Fully Understanding the Hashing Trick

Lior Kamma, Aarhus University

Joint work with **Casper Freksen** and **Kasper Green Larsen.**

AARHUS
UNIVERSITY

Recommendation and Classification

- PG-13
- Comic Book
- Super Hero
- Sci Fi
- Adventure
- Action

- Violent
- Scary
- Comedy
- Drama
- Horror

Recommendation and Classification

PG-13

Comic Book

Super Hero

Sci Fi

Adventure

Action

Violent

Scary

Comedy

Drama

Horror

Categorical Variables

How do we decide these are “close”?

Feature Vectors

Comic Book Comedy Super Hero Sci Fi Scary Drama Adventure Leading Woman Leading Man

(1 0 1 1 0 0 1 1 0. . .)

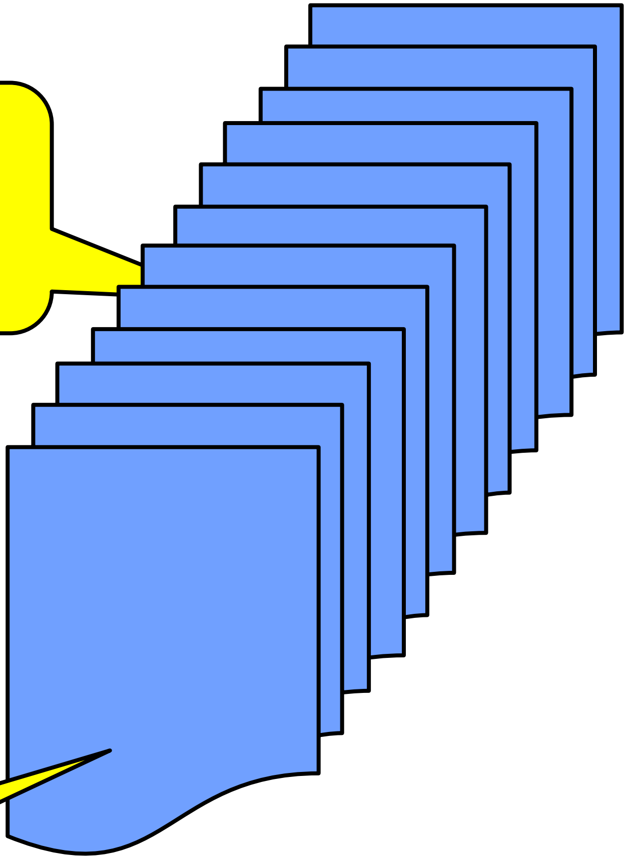
(1 0 1 1 0 0 1 0 1. . .)

