Data Mining Seminar : Sampling
Jeff M. Phillips

OVERVIEW
Aug 22, 2012
---

0. Aug 22 (today)
Overview

0.5 Aug 29
Jeff is at conference, not seminar.

---
1. Sept 05
What properties can be recovered from random sampling? What cannot?
 - if data is from much bigger distribution, only the first type interesting !

 ==== density based estimates ====
   + what fraction of points satisfy this property?
   + do more that X fraction of points satisfy this property?
   + what objects have high frequency?

 ==== shape estimates ====
   + what is most extreme point?
   + score of k-means clustering?

---
2. Sept 12
Importance and Rejection Sampling:
  - how to improve sampling with clever weighting.

Importance Sampling:
  Some areas more important than others (specific questions)
  a. select proportionally more points from important areas
  b. proportionally down-weight points from important areas
   -> lower variance in important areas

Rejection Sampling:
  Computational Tool for complex distribution.
  a. Sample from simple distribution close to true distribution.
  b. Keep those points with some probability (based on relative
probabilities).

---
3. Sept 19 (Yan)

Markov Chains: Definitions
  - powerful rejection sampling for really crazy distributions.

X = state space
P : X -> X = transition probability matrix
q = state

Starting at q_0 in X, move q_{i+1} = P q_i. Where is q_\infty likely to end up?
What are high-level properties or when this is not "strange"

----
4. Sept 26 (Haibo)
Markov Chains: Rapidly Mixing + Convergence
  - how long do we run this thing?

q_n = P^n q_0. As n->nifty, for any start q_0, the same q_{infty} is reached.
How long does this take.

If for **all** states x in X, and any q_0, if
   |Pr[q_infty = x] - Pr[q_n = x]| / |Pr[q_infty = x] - Pr[q_{2n} = x]| < 2
for all n > n_0, then (P,X,q_0) is said to be "rapidly mixing".

What does this mean?

Some scenarios...

----
5. Oct 03 (Parasaran)
Metropolis Algorithm + Gibbs
   - how do we use this to sample from weird distributions?

Given "probe-only distribution"   mu
  - can only pick point x, and evaluate phi(x)
    such that phi(x) = c mu(x)   (c is fixed but unknown)

Then y = P * x is an algorithm:
  - choose random q (nearby x perhaps)
    if (phi(q) > phi(x)) -> y=q
    else (w.p. phi(q)/phi(x)) y=q
         otherwise y=x   (stays put)

Gibbs, X = multi-dimensional:
  -> Speed up if, we fix subset of dimensions, we can sample from non-fixed set exactly and efficiently (rejection sampling, or direct).
  -> Fix each set in turn = one-step.
      No rejection...

---
6. Oct 17 (XXXXX)
Advanced Sampling
- what techniques are state of the art for speeding this up?

Main problem is "auto-correlation".
a) trying to move too far, and not moving
b) only trying nearby, and not exploring space.

Think of q as random Gaussian kernel around x. What is variance?
- if change dynamically, may not converge!

Have small probability of changing variance in P. (e.g. double, cut-by-half)
(like running many in parallel, at diff variance)

---
7. Oct 24 (Prasana)
Coupling from the Past
- selecting a "true" random sample

To sample: run many many steps, then take 1 sample, start over again for next.
in practice: run 1000 steps (burn-in), then retain all following samples (e.g. 5000)

What can we say in theory - can we ever guarantee a truly random sample: Yes!

Pretend we started chains at all q_0 in X. Use same randomness for all.
If they all reach same spot, the next is random.
(bit more subtle)

continuous state spaces X too.

---
8. Oct 31 (Reza)
Reservoir Sampling
- one-pass random sample

Streaming model: get to read each data point once. Memory too small to store all.
Important: (a) network traffic (actually like this), (b) I/O Efficient.

How do we draw a random sample: Keep ith point w.p. k/i (if kept, kick out random point)

How do we do this in parallel, when data is distributed.

... or L_p sampling.

---
9. Nov 7   (MiaoMiao)
Variance-Optimal Sampling
  - best way to sample weighted points

Stream of points, with weights -> output: weighted subset of points (diff weight)

Want subset, so estimate of sum of weights in any subset preserved.
 - select high weight points with prob 1.
 - select remaining points proportional to weight.

For any subset query
 - right expected value
 - minimize variance of estimate

can you trade off zero-bias for less variance?
    "regularization"

---
10. Nov 14 (Suresh)
Precision Sampling
  - variation of VarOpt Sampling with negative weights

Maintains set estimates hat a_i with "precision" u_i (error in count)
 - needs 1/u_i space for each
 - averages them to get error bound
 - each an improvement over random sample

---
11. Nov 21   (Samira)
Discrepancy
  - get same (density) guarantees, with fewer points.

(P, R) is range space (subsets R of P, e.g. contained in ball)
 chi: P -> {-1,+1}

disc_chi(P,R) = max_{r in R} | sum_{p in P cap r} chi(p) |
  (as close to 0 as possible)

Remove points colored -1, now size |P|/2.
 Repeat on these points for about log(|P|) rounds

--> for good chi, better than random sampling on (P,R).

```
----------------------
12.  Nov 28   (XXXXX)
Constructing the Coloring
  - how to construct this coloring

Hyperbolic Cosine:
 H(r) = cosh( (sqrt(ln |R| / |P|) * disc_chi(P,r) )
 H = sum_{r in R} H(r)
 choose point that decreases H most

 cosh(x) = (e^x + e^{-x})/2

Edge Walking:
  produces "optimal" discrepancy
  Special Markov Chain on relaxed coloring   chi : P -> [-1,1]
   when point hits {-1,1} becomes fixed

-----------------------
13.  Dec 05   (XXXXX)
Streaming Weighted points for Ranges
  - beats random sampling in stream with structure

Pair-aggregation.  Pairs points in similar ranges.
  if p_i + p_j < 1  : p_i = p_i + p_j // p_j = 0
So don't sample both.  Reduces variance.

  (idea should be similar to discrepancy -- often constructs pairing +1,-1)
```