Internet Measurement and Data Analysis (7)

Kenjiro Cho

2012-11-14

review of previous class

Class 6 Correlation (11/7)

- Online recommendation systems
- Distance
- Correlation coefficient
- exercise: correlation analysis

today's topics

Class 7 Multivariate analysis

- Data sensing
- Linear regression
- Principal Component Analysis
- exercise: linear regression

multivariate analysis

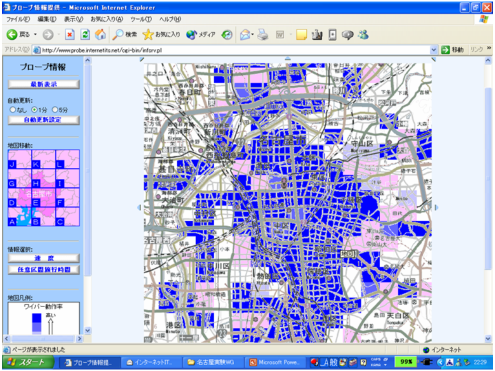
- univariate analysis
  - explores a single variable in a data set, separately
- multivariate analysis
  - looks at more than one variables at a time
  - enabled by computers
  - finding hidden trends (data mining)

data sensing

- data sensing: collecting data from remote site
- it becomes possible to access various sensor information over the Internet
  - weather information, power consumption, etc.

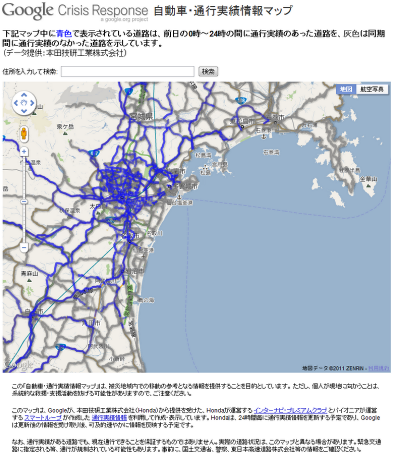
example: Internet vehicle experiment

- by WIDE Project in Nagoya in 2001
  - location, speed, and wiper usage data from 1,570 taxis
  - blue areas indicate high ratio of wiper usage, showing rainfall in detail



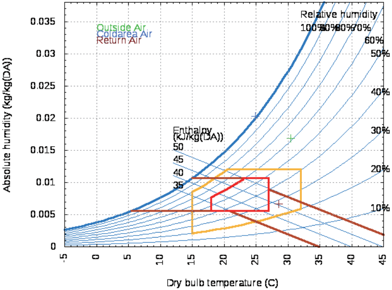
Japan Earthquake

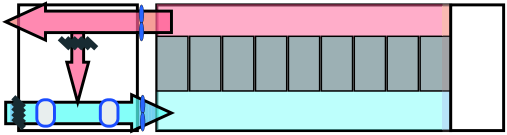
- the system is now part of ITS
- usable roads info released 3 days after the quake
  - data provide by HONDA (TOYOTA, NISSAN)

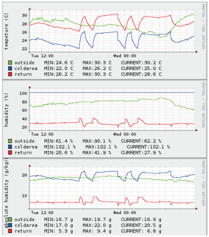


source: google crisis response

example: data center as data







measurement metrics of the Internet

measurement metrics

- link capacity, throughput
- delay
- jitter
- packet loss rate

methodologies

- active measurement: injects measurement packets (e.g., ping)
- passive measurement: monitors network without interfering in traffic
  - monitor at 2 locations and compare
  - infer from observations (e.g., behavior of TCP)
  - collect measurements inside a transport mechanism

delay measurement

- delay components
  - delay = propagation delay + queueing delay + other overhead
  - if not congested, delay is close to propagation deley
- methods
  - round-trip delay
  - one-way delay requires clock synchronization

  - average delay
  - max delay: e.g., voice communication requires $< 400ms$
  - jitter: variations in delay

some delay numbers

- packet transmission time (so called wire-speed)
  - 1500 bytes at 10Mbps: 1.2msec
  - 1500 bytes at 100Mbps: 120usec
  - 1500 bytes at 1Gbps: 12usec
  - 1500 bytes at 10Gbps: 1.2usec
- speed of light in fiber: about 200,000 km/s
  - 100km round-trip: 1 msec
  - 20,000km round-trip: 200msec
- satellite round-trip delay
  - LEO (Low-Earth Orbit): 200 msec
  - GEO (Geostationary Orbit): 600msec

