Long Live (Big Data-Fied) Statistics!

Norm Matloff
University of California at Davis

Joint Statistical Meetings
Montreal, August 4, 2013

Prolog

Where are we with Big Data?

Prolog

Where are we with Big Data?

- Role of statistics?
- Role of parallel computation?
- Interactions between the two?

Where are we?

Where are we?

Attitudes and worries:

Where are we?

Attitudes and worries:

- “With Big Data, you don’t need inference methods.”

Where are we?

Attitudes and worries:

- “With Big Data, you don’t need inference methods.”
- “With Machine Learning, you don’t need statistics.”

Where are we?

Attitudes and worries:

- "With Big Data, you don't need inference methods."
- "With Machine Learning, you don't need statistics."
- Stat community left out of the Big Data revolution (e.g. *Amstat News*, June 2013).

Yes, stat is still needed!

Yes, stat is still needed!

Actually, stat is needed more than ever, e.g.:

Yes, stat is still needed!

Actually, stat is needed more than ever, e.g.:

- Inference is an issue even for big n, once one considers subsets, where n becomes smaller. Same if do not have $p << n$.

Yes, stat is still needed!

Actually, stat is needed more than ever, e.g.:

- Inference is an issue even for big n, once one considers subsets, where n becomes smaller. Same if do not have $p << n$.
- Almost all machine learning techniques are revivals of old stat methods.

Yes, stat is still needed!

Actually, stat is needed more than ever, e.g.:

- Inference is an issue even for big n, once one considers subsets, where n becomes smaller. Same if do not have $p << n$.
- Almost all machine learning techniques are revivals of old stat methods. And if you don't understand stat, you won't be able to use ML methods effectively.

Yes, stat is still needed!

Actually, stat is needed more than ever, e.g.:

- Inference is an issue even for big n, once one considers subsets, where n becomes smaller. Same if do not have $p << n$.
- Almost all machine learning techniques are revivals of old stat methods. And if you don't understand stat, you won't be able to use ML methods effectively.
- An old stat technique—nonparametric curve estimation—now more useful than ever, for Big Data Graphics.

Yes, stat is still needed!

Actually, stat is needed more than ever, e.g.:

- Inference is an issue even for big n, once one considers subsets, where n becomes smaller. Same if do not have $p << n$.
- Almost all machine learning techniques are revivals of old stat methods. And if you don't understand stat, you won't be able to use ML methods effectively.
- An old stat technique—nonparametric curve estimation—now more useful than ever, for Big Data Graphics.
- The Curse of Dimensionality hasn't gone away.

Yes, stat is still needed!

Actually, stat is needed more than ever, e.g.:

- Inference is an issue even for big n, once one considers subsets, where n becomes smaller. Same if do not have $p << n$.
- Almost all machine learning techniques are revivals of old stat methods. And if you don't understand stat, you won't be able to use ML methods effectively.
- An old stat technique—nonparametric curve estimation—now more useful than ever, for Big Data Graphics.
- The Curse of Dimensionality hasn't gone away. Impossible to understand without stat.

Setting

Setting

Consider the classical (though not universal) data format:

Setting

Consider the classical (though not universal) data format:

- n observations/cases/instances/...

Setting

Consider the classical (though not universal) data format:

- n observations/cases/instances/...
- p variables/features/attributes/...

Setting

Consider the classical (though not universal) data format:

- n observations/cases/instances/...
- p variables/features/attributes/...
- Assumed i.i.d.

Setting

Consider the classical (though not universal) data format:

- n observations/cases/instances/...
- p variables/features/attributes/...
- Assumed i.i.d.

(Here I'm trying to include terminology from the nonstat communities.)

Setting

Consider the classical (though not universal) data format:

- n observations/cases/instances/...
- p variables/features/attributes/...
- Assumed i.i.d.

(Here I'm trying to include terminology from the nonstat communities.)
Does "Big" Data mean big n or big p or both?

Setting

Consider the classical (though not universal) data format:

- n observations/cases/instances/...
- p variables/features/attributes/...
- Assumed i.i.d.

(Here I'm trying to include terminology from the nonstat communities.)
Does "Big" Data mean big n or big p or both?
This talk will contain one "big n" section and two "big p" section.

Big n

Part I: Big-n Problem

Big-n can be handled

Big-n can be handled

Big n can generally be handled:

Big-n can be handled

Big n can generally be handled:

- Many (though certainly not all) computations "additive," thus "embarrassingly parallel."

