to understand the response of a model to changes in input parameters.

Overweighting or underweighting significant model behaviors can yield invalid conclusions, thus some method is required to identify precisely the model outputs that correspond to each significant behavior. Fodor (2002) describes this mathematically as a dimension reduction problem: "given the  $p$ -dimensional random variable  $\mathbf{x} = (x_1, \dots, x_p)^T$ , find a lower dimensional representation,  $\mathbf{s} = (s_1, \dots, s_k)^T$  with  $k \leq p$ , that captures the content in the original data, according to some criterion." Fodor goes on to survey numerous linear and non-linear techniques that may be applied to reduce the dimension of high-dimensional data sets. Adopting any of these techniques would provide a principled approach that experimenters could use to identify significant model behaviors from a large collection of model output data. Of course, one wonders whether some techniques are superior to others. Fodor identifies principal components analysis (PCA) as the best (in terms of mean-square error) linear dimension reduction technique.

In this paper, we combine correlation analysis and clustering (CAC) to identify significant behaviors in MesoNet (Mills, Schwartz and Yuan 2010), a network simulation model implemented in SLX<sup>1</sup> (Henriksen 2000). We also use PCA to identify an alternative view of significant behaviors. In applying each method, we identify issues that an analyst must decide and we compare the behaviors identified by the methods. The paper makes three contributions: (1) describes and applies two methods to identify significant behaviors in network simulators, (2) identifies specific judgments that must be made by an analyst when applying each method and (3) compares the dimension reductions achieved by the two methods. The ideas con-

---

<sup>1</sup> Any mention of commercial products within this paper is for information only; it does not imply recommendation or endorsement by NIST.

tained in this paper facilitate effective reduction in response dimension for large simulations and should improve the ability of researchers and practitioners to analyze results from simulation experiments.

The paper is organized in six main sections. In Sec. 2 we explain the general idea underlying dimension reduction for multivariate responses, as applied to simulation outputs. In Sec. 3 we identify and summarize candidate MesoNet responses, showing how those responses form a multivariate dataset. MesoNet is described more fully elsewhere (Mills, Schwartz and Yuan 2010). Sec. 4 applies correlation analysis and clustering (CAC) to identify significant MesoNet behaviors, while also identifying decisions an analyst must make. Sec. 5 applies principal components analysis (PCA), along with key analyst decisions, to produce two alternate views of significant MesoNet responses. In Sec. 6 we compare and contrast the significant behaviors found by CAC and PCA, we discuss pros and cons of the two methods, and we give our reasons for preferring the reduction proposed by CAC. We conclude in Sec. 7.

## 2 DIMENSION REDUCTION FOR MULTIVARIATE RESPONSES

As illustrated in eq. 1, a simulation model can be viewed as a function transforming a set of  $n$  input parameters,  $x_1$  to  $x_n$ , into a set of  $m$  responses,  $y_1$  to  $y_m$ . Each input parameter can take on a range of values, e.g., 1 to  $k$ , defining a parameter space of size  $k^n$ , which can be very large. Elsewhere (Mills and Filliben 2010) we explain methods to provide significant information while simulating only a reduced number ( $2^{n-r}$ ) of parameter combinations. In this paper, we focus on methods to reduce the number of responses that must be analyzed, while still reflecting the most significant model behaviors.

$$y_1, \dots, y_m = \mathbf{f}(x_{1[1, \dots, k]}, \dots, x_{n[1, \dots, k]}) \quad (1)$$

In this paper, we apply two methods that can reduce dimensionality of a model response space: correlation analysis combined with clustering (CAC) and principal components analysis (PCA). First, CAC suggests a reduction in the number of responses from  $m$  to  $k_1$ . We use domain expertise to select  $k_1$  specific responses from among the responses shown in Table 1. Second, PCA suggests a reduction from  $m$  to  $k_2$ . We use heuristics to select  $k_2$  specific responses from Table 1. Subsequently, we compare the specific responses selected by each method.

Table 1: Candidate MesoNet Responses

Response	Definition
y1	Active Sources – sources attempting to transfer data
y2	Proportion of total sources that are active: Active Flows/All Sources
y3	Data packets entering the network per second
y4	Data packets leaving the network per second
y5	Packet Loss Rate: $y_4/(y_3+y_4)$
y6	Flows Completed per second
y7	Flow Completion Rate: $y_6/(y_6+y_1)$
y8	Connection Failures per second
y9	Connection Failure Rate: $y_8/(y_8+y_1)$
y10	Retransmission Rate (ratio)
y11	Average Per Flow Congestion Window (packets)
y12	Average Number of Increases in Congestion Window Per Flow Per Second
y13	Average Number of Negative Acknowledgments Per Flow Per Second
y14	Average Number of Timeouts Per Flow Per Second
y15	Average Round-trip Time (ms)
y16	Relative Queuing Delay ( $y_{15}$ /average propagation delay)
y17	Average Throughput (packets/second) for <b>DD</b> Flows
y18	Average Throughput (packets/second) for <b>DF</b> Flows
y19	Average Throughput (packets/second) for <b>DN</b> Flows
y20	Average Throughput (packets/second) for <b>FF</b> Flows
y21	Average Throughput (packets/second) for <b>FN</b> Flows
y22	Average Throughput (packets/second) for <b>NN</b> Flows

### 3 OVERVIEW OF CANDIDATE MESONET RESPONSES

For purposes of our case study, we selected 22 candidate responses from among those generated by MesoNet, a data communications network simulator. Sixteen responses (y1 – y16), see Table 1, characterize various aspects of network-wide behavior. The flow of data traffic in the network is regulated with the transmission control protocol (TCP), which requires a source and receiver to exchange connection packets, after which the source sends a flow of data packets to a receiver, who responds with acknowledgments. Inability to exchange connection packets prior to a deadline causes a connection attempt to fail. For successful connections, the rate of transmission on a flow is controlled by a congestion window (*cwnd*) that defines the number of packets a source may transmit before receiving an acknowledgment. The time between sending a data packet and receiving an acknowledgment is known as the round-trip time, which is composed of propagation delays on network links plus queuing delays incurred when waiting in routers for transmission. In general, TCP flows begin with a low *cwnd* and then increase the *cwnd* as acknowledgments arrive. Failure to receive acknowledgments leads to a timeout, which causes a source to retransmit unacknowledged data packets. Similarly, receipt of a negative acknowledgment, indicating a specific packet loss, will also stimulate retransmission by a source. Responses y5-y16 capture most aspects of these processes involved in transmitting data between sources and receivers. Responses y1-y4 deal with macroscopic load on the network: the number (and proportion) of sources transmitting data packets and the rate at which data packets enter and exit the network.

