Counting with Combined Splitting and Capture-Recapture Methods

Ad Ridder (*Vrije Universiteit Amsterdam*)

Paul Dupuis (Brown), Bahar Kaynar (VU), Reuven Rubinstein (Technion), Radislav Vaisman (Technion)

INFORMS Applied Probability
Stockholm; 7 July 2011

Introduction

vrije Universiteit amsterdam

- A (computional) *decision* problem in NP (satisfiability, graph with prescribed degrees, contingency table, permanent, graph coloring, Hamilton cycle, ...).
- Correponding *counting* problem asks for the total number of solutions to a given instance.
- Most often #P-complete.
- Approximate counting by randomized algorithms based on *splitting* and MCMC.
- Relation to *rare event simulation*.
- Presentation of an empirical study.
- Discussion of several algorithms and enhancement techniques.

Example (3-SAT problems)

vrije Universiteit amsterdam

$\mathbf{x} \in \{0,1\}^7$ to satisfy CNF with clauses of size 3, e.g.,

$$\left(x_1 \vee x_4 \vee \overline{x_7}\right) \wedge \left(x_2 \vee \overline{x_3} \vee \overline{x_5}\right) \wedge \left(x_1 \vee x_3 \vee x_6\right) \wedge \left(\overline{x_3} \vee x_5 \vee x_7\right).$$

Many vectors might satisfy, e.g., $\mathbf{x} = (0,1,1,0,1,0,0)$ satisfy these four clauses.

- Decision problem: determine whether (or not) there is a satisfiable vector $\mathbf{x}$.
- Counting problem: determine how many satisfiable vectors $\mathbf{x}$ there are.

Generally, n variables and m clauses. For instance $n = 75, m = 325$. Library of benchmark problems SATLIB.

Literature overview

vrije Universiteit amsterdam

1. Splitting techniques: well-known for rare-event simulation
   - Villén-Altamirano & Villén-Altamirano 1991-present (RESTART, queueing networks);
   - Melas 1993-2003 (mathematical theory related to branching processes);
   - Glasserman, Heidelberger, Shahabuddin & others 1995-1999 (multilevel-splitting, large deviations);
   - Cérou, Guyader, del Moral & others 2006-present (Feynman-Kac models, particle methods);
   - L'Ecuyer, Tuffin & others 2006-present (efficiency analysis);
   - Dean & Dupuis 2008 (subsolution; asymptotic optimality);
   - Botev & Kroese 2008-present (generalized splitting method);
   - Lagnoux-Renaudie 2007-present (two-phase approach; cost constraints);
   - Rubinstein 2006-present (Gibbs sampler; stochastic enumeration);
   - Other ...

Literature overview

vrije Universiteit amsterdam

2. Approximate counting using randomized algorithms
   - Dyer & others 1995-present (complexity analysis; Markov chain methods);
   - Selman, Wei, Gomes & others 2006-2010 (model counting; random walk; no uniformity of the samples);
   - Gogate & Dechter 2005-present (importance sampling; graphical models);
   - Blanchet 2007-present (contingency tables; importance sampling);
   - Rubinstein 2006-present (Gibbs cloner; stochastic enumeration);
   - Other ...

Counting and probability estimating

vrije Universiteit amsterdam

- *Solution set* (of the counting problem) is $\mathcal{X}^*$ of unknown size $\ell = |\mathcal{X}^*|$.
- Larger set (called *state space*) $\mathcal{X} \supset \mathcal{X}^*$ of known and computable size $|\mathcal{X}| < \infty$.
- Fraction $p = |\mathcal{X}^*| \,/\, |\mathcal{X}|$ is viewed as *probability*.
- Estimate ℓ by estimating p by $\hat{p}$ and setting $\hat{\ell} = |\mathcal{X}|\,\hat{p}$.
- Crude Monte Carlo: assume uniform sampling on $\mathcal{X}$ is easy. Then $\hat{p} = \frac{1}{N} \sum_{k=1}^{N} 1\{\mathbf{X}_k \in \mathcal{X}^*\}$.
- When p is small, we say $\mathcal{X}^*$ is a *rare event* in the state space. Crude Monte Carlo fails.

