Intelligent Sampling of Big Data

Ashwin Satyanarayana

Department of Computer Systems Technology (CST)

Intelligent Sampling for Big Data using Bootstrap Sampling and Chebyshev Inequality

Ashwin Satyanarayana
Computer Systems Technology Dept.
New York City College of Technology (CUNY)

IEEE Canada CCECE 2014
Canadian Conference on Electrical and Computer Engineering

Toronto, ON, Canada
4 to 7 May, 2014

The Curse of Large Datasets

There are two main challenges of dealing with large datasets.

- Running Time: Reducing the amount of time it takes for the mining algorithm to train.
- Predictive Accuracy: Poor quality of instances in training data lead to lower predictive accuracies. Large datasets are usually heterogeneous, and subjected to more noisy instances [Liu, Motoda DMKD 2002].

The goal of this work is to address these questions and provide some solutions.

Two Potential Solutions

- There are two potential solutions to the problems
- Running time:
  - Scale up existing data mining algorithms (for e.g. parallelize them)
  - Scale down the data

Intelligent Sampling

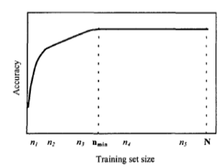
- Predictive accuracy:
  - Better mining algorithms
  - Improve the *quality* of the training data

Bootstrapping

We shall focus on first scaling down the data, while improving the quality

Learning Curve Phenomenon

- Is it necessary to apply a pattern recognition algorithm to *all* of the available data?
- A *learning curve* depicts the relationship between sample size and accuracy [Provost, Jensen & Oates 99].
- Problem: Determining n_{min} efficiently



Given a data mining algorithm M, a dataset D of N instances, we would like the smallest sample D_i of size n_{min} such that:

$$Pr(acc(D) - acc(D_i) > e) = d$$

e is the maximum acceptable decrease in accuracy (approximation) and d is the probability of failure

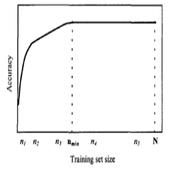
Definition: Chebyshev Inequality

- Definition 1: *Chebyshev Inequality* [Bertrand and Chebyshev-1845]: In *any* probability distribution, the probability of the estimated quantity p' being more than epsilon far away from the true value p after m independently drawn points is bounded by:

$$\Pr[|p - p'| \geq \varepsilon] \leq \left(\frac{\sigma^2}{\varepsilon^2 \cdot p'^2}\right)\frac{1}{m} \leq \delta$$

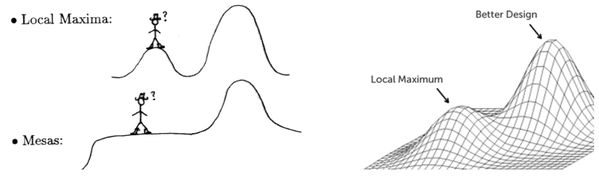
What are we trying to solve?
$Pr(acc(D) - acc(D_{nmin}) > e) = d$

Challenge: how do we compute the accuracy of the entire dataset* acc(D)*?

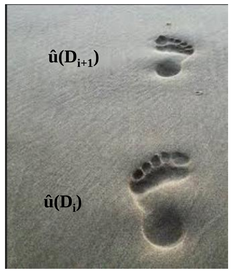


(a) **Sampling schedule** ? ***adaptive*** to the dataset under consideration.
(c) Adaptive **stopping criterion** to determine when the learning curve has reached a point of diminishing returns.

Intelligent Dynamic Adaptive Sampling



Myopic Strategy: One Step at a time



$$\hat{u}(D_i) = \frac{1}{|D_i|}\sum_{i=1}^{|D_i|} f(x_i)$$

[Average over the sample D_i]

The instance function $f(x_i)$ used here is a Bernoulli trial, 0-1 ***classification accuracy***

Approximation parameter — **Confidence parameter**

$$\Pr[|p - p'| \geq \varepsilon] \leq \left(\frac{\sigma^2}{\varepsilon^2 \cdot p'^2}\right)\frac{1}{m} \leq \delta$$

Plateau Region: p=0

True value $u(D_a)$-$u(D_b)$; where $|D_a|>|D_b|>n_{min}$

Myopic Strategy: $u(D_i)$ - $u(D_{i-1})$

Estimated value $u(D_a)$-$u(D_b)$; where $|D_a|>|D_b|=1$

Bootstrapped Variance p' *(to improve the quality of the training data)*

$$\Pr\left[\left|0 - \left(u(D_i) - u(D_{i-1})\right)\right| \geq \varepsilon\right] \leq \left(\frac{\sigma^2_{BOOT}}{\varepsilon^2 \cdot (u(D_i) - u(D_{i-1}))^2}\right)\frac{1}{m} \leq \delta$$

$$m \geq \left(\frac{\sigma^2_{BOOT}}{\varepsilon^2 \cdot (u(D_i) - u(D_{i-1}))^2}\right)\frac{1}{\delta}$$