Boolean vectors

Denote the *feature dimension* by n

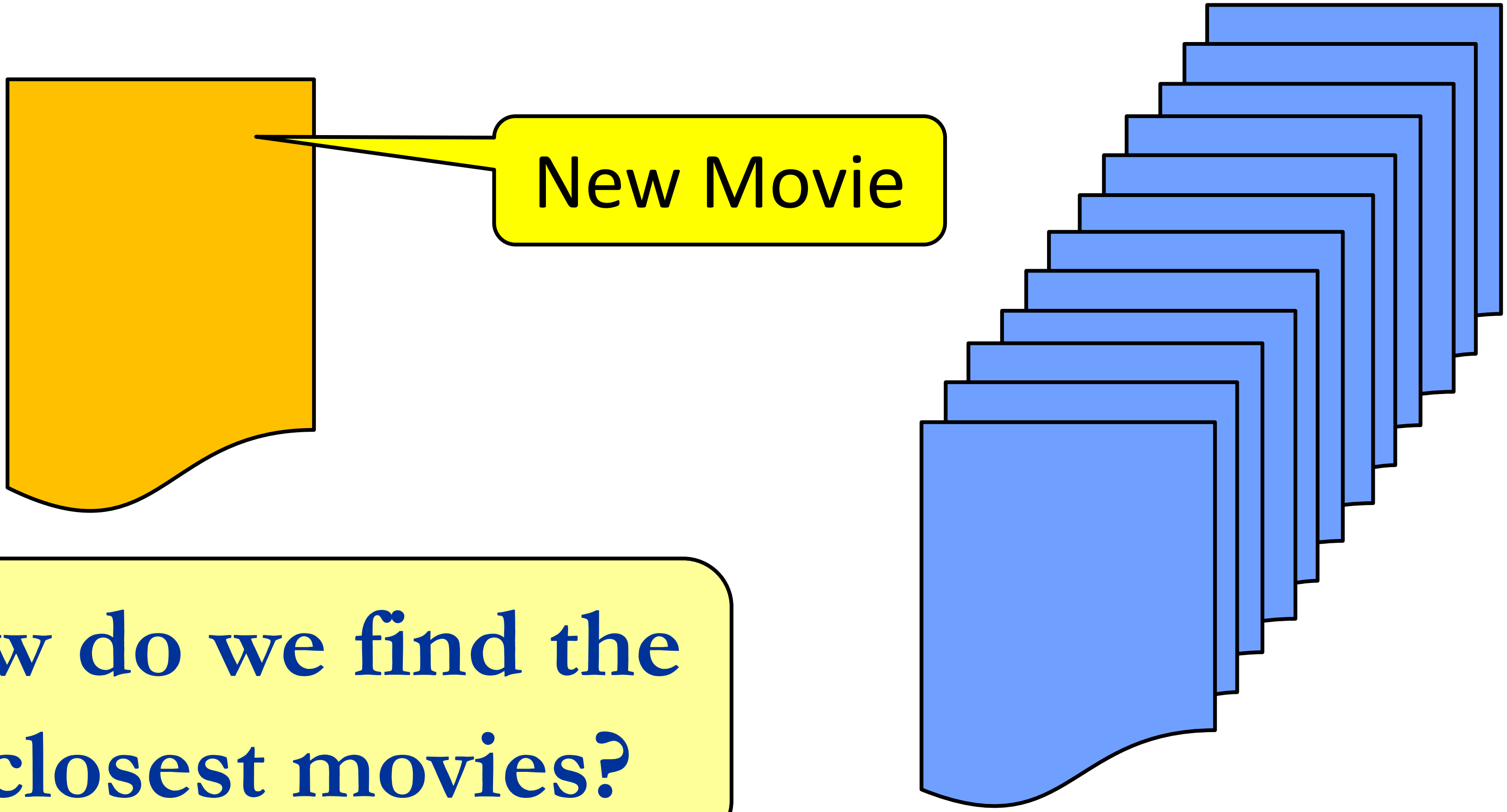
k-Nearest Neighbours

Storing a corpus of M items requires $\Omega(nM)$ memory



Corpus

k-Nearest Neighbours



How do we find the *k* closest movies?

Dimensionality Reduction

- Given $\varepsilon, \delta \in (0,1)$ find

Approximation Ratio

Error Probability

Dimensionality Reduction

For some small m

- Given $\varepsilon, \delta \in (0,1)$ find random $f: \mathbb{R}^n \to \mathbb{R}^m$ such that for every $x, y \in \mathbb{R}^n$

Think of n as **HUGE**

Dimensionality Reduction

- Given $\varepsilon, \delta \in (0,1)$ find random $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that for every $x, y \in \mathbb{R}^n$

$$\Pr[\|f(x) - f(y)\|_2^2 \in (1 \pm \varepsilon)\|x - y\|_2^2] \geq 1 - \delta$$

Dimensionality Reduction

- Given $\varepsilon, \delta \in (0,1)$ find random $A \in \mathbb{R}^{m \times n}$ such that for every $x, y \in \mathbb{R}^n$

$$\Pr[\|A(x-y)\|_2^2 \in (1 \pm \varepsilon)\|x-y\|_2^2] \geq 1-\delta$$

Focus on linear projections

Why linear?

- Cool Math
- Streaming (updates).
- Good in practice

Dimensionality Reduction

- Given $\varepsilon, \delta \in (0,1)$ find random $A \in \mathbb{R}^{m \times n}$ such that for every $x \in \mathbb{R}^n$

$$\Pr[\|A(x)\|_2^2 \in (1 \pm \varepsilon)\|x\|_2^2] \geq 1 - \delta$$

Focus on linear projections

Why linear?

- Cool Math
- Streaming (updates).
- Good in practice

Johnson Lindenstrauss Lemma [JL'84]

- Given $\varepsilon, \delta \in (0,1)$ there exists a random linear $A \in \mathbb{R}^{m \times n}$ such that for every x

$$\Pr[\|A(x)\|_2^2 \in (1 \pm \varepsilon)\|x\|_2^2] \geq 1 - \delta$$

$$m = O\left(\frac{\lg 1/\delta}{\varepsilon^2}\right)$$

In most proofs matrix is as dense as possible. Embedding takes $O(mn)$ operations.

Johnson Lindenstrauss Lemma [JL'84]

- Given $\varepsilon, \delta \in (0,1)$ there exists a random linear $A \in \mathbb{R}^{m\times n}$ such that for every x