The idea of the splitting method

vrije Universiteit amsterdam

1. Find a sequence of sets $\mathcal{X} = \mathcal{X}_0, \mathcal{X}_1, \ldots, \mathcal{X}_T$ such that $\mathcal{X}_0 \supset \mathcal{X}_1 \supset \cdots \supset \mathcal{X}_T = \mathcal{X}^*$. ($T$ might be random.)
2. Write $|\mathcal{X}^*| = |\mathcal{X}_T|$ as the telescoping product

$$|\mathcal{X}^*| = |\mathcal{X}_0| \prod_{t=1}^{T} \frac{|\mathcal{X}_t|}{|\mathcal{X}_{t-1}|},$$

   thus the target probability becomes a product $p = \prod_{t=1}^{T} c_t$, with ratio factors (conditional probabilities)

$$c_t = \frac{|\mathcal{X}_t|}{|\mathcal{X}_{t-1}|}.$$

3. Develop an efficient estimator $\hat{c}_t$ for each c_t and estimate $|\mathcal{X}^*|$ by

$$\hat{\ell} = \widehat{|\mathcal{X}^*|} = |\mathcal{X}_0|\,\hat{p} = |\mathcal{X}_0| \prod_{t=1}^{T} \hat{c}_t.$$

Type of problems

We consider problems for which

- statespace $\mathcal{X} \subset \mathbb{Z}^n$;
- a *performance function* $S : \mathcal{X} \to \mathbb{R}$;
- the rare event is represented by $\mathcal{X}^* = \{S(\mathbf{x}) \geq m\}$ for some *level* m.

Adaptive splitting method

vrije Universiteit amsterdam

- The subsets $(\mathcal{X}_t)_{t=0}^{T}$ are given by $\mathcal{X}_t = \{S(\mathbf{x}) \geq m_t\}$.
- The levels $m_1 \leq \cdots \leq m_T = m$ are not preset in advance, but are determined during the simulation based on the performance values of the current sample. The initial level is $m_0 = -\infty$.
- More specific:
  - suppose $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is a sample set of N points randomly distributed in $\mathcal{X}_t$; thus $\mathbf{X}_k \overset{\mathrm{d}}{\sim} U(\mathcal{X}_t)$ and $S_k = S(\mathbf{X}_k) \geq m_t$ for all $k = 1, \ldots, N$;
  - determine the order statistics of the performance values $S_{(1)}, \ldots, S_{(N)}$;
  - let there be given a *splitting control parameter* $\rho \in (0, 1)$;
  - set the next level by $m_{t+1} = S_{(\lceil (1-\rho)N \rceil)}$;
  - all points in the sample set for which $S(\mathbf{X}_k) \geq m_{t+1}$ are called *elite points*.

Level simulation procedure

vrije Universiteit amsterdam

Let be given a point $\mathbf{x} \in \mathcal{X}_t$, i.e., $S(\mathbf{x}) \geq m_t$.

We assume that we have at our disposal a random mapping $\Psi_t : \mathcal{X}_t \to \mathcal{X}_t$ such that

$$\mathbf{X} \overset{\mathrm{d}}{\sim} U(\mathcal{X}_t) \;\Rightarrow\; \Psi_t(\mathbf{X}) \overset{\mathrm{d}}{\sim} U(\mathcal{X}_t).$$

Typically this is obtained by a MCMC simulation such as a Gibbs sampler.

An algorithm with cloning

vrije Universiteit amsterdam

Given are $\rho \in (0,1)$ splitting control parameter and $N_0 = N$ initial sample size.

1. *Initialisation*. Generate a sample set of N_0 points uniformly distributed in $\mathcal{X}_0$. Determine level m_1, and the associated elite sample set of size $N_1^{(e)}$. Set a counter $t = 1$.
2. *Cloning*. Given elite size $N_t^{(e)}$ of points $S(\mathbf{X}_k) \geq m_t$. Set *cloning parameter* $\eta_t = \lceil N_{t-1}/N_t^{(e)} \rceil$. Reproduce each elite point η_t times to obtain a sample set of $N_t = \eta_t N_t^{(e)}$ points in $\mathcal{X}_t$.
3. *Simulation*. Apply the level simulation procedure Ψ_t to each of the point $\mathbf{X}_k$ of the sample set. Denote the generated points again $\mathbf{X}_1, \ldots, \mathbf{X}_{N_t}$.

