Mining Frequent Itemsets through Sampling with Rademacher Averages

Matteo Riondato (Two Sigma Investments, matteo@twosigma.com) and Eli Upfal (Department of Computer Science, Brown University, eli@cs.brown.edu)

1. What problem to we study?

Frequent Itemsets Mining: classic problem with many applications

Dataset $\mathcal{D}$
bread, milk
bread, beer, eggs
milk, beer, coke
bread, milk, beer
bread, milk, coke

Each line is a transaction of items from a set $\mathcal{I}$

An itemset is a subset of $\mathcal{I}$

The frequency $f_{\mathcal{D}}(A)$ of itemset A is the fraction of transactions containing A

E.g.: $f_{\mathcal{D}}(\{\text{bread,milk}\}) = 3/5 = 0.6$

Problem (Frequent Itemsets (FI) Mining):
Given $\theta \in [0,1]$ find (i.e., mine) all itemsets $A \subseteq \mathcal{I}$ with $f_{\mathcal{D}}(A) \geq \theta$
I.e., compute the set $\mathbf{FI}(\mathcal{D},\theta) = \{A \subseteq \mathcal{I} : f_{\mathcal{D}}(A) \geq \theta\}$

There are exact algorithms for FI mining (Apriori, FP-Growth, ...)

2. How to make FI Mining faster?

Exact algorithms do not scale with $|\mathcal{D}|$:
They scan $\mathcal{D}$ multiple times: slow on large $\mathcal{D}$

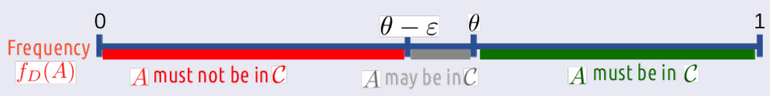
How to get faster? Trade-off accuracy for speed:
Only mine random samples of $\mathcal{D}$ that fit in main memory
Results are approximate but fast to compute
Approximation is OK: FI mining is a exploratory task

Key question:
How much to sample to get an approximation of the desired quality?

Our contribution:
A fast approximation algorithm for FI mining that uses Progressive Random Sampling and Rademacher Averages and has probabilistic guarantees on the accuracy of the output

3. How to define an approximation of the FIs?

For $\varepsilon,\delta \in (0,1)$, a (ε,δ)-approximation to $\mathbf{FI}(\mathcal{D},\theta)$ is a collection $\mathcal{C}$ of itemsets s.t., with prob. $\geq 1-\delta$:



"Close" False Positives are allowed, but no False Negatives
$\mathcal{C}$ can act as set of candidate FIs to prune with fast scan of $\mathcal{D}$

Our approximation algorithm for FI mining returns an (ε,δ)-approximation to $\mathbf{FI}(\mathcal{D},\theta)$ by using Progressive Random Sampling

4. What is Progressive Random Sampling (PRS)?

How much to sample from $\mathcal{D}$ to obtain an (ε,δ)-approximation?
Prev. works: loose sample size for worst-case dataset
Instead, let's start sampling: the data will tell us when to stop

Progressive Random Sampling is an iterative sampling scheme

Outline of PRS algorithm for approximating $\mathbf{FI}(\mathcal{D},\theta)$
At each iteration,

1. create sample $\mathcal{S}$ by drawing transactions from $\mathcal{D}$ uniformly and independently at random
2. Check a stopping condition on $\mathcal{S}$, to see if can get (ε,δ)-approximation from it
3. If stopping condition is satisfied, mine $\mathbf{FI}(\mathcal{S},\gamma)$ for some $\gamma < \theta$ and output it
4. Else, iterate with a larger sample

5. What are the challenges? What is our contribution?

The challenges are:

- Developing a stopping condition that
  - can be checked without expensive mining of each sample
  - guarantees that the output is a (ε,δ)-approximation
  - can be satisfied at small sample sizes
- Devising a method to choose the next sample size

Our contribution: We present the first algorithm that

- uses a stopping condition that does not mine each sample
- uses PRS to obtain an (ε,δ)-approximation of $\mathbf{FI}(\mathcal{D},\theta)$
- computes the optimal next sample size on the fly

Previous contributions gave no guarantees and/or required mining FIs from each sample (too expensive). They used predefined sample sizes