$$\Pr[\|A(x)\|_2^2 \in (1 \pm \varepsilon)\|x\|_2^2] \geq 1 - \delta$$

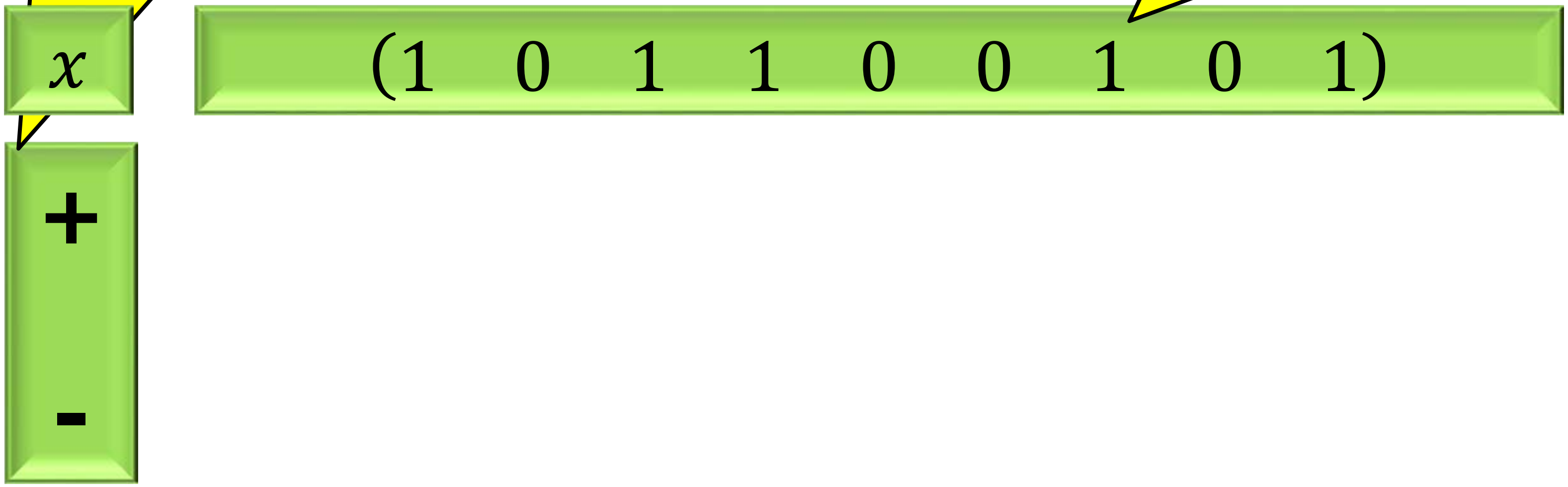
In most proofs matrix is as dense as possible. Embedding takes $O(mn)$ operations.

If A is sparse, this can be made faster.

Feature Hashing [Weinberger *et al.* 2009]