Four Possible cases for Convergence

(a) $|p - p'| \leq \varepsilon$ and $\left(\frac{\sigma^2}{\varepsilon^2 \cdot p'^2}\right)\frac{1}{m} \leq \delta \Rightarrow$ Converged

(b) $|p - p'| > \varepsilon$ and $\left(\frac{\sigma^2}{\varepsilon^2 \cdot p'^2}\right)\frac{1}{m} > \delta \Rightarrow$ Add more instances

(c) $|p - p'| \leq \varepsilon$ and $\left(\frac{\sigma^2}{\varepsilon^2 \cdot p'^2}\right)\frac{1}{m} > \delta \Rightarrow$ False Positive

(d) $|p - p'| > \varepsilon$ and $\left(\frac{\sigma^2}{\varepsilon^2 \cdot p'^2}\right)\frac{1}{m} \leq \delta \Rightarrow$ False Negative

Algorithm IDASA(D, ε, δ)

Input: Training dataset D, ε and δ

Output: Total number of instances and mean computation time (for convergence)

Step 0: $\hat{u}(D_0) \leftarrow 0$

Step 1: Randomly select (1/10)|D| instances (=$|D_1|$). Apply the learner (Decision Tree) and determine $\hat{u}(D_1)$

Step 2: For each iteration i (≥ 1) do:

(a) Check for convergence using the criteria:

Test	$\left(\frac{\sigma}{\varepsilon \cdot p'}\right)^2 \frac{1}{m} < \delta$	$\left(\frac{\sigma}{\varepsilon \cdot p'}\right)^2 \frac{1}{m} \geq \delta$
$\lvert\hat{u}(D_i) - \hat{u}(D_{i+1})\rvert < \varepsilon$	**Yes, EXIT**	**(False Positive)**
$\lvert\hat{u}(D_i) - \hat{u}(D_{i+1})\rvert \geq \varepsilon$	**(False Negative)**	Continue (Go to Step (b))

(b) Add $m \geq \left(\frac{\sigma^2_{BOOT}}{\varepsilon^2 \cdot (u(D_i) - u(D_{i-1}))^2}\right)\frac{1}{\delta}$ instances using to form the new sample D_{i+1}.

(c) Apply the classification mining algorithm on the sample and determine $\hat{u}(D_{i+1})$.

Empirical Results (for Non-incremental Learner)

Table 3.4. Comparison of the mean computation time required for the different methods to obtain the same accuracy (averaged over 20 runs of the experiment).

Mean Computation Time	Full: $S_N=\{N\}$	Arith: $S_a=\lvert D_1\rvert+k.\beta$	Geo $S_g=a^k.\lvert D_1\rvert$	DASA	Oracle $S_O=\{n_{min}\}$
DB3	**1821.3**	**956.7**	**870.2**	**368.2**	**312.1**
DB4	**1188.7**	**854.3**	**765.2**	**312.1**	**265.6**
NIST Special DB	**3580.4**	**4730.7**	**2082.3**	**989.7**	**887.6**
NIST Special DB	**2591.6**	**1753.9**	**1387.9**	**623.7**	**564.2**

Table 3.5. Comparison of the total number of instances required for the different methods to reach convergence.

Average No. of instances	Full: $S_N=\{N\}$	Arith: $S_a=\lvert D_1\rvert+k.\beta$	Geo $S_g=a^k.\lvert D_1\rvert$	DASA	Oracle $S_O=\{n_{min}\}$
DB3	**880**	**300**	**300**	**136**	**116**
DB4	**880**	**600**	**600**	**269**	**233**
NIST Special DB	**2000**	**2100**	**1500**	**686**	**585**
NIST Special DB	**1500**	**600**	**600**	**316**	**232**

Empirical Results (for Incremental Learners)

Table 1: Comparison of the total number of instances required for convergence by the different methods to obtain the same accuracy

Dataset	Full: $S_N=\{N\}$	Geo $S_g=a^k.\lvert D_1\rvert$	IDASA	Oracle $S_O=\{n_{min}\}$
LED	**100,000**	**6,300**	**5,100**	**2,000**
WAVEFORM	**100,000**	**25,500**	**16,108**	**12,000**
CENSUS	**32,000**	**25,500**	**10,014**	**8,000**
KDD CUP	**235,000**	**204,700**	**67,800**	**56,600**
NASA-HTTP	**461,612**	**409,500**	**158,345**	**130,645**

Table 2: Comparison of the mean computation time required for convergence by the different methods to obtain the same accuracy. Time is in CPU seconds (by the linux time command)

Dataset	Full: $S_N=\{N\}$	Geo $S_g=a^k.\lvert D_1\rvert$	IDASA	Oracle $S_O=\{n_{min}\}$
LED	**46.51**	**15.67**	**25.87**	**5.72**
WAVEFORM	**558.91**	**89.76**	**156.73**	**32.85**
CENSUS	**48.76**	**10.77**	**27.84**	**13.87**
KDD CUP	**17,870.59**	**5,616.89**	**3,116.84**	**1,826.16**
NASA-HTTP	**38,160.78**	**13,482.49**	**8,713.00**	**4,719.85**