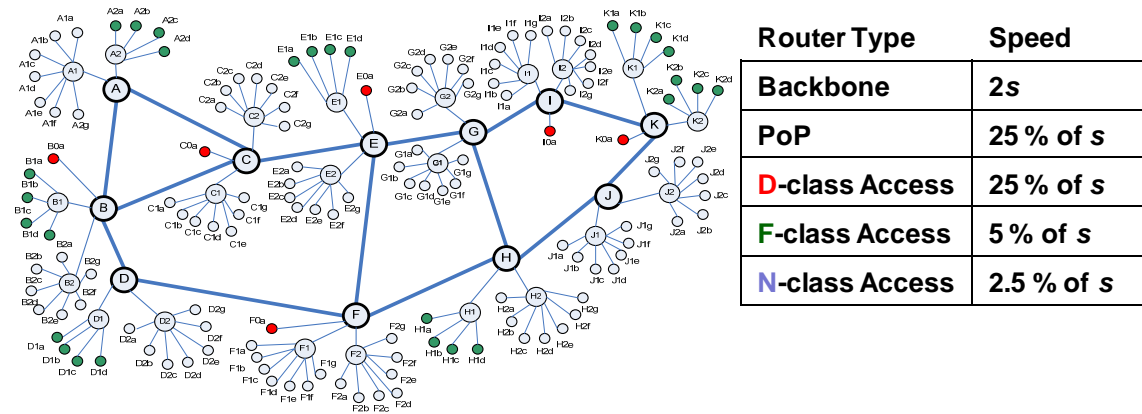


Figure 1: Example MesoNet Topology and Relative Router Speeds

The remaining six responses (y17 – y22) measure the average throughput achieved on flows transiting specific types of paths in a network topology, such as illustrated in Fig. 1. MesoNet topologies consist of three router tiers – backbone, point of presence (PoP) and access – providing transit paths for sources and receivers connected in a fourth tier. The speed of various router classes can be computed relative to a network speed,  $s$ , which is one of the input parameters of MesoNet. Access routers, which constrain transmission capacity available to sources and receivers, can be one of three types: (1) **D**-class, connecting directly to the network backbone and operating as fast as POP routers, (2) **F**-class, sharing a POP router with three other access routers, and (3) **N**-class, sharing a POP router with six other access routers and operating at the slowest speed among all access routers. Flows enter and exit the topology at access routers, thus access routers may be used to classify flows into **DD**, **DF**, **DN**, **FF**, **FN** and **NN** flows, where the flow class denotes the potential throughput (e.g., **DD** flows can achieve higher throughput than **DF** flows and so on).

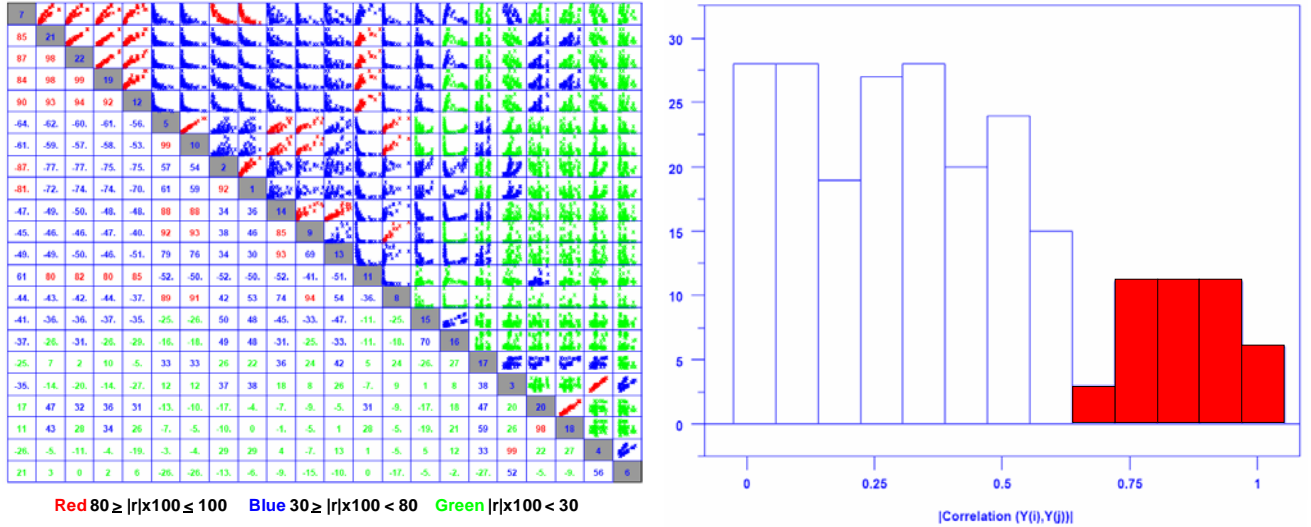
Run	y1	y2	...	y21	y22
1	4680.619	0.168126	...	92.034	89.785
2	6654.512	0.239371	...	72.596	57.738
3	9431.405	0.339259	...	29.569	13.963
4	11565.81	0.415439	...	23.427	19.882
...	...	...	...	...	...
61	10319.55	0.247471	...	87.969	41.573
62	1738.469	0.093668	...	159.298	161.602
63	1783.509	0.096094	...	148.395	161.36
64	21467.6	0.514811	...	26.159	9.981

Figure 2: Sample (Partial) Multivariate Dataset Reflecting 22 Responses for each of 64 MesoNet Simulations

We generated a 64 x 22 multivariate dataset (Fig. 2) by running 64 simulations, where each simulation consisted of different combinations of values for 11 of the 20 MesoNet parameters. Values for the remaining nine parameters were fixed in all runs. For each simulation we measured the 22 responses given in Table 1. We used a  $2^{11-5}$  orthogonal fractional factorial design (Box, Hunter and Hunter 2005) to ensure that we simulated a balanced arrangement of parameter combinations. Next we apply two different methods to reduce the dimension of the data in Fig. 2.

