Sublinear Time Orthogonal Tensor Decomposition

Zhao Song*, David P. Woodruff†, Huan Zhang‡

*UT-Austin, zhaos@utexas.edu †IBM Almaden, dpwoodru@us.ibm.com ‡UC Davis, ecezhang@ucdavis.edu

Introduction

In orthogonal tensor decomposition, we observe a tensor $\mathbf{T} = \mathbf{T}^* + \mathbf{E} \in \mathbb{R}^{n\times n\times n}$ where $\mathbf{E}$ is arbitrary noise, and the true tensor $\mathbf{T}^*$ can be decomposed into an orthonormal basis $\{v_t\}$ of $\mathbb{R}^n$,

$$\mathbf{T}^* = \sum_{t=1}^{n} \lambda_t v_t \otimes v_t \otimes v_t$$

Where $\{\lambda_t\}$ is the set of eigenvalues.

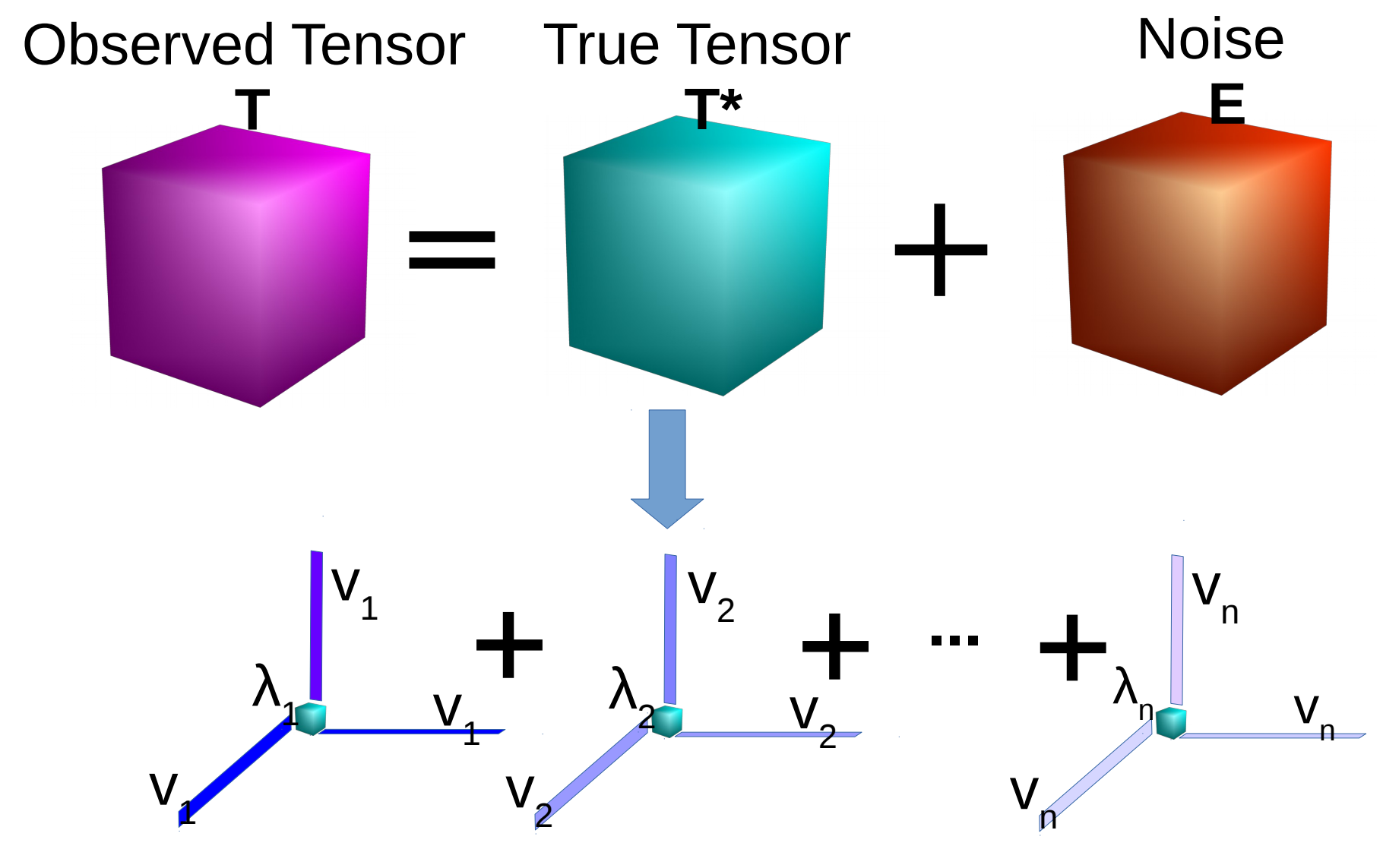

Figure 1: Orthogonal Tensor Decomposition

Analogous to the eigen-decomposition problem for matrices, the tensor power method can be used to find the decomposition efficiently.

Algorithm 1 Tensor Power Method (simplified)

```
for $t = 1 \to T$ do
    $u \leftarrow \mathrm{TIvw}(\mathbf{T}, u, u)$
    $u \leftarrow u/\|u\|_2$
end for
$\lambda_1 \leftarrow \mathrm{Tuvw}(\mathbf{T}, u, u, u)$
```

TIvw and Tuvw are subroutines to compute tensor contractions $\mathbf{T}(u, v, w)$ and $\mathbf{T}(I, v, w)$, defined as:

$$\mathbf{T}(u, v, w) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \mathbf{T}_{i,j,k}\, u_i v_j w_k$$

$$\mathbf{T}(I, v, w) = \left[\textstyle\sum_{j=1}^{n}\sum_{k=1}^{n} \mathbf{T}_{1,j,k}\, v_j w_k, \cdots, \sum_{j=1}^{n}\sum_{k=1}^{n} \mathbf{T}_{n,j,k}\, v_j w_k\right]$$

Figure 2 visualizes $\mathbf{T}(I, v, w)$.