packet loss measurement

packet loss rate

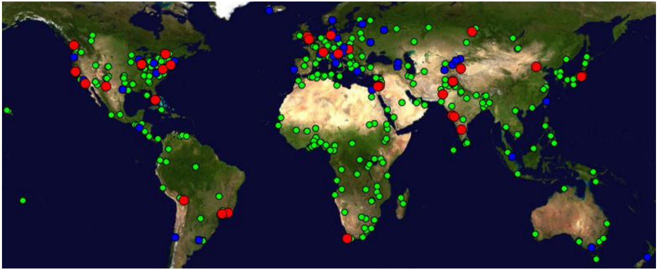
- loss rate is enough if packet loss is random...
- in reality,
  - bursty loss: e.g., buffer overflow
  - packet size dependency: e.g., bit error rate in wireless transmission

pingER project

- the Internet End-to-end Performance Measurement (IEPM) project by SLAC
- using ping to measure rtt and packet loss around the world
  - http://www-iepm.slac.stanford.edu/pinger/
  - started in 1995
  - over 600 sites in over 125 countries

pingER project monitoring sites

- monitoring (red), beacon (blue), remote (green) sites
  - beacon sites are monitored by all monitors



from pingER web site

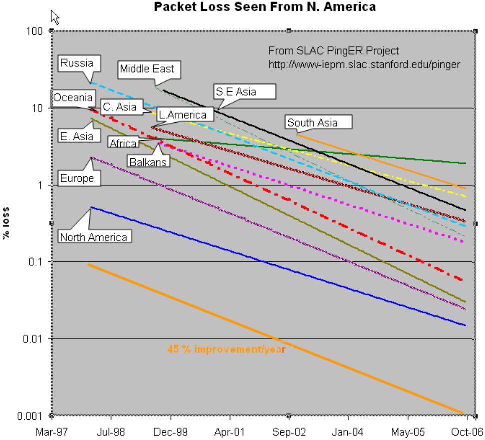
pingER project monitoring sites in east asia

- monitoring (red) and remote (green) sites

from pingER web site

pingER packet loss

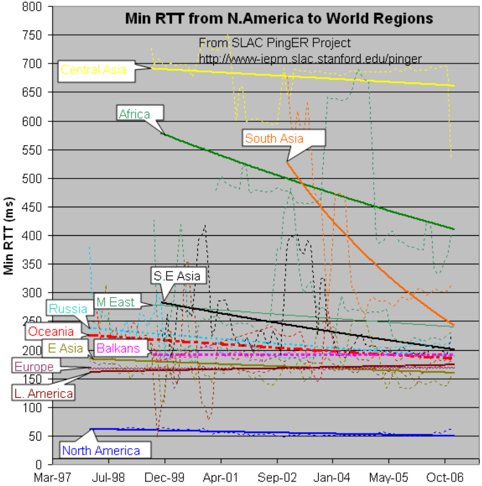
- packet loss observed from N. Ameria
- exponential improvement in 10 years



from pingER web site

pinger minimum rtt

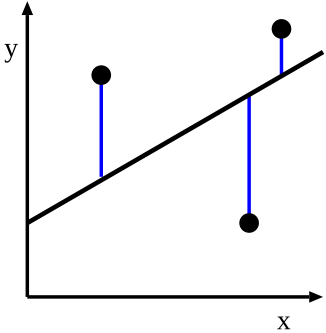
- minimum rtts observed from N. America
- gradual shift from satellite to fiber in S. Asia and Africa

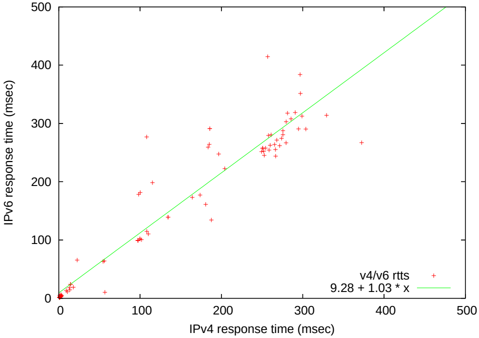


from pingER web site

linear regression

- fitting a straight line to data
  - least square method: minimize the sum of squared errors





least square method

a linear function minimizing squared errors

$$f(x) = b_0 + b_1 x$$

2 regression parameters can be computed by

$$b_1 = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n(\bar{x})^2}$$

$$b_0 = \bar{y} - b_1\bar{x}$$

where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\sum xy = \sum_{i=1}^{n} x_i y_i \qquad \sum x^2 = \sum_{i=1}^{n} (x_i)^2$$

a derivation of the expressions for regression parameters

The error in the ith observation: $e_i = y_i - (b_0 + b_1 x_i)$
For a sample of n observations, the mean error is

$$\bar{e} = \frac{1}{n}\sum_i e_i = \frac{1}{n}\sum_i (y_i - (b_0 + b_1 x_i)) = \bar{y} - b_0 - b_1\bar{x}$$