Big-n can be handled

Big n can generally be handled:

- Many (though certainly not all) computations "additive," thus "embarrassingly parallel."
- Thus amenable to parallel computation, especially distributed data, MapReduce etc.

Big-n can be handled

Big n can generally be handled:

- Many (though certainly not all) computations "additive," thus "embarrassingly parallel."
- Thus amenable to parallel computation, especially distributed data, MapReduce etc.
- "Chunks averaging method" (CAM) (Fan *et al*, 2007; Matloff, 2010; etc.) can turn most statistical computations into embarrassingly parallel.

Big-n can be handled

Big n can generally be handled:

- Many (though certainly not all) computations "additive," thus "embarrassingly parallel."
- Thus amenable to parallel computation, especially distributed data, MapReduce etc.
- "Chunks averaging method" (CAM) (Fan *et al*, 2007; Matloff, 2010; etc.) can turn most statistical computations into embarrassingly parallel. ("Software alchemy.")

Chunk Averaging Method

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.
  - Break observations into chunks.

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.
  - Break observations into chunks.
  - Fit regression equation to each chunk.

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.
  - Break observations into chunks.
  - Fit regression equation to each chunk.
  - Average the results.

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.
    - Break observations into chunks.
    - Fit regression equation to each chunk.
    - Average the results.
- Produces statistically equivalent results for large n.

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.
  - Break observations into chunks.
  - Fit regression equation to each chunk.
  - Average the results.
- Produces statistically equivalent results for large n.
- Essentially and i.i.d.-based method, e.g. quantile regression, hazard function estimation, tree methods, etc.

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.
    - Break observations into chunks.
    - Fit regression equation to each chunk.
    - Average the results.
- Produces statistically equivalent results for large n.
- Essentially and i.i.d.-based method, e.g. quantile regression, hazard function estimation, tree methods, etc.
- *Superlinear* speedup.

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.
  - Break observations into chunks.
  - Fit regression equation to each chunk.
  - Average the results.
- Produces statistically equivalent results for large n.
- Essentially and i.i.d.-based method, e.g. quantile regression, hazard function estimation, tree methods, etc.
- *Superlinear* speedup. E.g. quantile regression, 5.31X for 4 threads.

Chunk Averaging Method

- Key point: CAM converts non-embaarrassingly parallel algs to additive ones that are **statistically equivalent** (same standard errors).
- Example: Regression.
  - Break observations into chunks.
  - Fit regression equation to each chunk.
  - Average the results.
- Produces statistically equivalent results for large n.
- Essentially and i.i.d.-based method, e.g. quantile regression, hazard function estimation, tree methods, etc.
- *Superlinear* speedup. E.g. quantile regression, 5.31X for 4 threads. Can be faster even for just one core.

Part II

Part II: A New Graphical Approach to Big-p Problem

Graphing Lots of Variables

Graphing Lots of Variables

Issues:

Graphing Lots of Variables

Issues:

- Can deal (a little bit) better with big-p if we can display lots of variables on the same graph.

Graphing Lots of Variables

Issues:

- Can deal (a little bit) better with big-p if we can display lots of variables on the same graph.
- Can't display all at once, but try to get at least several.

Graphing Lots of Variables

Issues:

- Can deal (a little bit) better with big-p if we can display lots of variables on the same graph.
- Can't display all at once, but try to get at least several.
- Problems:

Graphing Lots of Variables

Issues:

- Can deal (a little bit) better with big-p if we can display lots of variables on the same graph.
- Can't display all at once, but try to get at least several.
- Problems:
  - Displaying > 2 variables on a 2-dimensional device.

Graphing Lots of Variables

Issues:

- Can deal (a little bit) better with big-p if we can display lots of variables on the same graph.
- Can't display all at once, but try to get at least several.
- Problems:
    - Displaying > 2 variables on a 2-dimensional device.
    - "Black screen problem"—with big n, at least parts of the screen become solid black.

Graphing Lots of Variables

Graphing Lots of Variables

Graphing Lots of Variables

Some existing methods:

Graphing Lots of Variables

Some existing methods:

- Grand tours:

Graphing Lots of Variables

Some existing methods:

- Grand tours: Show sequence of rotations and projections, e.g. **tourr** in R.

Graphing Lots of Variables

Some existing methods:

- Grand tours: Show sequence of rotations and projections, e.g. **tourr** in R.
  Nice, visually appealing.

Graphing Lots of Variables

Some existing methods:

- Grand tours: Show sequence of rotations and projections, e.g. **tourr** in R.
  Nice, visually appealing. But hard to discern exact relations, and suffers from black-screen problem.

