EEC 686/785
Modeling & Performance Evaluation of Computer Systems

Lecture 10

Wenbing Zhao
Department of Electrical and Computer Engineering
Cleveland State University
wenbing@ieee.org
(based on Dr. Raj Jain's lecture notes)

Outline

- Midterm#1 grade
- Workload Characterization Techniques

Midterm #1

P1	P2	P3	P4	P5	P6	P7	P8	Total
14	14	10	7	7.5	10	10	7.5	80
18	10	10	4.5	7.5	9	10	9.5	78.5
18	10	10	4.5	7.5	10	10	8	78
12	16	10	1.5	5	10	5	4.5	64
12	8	10	3	5	10	5	8.5	61.5
8	10	10	0	10	10	10	3.5	61.5

Workload Characterization Techniques

- **Workload characterization**: the process of studying the real-user environments, observe the key characteristics, and develop a workload model that can be used repeated
- The measured workload data consists of services requested or the resource demands of a number of users on the system
- The term "**user**" denotes the entity that makes the service requests at the SUT interface

Workload Characterization Techniques

- In workload characterization literature, the term **workload component** or **workload unit** is used instead of the user
- Workload component
  - Applications
  - User sessions
  - Sites
- **Workload parameters** or **workload features**
  - Measured quantities, service requests, or resource demands
  - For example: transaction types, instructions, packet sizes, source-destinations of a packet, and page reference pattern

Components and Parameter Selection

- The **workload component** should be at the SUT interface
- Each component should represent as homogeneous a group as possible. Combining very different users into a site workload may not be meaningful
- Domain of the control affects the component
  - Example: mail system designer are more interested in determining a typical mail session than a typical user session
- Do not use parameters that depend upon the system, e.g., the elapsed time, CPU time

Components and Parameter Selection

- Characteristics of service requests
  - Arrival time
  - Type of request or the resource demanded
  - Duration of the request
  - Quantity of the resource demanded, for example, pages of memory
- Exclude those parameters that have little impact

Workload Characterization Techniques

- Averaging
- Single-parameter histograms
- Multiparameter histograms
- Principal component analysis
- Markov models
- Clustering

Averaging

- Mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Standard deviation $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$
- Coefficient Of Variation (COV): $s / \bar{x}$
- Mode (for categorical variables): most frequent value
- Median: 50-percentile

Characteristics of an Average Editing Session

Single Parameter Histograms

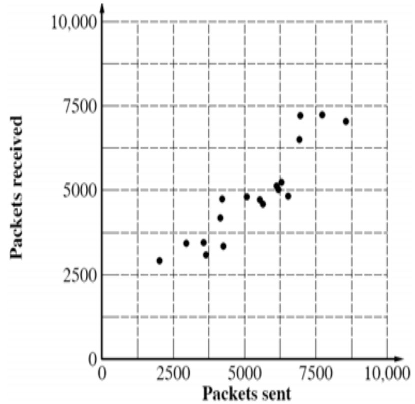
- A histogram shows the relative frequencies of various values of a parameter.
- For continuous-value parameters, this requires dividing the complete parameter range into several smaller subranges called *buckets* or *cells*) and counting the observations that fall in each cell

Single Parameter Histograms

- n buckets × m parameters × k components values
- Use only if the variance is high
- Ignores correlation among parameters

Multiparameter Histograms

- Two-parameter histogram



Principal Component Analysis

- Key idea: use a weighted sum of parameters to classify the components
- Let x_{ij} denote the ith parameter for jth component

$$y_j = \sum_{i=1}^{n} w_i x_{ij}$$

- Principal component analysis assigns weights w_i's such that y_j's provide the maximum discrimination among the components
- The quantity y_j is called the **principal factor**

Principal Component Analysis

- The factors are ordered. First factor explains the highest percentage of the variance.
- Statistically, given a set of n parameters $\{x_1, x_2, \ldots, x_n\}$, the principal-component analysis produces a set of factors $\{y_1, y_2, \ldots, y_n\}$ such that the following holds