6. What do we really need?

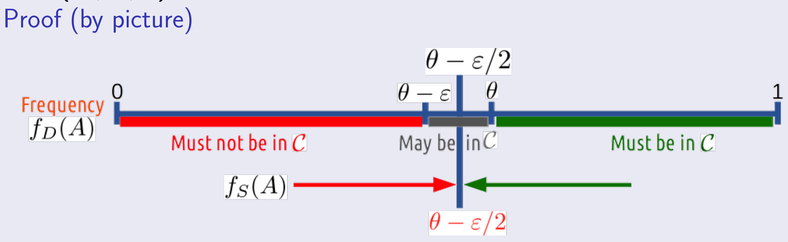
We need an efficient procedure that, given a sample $\mathcal{S}$ of $\mathcal{D}$, computes a value η s.t.

$$\Pr\left(\sup_{A\subseteq\mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)| \leq \eta\right) \geq 1-\delta$$

Then the stopping condition just tests if $\eta \leq \varepsilon/2$

Theorem: If $\eta \leq \varepsilon/2$, then $\mathbf{FI}(\mathcal{S}, \underbrace{\theta-\varepsilon/2}_{\gamma})$ is an (ε,δ)-approximation to $\mathbf{FI}(\mathcal{D},\mathcal{I},\theta)$

Proof (by picture)



How to compute η? Study statistics, learn about Rademacher Averages!

6. What are Rademacher Averages?

The behavior of $|\sup_{A\subseteq\mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)|$ has been extensively studied using VC-dimension, covering numbers, and Rademacher averages

For any itemset A and any transaction $\tau \in \mathcal{D}$, let

$$\phi_A(\tau) = \begin{cases} 1 \text{ if } A \subseteq \tau \\ 0 \text{ otherwise} \end{cases}$$

Let $\mathcal{S} = \{\tau_1,\ldots,\tau_n\}$ be a random sample of $\mathcal{D}$, and let $\sigma_1,\ldots,\sigma_n$ be independent Rademacher r.v. (1 with prob. $1/2$, -1 othw.)
The Rademacher Average on $\mathcal{S}$ is:

$$\mathbf{R}(\mathcal{S}) = \mathbb{E}_\sigma\left[\sup_{A\subseteq\mathcal{I}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \phi_A(\tau_i) \mid \mathcal{S}\right]$$

The expectation is taken wrt the σ_i, i.e., conditionally on $\mathcal{S}$

7. How to use the Rademacher average?

Theorem (Key result from Statistical Learning Theory): Let

$$\eta = 2\mathbf{R}(\mathcal{S}) + \sqrt{\frac{2\ln(2/\delta)}{n}}$$

Then

$$\Pr(\sup_{A\subseteq\mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)| \leq \eta_{\mathcal{S}}) \geq 1-\delta$$

η is a sample-dependent upper bound to $\sup_{A\subseteq\mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)|$

The key question is: how do we compute or bound $\mathbf{R}(\mathcal{S})$?
Computing $\mathbf{R}(\mathcal{S})$ efficiently allows us to compute η and check our stopping condition efficiently

8a. How can we bound the Rademacher average? (High-level)

We compute an upper bound to the distribution of the frequencies in $\mathcal{S}$ of the Closed Itemsets (CIs) in $\mathcal{S}$ (An itemset is closed iff none of its supersets has the same frequency)

Connection with the CIs:

$$\sup_{A\subseteq\mathcal{I}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)| = \sup_{A\in\text{CIs}} |f_{\mathcal{D}}(A) - f_{\mathcal{S}}(A)|$$

Efficiency Constraint: use only information that can be obtained with a single scan of $\mathcal{S}$

How:

1. We use the frequency of the single items and the lengths of the transactions to define a (conceptual) partitioning of the CIs into classes, and to compute upper bounds to the size of each class and to the frequencies of the CIs in the class
2. We use these bounds to compute an upper bound to $\mathbf{R}(\mathcal{S})$ by minimizing a convex function in $\mathbb{R}^+$ (no constraints)

8b. How to bound the Rademacher Averages? (In-depth)

For any itemset $A \subseteq \mathcal{I}$, let $\mathbf{v}_{\mathcal{S}}(A)$ be the n-dimensional vector

$$\mathbf{v}_{\mathcal{S}}(A) = (\phi_A(\tau_1),\ldots,\phi_A(\tau_n)),$$

and let $V_{\mathcal{S}} = \{\mathbf{v}_{\mathcal{S}}(A), A \subseteq \mathcal{I}\}$ ($V_{\mathcal{S}}$ is a set)