Graphing Lots of Variables

Some existing methods:

- Grand tours: Show sequence of rotations and projections, e.g. **tourr** in R.
  Nice, visually appealing. But hard to discern exact relations, and suffers from black-screen problem.
- Parallel coordinates:

Graphing Lots of Variables

Some existing methods:

- Grand tours: Show sequence of rotations and projections, e.g. **tourr** in R.
  Nice, visually appealing. But hard to discern exact relations, and suffers from black-screen problem.
- Parallel coordinates: Draw one vertical axis for each variable. Draw a set of connecting lines for each data point.

Graphing Lots of Variables

Some existing methods:

- Grand tours: Show sequence of rotations and projections, e.g. **tourr** in R.
  Nice, visually appealing. But hard to discern exact relations, and suffers from black-screen problem.
- Parallel coordinates: Draw one vertical axis for each variable. Draw a set of connecting lines for each data point.
  Hard to understand noncontiguous axes, black screen problem.

Statistics to the rescue!

Norm Matloff
University of California at Davis

Statistics to the rescue!

An obvious solution to the black-screen problem:

Statistics to the rescue!

An obvious solution to the black-screen problem: nonparametric curve estimation.

Statistics to the rescue!

An obvious solution to the black-screen problem: nonparametric curve estimation.
Example: Scatter plots.

Statistics to the rescue!

An obvious solution to the black-screen problem: nonparametric curve estimation.
Example: Scatter plots.

- Ordinary plot would fill the screen.

Statistics to the rescue!

An obvious solution to the black-screen problem: nonparametric curve estimation.
Example: Scatter plots.

- Ordinary plot would fill the screen.
- Solution: Draw the nonpar. 2-dim. density estimate instead.

Norm Matloff
University of California at Davis

Statistics to the rescue!

An obvious solution to the black-screen problem: nonparametric curve estimation.
Example: Scatter plots.

- Ordinary plot would fill the screen.
- Solution: Draw the nonpar. 2-dim. density estimate instead.
- That makes 3 dimensions, but code third dimension (density height) via color.

Statistics to the rescue!

An obvious solution to the black-screen problem: nonparametric curve estimation.
Example: Scatter plots.

- Ordinary plot would fill the screen.
- Solution: Draw the nonpar. 2-dim. density estimate instead.
- That makes 3 dimensions, but code third dimension (density height) via color.
- E.g. **scatterSmooth()** in R.

Displaying 3 Vars. in 2 Dims.

Norm Matloff
University of California at Davis

Displaying 3 Vars. in 2 Dims.

The **scatterSmooth()** example actually shows how to display 3 variables in 2 dimensions:

Displaying 3 Vars. in 2 Dims.

The **scatterSmooth()** example actually shows how to display 3 variables in 2 dimensions:
Say have variables X, Y, Z.

Displaying 3 Vars. in 2 Dims.

The **scatterSmooth()** example actually shows how to display 3 variables in 2 dimensions:
Say have variables X, Y, Z.

- Plot regression function of Z, **color coded**, against X and Y.

Displaying 3 Vars. in 2 Dims.

The **scatterSmooth()** example actually shows how to display 3 variables in 2 dimensions:
Say have variables X, Y, Z.

- Plot regression function of Z, **color coded**, against X and Y.
- Regression function: $m(s,t) = E(Z \mid X = s, Y = t)$ (i.e. general, not assuming param. model).

Displaying 3 Vars. in 2 Dims.

The **scatterSmooth()** example actually shows how to display 3 variables in 2 dimensions:
Say have variables X, Y, Z.

- Plot regression function of Z, **color coded**, against X and Y.
- Regression function: $m(s,t) = E(Z \mid X = s, Y = t)$ (i.e. general, not assuming param. model).
- Use nonpar. estimation, e.g. nearest-neighbor.

Displaying 3 Vars. in 2 Dims.

The **scatterSmooth()** example actually shows how to display 3 variables in 2 dimensions:
Say have variables X, Y, Z.

- Plot regression function of Z, **color coded**, against X and Y.
- Regression function: $m(s,t) = E(Z \mid X = s, Y = t)$ (i.e. general, not assuming param. model).
- Use nonpar. estimation, e.g. nearest-neighbor.
- 3 vars. in 2 dims.!

Displaying 3 Vars. in 2 Dims.

The **scatterSmooth()** example actually shows how to display 3 variables in 2 dimensions:
Say have variables X, Y, Z.