Add random signs

General Idea: Shuffle the entries of x



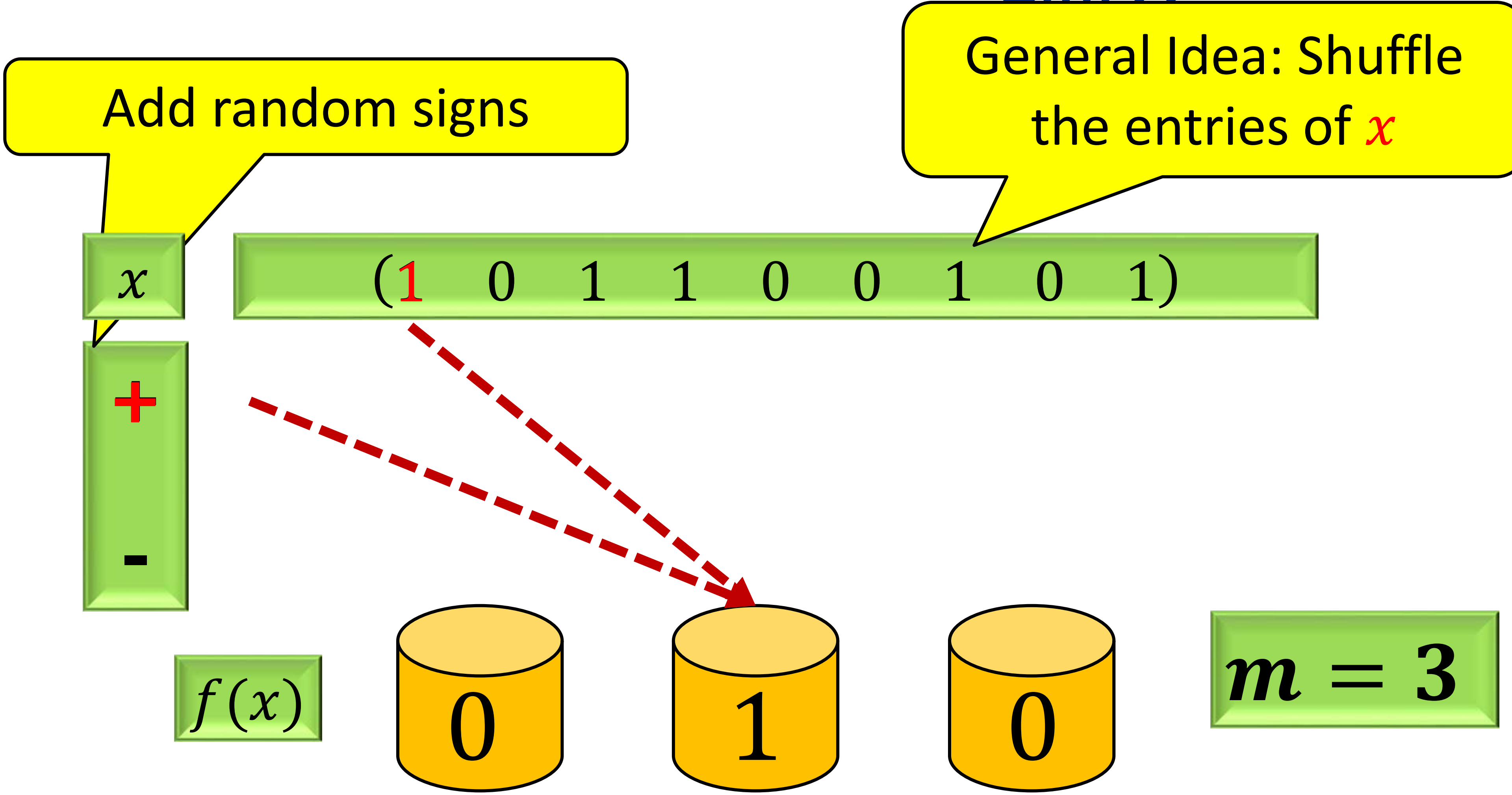
x

$(1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1)$

+

-

Feature Hashing [Weinberger *et al.* 2009]

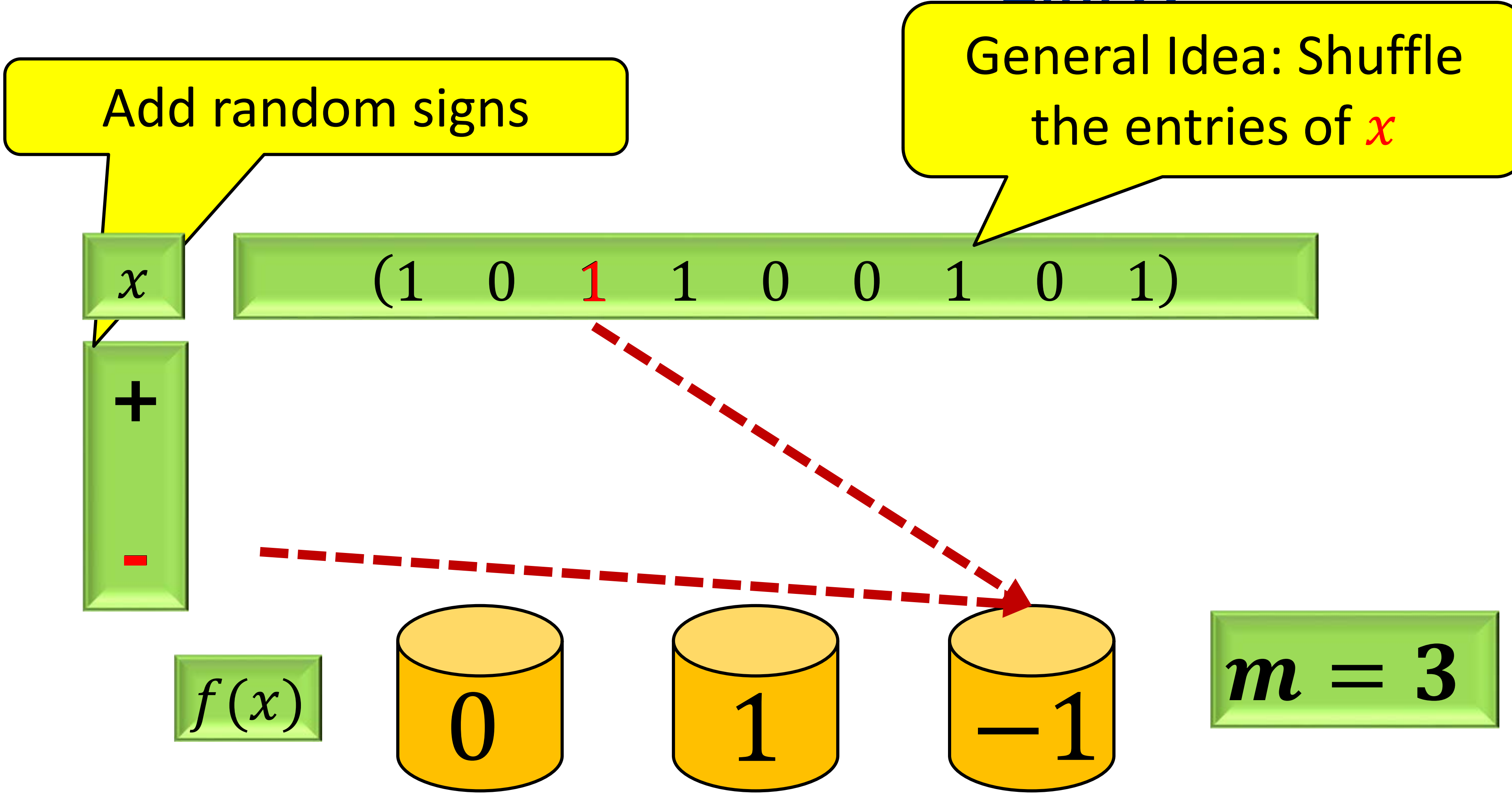


Add random signs

General Idea: Shuffle the entries of x

x
$(1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1)$
+
-
$f(x)$
0
1
0
$m = 3$

Feature Hashing [Weinberger *et al.* 2009]