Theorem (Variant of Massart's Lemma):
Let $w : \mathbb{R}^+ \to \mathbb{R}^+$ be the function

$$w(s) = \frac{1}{s}\ln \sum_{\mathbf{v}\in V_{\mathcal{S}}} \exp(s^2\|\mathbf{v}\|^2/(2n^2))$$

Then

$$\mathbf{R}(\mathcal{S}) \leq \min_{s\in\mathbb{R}^+} w(s)$$

Since $\tilde{w}$ is convex, its global minimum can be found efficiently

What does the set of vectors $V_{\mathcal{S}}$ look like?
Let $\mathbf{CI}(\mathcal{S})$ be the set of all Closed Itemsets in $\mathcal{S}$
Lemma: $V_{\mathcal{S}}$ contains all and only the vectors $\mathbf{v}_{\mathcal{S}}(A)$ for all $A \in \mathbf{CI}(\mathcal{S})$.
Issue: Can not mine $\mathbf{CI}(\mathcal{S})$ to compute $w(s)$: it is too expensive!

Solution: Define a function $\tilde{w}(s)$ efficient to compute and minimize and s.t. $\tilde{w}(s) \geq w(s)$ for all s. Then use $\tilde{w}(s)$ to compute $\eta_{\mathcal{S}}$

We define a partitioning $\mathcal{P}$ of $\mathbf{CI}(\mathcal{S})$

- Assume an ordering $<_{\mathcal{I}}$ of $\mathcal{I}$. For any $a \in \mathcal{I}$, assume an ordering $<_a$ of the transactions of $\mathcal{S}$ that contain a
- For any $A \in \mathbf{CI}(\mathcal{S})$, let $a \in A$ be the item in A that comes first wrt $<_{\mathcal{I}}$, and let τ be the transaction containing A that comes first wrt $<_a$. We assign A to the class $\mathcal{P}_{a,\tau}$

For each class $\mathcal{P}_{a,\tau}$ we

- compute an upper bound to $|\mathcal{P}_{a,\tau}|$ using $|\tau|$ and $<_a$
- use $f_{\mathcal{S}}(a)$ as upper bound to $f_{\mathcal{S}}(A)$, for $A \in \mathcal{P}_{a,\tau}$
  Very efficient to compute $f_{\mathcal{S}}(a)$ while creating the sample

The new function $\tilde{w}$ used to compute $\mathbf{R}(\mathcal{S})$ is:

$$\tilde{w}(s) = \frac{1}{s}\ln \sum_{a\in\mathcal{I}_{\mathcal{S}}} \left(\left(1 + \sum_{r=1}^{\chi_a}\sum_{j=1}^{g_{a,r}} 2^{\min\{r,h_{a,r}-j\}}\right) e^{\frac{s^2 f_{\mathcal{S}}(a)}{2n}}\right)$$

Then

$$\eta_{\mathcal{S}} = \min_{s\in\mathbb{R}^+} \tilde{w}(s) + \sqrt{\frac{2\ln(2/\delta)}{n}}$$

9. How do we choose the next sample size?

We can compute the next sample size on the fly

First iteration: Use a sample of size at least $8\ln(2/\delta)\varepsilon^{-2}$
Why? It is impossible that $\eta \leq \varepsilon/2$ at smaller sample sizes

Successive iterations:
multiply the sample size from the previous iteration by $(2\eta/\varepsilon)^2$

Intuition: If the frequencies of the items in the current iteration and the distribution of the transaction lengths are the same as in the previous iteration, then the stopping condition will be satisfied at this iteration

12. Experimental Evaluation

Implementation and Datasets: Sampling and stopping condition implemented in C++11, FIs mining using existing C implementation, NLOpt for minimization step. Implementation available from http://cs.brown.edu/~matteo/radeprogrfi.tar.bz2. Datasets from the FIMI'03 repository. Experiments run on a machine with a quad-core AMD PhenomTM II X4 955 processor and 16GB of RAM, running GNU/Linux 3.2.0

Recall: in 10K+ runs, always returned an ε-approx., not just with prob. $1-\delta$. All itemsets from $\mathbf{FI}(\mathcal{D},\theta)$ are in output: the recall is always 100%