#### 4 REDUCTION VIA CORRELATION ANALYSIS AND CLUSTERING

As a preliminary step, we computed the correlation coefficients ( $r$ ) for each pair of the 22 responses, and summarized the results in a matrix (Fig. 3a), which gives 231 pair-wise scatter plots in cells above and to the right of the diagonal and corresponding correlation coefficients (scaled  $\times 100$ ) below and to the left of the diagonal. We color the scaled correlations: above 80 red, below 30 green and intermediate values blue. We order the matrix by decreasing value of mean correlation for each response. In Fig. 3a, response y7 (gray cell labeled 7 in the upper left hand corner) has largest mean correlation and response y6 (gray cell labeled 6 in lower right hand corner) has smallest mean correlation. Cells on the diagonal identify the response related to cells in the corresponding row and column. For example, consider the gray cell labeled 13 (response y13) in the middle of Fig. 3a. The column of cells moving upward give scatter plots for y13 and each of the 11 responses (y9 – y7) higher on the diagonal; the row moving leftward gives the corresponding correlation coefficients. The row of cells moving rightward from 13 give scatter plots for y13 and each of the 10 responses (y11 – y6) lower on the diagonal; the column moving downward gives the corresponding correlation coefficients. Given pair-wise correlations, an analyst must decide which pairs to include in further analysis and which pairs to discard. To help with this decision, we plot a frequency distribution (Fig. 3b) of the absolute values,  $|r|$ , of correlation coefficients for all response pairs. We use the frequency plot to select a threshold for correlations to consider further. Here, we chose  $|r| > 0.65$  because the histogram shows a notable change in pattern above that value, appearing as a separate (sub)distribution of 42 pair-wise correlations centered on a different mode.



(a) Pair-wise Correlation Matrix

(b) Histogram: bins where  $|r| > 0.65$  highlighted in red

Figure 3: (a) Pair-wise correlation matrix: scatter plots above diagonal, correlation coefficient ( $r \times 100$ ) below diagonal, response number on the diagonal and (b) Frequency distribution of  $|r|$  for pair-wise correlations

Next, we construct an index-index plot, where both the x and y axes indicate the index of corresponding responses (1 – 22). We plot a blue dot for each of the 42  $y_{ij}$  pairs where  $|r| > 0.65$ . We identify the response that is correlated with the most other responses and create a self-contained subset from those responses. We then repeat the process for those responses remaining outside the subset, forming a second self-contained subset. We continue repeating the process until all responses have been allocated to a subset. Subsequently, we reorder the axes of the index plot so that response identifiers are arrayed in increasing order of the cardinality of the subset of which they are a member. Response identifier within each subset are ordered arbitrarily. Fig. 4, which shows the resulting sorted index-index plot generated from the 42 correlation pairs selected

from Fig. 3, reveals five subsets (or clusters) of mutually correlated responses, and two responses that were not correlated with any others. The largest cluster (25 correlation pairs) includes responses that reflect network congestion. The second largest group (14 correlation pairs) includes responses reflecting packet losses. Three pair-wise correlations reflect: (1) throughput on flows constrained by **F**-class access routers, (2) network delay and (3) packets entering and leaving the network. One uncorrelated response (y17) measures average throughput on **DD** flows, while the other (y6) measures flows completed per second. We summarize these response groupings under the CAC half of Table 2, which identifies seven significant behaviors, a reduction of 15 dimensions. For each behavior that includes multiple responses, an analyst must select one response to represent the behavior.

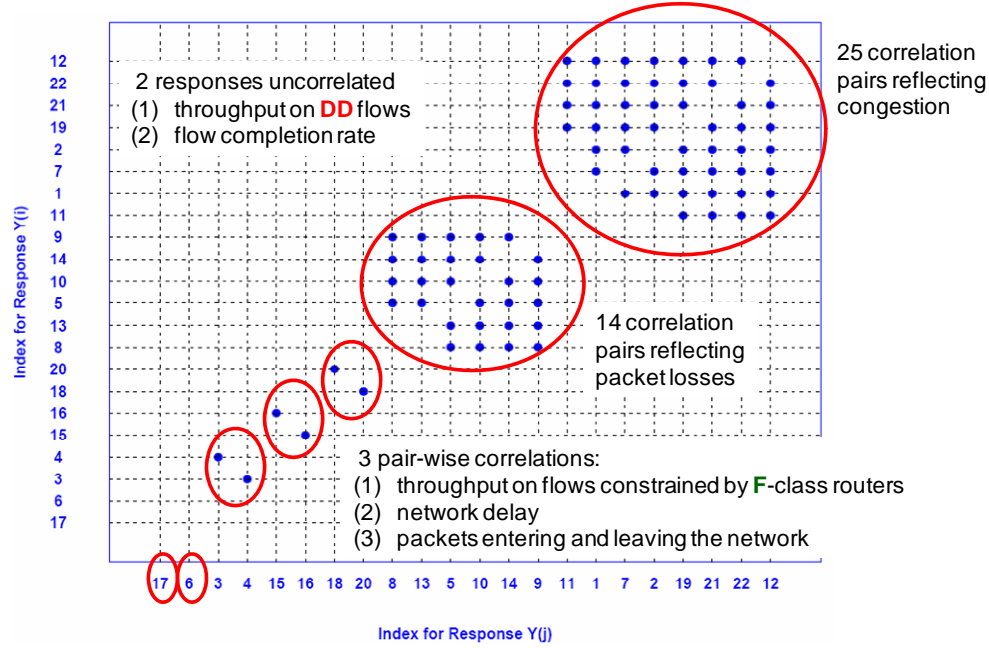


Figure 4: Index-Index plot sorted by increasing count of correlation pairs to identify clusters of mutual correlations

