Understanding Sparse JL for Feature Hashing

Meena Jagadeesan

Harvard University (Class of 2020)

NeurIPS 2019 (Poster #59)

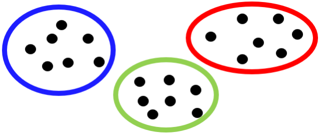
Dimensionality reduction (ℓ_2-to-ℓ_2)

A randomized map $\mathbb{R}^n \to \mathbb{R}^m$ (where $m \ll n$) that preserves distances.

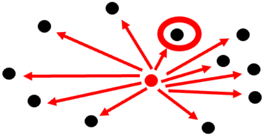
Dimensionality reduction (ℓ_2-to-ℓ_2)

A randomized map $\mathbb{R}^n \to \mathbb{R}^m$ (where $m \ll n$) that preserves distances.

A pre-processing step in many applications:



clustering

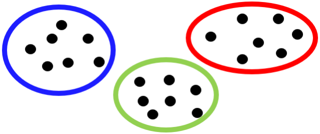


nearest neighbors

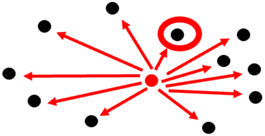
Dimensionality reduction (ℓ_2-to-ℓ_2)

A randomized map $\mathbb{R}^n \to \mathbb{R}^m$ (where $m \ll n$) that preserves distances.

A pre-processing step in many applications:



clustering



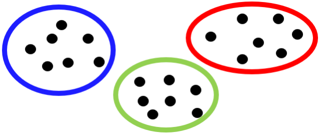
nearest neighbors

Key question: What is the tradeoff between the dimension m, the performance in distance preservation, and the projection time?

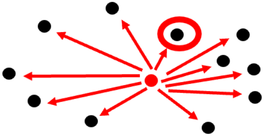
Dimensionality reduction (ℓ_2-to-ℓ_2)

A randomized map $\mathbb{R}^n \to \mathbb{R}^m$ (where $m \ll n$) that preserves distances.

A pre-processing step in many applications:



clustering



nearest neighbors

Key question: What is the tradeoff between the dimension m, the performance in distance preservation, and the projection time?

This paper: A theoretical analysis of this tradeoff for a state-of-the-art dimensionality reduction scheme on feature vectors.

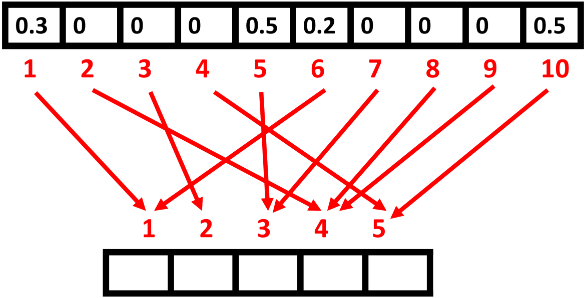
Feature hashing (Weinberger et al. '09)

One standard dimensionality reduction scheme is feature hashing.

Feature hashing (Weinberger et al. '09)

One standard dimensionality reduction scheme is feature hashing.

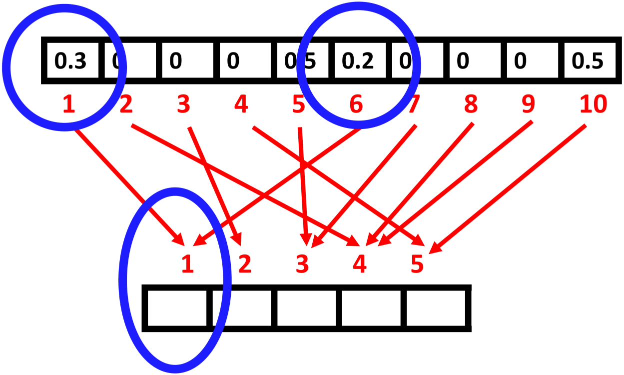
Use a **hash function** $h : \{1, \ldots, n\} \to \{1, \ldots, m\}$ on coordinates.



Feature hashing (Weinberger et al. '09)

One standard dimensionality reduction scheme is feature hashing.