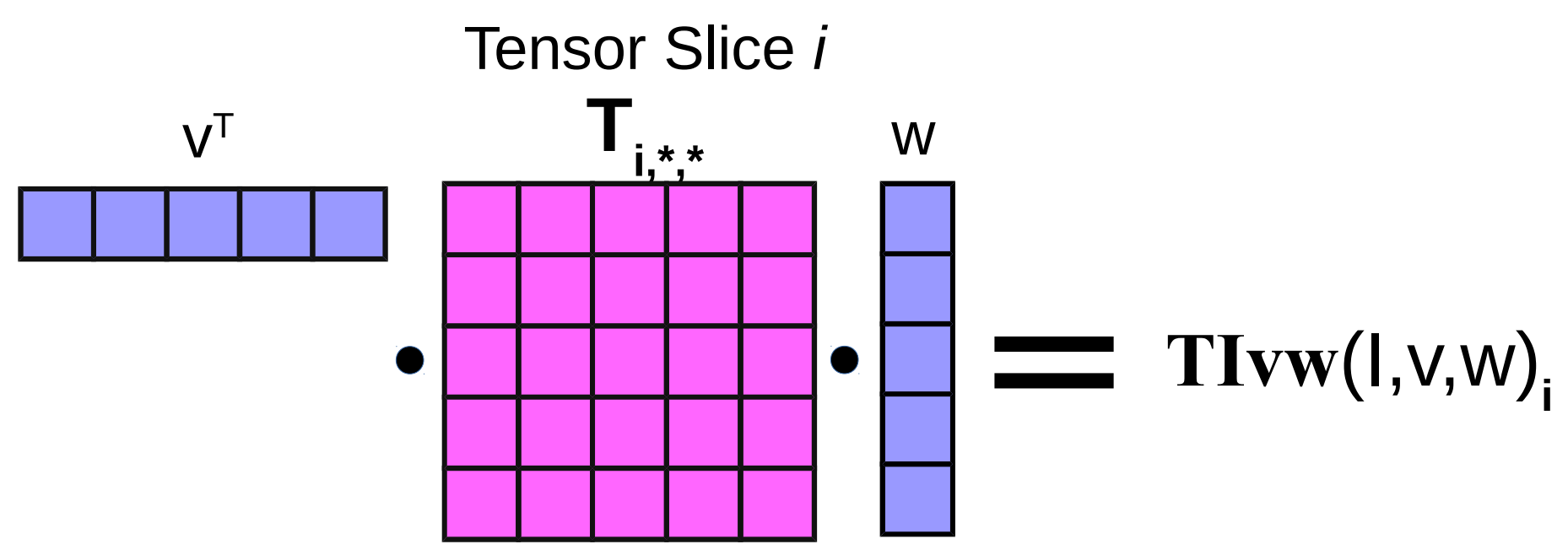

Figure 2: Each coordinate of $\mathbf{T}(I, v, w)$ is $v^T \mathbf{T}_{i,*,*}\, w$

Computing $\mathbf{T}(I, u, u)$ and $\mathbf{T}(u, u, u)$ are the most expensive steps in the tensor power method. We want to approximate these tensor contractions in sublinear time for large tensors.

Approximate Tensor Contractions

Previous work [1] approximates tensor contractions by taking a count-sketch of $\mathbf{T}$, and approximating tensor contractions using this sketch. [2] uses uniform sampling and does not achieve our guarantees.

In our work, we conduct importance sampling based on the magnitude of elements in iterate variable u. Because we know u, we can take more samples from $\mathbf{T}$ where the corresponding element in u is larger. We do not need to scan all elements of $\mathbf{T}$. Its correctness is guaranteed by the following lemma:

Lemma 1 (Importance sampling for TIvw). *For all $i \in [n]$, suppose random variable $X^i = \mathbf{T}_{i,j,k} v_j w_k/(q_j r_k)$ with probability $q_j r_k$, where $q_j = |v_j|^2/\|v\|_2^2$ and $r_k = |w_k|^2/\|w\|_2^2$, and we take L_i i.i.d. samples of X^i, say $X_1^i, X_2^i, \cdots, X_{L_i}^i$. Let $Y^i = \frac{1}{L_i}\sum_{\ell=1}^{L} X_\ell^i$. Then:*

- $\mathbb{E}[Y^i] = \langle \mathbf{T}_{i,*,*}, v \otimes w\rangle$
- $\mathbb{V}[Y^i] \le \frac{1}{L_i}\|\mathbf{T}_{i,*,*}\|_F^2 \|v \otimes w\|_F^2$.

In [1], a similar lemma is given but the variance bound is a factor of 4^p (where p is the order of tensor) larger than us. For a 3rd order tensor, the number of samples we need to take is a factor of 64 smaller than the size of sketch in [1].

