Understanding Sparse JL for Feature Hashing

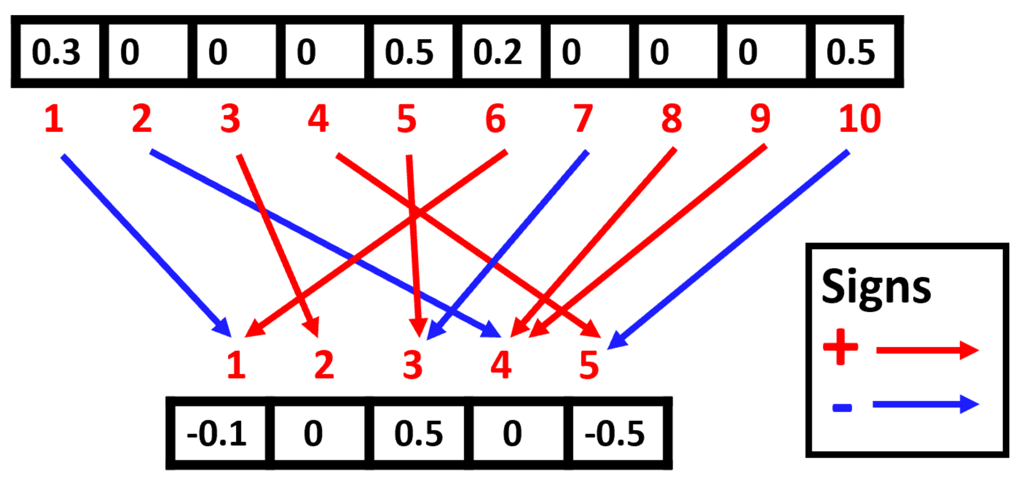
Meena Jagadeesan (Harvard University)

NeurIPS 2019

Feature Hashing

Feature hashing [7] is a commonly used technique to reduce the dimensionality of feature vectors.

Goal: Map vectors in $\mathbb{R}^n$ into $\mathbb{R}^m$ for $m \ll n$ while preserving Euclidean (ℓ_2-norm) distances.



❶ *Hash function* $h : [n] \to [m]$ on coordinates

❷ *Random signs* to handle collisions

Sparse Johnson-Lindenstrauss

Use s (anti-correlated) hash functions $h_1, \ldots, h_s : [n] \to [m]$. Use random signs σ_j^k for collisions:

$$f(x)_i = \frac{1}{\sqrt{s}} \sum_{k=1}^{s} \left(\sum_{j \in h_k^{-1}(i)} \sigma_j^k x_j \right).$$

Our contribution (Informal)

Analysis of sparse JL on feature vectors. Elucidates the tradeoff between projection time, dimension, and performance in preserving distances.

Mathematical Framework

Let $\mathcal{F}_{s,m}$ be a sparse JL distribution with parameters s and m, over linear maps $f : \mathbb{R}^n \to \mathbb{R}^m$.

The goal is $\mathbb{P}_{f \in \mathcal{F}_{s,m}}[\|f(x)\|_2 \in (1 \pm \epsilon) \|x\|_2] > 1 - \delta$ on each $x \in \mathbb{R}^n$, for error ϵ and failure probability δ.

Model for feature vectors

Feature vectors may have "well-spread" mass. Consider vectors with small ℓ_∞-to-ℓ_2 norm ratio:

$$S_v = \{x \in \mathbb{R}^n \mid \|x\|_\infty \le v \|x\|_2\}.$$

Definition [7]

$v(m, \epsilon, \delta, s)$ is the supremum over $v \in [0, 1]$ where: $\mathbb{P}_{f \in \mathcal{F}_{s,m}}[\|f(x)\|_2 \in (1 \pm \epsilon) \|x\|_2] > 1 - \delta$ holds for each $x \in S_v$.

$v(m, \epsilon, \delta, s)$ captures the performance of sparse JL.