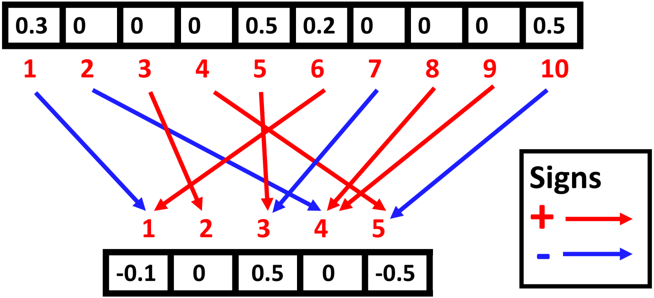
Use a **hash function** $h : \{1, \ldots, n\} \to \{1, \ldots, m\}$ on coordinates.



Feature hashing (Weinberger et al. '09)

One standard dimensionality reduction scheme is feature hashing.

Use a **hash function** $h : \{1, \dots, n\} \to \{1, \dots, m\}$ on coordinates.



Use **random signs** to handle collisions: $f(x)_i = \sum_{j \in h^{-1}(i)} \sigma_j x_j$.

Sparse Johnson-Lindenstrauss transform (KN '12)

Sparse JL is a state-of-the-art sparse dimensionality reduction.

Sparse Johnson-Lindenstrauss transform (KN '12)

Sparse JL is a state-of-the-art sparse dimensionality reduction.

Use many (anti-correlated) hash fns $h_1, \ldots, h_s : \{1, \ldots, n\} \to \{1, \ldots, m\}$.
$\implies$ Each input coordinate is mapped to s output coordinates.

Sparse Johnson-Lindenstrauss transform (KN '12)

Sparse JL is a state-of-the-art sparse dimensionality reduction.

Use many (anti-correlated) hash fns $h_1, \ldots, h_s : \{1, \ldots, n\} \to \{1, \ldots, m\}$.
$\implies$ Each input coordinate is mapped to s output coordinates.

Use random signs to deal with collisions.

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

Sparse Johnson-Lindenstrauss transform (KN '12)

Sparse JL is a state-of-the-art sparse dimensionality reduction.

Use many (anti-correlated) hash fns $h_1, \ldots, h_s : \{1, \ldots, n\} \to \{1, \ldots, m\}$.
$\implies$ Each input coordinate is mapped to s output coordinates.

Use random signs to deal with collisions.

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

(Alternate view: a random sparse matrix w/ s nonzero entries per column.)

Sparse Johnson-Lindenstrauss transform (KN '12)

Sparse JL is a state-of-the-art sparse dimensionality reduction.

Use many (anti-correlated) hash fns $h_1, \ldots, h_s : \{1, \ldots, n\} \to \{1, \ldots, m\}$.
$\implies$ Each input coordinate is mapped to s output coordinates.

Use random signs to deal with collisions.

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

(Alternate view: a random sparse matrix w/ s nonzero entries per column.)

The tradeoff: higher s preserves distances better, but takes longer.

Sparse Johnson-Lindenstrauss transform (KN '12)

Sparse JL is a state-of-the-art sparse dimensionality reduction.

Use many (anti-correlated) hash fns $h_1, \ldots, h_s : \{1, \ldots, n\} \to \{1, \ldots, m\}$.
$\implies$ Each input coordinate is mapped to s output coordinates.

Use random signs to deal with collisions.

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

(Alternate view: a random sparse matrix w/ s nonzero entries per column.)

The tradeoff: higher s preserves distances better, but takes longer.

This work

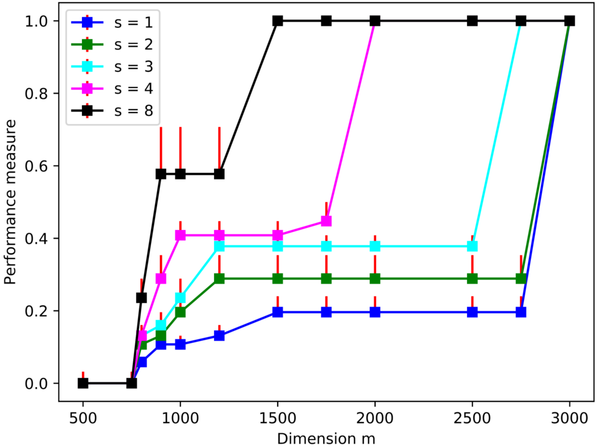
Analysis of tradeoff for sparse JL between # of hash functions s, dimension m, and performance in ℓ_2-distance preservation.

