Introduction to Counting

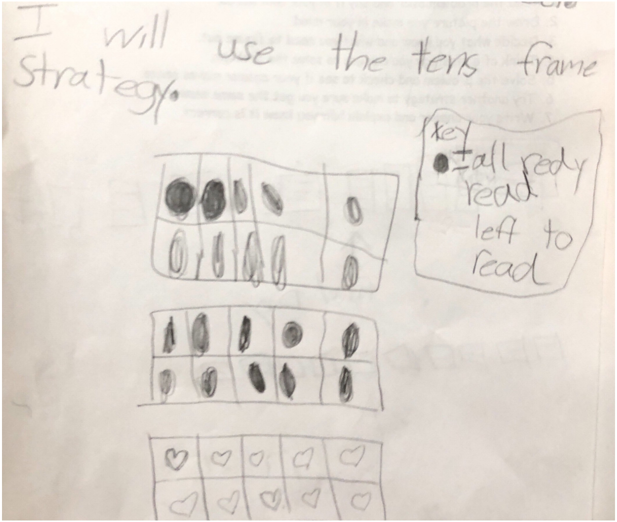
APAM E4990
Modeling Social Data

Jake Hofman

Columbia University

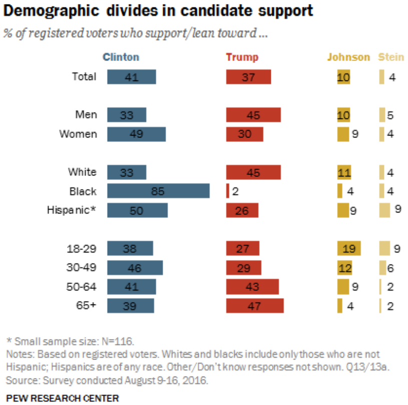
February 1, 2019

Why counting?



Why counting?

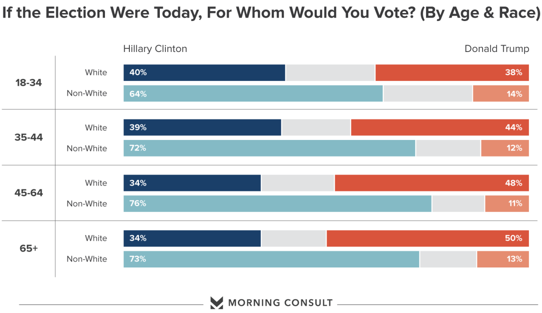
http://bit.ly/august2016poll



$$p(\underbrace{y}_{\text{support}} \mid \underbrace{x}_{\text{age}})$$

Why counting?

`http://bit.ly/ageracepoll2016`



$$p(\underbrace{y}_{\text{support}} \mid \underbrace{x_1, x_2}_{\text{age, race}})$$

Why counting?

?

$$p(\underbrace{y}_{\text{support}} \mid \underbrace{x_1, x_2, x_3, \ldots}_{\text{age, sex, race, party}})$$

Why counting?

How many responses do we need to estimate $p(y)$ with a 5% margin of error?

Why counting?

How many responses do we need to estimate $p(y)$ with a 5% margin of error?

What if we want to split this up by age, sex, race, and party?

Assume $\approx$ 100 age, 2 sex, 5 race, 3 party

Why counting?

Problem:

Traditionally difficult to obtain reliable estimates due to small sample sizes or sparsity

(e.g., $\sim$ 100 age $\times$ 2 sex $\times$ 5 race $\times$ 3 party $=$ 3,000 groups, but typical surveys collect $\sim$ 1,000s of responses)

Why counting?

Potential solution:

Sacrifice granularity for precision, by binning observations into larger, but fewer, groups

(e.g., bin age into a few groups: 18-29, 30-49, 50-64, 65+)

Why counting?

Potential solution:

Develop more sophisticated methods that generalize well from small samples

(e.g., fit a model: $\text{support} \sim \beta_0 + \beta_1 \text{age} + \beta_2 \text{age}^2 + \ldots$)

Why counting?