- Plot regression function of Z, **color coded**, against X and Y.
- Regression function: $m(s,t) = E(Z \mid X = s, Y = t)$ (i.e. general, not assuming param. model).
- Use nonpar. estimation, e.g. nearest-neighbor.
- 3 vars. in 2 dims.! (No perspective plotting.)

Proposed Boundary Method

Proposed Boundary Method

I introduce here a new approach to plotting multiple variables in 2 dims.,

Proposed Boundary Method

I introduce here a new approach to plotting multiple variables in 2 dims., based on "boundary curves."

Proposed Boundary Method

I introduce here a new approach to plotting multiple variables in 2 dims., based on "boundary curves."

First, again consider 3 variables, X, Y and Z.

Proposed Boundary Method

I introduce here a new approach to plotting multiple variables in 2 dims., based on "boundary curves."
First, again consider 3 variables, X, Y and Z.

- For user-chosen b, boundary is the set

$$\{(s,t) : E(Z|X=s, Y=t) = b\} \tag{1}$$

Proposed Boundary Method

I introduce here a new approach to plotting multiple variables in 2 dims., based on "boundary curves."
First, again consider 3 variables, X, Y and Z.

- For user-chosen b, boundary is the set

$$\{(s,t) : E(Z|X = s, Y = t) = b\} \qquad (1)$$

- User might set b = E(Z) (overall, unconditional mean).

Norm Matloff
University of California at Davis

Proposed Boundary Method

I introduce here a new approach to plotting multiple variables in 2 dims., based on "boundary curves."
First, again consider 3 variables, X, Y and Z.

- For user-chosen b, boundary is the set

$$\{(s,t) : E(Z|X=s, Y=t) = b\} \tag{1}$$

- User might set b = E(Z) (overall, unconditional mean).
- Plot estimate of the boundary curve.

Proposed Boundary Method

I introduce here a new approach to plotting multiple variables in 2 dims., based on "boundary curves."
First, again consider 3 variables, X, Y and Z.

- For user-chosen b, boundary is the set

$$\{(s,t) : E(Z|X = s, Y = t) = b\} \tag{1}$$

- User might set b = E(Z) (overall, unconditional mean).
- Plot estimate of the boundary curve.
- Displaying 3 vars. in 2 dims.

Boundary Plot Example

Boundary Plot Example

- Bank account data, UCI repository.

Boundary Plot Example

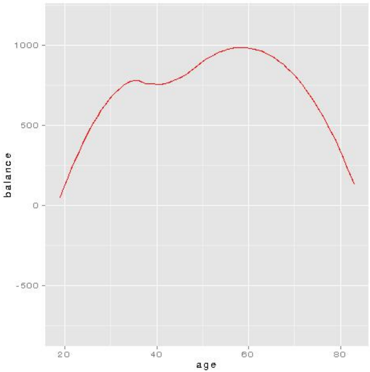
- Bank account data, UCI repository.
- X = age of customer, Y = current bank account, Z = say Yes to open new type of account

Boundary Plot Example

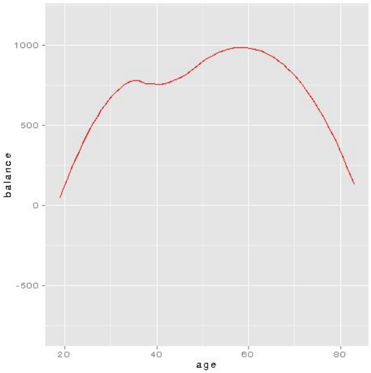
- Bank account data, UCI repository.
- X = age of customer, Y = current bank account, Z = say Yes to open new type of account
- $b = EZ = P(Z = 1)$

Bank Example

Bank Example

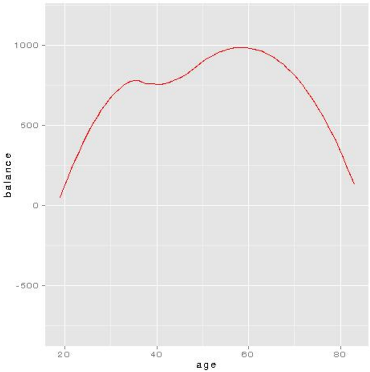


Bank Example



- Above line means, above-avg. prob. sign up for new account.

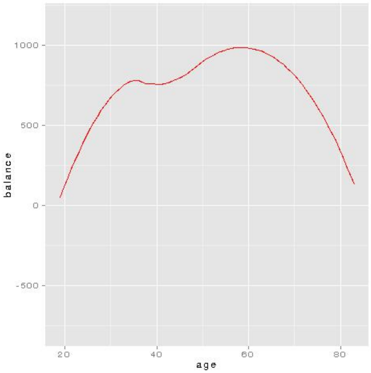
Bank Example



- Above line means, above-avg. prob. sign up for new account.
- Near retire ⇒ "hardest sell"!