Principal Component Analysis

- The y's are linear combinations of x's: $y_i = \sum_{j=1}^{n} a_{ij} x_j$
- Here, a_{ij} is called the loading of variable x_j on factor y_i
- The y's form an orthogonal set, that is, their inner product is zero:

$$< y_i, y_j >= \sum_k a_{ik} x_{kj} = 0$$

- This is equivalent to stating that y_i's are uncorrelated to each other
- The y's form an ordered set such that y_1 explains the highest percentage of the variance in resource demands

Finding Principal Factors

- Find the correlation matrix
- Find the eigenvalues of the matrix and sort them in the order of decreasing magnitude
- Find corresponding eigenvectors. These give the required loadings

Principal Component Example

Principal Component Example

1) Compute the mean and standard deviations of the variables:

$$\bar{x}_s = \frac{1}{n}\sum_{i=1}^{n} x_{si} = \frac{96336}{18} = 5352.0$$

$$\bar{x}_r = \frac{1}{n}\sum_{i=1}^{n} x_{ri} = \frac{88009}{18} = 4889.4$$

$$s_{x_s}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_{si}-\bar{x}_s)^2$$

$$= \frac{1}{n-1}\left[\left(\sum_{i=1}^{n} x_{si}^2\right) - n * \bar{x}_s^2\right]$$

$$= \frac{567119488 - 18 \times 5353^2}{17} = 1741.0$$

Similarly:

$$s_{x_r}^2 = \frac{462661024 - 18 \times 4889.4^2}{17} = 1379.5$$

Principal Component Example

2) Normalize the variables to zero mean and unit standard deviation. The normalized values x'_s and x'_r are given by:

$$x'_s = \frac{x_s - \bar{x}_s}{s_{x_s}} = \frac{x_s - 5352}{1741}$$

$$x'_r = \frac{x_r - \bar{x}_r}{s_{x_r}} = \frac{x_r - 4889}{1380}$$

3) Compute the correlation among the variables:

$$R_{x_s,x_r} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_{si}-\bar{x}_s)(x_{ri}-\bar{x}_r)}{s_{x_s} s_{x_r}} = 0.916$$

Principal Component Example

4) Prepare the correlation matrix

$$\mathbf{C} = \begin{bmatrix} 1.000 & 0.916 \\ 0.916 & 1.000 \end{bmatrix}$$

5) Compute the eigenvalues of the correlation matrix: by solving the characteristic equation:

$$|\lambda I - C| = \begin{vmatrix} \lambda - 1 & -0.916 \\ -0.916 & \lambda - 1 \end{vmatrix} = 0$$

or:

$$(\lambda - 1)^2 - 0.916^2 = 0$$

The eigenvalues are 1.916 and 0.084

Principal Component Example

6) Compute the eigenvectors of the correlation matrix. The eigenvectors $\mathbf{q}_1$ corresponding to λ_1 = 1.916 are defined by the following relationship:

$$\mathbf{C}\mathbf{q}_1 = \lambda_1 \mathbf{q}_1$$

or:

$$\begin{bmatrix} 1.000 & 0.916 \\ 0.916 & 1.000 \end{bmatrix} \times \begin{bmatrix} q_{11} \\ q_{21} \end{bmatrix} = 1.916 \begin{bmatrix} q_{11} \\ q_{21} \end{bmatrix}$$

or:

$$q_{11} = q_{21}$$

Principal Component Example

Restricting the length of the eigenvector to one, the following vector is the first eigenvector

$$\mathbf{q}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Similarly, the second eigenvector is:

$$\mathbf{q}_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Principal Component Example

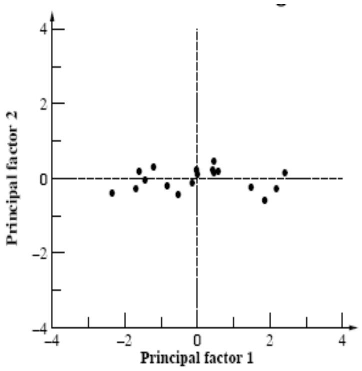
7) Obtain principal factors by multiplying the eigenvectors by the normalized vectors:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{x_s - 5352}{1741} \\ \frac{x_r - 4889}{1380} \end{bmatrix}$$