Intuition for this paper

Analysis of sparse JL with respect to a performance measure:

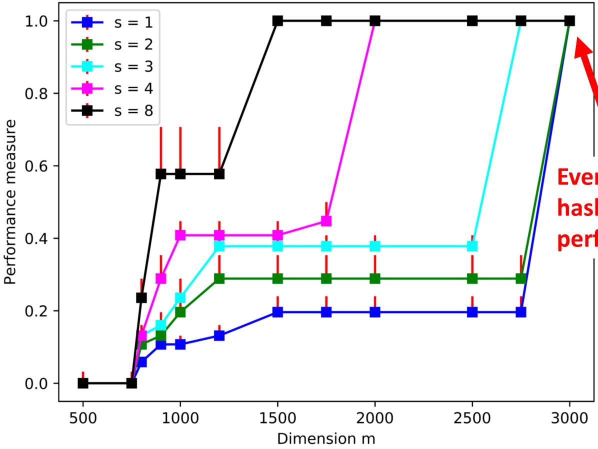
Intuition for this paper

Analysis of sparse JL with respect to a performance measure:



Intuition for this paper

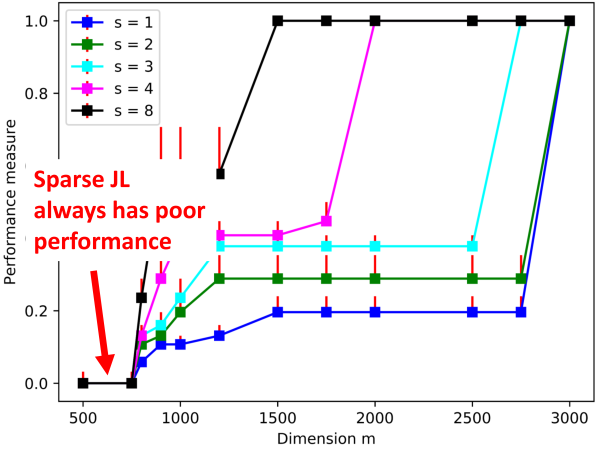
Analysis of sparse JL with respect to a performance measure:



Even feature hashing has full performance

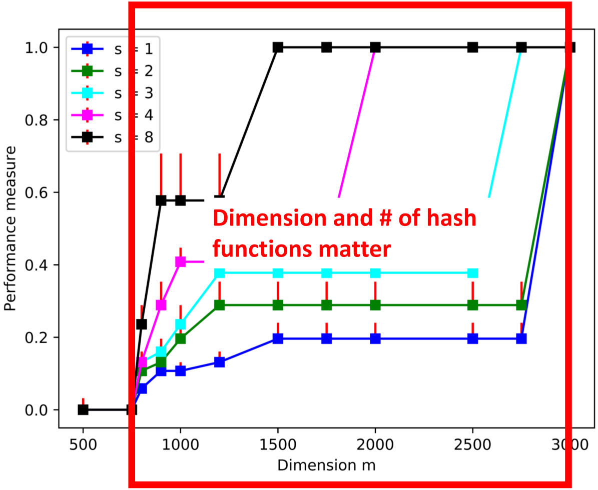
Intuition for this paper

Analysis of sparse JL with respect to a performance measure:



Intuition for this paper

Analysis of sparse JL with respect to a performance measure:



Traditional mathematical framework

Consider a probability distribution $\mathcal{F}$ over linear maps $f : \mathbb{R}^n \to \mathbb{R}^m$.

Traditional mathematical framework

Consider a probability distribution $\mathcal{F}$ over linear maps $f : \mathbb{R}^n \to \mathbb{R}^m$.

Geometry-preserving condition. For *each* $x \in \mathbb{R}^n$:

$$\mathbb{P}_{f \in \mathcal{F}}[\|f(x)\|_2 \in (1 \pm \epsilon)\|x\|_2] > 1 - \delta,$$

for ϵ target error, δ target failure probability.