Add random signs

General Idea: Shuffle the entries of x

x

$(1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1)$

$f(x)$

$\boldsymbol{m = 3}$

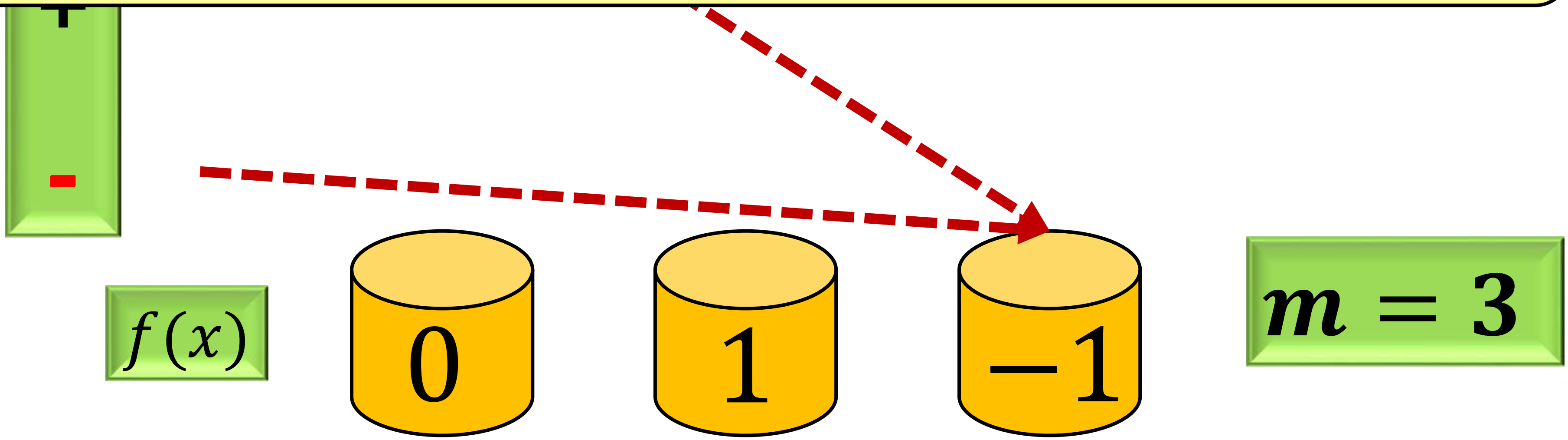
Feature Hashing [Weinberger *et al.* 2009]

Add random signs

General Idea: Shuffle the entries of x

Observation: **This operation is linear.**

Moreover, every column has exactly one non-zero entry.



The Hashing Trick – With High Prob.

- Observation: If m is large enough, and the "mass" of x is not concentrated in few entries, then the trick works with high probability.

$$\varepsilon = 0.1$$

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\Pr_{h:\{1,2,\ldots,n\}\to\{1,2,\ldots,m\}}[h(1) = h(2)] = \frac{1}{m}$$

$$\frac{\|x\|_\infty}{\|x\|_2} = \frac{1}{\sqrt{2}}.$$

The Hashing Trick – With High Prob.

nough, and the

m x is not concentrated in few entries,

the e trick works with high probability.

Success iff no collision occurs

$\varepsilon = 0.1$

$$\begin{pmatrix}1\\1\\0\\0\\\vdots\\0\end{pmatrix}$$

$$\Pr_{h:\{1,2,\ldots,n\}\to\{1,2,\ldots,m\}}[h(1) = h(2)] = \frac{1}{m}$$

$\frac{\|x\|_\infty}{\|x\|_2} = \frac{1}{[illegible]}$

To succeed we need $m \geq \frac{1}{\delta}$

Tight Bounds – Formal Problem

- Fix m, ε, δ.
- Define $\nu(m, \varepsilon, \delta)$ to be the maximum ν such that whenever $\|x\|_\infty \leq \nu \|x\|_2$ then feature hashing works.

Tight Bounds – Formal Problem

- Fix m, ε, δ.
- Define $\nu(m, \varepsilon, \delta)$ to be the maximum ν such that whenever $\|x\|_\infty \leq \nu \|x\|_2$ then feature hashing w

We have a fixed budget, and a fixed room for error.

Evaluating ν has been an open question for almost a decade.

Tight Bounds – Our Result

- Fix m, ε, δ.

Theorem.

1. If $m < \frac{c\ \log\frac{1}{\delta}}{\varepsilon^2}$ then $\nu = 0$.

Essentially, this means our budget is too small to do anything meaningful.

Tight Bounds – Our Result

- Fix m, ε, δ.

Theorem.

1. If $m < \frac{c\ \log\frac{1}{\delta}}{\varepsilon^2}$ then $\nu =$ (
2. If $m \geq \frac{2}{\delta\varepsilon^2}$ then $\nu = 1$.

Essentially, this means our budget is rich enough to do anything.

Tight Bounds – Our Result

- Fix m, ε, δ.

Theorem

This is tight,
which means this is the *right*
expression.