Table 2: Response Groupings by Correlation Analysis and Clustering (CAC) and by Principal Components Analysis (PCA)

CAC		PCA	
Dimension	Responses	Dimension (PC)	Responses
Congestion	y1, y2, y7, y11, y12, y19, y21, <b>y22</b>	Congestion	y1, y2, y5, y7, y8, y9, y10, y11, y12, y13, y14, y19, y21, y22
Losses	y5, y8, y9, <b>y10</b> , y13, y14	Delay	y15, y16
Delay	<b>y15</b> , y16	<b>D</b> -class & <b>F</b> -class Throughput	y3, y4, y17, y18, y20
<b>F</b> -class Throughput	y18, <b>y20</b>	Flows Per Second	y3, y4, y6
<b>D</b> -class Throughput	<b>y17</b>		
Packet Throughput	y3, <b>y4</b>		
Flows Per Second	<b>y6</b>		

We used domain-specific reasoning to select the responses highlighted in red bold under the CAC column of Table 2. Regarding congestion, response y22 measured throughput on the most numerous flows in the simulation, which transited the most congested paths, so all aspects of network congestion were likely to influence this response. Regarding losses, response y10 measured the retransmission rate on all data packets, which would be caused by losses of data, acknowledgments and negative acknowledgments, and so would be approximately twice the loss rate (y5) on data packets. Negative acknowledgments (y13) would be stimulated by lost data packets but could also be lost, as could acknowledgments. Timeouts (y14) would occur only for severely disrupted packet flows. Connection failures (y8 and y9) would measure only the influence of lost connection-establishment packets. For these reasons, we decided that response y10 should give the most comprehensive reflection of packet losses. Regarding delay, the smoothed round-trip time (y15) would be influenced by both propagation and queuing delay, while response y16 reflects only queuing delay, so we selected y15 as a more comprehensive measure of delay. Regarding **F**-class throughput, we selected response y20 (throughput on **FF** flows) because the related flows do not transit any **N**-class or **D**-class access routers, which should allow **FF** flows to give the best representation of **F**-class through-

put. Regarding packet throughput, we reasoned that packets could not leave the network (y4) unless they were injected (y3), so we decided that y4 could represent both the rate of packets injected and influence of other factors on the rate of packets leaving the network. Responses y17 and y6 each represent an additional dimension.

The PCA half of Table 2 indicates the responses grouped by PCA into the first four principal components, as explained below in Sec. 5. As we will show, selecting a response to represent each PC relies on a choice among heuristics that have no specific domain interpretation. For this reason, we will compare PC response selections made by two different heuristics. We will also compare the responses we selected from the CAC half of Table 2 against responses representing PCs, as identified by the two heuristics we used.

## 5 REDUCTION VIA PRINCIPAL COMPONENTS ANALYSIS

Principal Components Analysis (PCA) transforms a set of possibly correlated variables (such as the 22 variables identified in Table 1 and analyzed above in Sec. 4) into a smaller number of uncorrelated variables called principal components (PCs), which consist of orthogonal linear combinations of the original values that yield the largest variance. PCA requires an iterative procedure performed on a normalized (zero mean and unitary standard deviation) version of the original dataset. The procedure first identifies a weighted linear combination of normalized data values that accounts for the largest variance in the data, and then proceeds to identify a weighted linear combination of the remaining normalized data values accounting for the largest variance, and so on. Fodor (2002) provides a mathematical explanation. The result of the process yields a square matrix (22 x 22 in our case) where the first column contains the weight vector for PC1, the second column contains the weight vector for PC2 and so on (up to PC22 in our case). The process also yields a vector containing the standard deviation accounted for by each PC. Table 3 shows (a) the standard deviation vector and (b) part of the matrix of weight vectors computed for our case study.

Table 3: (a) standard deviation vector resulting from PCA in our case study, along with a cumulative distribution of % variance and (b) elided matrix of vectors (one per PC), where each vector gives the weight assigned to each of the 22 responses

(a) Distribution of Variance			(b) Weight Vector Matrix						
PC	Std. Dev.	Cumulative %	Response	PC1	PC2	PC3	PC4	...	PC22
PC1	9.7091	0.441325	y1	0.2581	0.1421	-0.1667	0.1126	...	-0.0236
PC2	4.0161	0.62388	y2	0.2608	0.1768	-0.1407	0.1058	...	-0.0002
PC3	3.2322	0.77079	y3	0.0864	0.0218	-0.4163	-0.3909	...	0.7017
PC4	2.0630	0.86457	y4	0.0435	0.0617	-0.4252	-0.4037	...	-0.6919
PC5	0.9716	0.90873	y5	0.2774	-0.2276	0.0363	0.0644	...	-0.0601
PC6	0.7585	0.94321	y6	-0.039	0.0914	-0.1246	-0.5636	...	0.0078
PC7	0.4537	0.96383	y7	-0.284	-0.1358	0.1004	-0.0815	...	-0.0072
PC8	0.2569	0.97551	y8	0.2261	-0.2452	0.0444	0.0589	...	-0.0472
PC9	0.1835	0.98385	y9	0.2358	-0.2775	0.0635	0.0236	...	0.0128
PC10	0.1254	0.98955	y10	0.2706	-0.2395	0.0344	0.0704	...	0.0401
PC11	0.0588	0.99222	y11	-0.245	-0.0838	-0.1081	0.0797	...	0.0026
PC12	0.0504	0.99451	y12	-0.283	-0.167	-0.0036	0.0441	...	-0.0061
PC13	0.0360	0.99615	y13	0.2241	-0.2678	-0.0174	-0.1225	...	-0.0005
PC14	0.0286	0.99745	y14	0.2387	-0.2965	0.0218	-0.0815	...	-0.0154
PC15	0.0225	0.99847	y15	0.0429	0.4426	-0.0415	0.1868	...	0.01
PC16	0.0113	0.99899	y16	0.0457	0.3242	-0.2348	0.3085	...	0.0236
PC17	0.0099	0.99944	y17	0.0682	-0.2008	-0.3694	0.2058	...	-0.0279
PC18	0.0060	0.99971	y18	-0.064	-0.1636	-0.4271	0.2485	...	0.0531
PC19	0.0032	0.99985	y19	-0.287	-0.1679	-0.0805	0.024	...	0.0433
PC20	0.0019	0.99994	y20	-0.084	-0.1519	-0.3983	0.2356	...	-0.062
PC21	0.0011	0.99999	y21	-0.29	-0.1716	-0.0989	0.0413	...	0.0704
PC22	0.0002	1.00000	y22	0.2581	-0.1667	-0.0332	0.0308	...	-0.074

