Experimental Analysis of Algorithms
A Statistical Perspective

Siddhartha Dalal

February 7, 2008

Outline

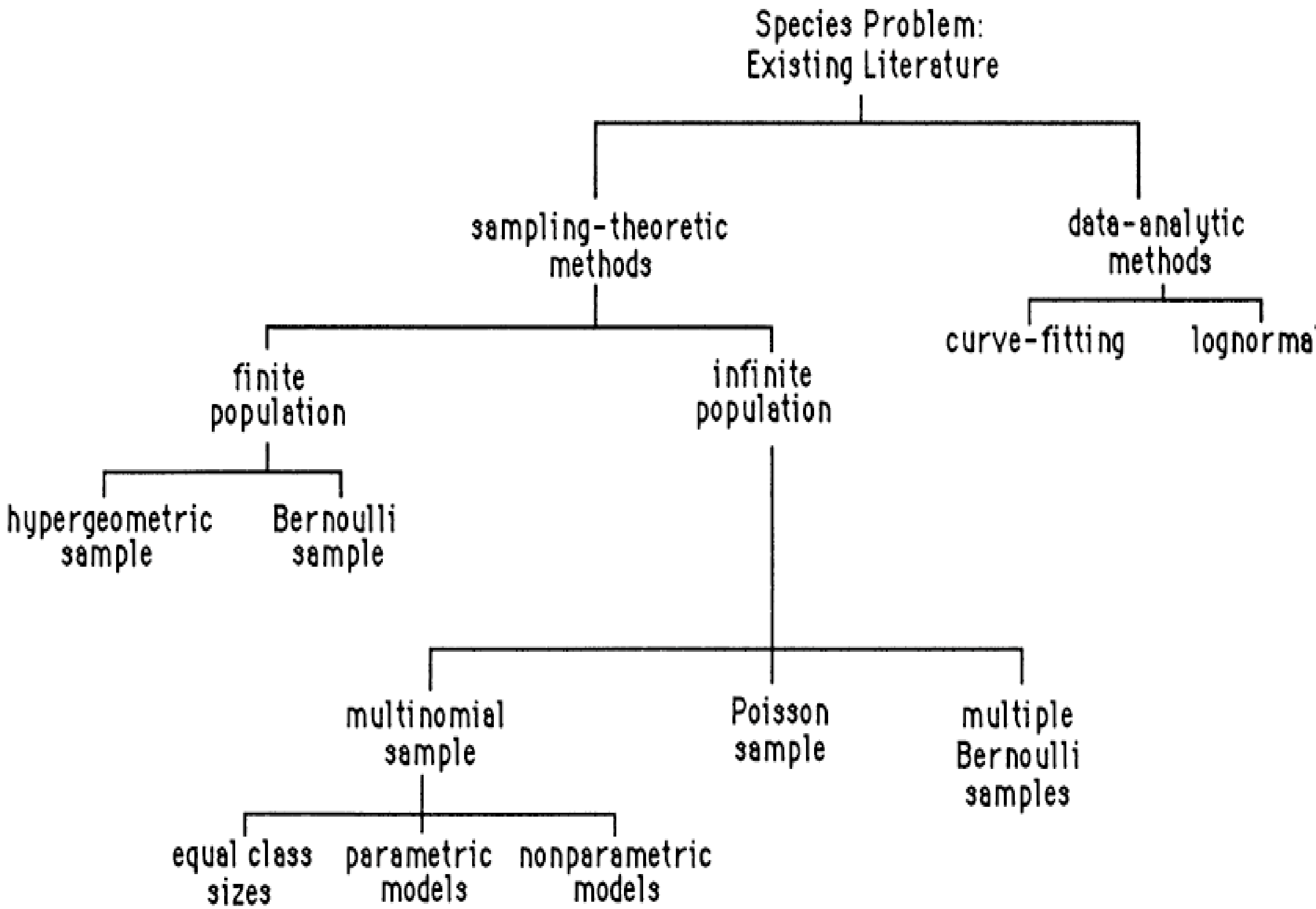
- Streaming: Distinct Value Problem
- Performance Tuning: Experimental Analysis- a case study and a tutorial
- Optimization and Testing: A new class of "Combinatorial Design" methodology which radically reduces the # of experiments to be done

Streaming: Distinct Value Problem

- Combinatorial Explosion Problem
- n parameters, k values each- k^n possible combinations: e.g., search engines, linguistics
- When one needs distinct values? Of what? Heavy hitters more important
- Hash function?
- Data is already a sample
- Related Problems: Infinite # Animals- Estimate how often a species occurs in the population based on a sample of size *n*
  - Species Problem: Good, Turing
  - Difficulty, what to do with values that do not occur in the sample. can not be solved by a single sample problem

Distinct Value Problem: Canonical Form

- N distinct values, each occurring $M_1,\ldots,M_N$
- Questions:
  - Estimate M_i
  - Estimate N
  - What can we do with single-stage sampling? Multistage sampling?
  - Do we need to sample all the entries?
- Examples:
  - Database, N for distinct values, M for heavy hitters Marios
  - Software Testing- N=2, M_2 is large, M_1 bugs



. *Existing Literature on the Problem of Estimating the Number of Classes in a Population, as Discussed in Section 1.*

Three Distinct Value Related Problems

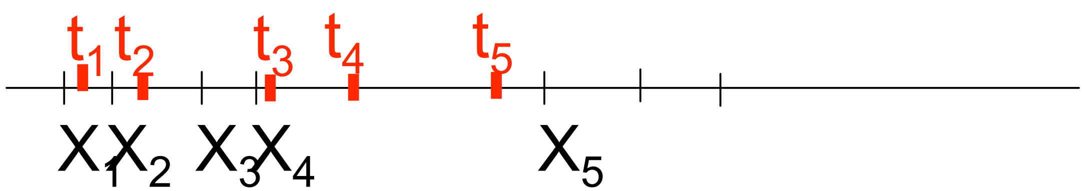
- Fixed Sample Problem:
  - n_r= # of species occurring *r* times in the sample of *n, n* is large
  - Good-Turing-Robins Estimate of $p_{r,}$ *the expected population probability is not r/n, but,* approx= r^*/n, $r^*=(r+1)n_{r+1}/n_r$

- Two sample problem
  - Capture-recapture problem (See Chao, A. (2001), An overview of closed capture-recapture models. J. Agricultural Biological Environmental Statist. v6. 138-155).