(Partial) solution:

Obtain larger samples through other means, so we can just count and divide to make estimates via relative frequencies

(e.g., with $\sim$ 1M responses, we have 100s per group and can estimate support within a few percentage points)

Why counting?

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

ELSEVIER

Forecasting elections with non-representative polls

CrossMark

Wei Wang[a,*], David Rothschild[b], Sharad Goel[b], Andrew Gelman[a,c]

[a] Department of Statistics, Columbia University, New York, NY, USA
[b] Microsoft Research, New York, NY, USA
[c] Department of Political Science, Columbia University, New York, NY, USA

Our analysis is based on an opt-in poll which was available continuously on the Xbox gaming platform during the 45 days preceding the 2012 US presidential election. Each day, three to five questions were posted, one of which gauged voter intention via the standard query, "If the election were held today, who would you vote for?". Full details of the questionnaire are given in the Appendix. The respondents were allowed to answer at most once per day. The first time they participated in an Xbox poll, respondents were also asked to provide basic demographic information about themselves, including their sex, race, age, education, state, party ID, political ideology, and who they voted for in the 2008 presidential election. In total, 750,148 interviews were conducted, with 345,858 unique respondents – over 30,000 of whom completed five or more polls – making this one of the largest election panel studies ever.

http://bit.ly/nonreppoll

Why counting?

The good:

Shift away from sophisticated statistical methods on small samples to simpler methods on large samples

Why counting?

The bad:

Even simple methods (e.g., counting) are computationally challenging at large scales

(1M is easy, 1B a bit less so, 1T gets interesting)

Why counting?

Claim:

Solving the counting problem at scale enables you to investigate many interesting questions in the social sciences

Learning to count

We'll focus on counting at small/medium scales on a single machine

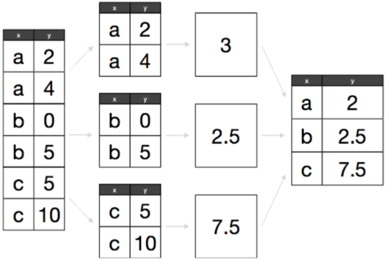
Learning to count

We'll focus on counting at small/medium scales on a single machine

But the same ideas extend to counting at large scales on many machines
(Hadoop, Spark, etc.)

Counting, the easy way

Split / Apply / Combine[1]



- Load dataset into memory
- Split: Arrange observations into groups of interest
- Apply: Compute distributions and statistics within each group
- Combine: Collect results across groups

[1] http://bit.ly/splitapplycombine

Examples

How much time and space do we need to compute per-group averages?

Examples

How much time and space do we need to compute per-group averages?

What about per-group variances?

The generic group-by operation

Split / Apply / Combine

```
for each observation as (group, value):
  place value in bucket for corresponding group

for each group:
  apply a function over values in bucket
  output group and result
```

The generic group-by operation

Split / Apply / Combine

```
for each observation as (group, value):
  place value in bucket for corresponding group

for each group:
  apply a function over values in bucket
  output group and result
```

Useful for computing arbitrary within-group statistics when we have required memory

(e.g., conditional distribution, median, etc.)

Why counting?

Anatomy of the Long Tail: Ordinary People with Extraordinary Tastes

Sharad Goel‡, Andrei Broder†, Evgeniy Gabrilovich†, Bo Pang†

‡ Yahoo! Research, 111 West 40th Street, New York, NY 10018, USA
† Yahoo! Research, 4301 Great America Parkway, Santa Clara, CA 95054, USA

{goel, broder, gabr, bopang}@yahoo-inc.com

ABSTRACT

The success of "infinite-inventory" retailers such as Amazon.com and Netflix has been ascribed to a "long tail" phenomenon. To wit, while the majority of their inventory is not in high demand, in aggregate these "worst sellers," unavailable at limited-inventory competitors, generate a significant fraction of total revenue. The long tail phenomenon, however, is in principle consistent with two fundamentally different theories. The first, and more popular hypothesis, is that a majority of consumers consistently follow the crowds and only a minority have any interest in niche content; the sec-