Given the matrix of weights and the vector of standard deviations, an analyst must choose how many PCs to use. An analyst might also decide which variables should be retained in each PC, because PC expressions may be simplified by ignoring the (low weight) contributions of certain of the variables. If the retained set of variables for the chosen PCs includes all or



most of the variables, then an analysis might seek to further reduce the set of variables to include only a handful of the most important. To choose the number of PCs to use, one credible source (Jolliffe 2002) suggests including the first  $n$  PCs that account for between 70 % and 90 % of the total variance in the data. Examining the cumulative % column of Table 3a suggests that the first four PCs (red) should be used (as we did in Table 2) and that PC5 (orange) might be used. We highlight PC6 and PC7 (purple), which lie beyond the range of the inclusion heuristic, because the CAC suggested seven dimensions were required to represent system behavior. In Sec. 6, when comparing the responses we selected using CAC against the responses recommended by PCA, we will consider PC1 through PC7, though the PCA inclusion heuristic suggests that only the first four or five PCs capture sufficient variation in the response data.

We tried several different statistical heuristics to prune variables from PCs. While some provided more clear cut results than others, all the heuristics generally pointed toward the same outcome. We obtained the most clear cut results using the following heuristic: (1) normalize the weights to zero mean and unit standard deviation, (2) take the absolute value of the normalized weights, (3) scale the results to the range 0 to 1, (4) plot the results, (5) visually identify a separating threshold and (6) use responses plotted above the threshold. Fig. 5 shows the application of this heuristic to the first four PCs, yielding the outcome shown in the PCA half of Table 2. What remains is to select specific responses to represent the PCs of interest.