Norm Matloff University of California at Davis

Bank Example



- Above line means, above-avg. prob. sign up for new account.
- Near retire ⇒ "hardest sell"!
- Those around 60 need a large balance before willing to try new account.

More Than 3 Vars. in 2 Dims.

More Than 3 Vars. in 2 Dims.

- Plotting boundaries has been done before.

More Than 3 Vars. in 2 Dims.

- Plotting boundaries has been done before.
- But the idea here is to display several boundaries at once, so as to display more variables in one 2-dim. graph.

Example: Adult Data

Example: Adult Data

- UCI Adult data

Example: Adult Data

- UCI Adult data
- X = age, Y = education, Z = high income

Example: Adult Data

- UCI Adult data
- X = age, Y = education, Z = high income
- But now add a 4th variable: W = gender

Example: Adult Data

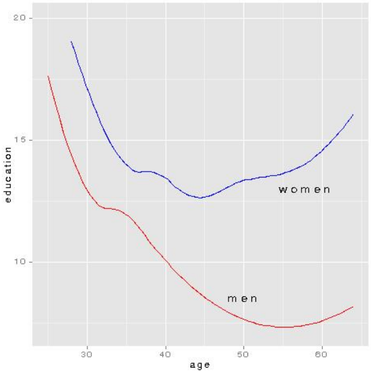
- UCI Adult data
- X = age, Y = education, Z = high income
- But now add a <u>4th</u> variable: W = gender
- Plot 2 boundary curves, one male and one female.

Example: Adult Data

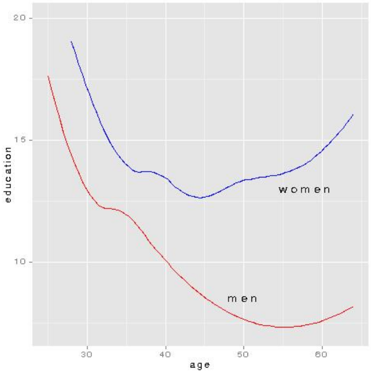
- UCI Adult data
- X = age, Y = education, Z = high income
- But now add a 4th variable: W = gender
- Plot 2 boundary curves, one male and one female.
- Thus display 4 variables in 2 dims.

Adult Example

Adult Example

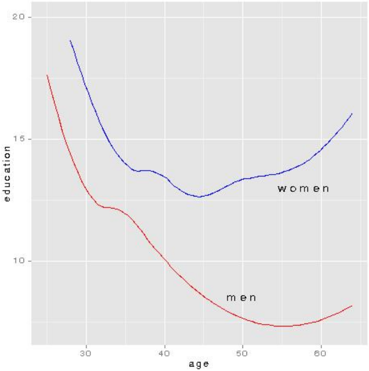


Adult Example



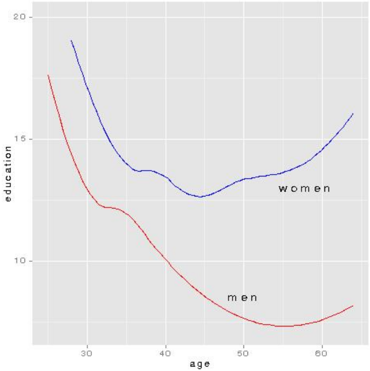
- Above line means, higher-than-avg. prob. of high income.

Adult Example



- Above line means, higher-than-avg. prob. of high income.
- Before age 35, not much difference.

Adult Example



- Above line means, higher-than-avg. prob. of high income.
- Before age 35, not much difference.
- After age 35, women need much more education than men to likely have high income.

Example: Flight Lateness

Norm Matloff
University of California at Davis

Example: Flight Lateness

- Airline lateness data.