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms

Economics, Measurement

Keywords

Long tail, infinite inventory

Example: Anatomy of the long tail

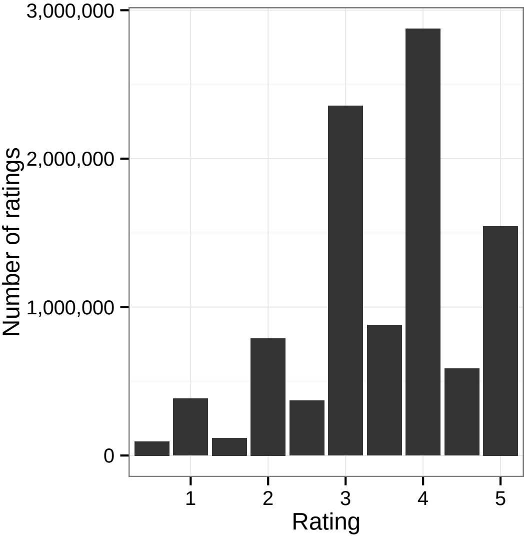
Dataset	Users	Items	Rating levels	Observations
Movielens	100K	10K	10	10M
Netflix	500K	20K	5	100M

Example: Anatomy of the long tail

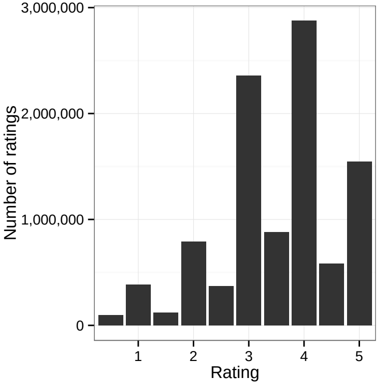
Dataset	Users	Items	Rating levels	Observations
Movielens	100K	10K	10	10M
Netflix	500K	20K	5	100M

Example: Movielens

How many ratings are there at each star level?



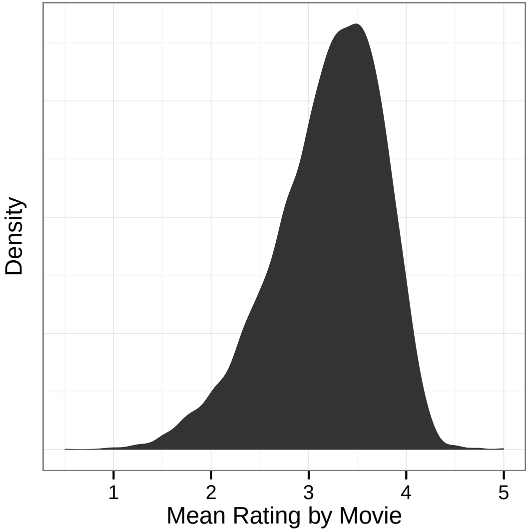
Example: Movielens




```
group by rating value
for each group:
  count # ratings
```

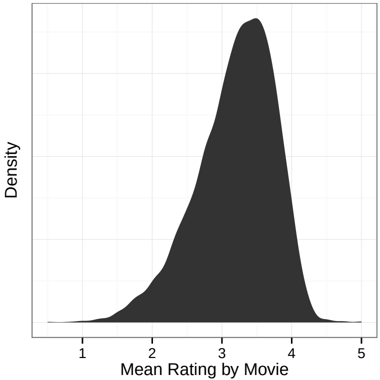
Example: Movielens

What is the distribution of average ratings by movie?