Our contribution

Tight bounds on $v(m, \epsilon, \delta, s)$ for a general s.

Main Result: Performance of Sparse JL on Feature Vectors

We analyze how sparse JL (a state-of-the-art dimensionality reduction scheme) performs on feature vectors.

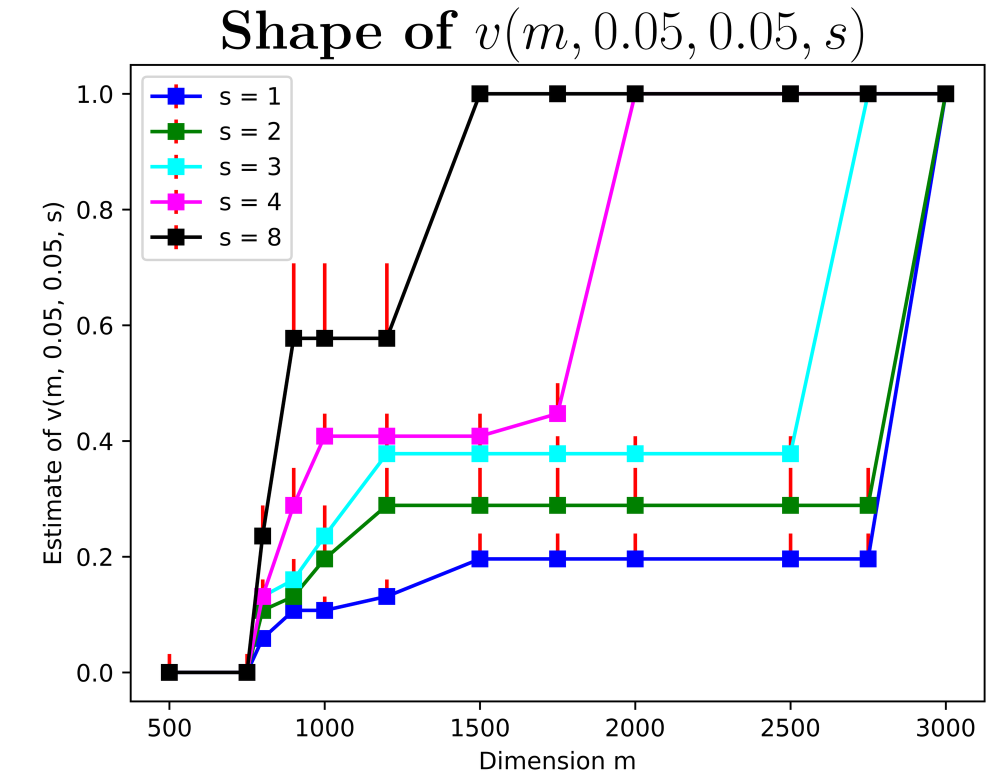
Theorem (Informal)

Sparse JL has **four regimes** in terms of how it performs on distance preservation. For error ϵ and failure probability δ, sparse JL with projected dimension m and s hash functions has performance $v(m, \epsilon, \delta, s)$ equal to:

$$\begin{cases} 1 \text{ (full performance)} & \text{High } m \\ \sqrt{s} B_1 \text{ (partial performance)} & \text{Middle } m \\ \sqrt{s} \min(B_1, B_2) \text{ (partial performance)} & \text{Middle } m \\ 0 \text{ (poor performance)} & \text{Small } m, \end{cases}$$

where B_1, B_2 are functions of m, ϵ, δ.

Shape of $v(m, 0.05, 0.05, s)$



Characterizes sparse JL tradeoff between the number of hash functions s, dimension m, and performance.