$\frac{C \log\frac{1}{\delta}}{\varepsilon^2} \le m < \frac{1}{\delta\varepsilon^2}$ then

$$\nu = \Theta\left(\sqrt{\varepsilon}\cdot\min\left\{\frac{\log\frac{\varepsilon m}{\log\frac{1}{\delta}}}{\log\frac{1}{\delta}}, \sqrt{\frac{\log\frac{\varepsilon^2 m}{\log\frac{1}{\delta}}}{\log\frac{1}{\delta}}}\right\}\right)$$

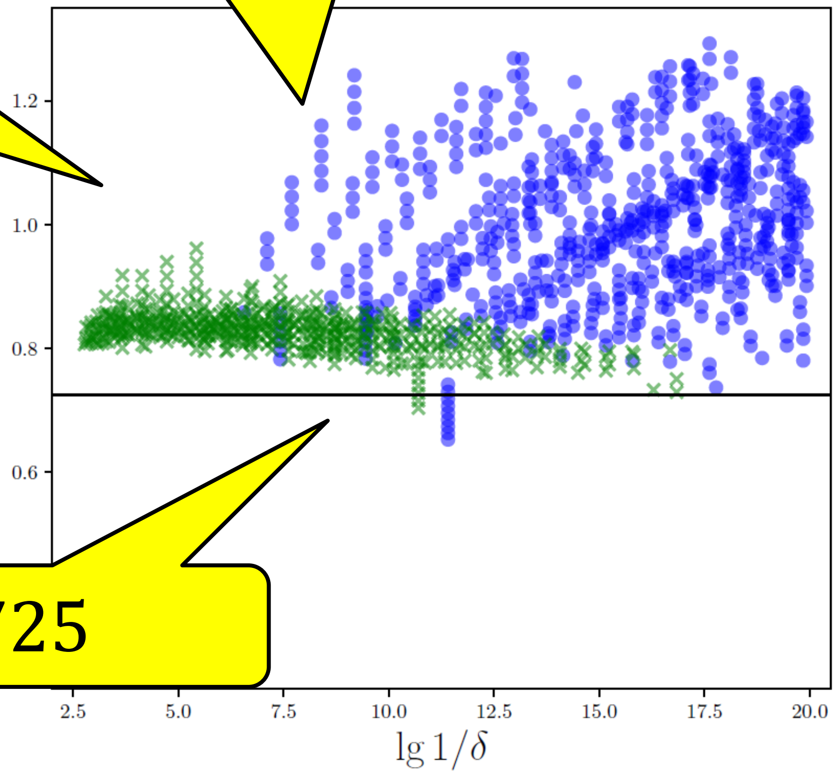
Empirical Analysis

Results show that the Θ-constant is close to 1.

This implies that Feature Hashing's performance can be very well predicted in practice using our formula.

$$\nu = \Theta\left(\sqrt{\varepsilon}\cdot\min\left\{\frac{\log\frac{\varepsilon m}{\log\frac{1}{\delta}}}{\log\frac{1}{\delta}}, \sqrt{\frac{\log\frac{\varepsilon^2 m}{\log\frac{1}{\delta}}}{\log\frac{1}{\delta}}}\right\}\right)$$

$$\frac{\nu}{\sqrt{\varepsilon}\ \min\left\{\left(\frac{\lg\frac{\varepsilon m}{\lg 1/\delta}}{\lg 1/\delta}\right), \sqrt{\frac{\lg\frac{\varepsilon^2 m}{\lg 1/\delta}}{\lg 1/\delta}}\right\}}$$



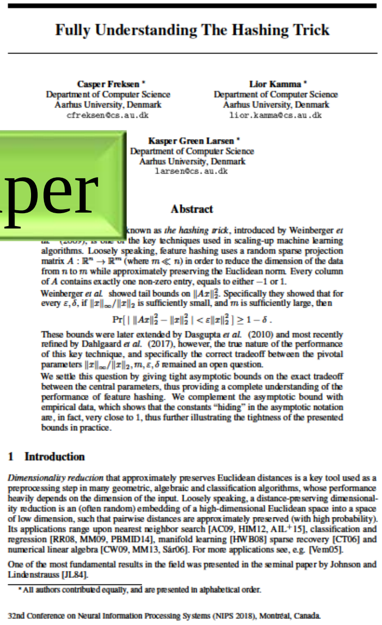
0.725

Questions?

Come see poster

1. Feature Hashing [Weinberger et al. '09]
Evaluating ε has been an open question for almost a decade
2. Motivation
feature dimension n
Storing a corpus of M movies requires O(nM) memory
Comparing two vectors takes O(n) operations
Example Problem: How do we find the k closest movies?
4. Proof Technique
Independent
6. Proof of Lower Bound
AARHUS UNIVERSITY

Read the paper



Fully Understanding The Hashing Trick

Casper Freksen *
Department of Computer Science
Aarhus University, Denmark
cfreksen@cs.au.dk

Lior Kamma *
Department of Computer Science
Aarhus University, Denmark
lior.kamma@cs.au.dk

Kasper Green Larsen *
Department of Computer Science
Aarhus University, Denmark
larsen@cs.au.dk

Abstract

known as *the hashing trick*, introduced by Weinberger *et al.* (2009), is one of the key techniques used in scaling-up machine learning algorithms. Loosely speaking, feature hashing uses a random sparse projection matrix $A : \mathbb{R}^n \to \mathbb{R}^m$ (where $m \ll n$) in order to reduce the dimension of the data from n to m while approximately preserving the Euclidean norm. Every column of A contains exactly one non-zero entry, equals to either -1 or 1.