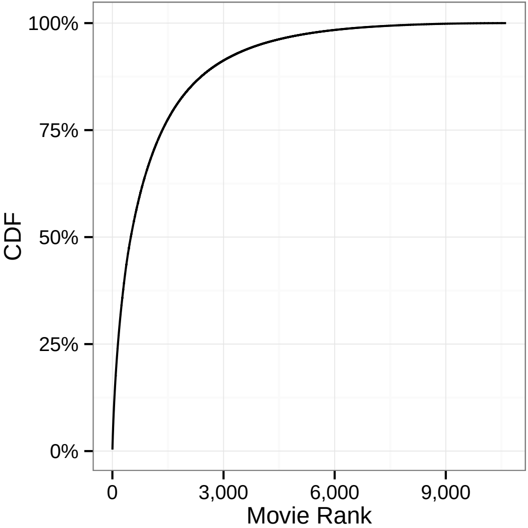
Example: Movielens

```
group by movie id
for each group:
  compute average rating
```

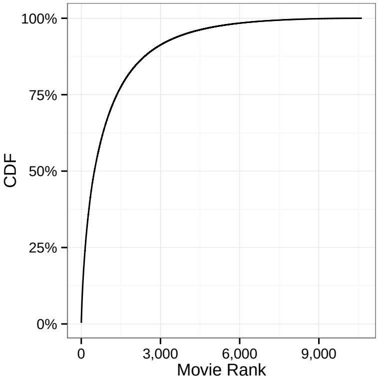


Example: Movielens

What fraction of ratings are given to the most popular movies?



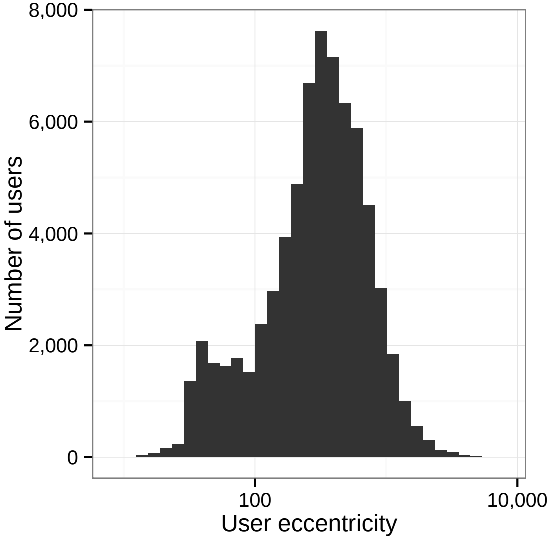
Example: Movielens




```
group by movie id
for each group:
  count # ratings

sort by group size

cumulatively sum group sizes
```

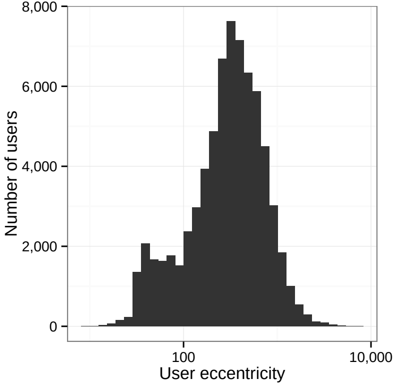
Example: Movielens

What is the median rank of each user's rated movies?



Example: Movielens

```
join movie ranks to ratings

group by user id
for each group:
  compute median movie rank
```



Example: Anatomy of the long tail

Dataset	Users	Items	Rating levels	Observations
Movielens	100K	10K	10	10M
Netflix	500K	20K	5	100M

What do we do when the full dataset exceeds available memory?

Example: Anatomy of the long tail

Dataset	Users	Items	Rating levels	Observations
Movielens	100K	10K	10	10M
Netflix	500K	20K	5	100M

What do we do when the full dataset exceeds available memory?

Sampling?

Unreliable estimates for rare groups

Example: Anatomy of the long tail

Dataset	Users	Items	Rating levels	Observations
Movielens	100K	10K	10	10M
Netflix	500K	20K	5	100M

What do we do when the full dataset exceeds available memory?