- Sequential Sampling:
  - If we don't want to check every entry then when can we stop and still guarantee that when we stop

  Pr{# of remaining entries for heavy hitter <= m} = 1- α

Optimal Sequential Sampling: How long to sample to estimate a particular distinct value?



Continue at t if we find many items/small gaps

- Time to find a particular item of class j, is approximately Exponential.

- Optimality: Optimal amongst a large class of sequential procedures with linear loss function- see referene

RAND

Dalal, S. R. and Mallows, C. L. (1992, 2008). Optimal Stopping with Exact Confidence on remaining defects. .

Performance Tuning: Experimental Design

- Quick Comments: Performance Tuning: Case Study:
  - How to improve performance of software systems, A methodology and a case study for tuning performance, Dalal, Hamada, Wang, *Annals of Software Engineering*

- Most of the work discuss designed for one, two or at most three parameters, e.g, Catherine- n, U[a,b]

- Real life algorithms need many parameters and then there is a combinatorial explosion

- Is there anyway to reduce the experimental runs?

- Combinatorial Designs, Factor Covering Designs-

A New Class of Combinatorial Designs for exploring large high dimensional spaces:

Potential Solution:
Orthogonal Arrays?
7 Fields 2 inputs: 2^7 cases

Tests	F1	F2	F3	F4	F5	F6	F7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

- Not efficient
- Often doesn’t exist
- Only for pairwise
- No constraint/netsting

- New Combinatorial Design Testing
- Forget about balance
- Valid for higher order interactions

Orthogonal Arrays vs. Combinatorial Designs in AETG System

7 Parameters 2 inputs

Tests	F1	F2	F3	F4	F5	F6	F7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

Tests	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2
3	1	2	2	2	1	1	1	2	2	2
4	2	1	2	2	1	2	2	1	1	2
5	2	2	1	2	2	1	2	1	2	1
6	2	2	2	1	2	2	1	2	1	1

- With 10 test cases can cover 126 parameters with 2 inputs
- General Question: Unequal *l*'s, constraints, nth order combinations?

Dalal-Mallows: 16 parameters 3 value design- A_{17}

Param. / Tests	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
T1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T2	1	2	2	2	1	2	2	2	1	2	2	2	1	2	2	2
T3	1	3	3	3	1	3	3	3	1	3	3	3	1	3	3	3
T4	2	1	2	3	2	1	2	3	2	1	2	3	2	1	2	3
T5	2	2	3	1	2	2	3	1	2	2	3	1	2	2	3	1
T6	2	3	1	2	2	3	1	2	2	3	1	2	2	3	1	2
T7	3	1	3	2	3	1	3	2	3	1	3	2	3	1	3	2
T8	3	2	1	3	3	2	1	3	3	2	1	3	3	2	1	3
T9	3	3	2	1	3	3	2	1	3	3	2	1	3	3	2	1
T10	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2
T11	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3
T12	2	2	2	2	1	1	1	1	2	2	2	2	3	3	3	3
T13	2	2	2	2	2	2	2	2	3	3	3	3	1	1	1	1
T14	2	2	2	2	3	3	3	3	1	1	1	1	2	2	2	2
T15	3	3	3	3	1	1	1	1	3	3	3	3	2	2	2	2
T16	3	3	3	3	2	2	2	2	1	1	1	1	3	3	3	3
T17	3	3	3	3	3	3	3	3	2	2	2	2	1	1	1	1

5314 Two level parameters can be added without increasing the experiment size

References

References on Stopping rules for determination of distinct values

- Bunge, J. M. Fitzpatrick Estimating the Number of Species: A Review, J. Am. Statist. Assoc., (Mar., 1993), pp. 364-373
- Chao, A. (2001), An overview of closed capture-recapture models. J. Agricultural Biological Environmental Statist. v6. 138-155
- Dalal, S. R. and Mallows, C. L. (1992) Buying with Exact Confidence. Ann. Appl. Prob. ,2, pp752-765.
- Dalal, S. R. and Mallows, C. M. (2008). Sequential screening for defects with exact confidence: Unknown scale, Technometrics

References on Parameter Tuning: Tutorial and a case study

1. Dalal, S. R., Hamada, M. and Wang, T. J. (1999) How to improve performance of software systems: A methodology and a case study for tuning performance. Annals of Software Engineering 8, 53-84

References on Combinatorial Design Testing:

1. Cohen, D. M., Dalal, S. R., Parelius J., and Patton G. C. (1996), The Combinatorial Design Approach to Automatic Test Generation, *IEEE Software*
2. Cohen, D. M., Dalal, S. R, Fredman M. L., AND Patton, G. C. (1997). The AETG system: An Approach to Testing Based on Combinatorial Designs, IEEE Transactions of Software Engineering, 23, 437-44
3. Dalal, S. R. and Mallows, C. M. (1998). Factor Covering Designs for Software Testing. Technometrics, 40, 234-243