Weinberger *et al.* showed tail bounds on $\|Ax\|_2^2$. Specifically they showed that for every ε, δ, if $\|x\|_\infty / \|x\|_2$ is sufficiently small, and m is sufficiently large, then

$$\Pr[\,|\,\|Ax\|_2^2 - \|x\|_2^2\,| < \varepsilon\|x\|_2^2\,] \geq 1 - \delta\,.$$

These bounds were later extended by Dasgupta *et al.* (2010) and most recently refined by Dahlgaard *et al.* (2017), however, the true nature of the performance of this key technique, and specifically the correct tradeoff between the pivotal parameters $\|x\|_\infty / \|x\|_2, m, \varepsilon, \delta$ remained an open question.

We settle this question by giving tight asymptotic bounds on the exact tradeoff between the central parameters, thus providing a complete understanding of the performance of feature hashing. We complement the asymptotic bound with empirical data, which shows that the constants "hiding" in the asymptotic notation are, in fact, very close to 1, thus further illustrating the tightness of the presented bounds in practice.

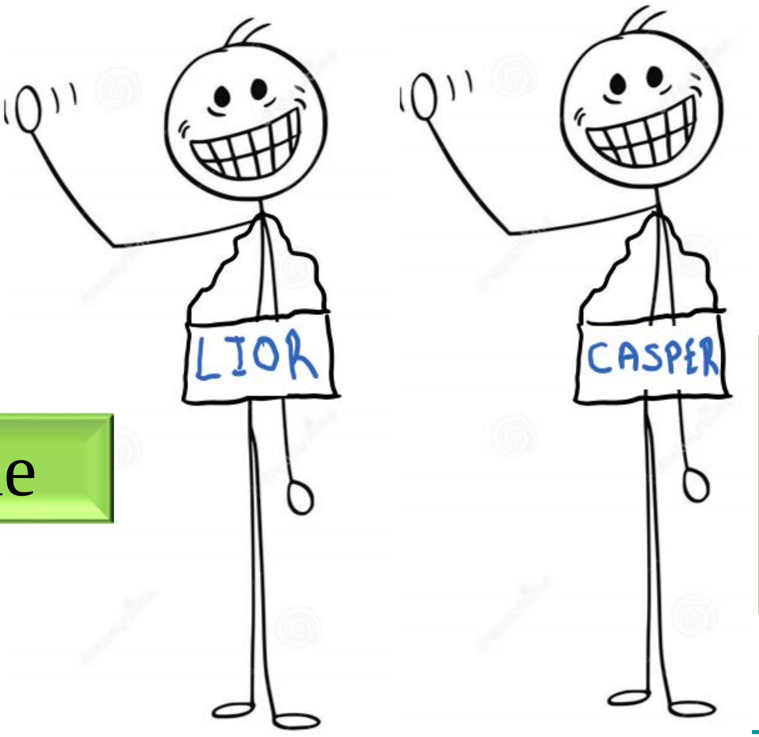
1 Introduction

Dimensionality reduction that approximately preserves Euclidean distances is a key tool used as a preprocessing step in many geometric, algebraic and classification algorithms, whose performance heavily depends on the dimension of the input. Loosely speaking, a distance-preserving dimensionality reduction is an (often random) embedding of a high-dimensional Euclidean space into a space of low dimension, such that pairwise distances are approximately preserved (with high probability). Its applications range upon nearest neighbor search [AC09, HIM12, AIL$^+$15], classification and regression [RR08, MM09, PBMID14], manifold learning [HWB08] sparse recovery [CT06] and numerical linear algebra [CW09, MM13, Sár06]. For more applications see, e.g. [Vem05].

One of the most fundamental results in the field was presented in the seminal paper by Johnson and Lindenstrauss [JL84].

* All authors contributed equally, and are presented in alphabetical order.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.



Talk offline

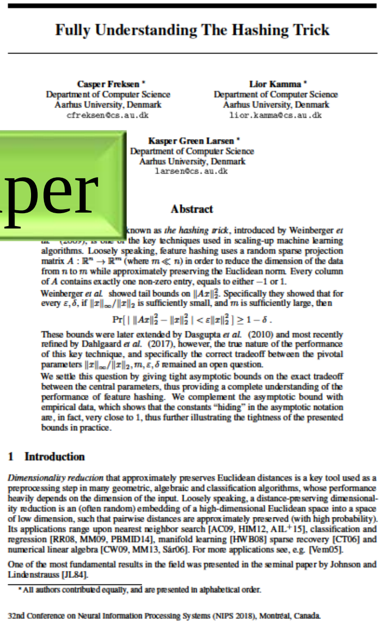
All of the above

Questions?

