Data Mining Seminar : Sampling
Jeff M. Phillips

What Properties are Maintained by Random Sampling
Sept 05, 2012

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?

Let P be the phenomenon we are sampling from.
  Q subset P is sample.

R = class of subsets of P "ranges"
  ~ geometric ranges (balls, rectangles, half-spaces) | intervals
  ~ kernels (weighted subsets)
  ~ "dual" P is rectangles and r in R is point "stabbing"
  ~ simple combination of these ranges

---- density -------
for r in R   let
 r(P) = | r cap P | / | P |
be fraction of objects from P in range r
  "density"

want property:
 ** for any range r in R **
 | r(P) - r(Q) | < eps
for some parameter eps in [0,1]   (think of eps = 0.01 = 1/100)

 Q1: Can we do this?
 Q2: How big does Q need to be?

 A1: Yes, if R is "reasonable"

basically if |P| = n , finite, then |R| < n^v
bounded VC-dimension (basis of learning theory)

balls v = d+1
axis-aligned rectangles v = 2d
half-spaces v = d+1
P = rectangles v = 2d
v ~ description complexity
~ degrees of freedom

most important:
intervals v = 2

A2: |Q| = (1/eps^2)(v + log (1/delta))
eps = maximum error
delta = probability of failure (success w.p. > 1-delta)
v = VC-dimension
[Vapnik-Chervonenkis 1971 --> Li, Long, Srinivasan 2001 (via Talagrand 1984)]

eps-sample aka eps-approximation aka agnostic sampling

Chernoff Bound:
r independent events X1, X2, ..., Xr
A = (1/t) sum_i Xi
Xi in [0,1]
Pr[A - E[A] < eps] < 2 exp(-2 eps^2 t) < delta

X_i is 1 if sample i in r, 0 if X_i not in r (contribution to r(Q))
solve for |Q| = t > (1/2) (1/eps^2) ln (2/delta)

Frequent Objects

Let R_eps subset R such that r(P) > eps
R_eps = {r in R | r(P) > eps}

Want Q such that
** for all r in R_eps **
r(Q) > 0

we "hit" all large enough ranges.

Q1: Can we do this?

Q2: How large does Q need to be?

A1: Clearly yes if R satisfies above (v is bounded)

if eps-sample
  |r(P) - r(Q)| < eps
-> if r(P) > eps -> r(Q) > 0

Small discrete sets (only m possible values) also work.
  Here also v = 1
  since at most n+1 distinct ranges with different subsets.

A2: |Q| = (v/eps) log (v/eps * delta)
  eps = maximum error
  delta = probability of failure (success w.p. > 1-delta)
  v = VC-dimension
  [Hausler + Welzl 1986]

eps-net aka heavy-hitters aka noiseless-learning

discrete sets: heavy-hitters are all sets which occur more than eps*n times.

Note: We might accidentally hit the small sets, or over-sample large sets.

.... proof from Chernoff bound - small sets are easier class .....
.... similar to Coupon Collectors problem .....

----

extreme points:
  max value of set. Will sample recover?
  Average value. <sometimes, if variance is low / bounded>
  k-means cluster. Can you sample Q subset P
  run C = k-means(Q)
  compare average cost | |P|/|Q| cost(C,Q) - cost(C,P) | ?

  No?

Trick: don't try to recover density, since won't work.
Sample C directly:

basic: for max, just choose maximum point.
  No need to sample.

approx-convex hull:
  Let u be a unit vector in some direction

```
        wid(u,P) = max_{p in P} <u,p>  -  min_{p in P} <u,p>
        eps-kernel Q:
            for **any** u :    (wid(u,P) - wid(u,Q))/wid(u,P)  < eps
        (other forms, but this settled upon)

        1.  normalize so P fits in [-1,1]^d
        2.  place (1/eps)^{(d-1)/2} points G evenly on each side of [-2,2]^d
        3.  Select to Q the closest point in P to each g in G

        or
        2b. Take one point from each [eps]^d grid cell

k-means clustering:

let phi_C : P -> C
    phi_C(p) = argmin_{c in C} ||p - c||

construct C_1  = c_1 at random (q in P)
   C_{i+1} = C_i cup c_{i+1}
      choose c_{i+1} proportional to (phi_{C_i}(p))^2

cost(C,P) is 8-approximation to optimal centers
  cost(C,P) = sum_{p in P} phi_C(p)


more general:
  phi(p)  = sensitivity of p
     ~  phi(p) proportional to how cost(Q) changes to cost(Q / p)
        where Q is random subset
     ~  "extreme" have higher phi(p)
     ~  points near many other points have lower phi(p)
  complicated to describe in some specific settings


-----------------------
Question:  MAP Estimate?
```