8) Compute the values of the principal factors (shown in the table)

Principal Component Example

9) Compute the sum and sum of squares of the principal factors
- The sum must be zero
- The sum of squares give the percentage of variation explained
- The first factor explains 32.565/(32.565+1.435) or 95.7% of the variation
- The second factor explains only 4.3% of the variation and can, thus, be ignored



Markov Models

- Markov => the next request depends only on the last request
- Described by a transition matrix

From/To	CPU	Disk	Terminal
CPU	0.6	0.3	0.1
Disk	0.9	0	0.1
Terminal	1	0	0

- Transition matrices can be used also for application transitions. E.g, P(Link|Compile)
- Used to specify page-reference locality. P(Reference module *i* | Referenced module *j*)

Transition Probability

- Given the same relative frequency of requests of different types, it is possible to realize the frequency with several different transition matrices
- If order is important, measure the transition probabilities directly on the real system

Transition Probability

- Examples: two packet sizes: small (80%), large (20%)
  - An average of four small packets are followed by an average of one big packet, e.g., ssssbssssbssss

Current Packet	Next packet Small	Next packet Large
Small	0.875	0.125
Large	0.5	0.5

Current Packet	Next packet Small	Next packet Large
Small	0.75	0.25
Large	1	0

  - Eight small packets followed by two big packets
  - Generate a random number *x*. $x <= 0.8$ => generate a small packet; otherwise generate a large packet

Current Packet	Next packet Small	Next packet Large
Small	0.8	0.2
Large	0.8	0.2

Clustering

- Steps
  - Take a sample, that is, a subset of workload components
  - Select workload parameters
  - Select a distance measure
  - Remove outliers
  - Scale all observations
  - Perform clustering
  - Interpret results
  - Change parameters, or number of clusters, and repeat 3-7
  - Select representative components from each cluster

Sampling

- In one study, 2% of the population was chosen for analysis; later 99% of the population could be assigned to the clusters obtained
- Random selection
- Select top consumers of a resource

Parameter Selection

- Criteria
  - Impact on performance
  - Variance
- Method: redo clustering with one less parameter
- Principal component analysis: identify parameters with the highest variance
- Data transformation may be required

Distance Metric

- Methods: given $\{x_{i1}, x_{i2},\ldots, x_{in}\}$ and $\{x_{j1}, x_{j2},\ldots, x_{jn}\}$
  - Euclidean distance: $d = \left\{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2\right\}^{0.5}$
  - Weighte-Euclidean distance: $d = \sum_{k=1}^{n}\left\{a_k(x_{ik} - x_{jk})^2\right\}^{0.5}$

    Here a_k, k=1,2,…,n are suitably chosen weights for the n parameters
  - Chi-Square distance: $d = \sum_{k=1}^{n}\left\{\frac{(x_{ik} - x_{jk})^2}{x_{ik}}\right\}$

Distance Metric

- Methods (continued):
  - The Euclidean distance is the most commonly used distance metric
  - The weighted Euclidean is used if the parameters have not been scaled or if the parameters have significantly different levels of importance
  - Used Chi-Square distance only if $x_{.k}$'s are close to each other. Parameters with low values of $x_{.k}$ get higher weights

Outliers

- **Outliers** = data points with extreme parameter values
- Affect normalization
- Can exclude only if that do not consume a significant portion of the system resources. Example, backup

Data Scaling

- Normalize to zero mean and unit variance:

$$x'_{ik} = \frac{x_{ik} - \bar{x}_k}{s_k}$$

- Weights:

$$x'_{ik} = w_k x_{ik}$$

$w_k \propto$ relative importance or $w_k = 1/s_k$

- Range normalization (affected by outliers):

$$x'_{ik} = \frac{x_{ik} - x_{min,k}}{x_{max,k} - x_{min,k}}$$

- Percentile normalization:

$$x'_{ik} = \frac{x_{ik} - x_{2.5,k}}{x_{97.5,k} - x_{2.5,k}}$$

Clustering Techniques

- Goal: partition into groups so the members of a group are as similar as possible and different groups are as dissimilar as possible
- Statistically: the intragroup variance should be as small as possible, and intergroup variance should be as large as possible