Come see poster

1. Feature Hashing [Weinberger et al. '09]
Approximation Ratio
Error Probability
Evaluating ε has been an open question for almost a decade
2. Motivation
Recommendation Systems
feature dimension n
Storing a corpus of M movies requires O(nM) memory
Comparing two vectors takes O(n) operations
Example Problem: How do we find the k closest movies?
4. Proof Technique
Independent
6. Proof of Lower Bound

Read the paper



Fully Understanding The Hashing Trick

Casper Freksen *
Department of Computer Science
Aarhus University, Denmark
cfreksen@cs.au.dk

Lior Kamma *
Department of Computer Science
Aarhus University, Denmark
lior.kamma@cs.au.dk

Kasper Green Larsen *
Department of Computer Science
Aarhus University, Denmark
larsen@cs.au.dk

Abstract

known as *the hashing trick*, introduced by Weinberger *et al.* (2009), is one of the key techniques used in scaling-up machine learning algorithms. Loosely speaking, feature hashing uses a random sparse projection matrix $A : \mathbb{R}^n \to \mathbb{R}^m$ (where $m \ll n$) in order to reduce the dimension of the data from n to m while approximately preserving the Euclidean norm. Every column of A contains exactly one non-zero entry, equals to either -1 or 1.

Weinberger *et al.* showed tail bounds on $\|Ax\|_2^2$. Specifically they showed that for every ε, δ, if $\|x\|_\infty / \|x\|_2$ is sufficiently small, and m is sufficiently large, then

$$\Pr[\,|\,\|Ax\|_2^2 - \|x\|_2^2\,| < \varepsilon \|x\|_2^2\,] \geq 1 - \delta\,.$$

These bounds were later extended by Dasgupta *et al.* (2010) and most recently refined by Dahlgaard *et al.* (2017), however, the true nature of the performance of this key technique, and specifically the correct tradeoff between the pivotal parameters $\|x\|_\infty / \|x\|_2, m, \varepsilon, \delta$ remained an open question.

We settle this question by giving tight asymptotic bounds on the exact tradeoff between the central parameters, thus providing a complete understanding of the performance of feature hashing. We complement the asymptotic bound with empirical data, which shows that the constants "hiding" in the asymptotic notation are, in fact, very close to 1, thus further illustrating the tightness of the presented bounds in practice.

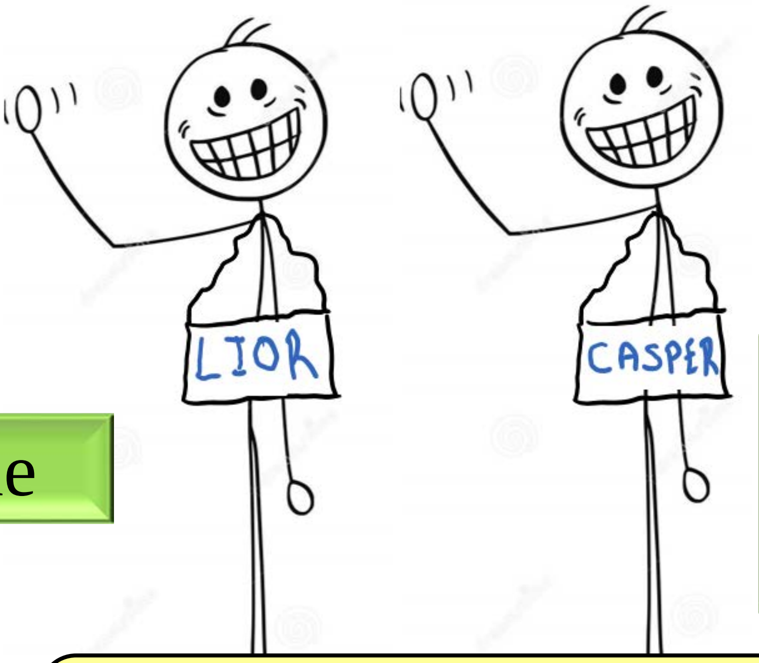
1 Introduction

Dimensionality reduction that approximately preserves Euclidean distances is a key tool used as a preprocessing step in many geometric, algebraic and classification algorithms, whose performance heavily depends on the dimension of the input. Loosely speaking, a distance-preserving dimensionality reduction is an (often random) embedding of a high-dimensional Euclidean space into a space of low dimension, such that pairwise distances are approximately preserved (with high probability). Its applications range upon nearest neighbor search [AC09, HIM12, AIL+15], classification and regression [RR08, MM09, PBMID14], manifold learning [HWB08] sparse recovery [CT06] and numerical linear algebra [CW09, MM13, Sár06]. For more applications see, e.g. [Vem05].

One of the most fundamental results in the field was presented in the seminal paper by Johnson and Lindenstrauss [JL84].

* All authors contributed equally, and are presented in alphabetical order.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.



Talk offline

All of the above

Thank you