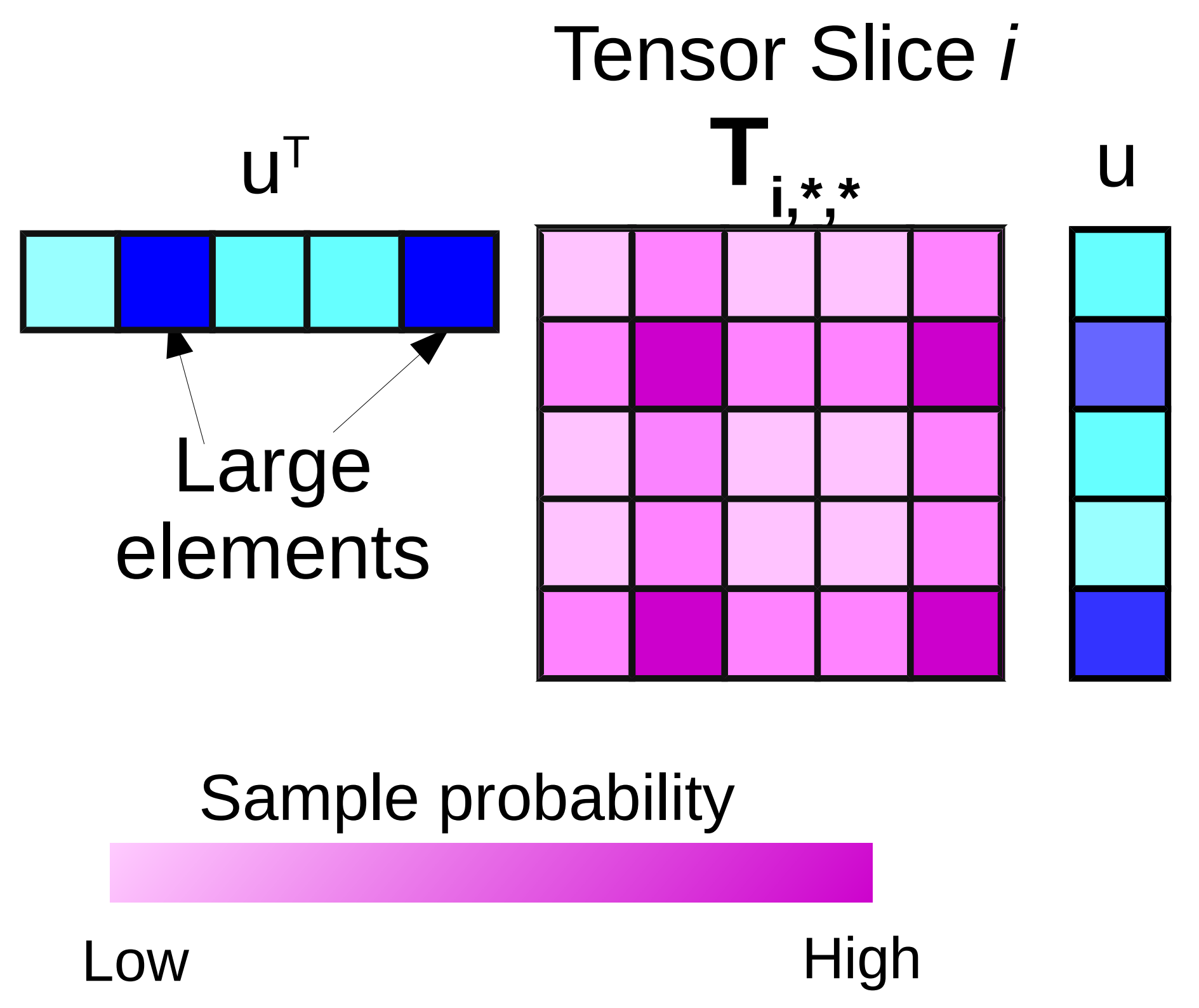

Figure 3: ApproxTIvw: TIvw with importance sampling

Figure 3 illustrates the importance sampling process. We can also derive similar importance sampling lemma for $\mathbf{T}(u, v, w)$.

We then use importance sampling based tensor contraction subroutines ApproxTuvw and ApproxTIvw to replace Tuvw and TIvw in Algorithm 1.

Main Results

As our importance sampling will incur addition noise during the power iteration, we must ensure that the noise is not too big by taking enough samples to guarantee that the power method will converge to the true eigenvector with high probability.