Algorithm 1 (cont'd)

vrije Universiteit amsterdam

4. *Elite selection*. Use the performance values $S_k = S(\mathbf{X}_k)$ for setting the next level $m_{t+1} = S_{(\lceil (1-\rho)N_t \rceil)}$. Determine the associated elite sample set of size $N_{t+1}^{(e)}$.
5. *Stopping rule*. Set $t = t+1$. If $m_t = m$ stop; else repeat from step 2.

Let $\hat{c}_t = N_t^{(e)}/N_{t-1}$ be an estimator of $c_t = |\mathcal{X}_t|/|\mathcal{X}_{t-1}|$, for $t = 1, 2, \ldots, T$. The final estimator of $p = |\mathcal{X}^*|/|\mathcal{X}|$ is

$$\hat{\ell} = \prod_{t=1}^{T} \hat{c}_t.$$

Adapted implementation of algorithm 1

vrije Universiteit amsterdam

A. Apply *screening* after elite selection and before cloning. This means that we delete all duplicates from the elite sample set.

B. Keep the sample size constant N. Hence, when $N_t^{(e)}$ is the elite size (after screening!), set the cloning parameter $\eta_t = \lfloor N/N_t^{(e)} \rfloor$, and choose randomly $N - \eta_t N_t^{(e)}$ elite points to clone one more time.

C. When level $m_t = m - 2$ (or $m - 1$) has attained, increase the sample size N and decrease splitting parameter ρ.

D. Adapt splitting parameter in all iterations, ρ_t.

Algorithm 1 with direct estimation

vrije Universiteit amsterdam

- Perform an extra iteration of size N_d at the end, after algorithm 1 has reached the desired level m.
- Hence, all points in the sample set satisfy $S(\mathbf{x}) \geq m$.
- Screen out the sample set to obtain $N_d^{(s)}$ distint points.
- Estimator $\hat{\ell} = N_d^{(s)}$.
- No need to keep track of intermediate conditional probability estimators $\hat{c}_t$.
- Applicable for small $|\mathcal{X}^*|$ sizes.

The capture-recapture method

vrije Universiteit amsterdam

- Consider an urn model with a total of M identical balls, where M is unknown.
- Execute two (independent) draws of sizes N_1 and N_2, respectively.
- The second draw takes place after all N_1 balls have been returned to the urn.
- Before returning the N_1 balls, each is marked, say we painted them a different color.
- Denote by R the number of balls from the first draw that reappear in the second.
- Then an (biased) estimate $\widetilde{M}$ of M becomes $\widetilde{M} = N_1 N_2 / R$.

Algorithm 1 with cap-recap estimation

vrije Universiteit amsterdam

- Apply capture-recapture with $M = |\mathcal{X}^*|$.
- Perform the two draws at the end, after algorithm 1 has reached the desired level m.
- No need to keep track of intermediate conditional probability estimators $\hat{c}_t$.
- Applicable for $|\mathcal{X}^*|$ sizes that are not too big.

Testing algorithm 1

vrije Universiteit amsterdam

A test problem with $n = 25$ variables for which exact count is tractable.

Regular sample $N = 1000$; increased to $N = 10000$ when $m_t = m - 1$.
Regular splitting parameter $\rho = 0.05$; decreased to $\rho = 0.005$ when $m_t = m - 1$.
Direct sample size $N_d = 2000$.
Capture-recapture sample sizes $N_1 = N_2 = 1000$.

Three estimators of $|\mathcal{X}^*|$:
product $\hat{\ell}_p$; direct $\hat{\ell}_d$; capture-recapture $\hat{\ell}_{cr}$.

$K = 100$ simulation runs per instance of the problem.
IID estimators $\hat{\ell}_e^{(i)}$, $i = 1, \ldots, K$, $e = p, d, cr$.
We report their averages $\overline{\ell}_e = (1/K) \sum_{i=1}^{K} \hat{\ell}_e^{(i)}$;
and the estimated relative errors of the $\hat{\ell}_e$ estimators (standard deviation over the mean).