Traditional mathematical framework

Consider a probability distribution $\mathcal{F}$ over linear maps $f : \mathbb{R}^n \to \mathbb{R}^m$.

Geometry-preserving condition. For *each* $x \in \mathbb{R}^n$:

$$\mathbb{P}_{f \in \mathcal{F}}[\|f(x)\|_2 \in (1 \pm \epsilon) \|x\|_2] > 1 - \delta,$$

for ϵ target error, δ target failure probability.

(Can apply to *differences* $x = x_1 - x_2$ since f is linear.)

Traditional mathematical framework

Consider a probability distribution $\mathcal{F}$ over linear maps $f : \mathbb{R}^n \to \mathbb{R}^m$.

Geometry-preserving condition. For *each* $x \in \mathbb{R}^n$:

$$\mathbb{P}_{f \in \mathcal{F}}[\|f(x)\|_2 \in (1 \pm \epsilon) \|x\|_2] > 1 - \delta,$$

for ϵ target error, δ target failure probability.

(Can apply to *differences* $x = x_1 - x_2$ since f is linear.)

Sparse JL can sometimes perform much better in practice on feature vectors than traditional theory suggests...

Performance on feature vectors (Weinberger et al. '09)

Consider vectors w/ small ℓ_∞-to-ℓ_2 norm ratio:

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

Performance on feature vectors (Weinberger et al. '09)

Consider vectors w/ small ℓ_∞-to-ℓ_2 norm ratio:

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

Let $\mathcal{F}_{s,m}$ be the distribution given by sparse JL with parameters s and m.

Performance on feature vectors (Weinberger et al. '09)

Consider vectors w/ small ℓ_∞-to-ℓ_2 norm ratio:

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

Let $\mathcal{F}_{s,m}$ be the distribution given by sparse JL with parameters s and m.

Definition

$v(m, \epsilon, \delta, s)$ is the supremum over $v \in [0, 1]$ such that:

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

Performance on feature vectors (Weinberger et al. '09)

Consider vectors w/ small ℓ_∞-to-ℓ_2 norm ratio:

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

Let $\mathcal{F}_{s,m}$ be the distribution given by sparse JL with parameters s and m.

Definition

$v(m, \epsilon, \delta, s)$ is the supremum over $v \in [0, 1]$ such that:

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

- $v(m, \epsilon, \delta, s) = 0 \implies$ poor performance
- $v(m, \epsilon, \delta, s) = 1 \implies$ full performance
- $v(m, \epsilon, \delta, s) \in (0, 1) \implies$ good performance on $x \in S_{v(m,\epsilon,\delta,s)}$

Performance on feature vectors (Weinberger et al. '09)

Consider vectors w/ small ℓ_∞-to-ℓ_2 norm ratio:

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

Let $\mathcal{F}_{s,m}$ be the distribution given by sparse JL with parameters s and m.

Definition

$v(m, \epsilon, \delta, s)$ is the supremum over $v \in [0, 1]$ such that:

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

- $v(m, \epsilon, \delta, s) = 0 \implies$ poor performance
- $v(m, \epsilon, \delta, s) = 1 \implies$ full performance
- $v(m, \epsilon, \delta, s) \in (0, 1) \implies$ good performance on $x \in S_{v(m,\epsilon,\delta,s)}$

We give a tight theoretical analysis of the function $v(m, \epsilon, \delta, s)$.

Informal statement of main result

Goal: $\mathbb{P}_{f \in \mathcal{F}}[\|f(x)\|_2 \in (1 \pm \epsilon) \|x\|_2] > 1 - \delta$.

$v(m, \epsilon, \delta, s) :=$ sup over $v \in [0, 1]$ s.t. sparse JL meets ℓ_2 goal on $x \in S_v$.

Informal statement of main result

Goal: $\mathbb{P}_{f\in\mathcal{F}}[\|f(x)\|_2 \in (1\pm\epsilon)\|x\|_2] > 1-\delta$.

$v(m,\epsilon,\delta,s) :=$ sup over $v \in [0,1]$ s.t. sparse JL meets ℓ_2 goal on $x \in S_v$.

Theorem (Informal)

For error ϵ and failure probability δ, sparse JL with projected dimension m and s hash functions has **four regimes** *in its performance: that is,*

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

where $p = \ln(1/\delta)$, $B_1 = \sqrt{\ln(m\epsilon^2/p)}/\sqrt{p}$ and $B_2 = \ln(m\epsilon/p)/p$.