Formal Statement of Main Result

Theorem

Consider a uniform sparse JL distribution with dimension m and s hash functions. For $s \le m/e$ and for small enough ϵ and δ, the function $v(m, \epsilon, \delta, s)$ is equal to $f'(m, \epsilon, \ln(1/\delta), s)$, where:

$$f'(m,\epsilon,p,s) = \begin{cases} 1 & \text{if } m \ge \min\left(2\epsilon^{-2}e^{p}, \epsilon^{-2}pe^{\Theta\left(\max\left(1, \frac{p\epsilon^{-1}}{s}\right)\right)}\right) \\ \Theta\left(\sqrt{\epsilon s}\frac{\sqrt{\ln(\frac{m\epsilon^2}{p})}}{\sqrt{p}}\right) & \text{else, if } \max\left(\Theta(\epsilon^{-2}p), s \cdot e^{\Theta\left(\max\left(1, \frac{p\epsilon^{-1}}{s}\right)\right)}\right) \le m \le \epsilon^{-2}e^{\Theta(p)} \\ \Theta\left(\sqrt{\epsilon s}\min\left(\frac{\ln(\frac{m\epsilon}{p})}{p}, \frac{\sqrt{\ln(\frac{m\epsilon^2}{p})}}{\sqrt{p}}\right)\right) & \text{else, if } \Theta(\epsilon^{-2}p) \le m \le \min\left(\epsilon^{-2}e^{\Theta(p)}, s \cdot e^{\Theta\left(\max\left(1, \frac{p\epsilon^{-1}}{s}\right)\right)}\right) \\ 0 & \text{if } m \le \Theta(\epsilon^{-2}p). \end{cases}$$

Our result gives tight bounds on the function $v(m, \epsilon, \delta, s)$ for sparse JL across most of the parameter space.

Relationship with Previous Bounds

Sparse JL on full space $\mathbb{R}^n$

❶ Can set $m \approx \epsilon^{-2}\log(1/\delta))$, $s \approx \epsilon^{-1}\log(1/\delta)$ [4].

❷ A lower s is possible with a higher m, which enables faster projection. That is, m can be set to $\min(2\epsilon^{-2}/\delta, \epsilon^{-2}\log(1/\delta)e^{\Theta(\epsilon^{-1}\log(1/\delta)/s)}$ [1].

Bounds on $v(m, \epsilon, \delta, s)$

❶ $v(m, \epsilon, \delta, 1)$ understood [7, 2, 3]

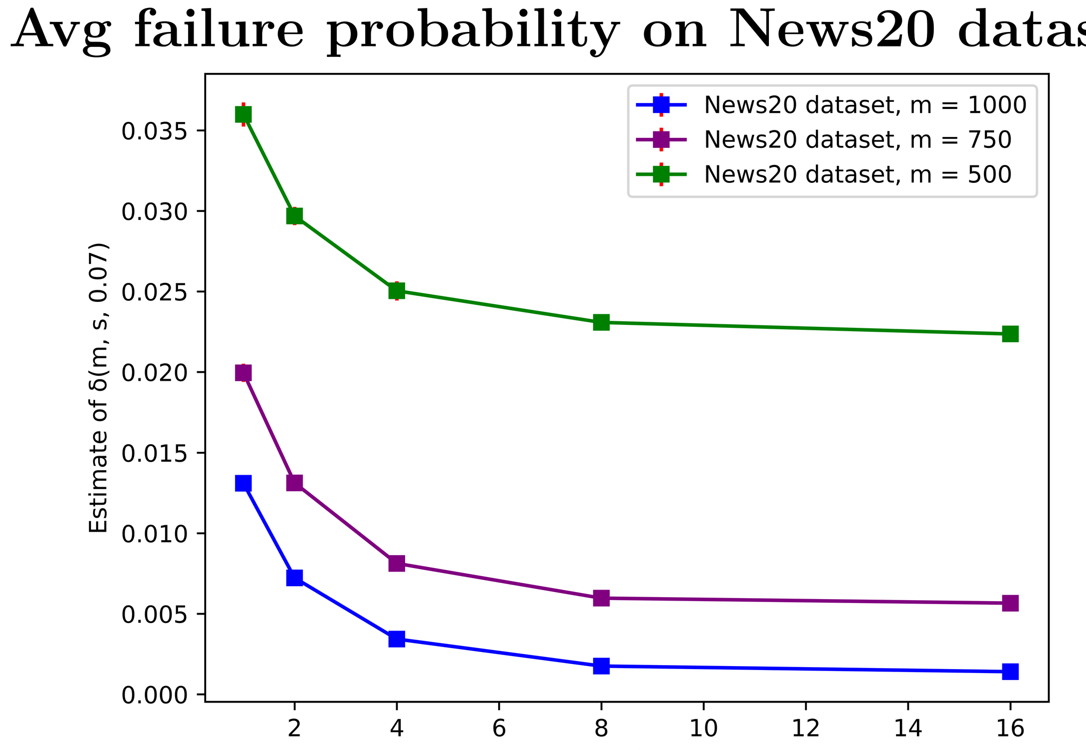
❷ $v(m, \epsilon, \delta, s)$ lower bound for *multiple hashing* [7]