Total Variance = Intragroup Variance + Intergroup Variance

Clustering Techniques

- Nonhierarchical techniques:
  start with a arbitrary set of *k* clusters, move members until the intragroup variance is minimum
- Hierarchical techniques:
  - **Agglomerative**: start with *n* clusters and merge
  - **Divisive**: start with one cluster and divide
- Two popular techniques
  - Minimum spanning tree method (agglomerative)
  - Centroid method (division)

Minimum Spanning Tree-Clustering Method

1. Start with *k*=*n* clusters
2. Find the centroid of the *i*th cluster, *i=1,2,…,k*
3. Compute the intercluster distance matrix
4. Merge the nearest clusters
5. Repeat steps 2 through 4 until all components are part of one cluster

Minimum Spanning Tree Example

Program	CPU Time	Disk I/O
A	2	4
B	3	5
C	1	6
D	4	3
E	5	2

- Step 1: consider five clusters with *i*th cluster consisting solely of *i*th program
- Step 2: the centroids are {2,4}, {3,5}, {1,6}, {4,3}, and {5,2}

Minimum Spanning Tree Example

- Step 3: the Euclidean distance is

Program	A	B	C	D	E
A	0	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{5}$	$\sqrt{13}$
B		0	$\sqrt{5}$	$\sqrt{5}$	$\sqrt{13}$
C			0	$\sqrt{18}$	$\sqrt{32}$
D				0	$\sqrt{2}$
E					0

- Step 4: minimum intercluster distance = $\sqrt{2}$
  Merge A+B, D+E

Minimum Spanning Tree Example

- Step 2: the centroid of cluster pair AB is {(2+3)/2, (4+5)/2}, that is, {2.5, 4.5}. Similarly, the centroid of pair DE is {4.5, 2.5}
- Step 3: the distance matrix is:

	Program		
Program	AB	C	DE
AB	0	$\sqrt{4.5}$	$\sqrt{10.25}$
C		0	$\sqrt{24.4}$
DE			0

- Merge AB and C

Minimum Spanning Tree Example

- Step 2: the centroid of cluster ABC is {(2+3+1)/3, (4+5+6)/3}, that is {2,5}
- Step 3: the distance matrix is

	Program	
Program	ABC	DE
ABC	0	$\sqrt{12.5}$
DE		0

- Step 4: minimum distance is $\sqrt{12.5}$. Merge ABC and DE

Dendogram

- **Dendrogram** = Spanning Tree
- Purpose: obtain clusters for any given maximum allowable intracluster distance

Nearest Centroid Method

- Start with k=1
- Find the centroid and intracluster variance for ith cluster, i=1,2,…,k
- Find the cluster with the highest variance and arbitrarily divide it into two clusters
  - Find the two components that are farthest apart, assign other components according to their distance from these points
  - Place all components below the centroid in one cluster and all components above this hyperplane in the other

Nearest Centroid Method

- Adjust the points in the two new clusters until the intercluster distance between the two clusters is maximum
- Set $k=k+1$. repeat steps 2 through 4 until $k=n$

Cluster Interpretation

- Assign all measured components to the cluster
- Clusters with very small populations and small total resource demands can be discarded (don’t just discard a small cluster)
- Interpret clusters in functional terms, e.g., a business application. Or label clusters by their resource demands, for example, CPU-bound, I/O-bound, and so forth
- Select one or more representative components from each cluster for use as test workload

Problems with Clustering

- Goal: minimize variance
- The results of clustering are highly variable. No rules for:
  - Selection of parameters
  - Distance measure
  - Scaling
- Labeling each cluster by functionality is difficult
- Requires many repetitions of the analysis