Precision: Algorithm gives no guarantee. Varied between 15% and 92%, depending on the parameters and on the dataset. Price to pay when mining a subset of transactions. The output can be used as a set of candidates from which to compute efficiently the exact collection of FIs with a single linear scan of the dataset (negligible cost)

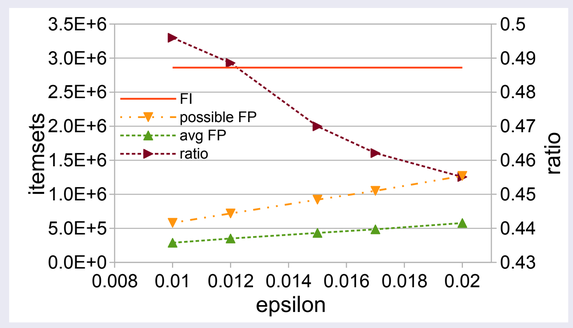


Figure: Precision for connect, $\theta = 0.72$

Frequency Estimation: For each itemset A in output, we measure the absolute frequency error. Average and maximum abs. error are almost 10x less than $\varepsilon/2$ and very concentrated. We also measure the relative frequency error (right axis): it is always less than 1.4%

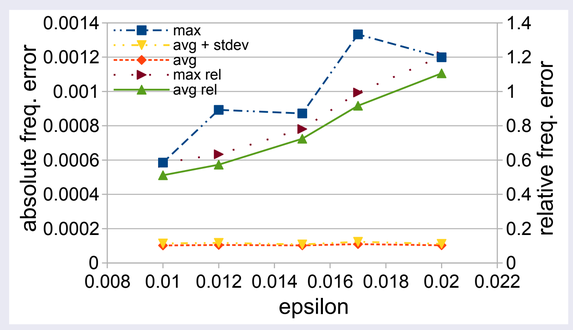


Figure: Frequency error for retail, $\theta = 0.015$

Runtime: We compare the running time of our algorithm to that of a one-shot sampling approach (VC), to that of FP-Growth, and to the running time of our algorithm using a geometric sampling schedule $|\mathcal{S}_i| = \alpha^i|\mathcal{S}_1|$ for $\alpha \in \{2, 2.5, 3\}$ (to evaluate the automatic sample schedule). Our algorithm (avg line) vastly outperforms the exact algorithm and VC. The automatic sampling schedule is more efficient than the geometric sample schedule by avoiding the creation and analysis of samples whose size is probably not sufficient for the stopping condition to be satisfied, based on information obtained from the current sample

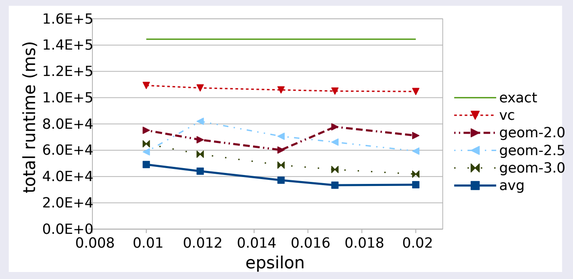


Figure: Running time for BMS-POS, $\theta = 0.015$

We also analyze the breakdown of the runtime of our algorithm, splitting it between time needed to sample the transaction, time needed to evaluate the stopping condition, and time needed to perform the mining of the sample after the stopping condition is satisfied

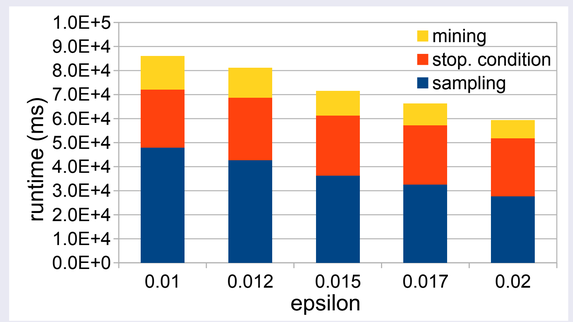


Figure: Breakdown of runtime for pumsb_star, $\theta = 0.32$.

11. Future Work

Improve bound to the Rademacher Average by studying the orderings and the function $\tilde{w}$

Improve the automatic sample schedule by better taking into account the behavior of $\tilde{w}$

Provide multiplicative error guarantees

Extend to other measures of interestingness not bounded by anti-monotonic functions

Acknowledgements

This work was supported, in part, by NSF grant IIS-1247581 and NIH grant R01-CA180776

Extended Version

Available from http://goo.gl/a7dceX