Numerical results

vrije Universiteit amsterdam

Results with algorithm 1.

m	exact	$\overline{\ell}_p$	$\overline{\ell}_d$	$\overline{\ell}_{cr}$	RE$(\hat{\ell}_p)$	RE$(\hat{\ell}_d)$	RE$(\hat{\ell}_{cr})$
87	6	6.7739	5.9000	5.9000	0.3882	0.0743	0.0743
75	60	59.4389	57.3800	58.3500	0.4243	0.0522	0.0455
72	175	169.7680	168.9600	170.4007	0.3680	0.0303	0.0270
70	404	415.6734	384.1200	395.2937	0.2869	0.0288	0.0187
			391.9300	397.7324		0.0245	0.0140

(Last row: $N_d = 3000, N_1 = N_2 = 1500$ in stead of $N_d = 2000, N_1 = N_2 = 1000$)

Results with algorithm 2 (next slide).

m	exact	$\overline{\ell}_p$	$\overline{\ell}_d$	$\overline{\ell}_{cr}$	RE$(\hat{\ell}_p)$	RE$(\hat{\ell}_d)$	RE$(\hat{\ell}_{cr})$
87	6	6.3270	6.0000	6.0000	0.2580	0.0000	0.0000
75	60	60.9494	59.9800	59.9800	0.2364	0.0033	0.0033
72	175	174.3973	174.6000	174.5933	0.2354	0.0082	0.0083
70	404	403.7925	398.9800	402.4376	0.1778	0.0075	0.0074
			402.8200	403.2146		0.0048	0.0048

(Last row: $N_d = 3000, N_1 = N_2 = 1500$ in stead of $N_d = 2000, N_1 = N_2 = 1000$)

An algorithm with MCMC

vrije Universiteit amsterdam

1. *Initialisation*. Similar as in algorithm 1.
2. *Screening*. Similar as in algorithm 1.
3. *Simulation*. Set the *burn in parameter* $b_t = \lfloor N/N_t^{(e)} \rfloor$. For all $i = 1, 2, \ldots, N_t^{(e)}$, starting at the i-th screened elite point run a Markov chain of length b_t on $\mathcal{X}_t$ with level procedure Ψ_t as its transition operator. Extend $N - b_t N_t^{(e)}$ randomly chosen sample paths with one point. Denote all the generated points on the sample paths again $\mathbf{X}_1, \ldots, \mathbf{X}_{N_t}$.
4. *Elite selection*. Similar as in algorithm 1.
5. *Stopping rule*. Similar as in algorithm 1.

Final estimator: similar as in algorithm 1 (product, direct, or capture-recapture).

Other problems

vrije Universiteit amsterdam

We have experimented with our algorithms for

- Larger 3-SAT problems, for instance $n = 75, m = 325$.

	$\overline{\ell}_p$	$\overline{\ell}_d$	$\overline{\ell}_{cr}$	RE($\hat{\ell}_p$)	RE($\hat{\ell}_d$)	RE($\hat{\ell}_{cr}$)
alg. 1	2100.7202	2132.3000	2191.1752	0.3182	0.0244	0.0159
alg. 2	2263.3366	2223.8000	2254.4950	0.1809	0.0038	0.0038

- Random graphs with prescribed degrees: given positive integers $d_1, \ldots, d_n$, how many graphs $G = (V, E)$ are feasible with vertex set $|V| = n$, and edge set E, such that vertex i has degree d_i, for all $i = 1, \ldots, n$?
- Binary contingency tables: given two vectors of positive integers $\boldsymbol{r} = (r_1, \ldots, r_m)$ and $\boldsymbol{c} = (c_1, \ldots, c_n)$ such that $r_i \leq n$ for all i, $c_j \leq n$ for all j, and $\sum_{i=1}^{m} r_i = \sum_{j=1}^{n} c_j$. How many $m \times n$ binary contingency tables of entries $x_{ij} \in \{0, 1\}$ are feasible with row sums $\boldsymbol{r}$ and column sums $\boldsymbol{c}$?

Conclusion and outlook

vrije Universiteit amsterdam

- Randomized algorithms based on MCMC, splitting and capture-recapture;
- fast and excellent performance;
- current status: empirical;
- rigorous analysis of complexity and efficiency under development;