Setting the mean error to 0, we obtain: $b_0 = \bar{y} - b_1\bar{x}$
Substituting b_0 in the error expression: $e_i = y_i - \bar{y} + b_1\bar{x} - b_1 x_i = (y_i - \bar{y}) - b_1(x_i - \bar{x})$
The sum of squared errors, SSE, is

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n}[(y_i - \bar{y})^2 - 2b_1(y_i - \bar{y})(x_i - \bar{x}) + b_1^2(x_i - \bar{x})^2]$$

$$\begin{aligned}\frac{SSE}{n} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 - 2b_1\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) + b_1^2\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \\ &= \sigma_y^2 - 2b_1\sigma_{xy}^2 + b_1^2\sigma_x^2\end{aligned}$$

The value of b_1, which gives the minimum SSE, can be obtained by differentiating this equation with respect to b_1 and equating the result to 0:

$$\frac{1}{n}\frac{d(SSE)}{db_1} = -2\sigma_{xy}^2 + 2b_1\sigma_x^2 = 0$$

That is: $b_1 = \frac{\sigma_{xy}^2}{\sigma_x^2} = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n(\bar{x})^2}$

principal component analysis; PCA

purpose of PCA

- convert a set of possibly correlated variables into a smaller set of uncorrelated variables

PCA can be solved by eigenvalue decomposition of a covariance matrix

applications of PCA

- demensionality reduction
  - sort principal components by contribution ratio, components with small contribution ratio can be ignored
- principal component labeling
  - find means of produced principal components

notes:

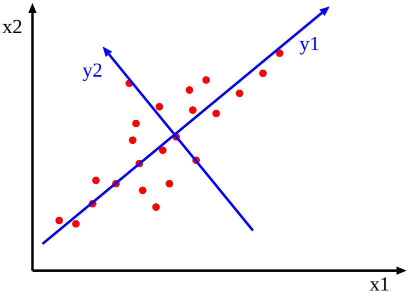
- PCA just extracts components with large variance
  - not simple if axes are not in the same unit
- a convenient method to automatically analyze complex relationship, but it does not explain the complex relationship

PCA: intuitive explanation

a view of cordinate transformation using a 2D graph

- draw the first axis (the 1st PCA axis) that goes through the centroid, along the direction of the maximal variability
- draw the 2nd axis that goes through the centroid, is orthogonal to the 1st axis, along the direction of the 2nd maximal variability
- draw the subsequent axes in the same manner

For example, “height” and “weight” can be mapped to “body size” and “slimness”. we can add “sitting height” and “chest measurement” in a similar manner



PCA (appendix)

principal components can be found as the eigenvectors of a covariance matrix.
let X be a d-demensional random variable. we want to find a dxd orthogonal transformation matrix P that convers X to its principal components Y.

$$\mathrm{Y} = \mathrm{P}^{\top}\mathrm{X}$$

solve this equation, assuming $cov(\mathrm{Y})$ being a diagonal matrix (components are independent), and P being an orthogonal matrix. ($\mathrm{P}^{-1} = \mathrm{P}^{\top}$)
the covariance matrix of Y is

$$\begin{aligned} cov(\mathrm{Y}) &= \mathrm{E}[\mathrm{Y}\mathrm{Y}^{\top}] = \mathrm{E}[(\mathrm{P}^{\top}\mathrm{X})(\mathrm{P}^{\top}\mathrm{X})^{\top}] = \mathrm{E}[(\mathrm{P}^{\top}\mathrm{X})(\mathrm{X}^{\top}\mathrm{P})] \\ &= \mathrm{P}^{\top}\mathrm{E}[\mathrm{X}\mathrm{X}^{\top}]\mathrm{P} = \mathrm{P}^{\top}cov(\mathrm{X})\mathrm{P} \end{aligned}$$

thus,

$$\mathrm{P}cov(\mathrm{Y}) = \mathrm{P}\mathrm{P}^{\top}cov(\mathrm{X})\mathrm{P} = cov(\mathrm{X})\mathrm{P}$$

rewrite P as a $dx1$ matrix:

$$\mathrm{P} = [\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_d]$$

also, $cov(\mathrm{Y})$ is a diagonal matrix (components are independent)

$$cov(\mathrm{Y}) = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_d \end{bmatrix}$$

this can be rewritten as

$$[\lambda_1\mathrm{P}_1, \lambda_2\mathrm{P}_2, \ldots, \lambda_d\mathrm{P}_d] = [cov(\mathrm{X})\mathrm{P}_1, cov(\mathrm{X})\mathrm{P}_2, \ldots, cov(\mathrm{X})\mathrm{P}_d]$$