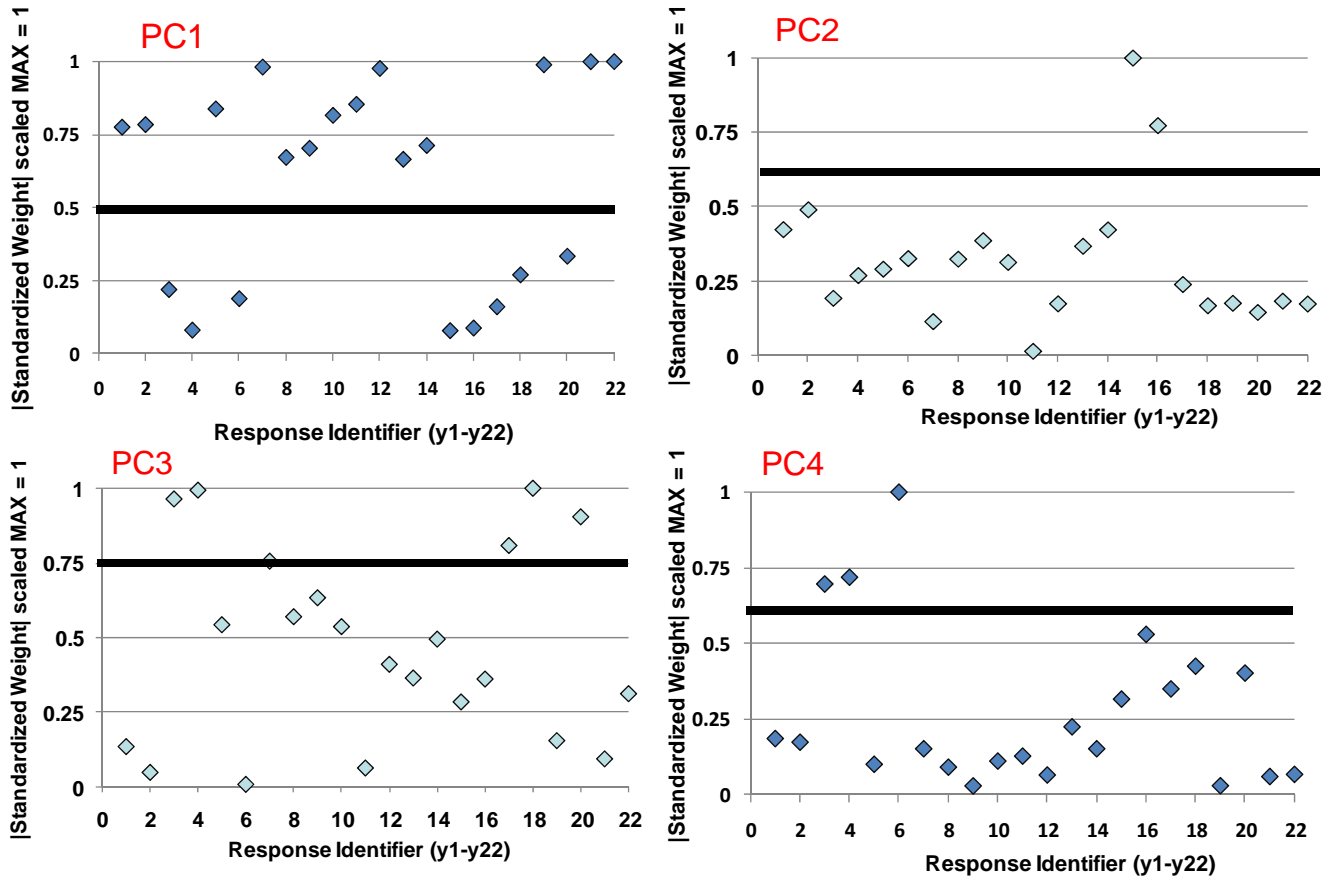


Figure 5: Plots of normalized and scaled weight vectors (y axis) against response variable identifiers (x axis) for the first four PCs. Each plot includes the visually selected threshold (black horizontal line) used to identify responses to include (above the line) and exclude (on or below the line) for each PC. For each plot, responses with normalized and scaled weights above the selected threshold were included for the corresponding PC in the PCA half of Table 2.

While one could choose to base subsequent data analyses on the first four PCs, interpretation of such analyses can be difficult because PCs do not necessarily represent any distinguishable domain concepts (though we attempt to supply a mapping to domain concepts in Table 2). As an alternative to using the first four PCs, we can instead select a subset of the variables. Jolliffe (2002) suggests that the number of variables to select should equal the number of PCs used, or perhaps a few more. Since the PCA analysis identified that we should use four or five PCs, we decided to select seven variables, which matches the number of variables we selected using CAC. Selecting specific variables requires an additional heuristic. We tried two

particular approaches in order to compare the outcomes. One algorithm, which we denote as MKB, described by Mardia, Kent and Bibby (1995), iterates over the weight vectors from the least significant (PC22 here) to the most significant (PC1). When examining each weight vector, the response corresponding to the weight with the largest magnitude is discarded from further consideration. This process continues until the remaining number of responses corresponds to the number (seven here) of variables sought. This seemingly counterintuitive algorithm appears to be based on the assumption that responses that contribute large weights to insignificant PCs must not be particularly salient with respect to the dataset. The second algorithm we used, which we denote here as JJF, was derived by inverting the MKB algorithm. The JJF algorithm iterates over the weight vectors from the most significant to the least significant. When examining each weight vector, the variable corresponding to the weight with the largest magnitude is withdrawn from further consideration. This process continues until the cardinality of the set of withdrawn responses equals the number of variables sought. The set of withdrawn responses constitutes the responses to be used in subsequent analyses. Table 4 shows the results of applying these two algorithms. The table also shows the seven responses we selected based on CAC. We compare and contrast these results in the next section.

Table 4: Two columns give the ordering of response significance identified from PCA weight vectors using two heuristics. One column repeats, for comparison, the specific responses identified by a domain analysis to represent each of the seven clusters of correlated responses using CAC. The first four responses (red) correspond to the number suggested by analysis of the cumulative distribution of the proportion of PC variance. The fifth response (orange) corresponds to a fifth PC, which was on the borderline for inclusion. The sixth and seventh responses (magenta) correspond to the two additional dimensions identified by CAC.

PC	JJF Heuristic	MKB Heuristic	CAC Domain Analysis
PC1	Y22 (NN flow TP)	Y4 (packets output)	Y22 (NN flow TP)
PC2	Y15 (SRTT)	Y19 (DN flow TP)	Y15 (SRTT)
PC3	Y18 (DF flow TP)	Y18 (DF flow TP)	Y4 (packets output)
PC4	Y6 (flows completed TP)	Y6 (flows completed TP)	Y6 (flows completed TP)
PC5	Y11 (CWND)	Y9 (connection failure rate)	Y10 (retransmission rate)
PC6	Y17 (DD flow TP)	Y17 (DD flow TP)	Y20 (FF flow TP)
PC7	Y16 (queuing delay)	Y16 (queuing delay)	Y17 (DD flow TP)
PC8	Y13 (NAKs)	Y13 (NAKs)	N/A
PC9	Y1 (active sources)	Y11 (CWND)	N/A
PC10	Y14 (timeouts)	Y15 (SRTT)	N/A
PC11	Y2 (% sources active)	Y7 (flow-completion rate)	N/A
PC12	Y9 (connection failure rate)	Y2 (% sources active)	N/A
PC13	Y10 (retransmission rate)	Y10 (retransmission rate)	N/A
PC14	Y8 (connection failures)	Y8 (connection failures)	N/A
PC15	Y7 (flow-completion rate)	Y14 (timeouts)	N/A
PC16	Y12 (CWND increases)	Y12 (CWND increases)	N/A
PC17	Y21 (FN flow TP)	Y1 (active sources)	N/A
PC18	Y20 (FF flow TP)	Y21 (FN flow TP)	N/A
PC19	Y19 (DN flow TP)	Y20 (FF flow TP)	N/A
PC20	Y4 (packets output)	Y22 (NN flow TP)	N/A
PC21	Y5 (loss rate)	Y5 (loss rate)	N/A
PC22	Y3 (packets input)	Y3 (packets input)	N/A

## 6 COMPARING DIMENSION REDUCTION METHODS

In Sec. 5, applying a heuristic threshold to the cumulative distribution of the proportion of variance led us to conclude that the first four PCs accounted for most of the significant variance in our dataset. As shown in Table 2, we were able to develop a domain interpretation of PC1 through PC4<sup>2</sup>, and we could relate that interpretation to the seven dimensions identified using

<sup>2</sup> We were unable to devise a domain interpretation for PC5 – PC7, which provides additional evidence that only the first four PCs should be used in this case study.