Informal statement of main result

Goal: $\mathbb{P}_{f \in \mathcal{F}}[\|f(x)\|_2 \in (1 \pm \epsilon)\|x\|_2] > 1 - \delta$.

$v(m, \epsilon, \delta, s) :=$ sup over $v \in [0,1]$ s.t. sparse JL meets ℓ_2 goal on $x \in S_v$.

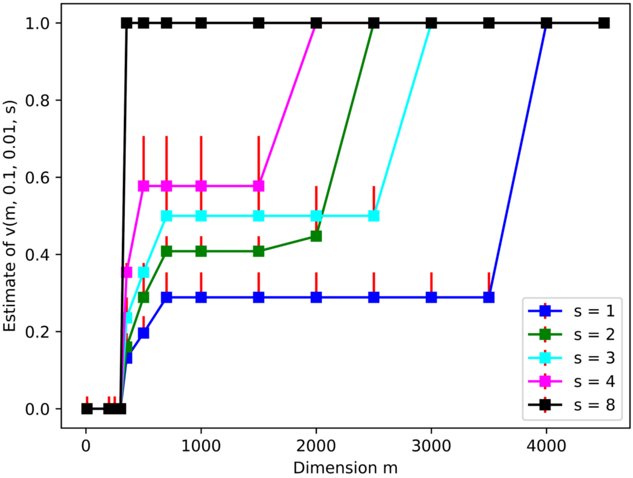
Theorem (Informal)

For error ϵ and failure probability δ, sparse JL with projected dimension m and s hash functions has **four regimes** *in its performance: that is,*

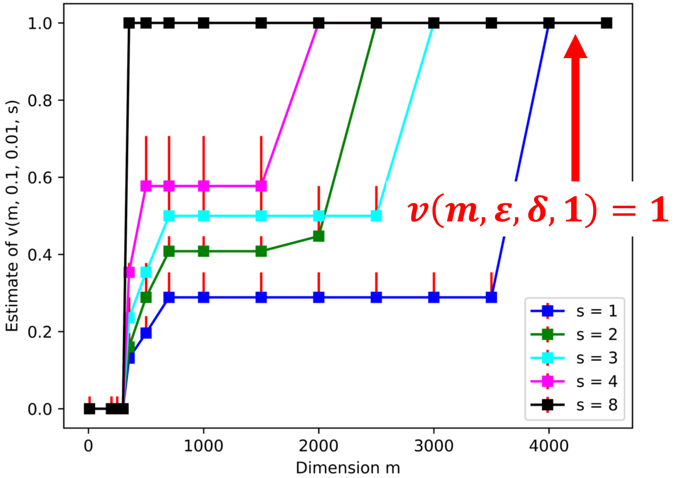
$$v(m, \epsilon, \delta, s) = \begin{cases} 1 \textit{ (full performance)} & \textit{High } m \\ \sqrt{s} B_1 \textit{ (partial performance)} & \textit{Middle } m \\ \sqrt{s} \min(B_1, B_2) \textit{ (partial performance)} & \textit{Middle } m \\ 0 \textit{ (poor performance)} & \textit{Small } m, \end{cases}$$

where $p = \ln(1/\delta)$, $B_1 = \sqrt{\ln(m\epsilon^2/p)}/\sqrt{p}$ and $B_2 = \ln(m\epsilon/p)/p$.

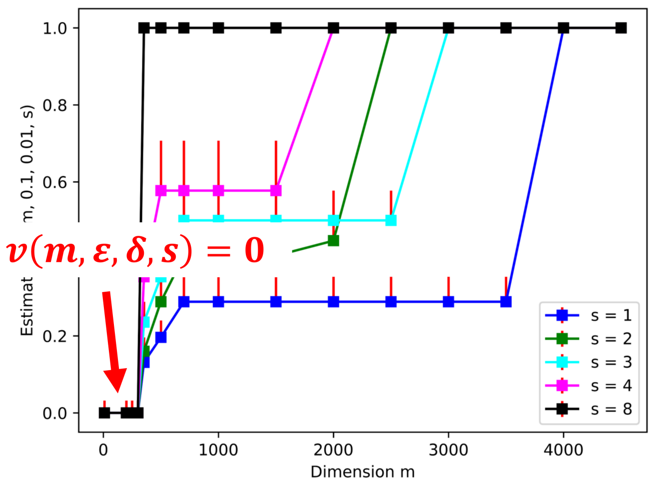
$v(m, \epsilon, \delta, s)$ on more synthetic data