Let $\widehat{b}_i$ be the number samples we take from slice $i \in [n]$ in ApproxTIvw, and let $\widehat{b}$ denote the total number of samples in our algorithm. The following theorem gives the requirement for $\widehat{b}_i$:

Theorem 2 (Informal main theorem, $k = 1$). *Suppose we take $\widehat{b} = \sum_{i=1}^{n} \widehat{b}_i$ samples during the power iterations for recovering $\widehat{\lambda}_1$ and $\widehat{v}_1$, the number of samples for slice i is $\widehat{b}_i \gtrsim b\|\mathbf{T}_{i,*,*}\|_F^2/\|\mathbf{T}\|_F^2$ where $b \gtrsim n\|\mathbf{T}\|_F^2/\epsilon^2 + \|\mathbf{T}\|_F^2/\min\{\epsilon, \lambda_{\min}/n\}^2$, and ϵ is the desired recovery error. Then the tensor power method recovers $\widehat{\lambda}_1$ and $\widehat{v}_1$ with small error in constant probability.*

The running time of our method is $O(\widehat{b})$. If we require $\widehat{b}_i = b\|\mathbf{T}_{i,*,*}\|_F^2/\|\mathbf{T}\|_F^2$, we need to *pre-scan* the tensor to compute $\|\mathbf{T}_{i,*,*}\|_F^2$, making our algorithm not sublinear time. However, with the following mild assumption, our algorithm is sublinear when sampling uniformly $(\widehat{b}_i = b/n)$ *without pre-scanning*:

Theorem 3 (Bounded slice norm, simplified). *If there is an $\alpha \in (0, 1]$ such that $\|\mathbf{T}_{i,*,*}\|_F^2 \le \frac{1}{n^\alpha}\|\mathbf{T}\|_F^2$ for all $i \in [n]$, and $\mathbf{E}$ satisfies certain noise bounds (the same bounds as in [1]), then Algorithm 1 with importance sampling takes $O(n^{3-\alpha})$ time.*

For certain cases, we can remove the bounded slice norm assumption by taking a sublinear number of samples from the tensor to obtain upper bounds on all slice norms. It is an interesting questions on its own. In this paper we have shown the following cases:

Theorem 4 (Even order). *There is a constant $\alpha > 0$ and a sufficiently small constant $\gamma > 0$, such that, for any even order-p tensor $\mathbf{T} = \mathbf{T}^* + \mathbf{E} \in \mathbb{R}^{n^p}$ with $\mathrm{rank}(\mathbf{T}^*) \le n^\gamma$ and $\mathbf{E}$ satisfies certain noise bounds (the same bounds as in [1]), Algorithm 1 with importance sampling runs in $O(n^{p-\alpha})$ time.*

Theorem 5 (Low rank). *There is a constant $\alpha > 0$ such that for any symmetric tensor $\mathbf{T} = \mathbf{T}^* + \mathbf{E} \in \mathbb{R}^{n^3}$ with $\mathbf{E}$ satisfying certain noise bounds (the same bounds as in [1]) and $\mathrm{rank}(\mathbf{T}^*) \le 2$, Algorithm 1 with importance sampling runs in $O(n^{3-\alpha})$ time.*

Experiments

Our code is based on the code released for [1]. We changed the ApproxTIvw and ApproxTuvw functions from sketching to importance sampling. The following figures show the time required to achieve a similar level of accuracy using both methods for synthetic datasets.