CAC. The results show relatively good alignment: PCA merged congestion and losses into a single dimension and merged throughput on flows constrained by either **D**-class or **F**-class routers. On the other hand, PCA spread packet throughput among two PCs, rather than retaining it as a unique dimension. In related work (Mills, Filliben, Cho, Schwartz and Genin 2010), guided by analysis of the responses grouped into PC1 – PC4, we were able to analyze and interpret the PCs themselves in one situation. In a second situation, we were unable to construct a domain interpretation of even the top four PCs. In situations where PCs cannot be interpreted successfully, one can resort to substituting a set of responses for the PCs. Here, guided by Table 4, we compare the responses identified by the BKM and JJF heuristics against each other and against the responses selected based on a domain analysis of the CAC results. We consider up to seven responses because CAC identified the need for seven dimensions to cover the response dataset.

The top four responses selected by the JJF heuristic can be interpreted to represent congestion (y22), delay (y15), **D**- and **F**-class throughput (y18) and flows completed per second (y6). Under such an interpretation, the JJF heuristic maps nicely to the first four PCs of Table 2. Indeed, three of the first four responses selected by the JJF heuristic matched responses selected by domain analysis of the CAC results. The top four responses selected by the MKB heuristic can be interpreted to represent congestion (y19), **D**- and **F**-class throughput (y18) and flows completed per second (y6). Interpreting the remaining response (y4) is more problematic because y4 appears in two PCs in Table 2, and both PCs have already been covered by other responses. Considering only the top four responses, the JJF heuristic gave results that were better aligned with the CAC results.

Expanding our comparison with the CAC results, we consider additional responses (five through seven) produced by each heuristic. The MKB heuristic identifies connection failure rate (y9), which can be interpreted as representing losses, **DD**-flow throughput (y17), which can be interpreted as representing **D**-class throughput, and y16, which can be interpreted as representing delay. In addition, as its first response, the MKB heuristic identifies y4, which corresponds to packet throughput. Under this interpretation, the top seven responses identified by the MKB heuristic correspond with the seven dimensions identified by CAC. The JJF heuristic also identifies y17 (**D**-class throughput) and y16 (delay) in its top seven responses. However, the JJF heuristic had already included a response (y15) representing delay, so y16 appears redundant. In addition, the JJF heuristic includes response y11 (congestion window size), which can be interpreted to provide a duplicate response representing congestion. Given the two duplicates, the JJF heuristic does not include responses representing losses or packet throughput. Of course, the PCA grouped losses together with congestion and spread packets output across two PCs, so the results of the JJF heuristic are consistent with the PCA response groupings, but not consistent with the seven CAC dimensions. The MKB heuristic appears consistent with the seven CAC dimensions. On the other hand, considering only the top four responses, the JJF heuristic provides a better correspondence with the CAC analysis.

The PCA method provided greater dimension reduction (22→4) than CAC (22→7). On the other hand, because PCs are uncorrelated variables created from a set of possibly correlated variables, PCA guarantees no obvious domain interpretation of even the top 2 or 3 PCs<sup>3</sup>, though in the case discussed in this paper we were able to arrive at a reasonable interpretation. In other cases (not described here) we were unable to infer any convincing domain interpretation. Even when a reasonable domain interpretation is possible, PCs may take on both positive and negative values for which domain analysts cannot determine any obvious interpretation, even after establishing a meaning for a PC itself. For example, based on the specific responses that showed significant weights for PC1, we inferred that the PC represented congestion (including its influence on packet losses). Determining whether positive or negative values for the PC represented higher or lower congestion proved quite challenging (Mills, Filliben, Cho, Schwartz and Genin 2010). To make such a determination we had to compare the parameter settings causing positive and negative PC values with the parameter settings for individual responses that were heavily weighted in the PC. Since we could easily interpret the domain meaning of the concrete values produced for the responses, we were usually able to infer some seemingly plausible mapping to positive and negative PC values. While in some situations such chains of inference may be possible, we remain uneasy about the validity of the resulting interpretations. To overcome these limitations with respect to PCs, we adopted two different heuristics to identify responses to substitute for the most significant four PCs. One heuristic (JJF) identified four responses that were within the set of seven responses identified using CAC, and identified a fifth response that could be mapped to a response identified using CAC. The other heuristic (MKB) identified four responses that were within the set of seven responses identified using CAC, and identified seven responses that could be mapped to responses identified using CAC. We have no rigorous criteria for judging which of the two heuristics produced results better aligned with the CAC results. Further, the general success of such heuristics is not guaranteed. In fact, we suspect that such heuristics might prove less successful when no convincing domain interpretation can be established for the PCs.

The CAC method provided effective dimension reduction through correlations that could be vetted easily by domain experts. Further, examining pair-wise response correlations helped to validate that MesoNet provides reasonable behavior as a simulation model for data communications networks. The CAC method allowed us to uncover the nuanced difference be-

<sup>3</sup> Note, however, that PC1 will almost always represent some kind of “global average”.

tween network-wide throughput in terms of packets per second versus flows completed per second, a difference that might otherwise have been overlooked. In addition, the CAC method allowed us to discern significant differences in factors influencing the throughput of **DD** flows, as compared to other **D**- and **F**-class flows. These differences were masked by the coarseness of the PCs identified using PCA. On the other hand, when CAC was applied to analyze data from a second simulation experiment (Appendix C in Mills, Filliben, Cho, Schwartz and Genin 2010) using the same parameters set to different values, the method identified some differences in response correlation, when compared with those shown in this case study. While these differences could be justified by domain analysis, the existence of such differences indicates that separate correlation analyses must be generated and examined for each set of simulation experiments conducted. We found this also true for PCA.