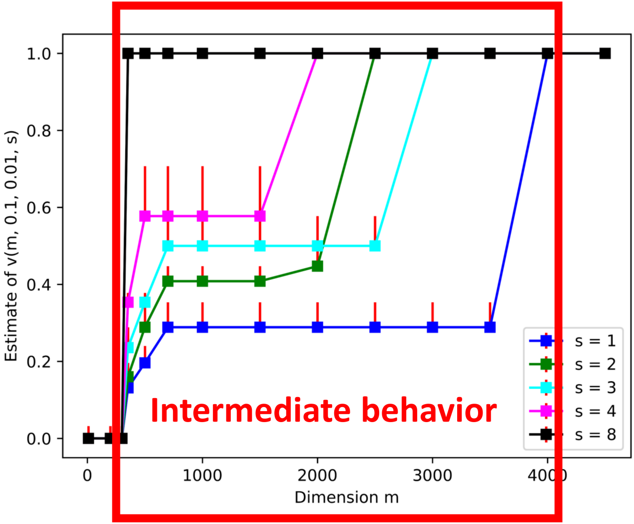
$v(m, \epsilon, \delta, s)$ on more synthetic data



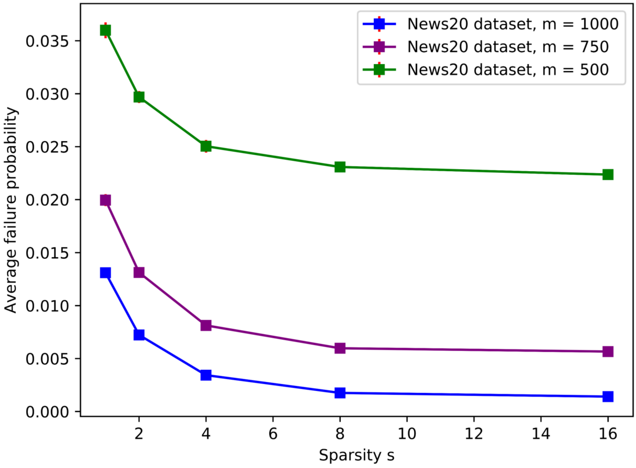
$v(m, \epsilon, \delta, s)$ on more synthetic data



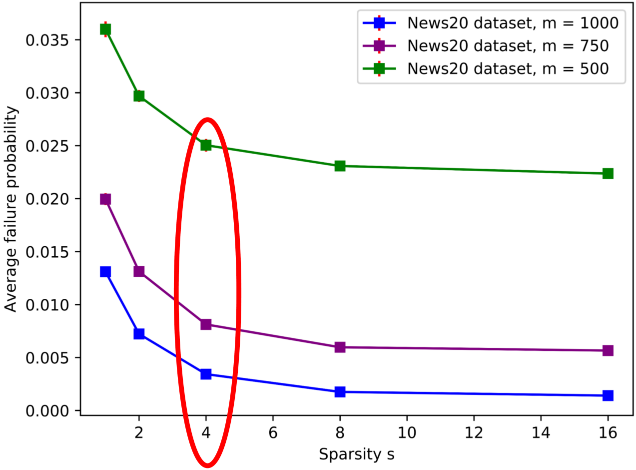
$v(m, \epsilon, \delta, s)$ on more synthetic data



Sparse JL on News20 dataset



Sparse JL on News20 dataset



Sparse JL with 4 hash fns can significantly outperform feature hashing!

Comparison to previous work

Goal: $\mathbb{P}_{f \in \mathcal{F}}[\|f(x)\|_2 \in (1 \pm \epsilon)\|x\|_2] > 1 - \delta$.

$v(m, \epsilon, \delta, s) :=$ sup over $v \in [0, 1]$ s.t. sparse JL meets ℓ_2 goal on $x \in S_v$.