Our tight bound on $v(m, \epsilon, \delta, s)$ for sparse JL for $s > 1$ significantly generalizes these results.

Evaluation on Real-World Data

Sparse JL with ≥ 4 hash functions can perform much better on feature vectors than feature hashing.

Avg failure probability on News20 dataset



Proof Approach: Moment Bounds

Analyze moments of "error" random variable:

$$R(x_1, \ldots, x_n) = \frac{1}{s}\sum_{r=1}^{m}\left(\sum_{1 \le i \ne j \le n} \eta_{r,i}\eta_{r,j}\sigma_{r,i}\sigma_{r,j}x_i x_j\right) =: \frac{1}{s}\sum_{r=1}^{m} Z_r(x_1, \ldots, x_n).$$

We show tight bounds on $\mathbb{E}[R(x_1, \ldots, x_n)^p]$ on $x \in S_v$ at every threshold v value (which are much more general than known bounds [4, 3, 1]).

Our key ingredient is a non-combinatorial approach with Rademacher-specific bounds.

Lower bound: Pick "worst" vector in each S_v

- View $Z_r(v, \ldots, v, 0, \ldots, 0)$ as a quadratic form of ± 1 rvs. Apply moments bounds in [6].
- Carefully combine over $r \in [m]$.

Upper bound: $R(x_1, \ldots, x_n)$ for every $x \in S_v$

- Create *tractable* versions of estimates in [6, 5]; Structure of $Z_r(x_1, \ldots, x_n)$ is helpful.
- Combine over $r \in [m]$ using bound in [5].

Challenges: Correlations between $\eta_{r,i}$; asymmetry imposed by the x_i values; need "tightness" to get matching upper and lower bounds.

Acknowledgements

I would like to thank Prof. Jelani Nelson for advising this project.

Selected References

[1] M. B. Cohen. "Nearly tight oblivious subspace embeddings by trace inequalities". In: *SODA*. 2016, 278–287.

[2] S. Dahlgaard, M. Knudsen, and M. Thorup. "Practical Hash Functions for Similarity Estimation and Dimensionality Reduction". In: *NIPS*. 2017, pp. 6618–6628.

[3] C. Freksen, L. Kamma, and K. G. Larsen. "Fully Understanding the Hashing Trick". In: *NeurIPS*. 2018, pp. 5394–5404.

[4] D. M. Kane and J. Nelson. "Sparser Johnson-Lindenstrauss transforms". In: *SODA*. 2012, 16872–16876.

[5] R. Latała. "Estimation of moments of sums of independent real random variables". In: *Annals of Probability* 25.3 (1997), pp. 1502–1513.

[6] R. Latała. "Tail and moment estimates for some types of chaos". In: *Studia Mathematica* 135.1 (1999), pp. 39–53.

[7] K. Weinberger et al. "Feature Hashing for Large Scale Multitask Learning". In: *ICML*. 2009, pp. 1113–1120.