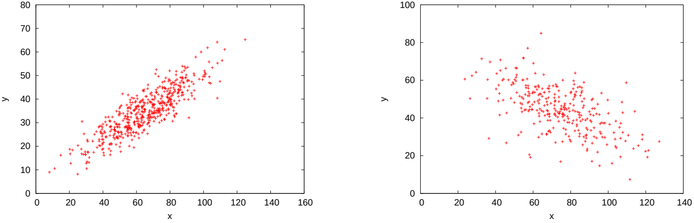
for $\lambda_i\mathrm{P}_i = cov(\mathrm{X})\mathrm{P}_i$, P_i is an eigenvector of the covariance matrix X
thus, we can find a transformation matrix P by finding the eigenvectors.

previous exercise: computing correlation coefficient

- compute correlation coefficient using the sample data sets
  - correlation-data-1.txt, correlation-data-2.txt

correlation coefficient

$$\rho_{xy} = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n} x_i y_i - \frac{(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n}}{\sqrt{(\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n})(\sum_{i=1}^{n} y_i^2 - \frac{(\sum_{i=1}^{n} y_i)^2}{n})}}$$



data-1:r=0.87 (left), data-2:r=-0.60 (right)

script to compute correlation coefficient

```
#!/usr/bin/env ruby

# regular expression for matching 2 floating numbers
re = /([-+]?\d+(?:\.\d+)?)\s+([-+]?\d+(?:\.\d+)?)/

sum_x = 0.0 # sum of x
sum_y = 0.0 # sum of y
sum_xx = 0.0 # sum of x^2
sum_yy = 0.0 # sum of y^2
sum_xy = 0.0 # sum of xy
n = 0 # the number of data

ARGF.each_line do |line|
    if re.match(line)
      x = $1.to_f
      y = $2.to_f
      sum_x += x
      sum_y += y
      sum_xx += x**2
      sum_yy += y**2
      sum_xy += x * y
      n += 1
    end
end

r = (sum_xy - sum_x * sum_y / n) /
  Math.sqrt((sum_xx - sum_x**2 / n) * (sum_yy - sum_y**2 / n))

printf "n:%d r:%.3f\n", n, r
```

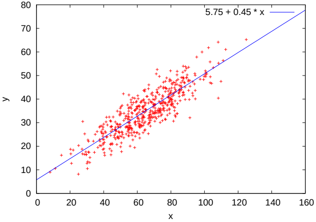
today's exercise: linear regression

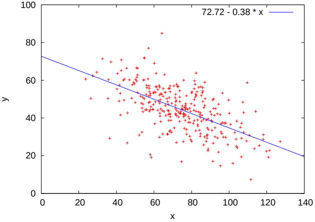
- linear regression by the least square method
- use the data for the previous exercise
  - correlation-data-1.txt, correlation-data-2.txt

$$f(x) = b_0 + b_1 x$$

$$b_1 = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n(\bar{x})^2}$$

$$b_0 = \bar{y} - b_1\bar{x}$$





data-1:r=0.87 (left), data-2:r=-0.60 (right)

script for linear regression

```
#!/usr/bin/env ruby

# regular expression for matching 2 floating numbers
re = /([-+]?\d+(?:\.\d+)?)\s+([-+]?\d+(?:\.\d+)?)/

sum_x = sum_y = sum_xx = sum_xy = 0.0
n = 0
ARGF.each_line do |line|
    if re.match(line)
      x = $1.to_f
      y = $2.to_f

      sum_x += x
      sum_y += y
      sum_xx += x**2
      sum_xy += x * y
      n += 1
    end
end

mean_x = Float(sum_x) / n
mean_y = Float(sum_y) / n
b1 = (sum_xy - n * mean_x * mean_y) / (sum_xx - n * mean_x**2)
b0 = mean_y - b1 * mean_x

printf "b0:%.3f b1:%.3f\n", b0, b1
```

adding the least squares line to scatter plot

```
set xrange [0:160]
set yrange [0:80]

set xlabel "x"
set ylabel "y"

plot "correlation-data-1.txt" notitle with points, \
5.75 + 0.45 * x lt 3
```

summary

Class 7 Multivariate analysis

- Data sensing
- Linear regression
- Principal Component Analysis
- exercise: linear regression

next class

Class 8 Time-series analysis (11/20) ***makeup class

- Nov 20 (Tue) 11:10-12:40 ϵ11

- Internet and time
- Network Time Protocol
- Time series analysis
- exercise: time-series analysis
- **assignment 2**