Comparison to previous work

Goal: $\mathbb{P}_{f \in \mathcal{F}}[\|f(x)\|_2 \in (1 \pm \epsilon)\|x\|_2] > 1 - \delta$.

$v(m, \epsilon, \delta, s) :=$ sup over $v \in [0, 1]$ s.t. sparse JL meets ℓ_2 goal on $x \in S_v$.

Bounds on v (Weinberger et al. '09,..., Freksen et al. '18):

- $v(m, \epsilon, \delta, \mathbf{1})$ understood
- $v(m, \epsilon, \delta, s)$ bound for *multiple hashing* (a suboptimal construction)

Bounds for sparse JL on full space $\mathbb{R}^n$:

- Can set $m \approx \epsilon^{-2} \log(1/\delta)$, $s \approx \epsilon^{-1} \log(1/\delta)$ (Kane and Nelson '12)
- Can set $m \approx \min(2\epsilon^{-2}/\delta, \epsilon^{-2} \log(1/\delta) e^{\Theta(\epsilon^{-1} \log(1/\delta)/s)})$ (Cohen '16)

Comparison to previous work

Goal: $\mathbb{P}_{f\in\mathcal{F}}[\|f(x)\|_2 \in (1\pm\epsilon)\|x\|_2] > 1-\delta$.

$v(m,\epsilon,\delta,s) :=$ sup over $v \in [0,1]$ s.t. sparse JL meets ℓ_2 goal on $x \in S_v$.

Bounds on v (Weinberger et al. '09,..., Freksen et al. '18):

- $v(m,\epsilon,\delta,\mathbf{1})$ understood
- $v(m,\epsilon,\delta,s)$ bound for *multiple hashing* (a suboptimal construction)

Bounds for sparse JL on full space $\mathbb{R}^n$:

- Can set $m \approx \epsilon^{-2}\log(1/\delta)$, $s \approx \epsilon^{-1}\log(1/\delta)$ (Kane and Nelson '12)
- Can set $m \approx \min(2\epsilon^{-2}/\delta, \epsilon^{-2}\log(1/\delta)e^{\Theta(\epsilon^{-1}\log(1/\delta)/s)})$ (Cohen '16)

This work

Tight bounds on $v(m,\epsilon,\delta,s)$ **for a general** $s > 1$ **for sparse JL**.

Comparison to previous work

Goal: $\mathbb{P}_{f \in \mathcal{F}}[\|f(x)\|_2 \in (1 \pm \epsilon)\|x\|_2] > 1 - \delta$.

$v(m, \epsilon, \delta, s) :=$ sup over $v \in [0,1]$ s.t. sparse JL meets ℓ_2 goal on $x \in S_v$.

Bounds on v (Weinberger et al. '09,..., Freksen et al. '18):

- $v(m, \epsilon, \delta, \mathbf{1})$ understood
- $v(m, \epsilon, \delta, s)$ bound for *multiple hashing* (a suboptimal construction)

Bounds for sparse JL on full space $\mathbb{R}^n$:

- Can set $m \approx \epsilon^{-2}\log(1/\delta)$, $s \approx \epsilon^{-1}\log(1/\delta)$ (Kane and Nelson '12)
- Can set $m \approx \min(2\epsilon^{-2}/\delta, \epsilon^{-2}\log(1/\delta)e^{\Theta(\epsilon^{-1}\log(1/\delta)/s)})$ (Cohen '16)

This work

Tight bounds on $v(m, \epsilon, \delta, s)$ **for a general** $s > 1$ **for sparse JL**.

$\implies$ *Characterization of sparse JL performance in terms of* ϵ, δ, *and* ℓ_∞*-to-*ℓ_2 *norm ratio for a general # of hash functions* s

Conclusion

Tight analysis of $v(m, \epsilon, \delta, s)$ for uniform sparse JL for a general s. Could inform how to optimally set s and m in practice.

Characterization of sparse JL performance in terms of ϵ, δ, and ℓ_∞-to-ℓ_2 norm ratio for a general # of hash functions s.

Evaluation on real-world and synthetic data (sparse JL can perform much better than feature hashing).

Proof technique involves a new perspective on analyzing JL distributions.

Thank you!