Example: Flight Lateness

- Airline lateness data.
- X = departure delay, Y = distance, Z = arrival lateness, W = originating airport (here, SFO, IAD, IAH),

Example: Flight Lateness

- Airline lateness data.
- X = departure delay, Y = distance, Z = arrival lateness, W = originating airport (here, SFO, IAD, IAH), so again, displaying 4 variables in 2 dims.
- 3 curves, one for each airport

Example: Flight Lateness

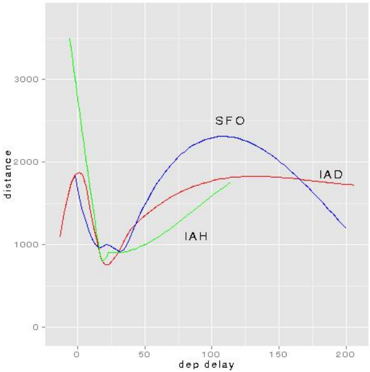
- Airline lateness data.
- X = departure delay, Y = distance, Z = arrival lateness, W = originating airport (here, SFO, IAD, IAH), so again, displaying 4 variables in 2 dims.
- 3 curves, one for each airport
- Could add V = daytime vs. evening, for 6 curves, thus displaying 5 variables in 2 dims.

Example: Flight Lateness

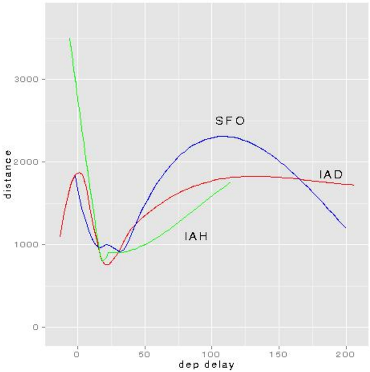
- Airline lateness data.
- X = departure delay, Y = distance, Z = arrival lateness, W = originating airport (here, SFO, IAD, IAH), so again, displaying 4 variables in 2 dims.
- 3 curves, one for each airport
- Could add V = daytime vs. evening, for 6 curves, thus displaying 5 variables in 2 dims.
- Could plot straight regressions too, but boundaries always enable us to plot "one more variable."

Airline Example

Airline Example

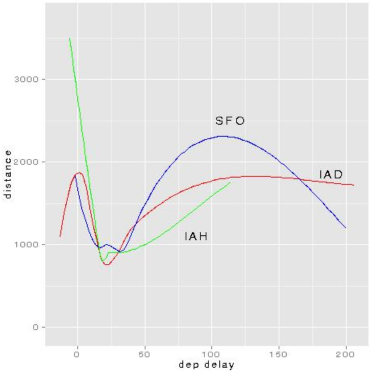


Airline Example



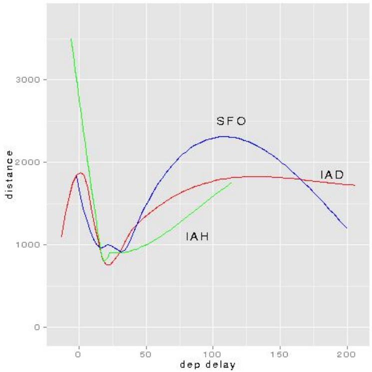
- Above line means, higher-than-avg. mean delay.

Airline Example



- Above line means, higher-than-avg. mean delay.
- SFO seems to be doing better.

Airline Example



- Above line means, higher-than-avg. mean delay.
- SFO seems to be doing better. Need a very long flight to have above-avg. delay, relative to the others.

Computation

Computation

- R's (**contour()** not used (don't want "islands").

Computation

- R's (**contour()** not used (don't want "islands").
- Estimate regression (via fast kNN, **FNN** library).

Norm Matloff
University of California at Davis

Computation

- R's (**contour()** not used (don't want "islands").
- Estimate regression (via fast kNN, **FNN** library).
- Find "boundary band," all points near the estimate boundary.

Computation

- R's (**contour()** not used (don't want "islands").
- Estimate regression (via fast kNN, **FNN** library).
- Find "boundary band," all points near the estimate boundary.
- Smooth the band.

Parallel Computation

Parallel Computation

Computation can be voluminous.

Parallel Computation

Computation can be voluminous.

- Parallel processing.

Parallel Computation

Computation can be voluminous.

- Parallel processing.
- Take advantage of superlinearity from CAM.

Parallel Computation

Computation can be voluminous.

- Parallel processing.
- Take advantage of superlinearity from CAM.
- Break into chunks, but only find near nghbrs. within chunks, not across chunks.

Norm Matloff
University of California at Davis

Parallel Computation

Computation can be voluminous.

- Parallel processing.
- Take advantage of superlinearity from CAM.
- Break into chunks, but only find near nghbrs. within chunks, not across chunks.
- The "A" part of CAM comes in the smoothing of the band.

Part III: Big p and the Curse of Dimensionality

Exorcizing the Curse of Dimensionality

Part III: Big p and the Curse of Dimensionality

Exorcizing the Curse of Dimensionality

Some small steps in that direction.