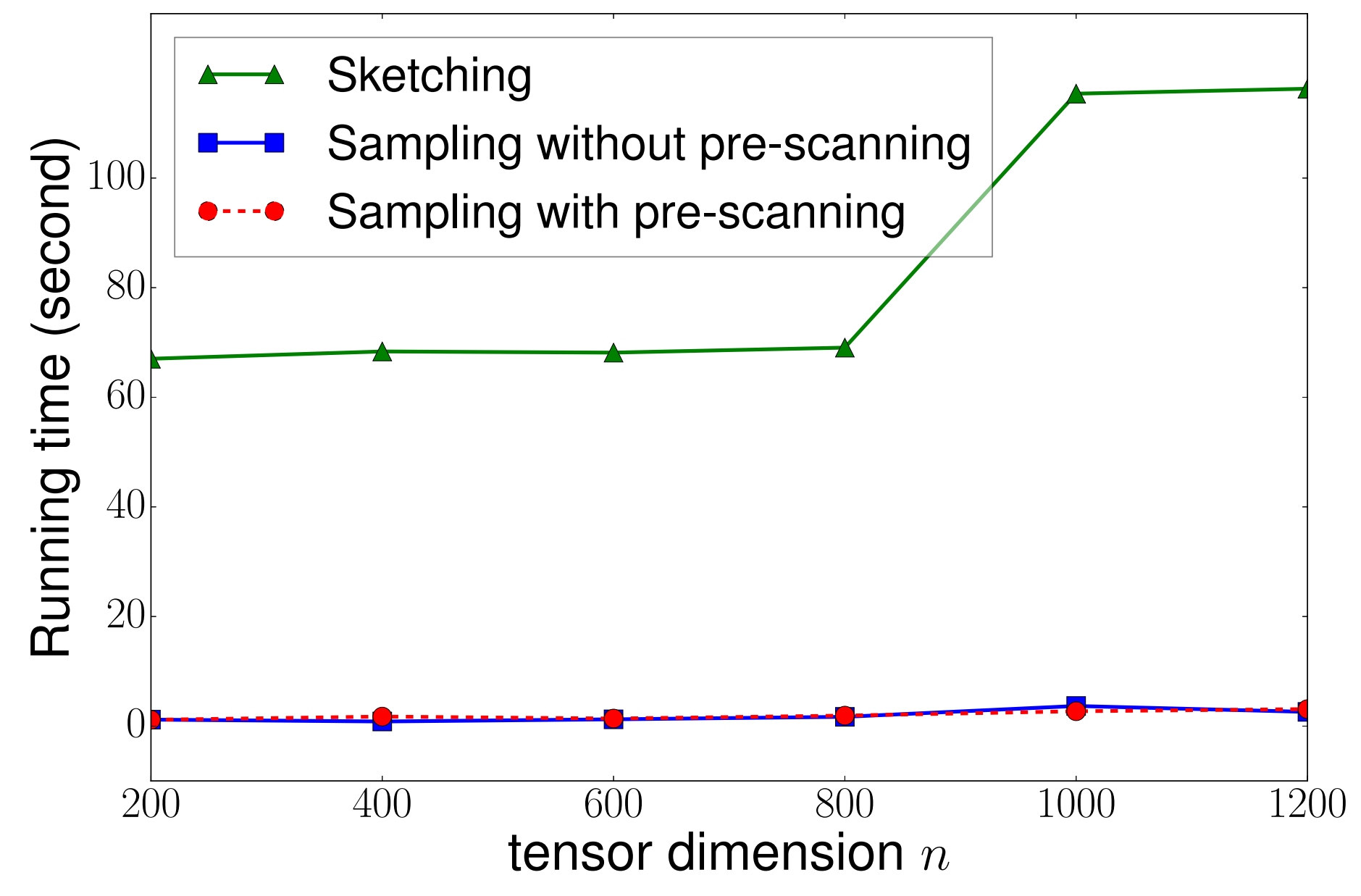

Figure 4: Running time vs tensor dimension

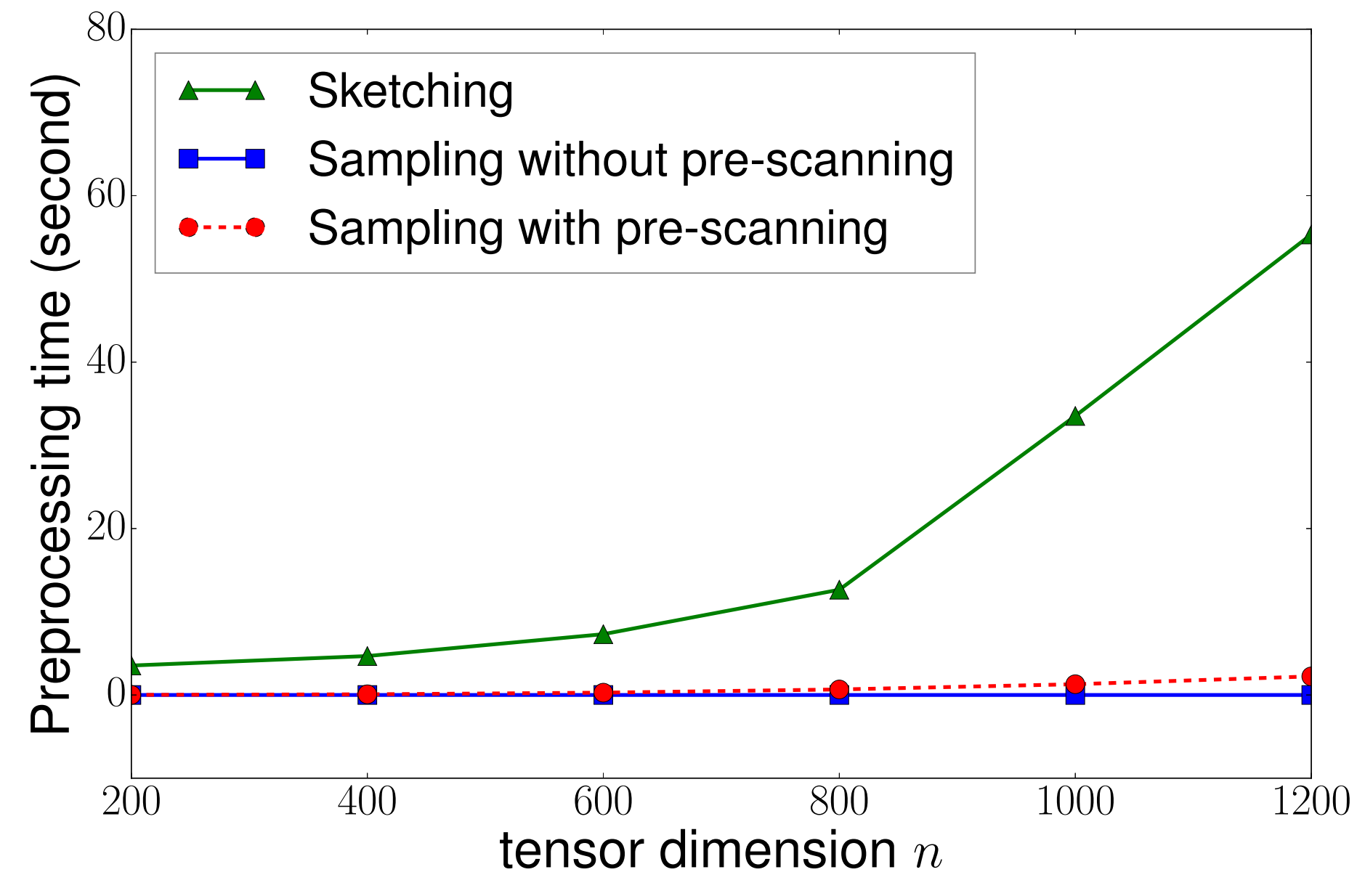

Figure 5: Preprocessing time vs tensor dimension

For $n = 200$ tensors from spectral-LDA application, we can achieve 3 to 5× speedup and our advantage is expected to increase if n is larger. Please visit our project page `https://github.com/huanzhang12/sampling_tensor_decomp` for code and more results.

The success of our algorithm can be attributed to the better variance bound, its sublinear running time with no pre-processing, and the efficient importance sampling implementation.

References

[1] Yining Wang, Hsiao-Yu Tung, Alex J Smola, and Anima Anandkumar.
Fast and guaranteed tensor decomposition via sketching.
In *NIPS*, pages 991–999, 2015.

[2] Charalampos E. Tsourakakis.
MACH: fast randomized tensor decompositions.
In *SDM*, pages 689–700, 2010.