Random access from disk?
1000x more storage, but 1000x slower[2]

[2] Numbers every programmer should know

Example: Anatomy of the long tail

Dataset	Users	Items	Rating levels	Observations
Movielens	100K	10K	10	10M
Netflix	500K	20K	5	100M

What do we do when the full dataset exceeds available memory?

Streaming

Read data one observation at a time, storing only needed state

The combinable group-by operation

Streaming

```
for each observation as (group, value):
  if new group:
    initialize result

  update result for corresponding group as function of
  existing result and current value

for each group:
  output group and result
```

The combinable group-by operation

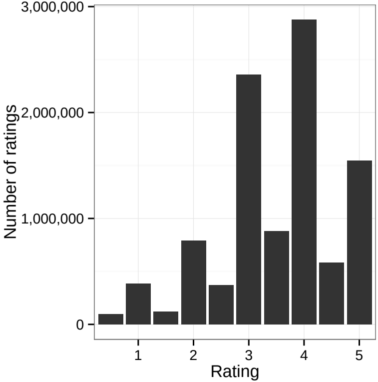
Streaming

```
for each observation as (group, value):
  if new group:
    initialize result

  update result for corresponding group as function of
  existing result and current value

for each group:
  output group and result
```

Useful for computing a subset of within-group statistics with a limited memory footprint

(e.g., min, mean, max, variance, etc.)

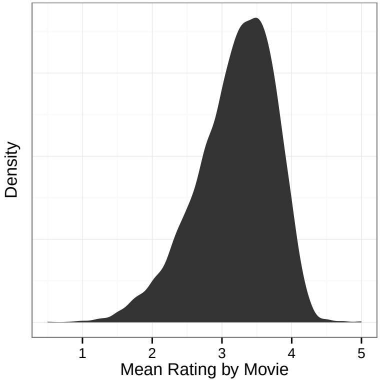
Example: Movielens




```
for each rating:
  counts[movie id]++
```

Example: Movielens

```
for each rating:
  totals[movie id] += rating
  counts[movie id]++

for each group:
  totals[movie id] /
  counts[movie id]
```



Yet another group-by operation

Per-group histograms

```
for each observation as (group, value):
  histogram[group][value]++

for each group:
  compute result as a function of histogram
  output group and result
```

Yet another group-by operation

Per-group histograms

```
for each observation as (group, value):
  histogram[group][value]++

for each group:
  compute result as a function of histogram
  output group and result
```

We can recover arbitrary statistics if we can afford to store counts of all distinct values within in each group

The group-by operation

For arbitrary input data:

Memory	Scenario	Distributions	Statistics
N	Small dataset	Yes	General
V*G	Small distributions	Yes	General
G	Small # groups	No	Combinable
V	Small # outcomes	No	No
1	Large # both	No	No

N = total number of observations
G = number of distinct groups
V = largest number of distinct values within group

Examples (w/ 8GB RAM)

Median rating by movie for Netflix

$N \sim$ 100M ratings
$G \sim$ 20K movies
$V \sim$ 10 half-star values

$V * G \sim$ 200K, store per-group histograms for arbitrary statistics

(scales to arbitrary N, if you're patient)

Examples (w/ 8GB RAM)

Median rating by video for YouTube

$N \sim$ 10B ratings
$G \sim$ 1B videos
$V \sim$ 10 half-star values

$V * G \sim$ 10B, fails because per-group histograms are too large to store in memory

$G \sim$ 1B, but no (exact) calculation for streaming median

Examples (w/ 8GB RAM)

Mean rating by video for YouTube

$N \sim 10$B ratings
$G \sim 1$B videos
$V \sim 10$ half-star values

$G \sim 1$B, use streaming to compute combinable statistics

The group-by operation

For pre-grouped input data:

Memory	Scenario	Distributions	Statistics
N	Small dataset	Yes	General
V*G	Small distributions	Yes	General
G	Small # groups	No	Combinable
V	Small # outcomes	Yes	General
1	Large # both	No	Combinable

N = total number of observations

G = number of distinct groups

V = largest number of distinct values within group