Big p

Big p

- Theoretical considerations imply that should have $p < \sqrt{n}$ in regression case (Portnoy, 1968).

Big p

- Theoretical considerations imply that should have $p < \sqrt{n}$ in regression case (Portnoy, 1968).
- Yet today $p >> n$ is commonplace.

Big p

- Theoretical considerations imply that should have $p < \sqrt{n}$ in regression case (Portnoy, 1968).
- Yet today $p >> n$ is commonplace.
- This causes "multiple inference" problems (e.g. familywise error rates).

Big p

- Theoretical considerations imply that should have $p < \sqrt{n}$ in regression case (Portnoy, 1968).
- Yet today $p >> n$ is commonplace.
- This causes "multiple inference" problems (e.g. familywise error rates).
- So, e.g., CI radii $1.96 \ \mathrm{std.err.}(\widehat{\theta})$ **might NOT be "essentially 0."**

Big p

- Theoretical considerations imply that should have $p < \sqrt{n}$ in regression case (Portnoy, 1968).
- Yet today $p >> n$ is commonplace.
- This causes "multiple inference" problems (e.g. familywise error rates).
- So, e.g., CI radii 1.96 $\mathrm{std.err.}(\widehat{\theta})$ **might NOT be "essentially 0."** I.e., Big n not big after all.

Big p

- Theoretical considerations imply that should have $p < \sqrt{n}$ in regression case (Portnoy, 1968).
- Yet today $p >> n$ is commonplace.
- This causes "multiple inference" problems (e.g. familywise error rates).
- So, e.g., CI radii $1.96 \ \mathrm{std.err.}(\widehat{\theta})$ **might NOT be "essentially 0."** I.e., Big n not big after all.
- And the ever-present Curse of Dimensionality.

Principle Components Analysis

Principle Components Analysis

- What sizes of p relative to n might be problematic for PCA?

Principle Components Analysis

- What sizes of p relative to n might be problematic for PCA?
  - Sample covariance matrix V has p(p-1)/2 distinct entries.

Norm Matloff University of California at Davis

Principle Components Analysis

- What sizes of p relative to n might be problematic for PCA?
  - Sample covariance matrix V has p(p-1)/2 distinct entries.
  - Data matrix has np entries.

Principle Components Analysis

- What sizes of p relative to n might be problematic for PCA?
  - Sample covariance matrix V has p(p-1)/2 distinct entries.
  - Data matrix has np entries.
  - So V is completely determined (except roundoff error) if np = p(p-1)/2.

Principle Components Analysis

- What sizes of p relative to n might be problematic for PCA?
  - Sample covariance matrix V has p(p-1)/2 distinct entries.
  - Data matrix has np entries.
  - So V is completely determined (except roundoff error) if np = p(p-1)/2.
  - So, have problem if p > 2n, roughly.

PCA Experiment

PCA Experiment

Simulation experiment:

PCA Experiment

Simulation experiment:

- Y_1, Y_2 indep. N(0,1); $X_1 = Y_1 + Y_2$, $X_2 = Y_1 - Y_2$, $X_3, ..., X_p$ iid N(0,1), indep. of X_1, X_2.

PCA Experiment

Simulation experiment:

- $Y_1, \ Y_2$ indep. N(0,1); $X_1 = Y_1 + Y_2$, $X_2 = Y_1 - Y_2$, $X_3, ..., X_p$ iid N(0,1), indep. of $X_1, \ X_2$.
- First PC should be (1,0,0,...) or (0,1,0,...).

PCA Experiment

Simulation experiment:

- Y_1, Y_2 indep. N(0,1); $X_1 = Y_1 + Y_2$, $X_2 = Y_1 - Y_2$, $X_3, ..., X_p$ iid N(0,1), indep. of X_1, X_2.
- First PC should be (1,0,0,...) or (0,1,0,...).

```
> sim
function(n,p) {
   y1 <- rnorm(n); y2 <- rnorm(n);
   x1 <- y1+y2; x2 <- y1-y2; p2 <- p - 2
   x <-
      cbind(x1,x2,matrix(rnorm(n*p2),ncol=p2))
   cvr <- cov(x)
   which.max(
      abs(eigen(cvr,symmetric=T)$vectors[,1]))
}
```

Simulation, cont'd.

Simulation, cont'd.

Return value from **sim()** should be 1 or 2. Let's see:

```
> sim(500,400)
[1] 1
> sim(500,800)
[1] 1
> sim(500,800)
[1] 2
> sim(500,1200)
[1] 439
> sim(500,1200)
[1] 2
> sim(500,1200)
[1] 1
> sim(500,1200)
[1] 905
```

Simulation, cont'd.