In the end, the CAC method proved more useful to domain analysts seeking to understand output from the MesoNet simulation. The CAC method identified and clustered correlated variables, whereas the PCA method constructed artificial uncorrelated variables that did not necessarily have a domain interpretation. The clusters identified by CAC uncovered some nuanced differences in network behavior, differences concealed within the coarser PCs produced by PCA. Further, the clustered correlated variables identified by CAC could be used in validating that MesoNet produced behavior reasonably aligned with a real network. The PCs produced by PCA could not be used for model validation. In addition, in applying CAC, an analyst need only make one heuristic decision: the threshold for correlation values to include in the clustering step. Domain analysis is used to select which response to adopt as a surrogate for each cluster containing multiple responses. In applying PCA, an analyst needs heuristics to decide: (1) which PCs to include, (2) which responses should be included within each PC, (3) whether PCs should be analyzed directly or whether surrogate responses should be selected, and (4) if surrogate responses are needed, how to identify which responses to use. The necessity to make four heuristic decisions (vs. one heuristic decision for CAC) leads to greater burden for an analyst, provides increased room for error and necessarily affects the repeatability of the PCA method.

## 7 CONCLUSIONS

We investigated CAC and PCA as two techniques to reduce the dimension of multivariate response data produced from simulation models. We applied each method to reduce a 22-dimensional response space generated by MesoNet, a simulator for data communications networks. While the PCA method usually suggests a smaller number of dimensions, we found that the CAC method distinguishes dimensions that better correspond with a domain analyst's understanding of the domain. Further, the results from CAC are much easier for a domain analyst to interpret, which aids in model validation. On the other hand, PCA and CAC appear to be reasonably complementary methods that, applied together, can provide alternate views of a multivariate data set. Comparing dimensions identified by both methods (as we did, for example, in Table 2) should provide a better understanding of responses from simulation experiments. Where only one method is used, we suggest that CAC provides response groupings that are straightforward for a domain analyst to interpret. Using only PCA leads to response groupings that are coarse, abstract and often difficult for a domain analyst to understand. No matter which method is applied, each new data set generated by different experiments with a given simulation model must be subjected to a separate analysis and interpretation because correlations and PCs may change with changes in parameter values.

## REFERENCES

- Box, G., J Hunter, and W. Hunter, 2005. *Statistics for Experimenters*. 2<sup>nd</sup> ed. Hoboken, New Jersey: Wiley.
- Fall, K. and K. Varadhan, eds. 2009. The *ns* Manual. Available via [http://www.isi.edu/nsnam/ns/doc/ns\\_doc.pdf](http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf) [accessed December 2, 2009].
- Fodor, I.K. 2002. A Survey of Dimension Reduction Techniques. Lawrence Livermore National Laboratory Technical Report no. UCRL-ID-148494.
- Henriksen, J.O. 2000. SLX: the X is for extensibility. In *Proceedings of the 32<sup>nd</sup> Winter Simulation Conference*, 183-190.
- Jolliffe, I. T. 2002. *Principal Components Analysis*. 2<sup>nd</sup> ed. Springer Series in Statistics.
- Mardia, K.V., Kent, J.T., and Bibby J.M. 1995. *Multivariate Analysis*. Probability and Mathematical Statistics. Academic Press.
- Mills, K. and J. Filliben. 2010. An Efficient Sensitivity Analysis Method for Network Simulation Models. Technical Report CxS-IMS-2010-2. NIST.
- Mills, K., J. Filliben, D. Cho, E. Schwartz and D. Genin. 2010. *Study of Proposed Internet Congestion-Control Mechanisms*. NIST Special Publication 500-282.
- Mills, K., E. Schwartz and J. Yuan. 2010. How to model a TCP/IP network using only 20 parameters. Submitted to the 41<sup>st</sup> Winter Simulation Conference.

- Paxson, V. and S. Floyd. 1997. Why we don't know how to simulate the Internet. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradottir, K. J. Healy, D. H. Withers, and B. L. Nelson, 1037-1044.
- Riley, G., M. Ammar, F. Fujimoto, A. Park, K. Perumalla and D. Xu. 2004. A Federated Approach to Distributed Network Simulation. In *ACM Transactions on Modeling and Computer Simulation*, 14:2, 116-148.
- SSFNet. 2009. How to use SSFNet. Available via <http://www.ssfnet.org/internetPage.html> [accessed December 2, 2009]
- Tan, K., J. Song, Q. Zhang and M. Sridharan. 2006. A Compound TCP Approach for High-Speed and Long Distance Networks. In *Proceedings of the 25<sup>th</sup> Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2006)*, pp. 1-12
- Tyan, H-Y., A. Sobeih and J. Hou. 2009. Design, Realization and Evaluation of a Component-based, Compositional Network Simulation Environment. In *Simulation*, 85:3, 159-181.
- Yaun, G., D. Bauer, H. Bhutada, C. Carothers, M. Yukel and S. Kalyanaraman. 2003. Large-Scale Network Simulation Techniques: Examples of TCP and OSFP Models. In *SIGCOM Computer Communications Review*, 33:3, 27-41.
- Zeng, X., R. Bagrodia and M. Gerla. 1998. GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks. In *Proceedings of the 12<sup>th</sup> Workshop on Parallel and Distributed Simulations*, 154-161.

## **AUTHOR BIOGRAPHIES**

**KEVIN L. MILLS** is a Senior Research Scientist at the U.S. National Institute of Standards and Technology (NIST). He received his Ph.D. in information technology from George Mason University (GMU). From 1996 to 2006 he served on the adjunct faculty of the department of computer science at GMU. From 1996 to 1999 he was a program manager at the Defense Advanced Research Projects Agency (DARPA). His research interests include complex distributed systems. His email is [kmills@nist.gov](mailto:kmills@nist.gov).

**JAMES J. FILLIBEN** is a Mathematical Statistician at NIST. He received his Ph.D. in statistics from Princeton University. He has collaborated with dozens of NIST scientists in a wide range of disciplines, for which and his achievements have been recognized with Gold, Silver and Bronze medals awarded by the U.S. Department of Commerce. His research interests include statistical experiment design and exploratory analysis of multidimensional data. His email is [james.filliben@nist.gov](mailto:james.filliben@nist.gov).