Norm Matloff
University of California at Davis

Simulation, cont'd.

- When $n < p/2$—very common in practice!—sometimes right but sometimes get phantom PCs.

Simulation, cont'd.

- When $n < p/2$—very common in practice!—sometimes right but sometimes get phantom PCs.
- On the other hand, results of Johnstone (2000) suggest that as long as $n > p/2$ we might be OK.

Simulation, cont'd.

- When $n < p/2$—very common in practice!—sometimes right but sometimes get phantom PCs.
- On the other hand, results of Johnstone (2000) suggest that as long as $n > p/2$ we might be OK.
- Moreover, in practice the variables are correlated, often very highly so, in regular patterns. I suspect this makes it "more OK."

Exorcizing the Curse?

Norm Matloff
University of California at Davis

Exorcizing the Curse?

- The term *curse of dimensionality* goes back 50 years.

Exorcizing the Curse?

- The term *curse of dimensionality* goes back 50 years.
- In last 10-15 years, it has gotten scarier:

Exorcizing the Curse?

- The term *curse of dimensionality* goes back 50 years.
- In last 10-15 years, it has gotten scarier: Berry(1999) proved that any 2 points are approximately the same distance from each other!

Exorcizing the Curse?

- The term *curse of dimensionality* goes back 50 years.
- In last 10-15 years, it has gotten scarier: Berry(1999) proved that any 2 points are approximately the same distance from each other!
- So, e.g., nearest-neighbor methods look iffy.

Exorcizing the Curse?

- The term *curse of dimensionality* goes back 50 years.
- In last 10-15 years, it has gotten scarier: Berry(1999) proved that any 2 points are approximately the same distance from each other!
- So, e.g., nearest-neighbor methods look iffy.
- My own rough derivation:

Exorcizing the Curse?

- The term *curse of dimensionality* goes back 50 years.
- In last 10-15 years, it has gotten scarier: Berry(1999) proved that any 2 points are approximately the same distance from each other!
- So, e.g., nearest-neighbor methods look iffy.
- My own rough derivation:
    - Suppose the p distance components are iid.

Exorcizing the Curse?

- The term *curse of dimensionality* goes back 50 years.
- In last 10-15 years, it has gotten scarier: Berry(1999) proved that any 2 points are approximately the same distance from each other!
- So, e.g., nearest-neighbor methods look iffy.
- My own rough derivation:
  - Suppose the p distance components are iid.
  - $\sqrt{Var(distance)}/E(distance) \rightarrow 0$ as $p- > \infty$

Exorcizing the Curse?

- The term *curse of dimensionality* goes back 50 years.
- In last 10-15 years, it has gotten scarier: Berry(1999) proved that any 2 points are approximately the same distance from each other!
- So, e.g., nearest-neighbor methods look iffy.
- My own rough derivation:
  - Suppose the p distance components are iid.
  - $\sqrt{Var(distance)}/E(distance) \rightarrow 0$ as $p- > \infty$
  - So, distances are approximately constant.

Some Hope

Some Hope

Some Hope:

- But all that involves equally-weighted components in distance.

Norm Matloff
University of California at Davis

Some Hope

Some Hope:

- But all that involves equally-weighted components in distance.
- Yet, arguably we should have weights, according to importance of the variables.

Norm Matloff
University of California at Davis

Some Hope

Some Hope:

- But all that involves equally-weighted components in distance.
- Yet, arguably we should have weights, according to importance of the variables.
- Then the above problem goes away. (Coef. of var. does not go to 0.)

Some Hope

Some Hope:

- But all that involves equally-weighted components in distance.
- Yet, arguably we should have weights, according to importance of the variables.
- Then the above problem goes away. (Coef. of var. does not go to 0.)
- But how set the weights?

Some Hope

Some Hope:

- But all that involves equally-weighted components in distance.
- Yet, arguably we should have weights, according to importance of the variables.
- Then the above problem goes away. (Coef. of var. does not go to 0.)
- But how set the weights?
- Stay tuned...

Norm Matloff
University of California at Davis

Misc.

Misc.

Online materiasl:
The visualization code is available for your use and comments/suggestions:
`http://heather.cs.ucdavis.edu/BigDataVis.html`
These slides are there too.

Acknowlegements:
The author would like to thank Noah Gift, Marnie Dunsmore, Nicholas Lewin-Koh, and David Scott for helpful discussions, and Hao Chen and Bill Hsu for use of their high-performance computing equipment.