DD2434 Machine Learning, Advanced Course

Module 8 : randomization techniques in machine learning

Aristides Gionis

`argioni@kth.se`

KTH Royal Institute of Technology

in previous lectures

- discussed common dimensionality-reduction methods, i.e., PCA, MDS, and Isomap
- methods rely on minimizing the reconstruction error
- a typical statement : find a mapping $f : \mathbb{R}^d \to \mathbb{R}^k$, with $k \ll d$ to minimize

$$E_f = \mathbb{E}[\|\mathbf{x} - \mathbf{y}\| - \|f(\mathbf{x}) - f(\mathbf{y})\|]$$

- so, reconstruction error guarantees distance preservation on expectation
- for a given pair of points (outliers) the error can be very large
- question : can we devise a mapping to preserve distances in the worst case?
- **yes!** the Johnson-Lindenstrauss lemma

overview of module 8

- essential probability tools for large-scale data analysis
- the Johnson-Lindenstrauss lemma
- data streams and computation of frequency moments

reading material

- your favorite book on probability, computing, and randomized algorithms
  - e.g., Motwani and Raghavan, Randomized algorithms (chapters 3 and 4)
- Dasgupta and Gupta. An Elementary Proof of a Theorem of Johnson and Lindenstrauss. 2002
- Achlioptas. Database-friendly Random Projections. 2003
- Alon, Matias, and Szegedy. The space complexity of approximating the frequency moments. 1999

essential probability tools

- union bound
- linearity of expectation
- concentration inequalities
  - Markov inequality, Chebyshev inequality, Chernoff bound

the union bound

- by the probability axioms we know that for any finite (or countably infinite) sequence of pairwise mutually disjoint events $E_1, E_2, \ldots$ it is

$$\Pr\left[\bigcup_{i\geq 1} E_i\right] = \sum_{i\geq 1} \Pr[E_i]$$

- moreover, for any events $E_1, E_2, \ldots, E_n$

$$\Pr\left[\bigcup_{i=1}^{n} E_i\right] \leq \sum_{i=1}^{n} \Pr[E_i]$$

how to apply the union bound

- consider a random process for which we can identify the possible "bad" events
- assume that "bad" event i happens with probability p_i
- union bound says that probability that any "bad" event happens is at most $\sum_i p_i$
- if we can show that $\sum_i p_i$ is (significantly) less than 1
- then, probability of success (no "bad" event) is at least $1 - \sum_i p_i$

random variable

- a random variable X on a sample space Ω is a function $X : \Omega \to \mathbb{R}$
- a discrete random variable takes only a finite (or countably infinite) number of values

expectation and variance of a random variable

- the expectation of a discrete random variable X, denoted by $\mathbb{E}[X]$, is given by

$$\mathbb{E}[X] = \sum_x x\,\mathbb{P}[X = x] = \mu_X$$

  where the summation is over all values in the range of X

- variance

$$\mathrm{Var}\,[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}\left[(X - \mu_X)^2\right] = \sigma_X^2$$

linearity of expectation

- for any two random variables X and Y

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

- for a constant c and a random variable X

$$\mathbb{E}[cX] = c\,\mathbb{E}[X]$$

linearity of expectations — application to coupon collector

- consider a collector of coupons (e.g., stamps, coins, football cards, etc.)
- assume n different coupon types
- in each trial, the collector picks a coupon type at random
- how many trials are needed, in expectation, until the collector gets all the coupon types?

analysis (1/2)

- let $c_1, c_2, \dots, c_X$ the sequence of coupon types picked, $c_i \in \{1, \dots, n\}$
- so, the random variable X denotes the total number of trials until all types are collected
- call c_i success if a new coupon type is picked
  - c_1 and c_X are always successes
- divide the sequence in epochs:
  - the i-th epoch starts after the i-th success and ends with the $(i+1)$-th success
  - thus, i ranges from 0 to $n-1$
- define X_i to be the length of the i-th epoch
- clearly $X = \sum_{i=0}^{n-1} X_i$

analysis (2/2)

- probability of success in the i-th epoch

$$p_i = \frac{n-i}{n}$$

- X_i is geometrically distributed with parameter p_i, and its expectation is

$$E[X_i] = \frac{1}{p_i} = \frac{n}{n-i}$$

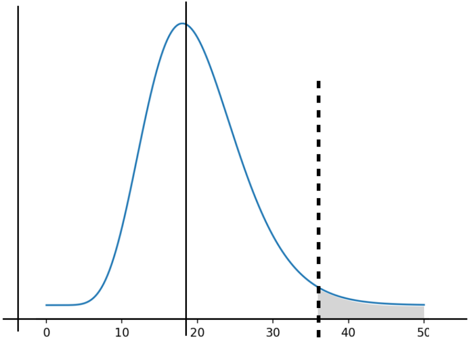
- from linearity of expectation

$$E[X] = E\left[\sum_{i=0}^{n-1} X_i\right] = \sum_{i=0}^{n-1} E[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=1}^{n} \frac{1}{i} = nH_n$$

where H_n is the harmonic number, asymptotically equal to $\ln n$

concentration inequalities

- also known as tail inequalities, or concentration bounds
- we want to bound the probability that a random variable deviates from its expectation
- many different bounds, depending on the kind of distribution we are interested in



Markov inequality

Theorem

let X *a random variable taking non-negative values*

for all $t > 0$

$$\Pr[X \geq t] \leq \frac{E[X]}{t}$$

or equivalently

$$\Pr[X \geq k\, E[X]] \leq \frac{1}{k}$$

- one of the simplest bounds, it makes no assumption other than non-negativity
- however, because of its generality, it is also a weak bound

Proof.

- by the definition of expectation $\mathbb{E}[f(X)] = \sum_x f(x)\,\mathbb{P}[X = x]$
- define $f(x) = 1$ if $x \geq t$ and 0 otherwise
- applying the first equation with this particular function f gives $\mathbb{E}[f(X)] = \mathbb{P}[X \geq t]$
- notice also that $f(x) \leq x/t$, which implies

$$\mathbb{E}[f(X)] \leq \mathbb{E}\left[\frac{X}{t}\right]$$

- putting everything together

$$\mathbb{P}[X \geq t] = \mathbb{E}[f(X)] \leq \mathbb{E}\left[\frac{X}{t}\right] = \frac{\mathbb{E}[X]}{t}$$

the last step is linearity of expectation

$\square$

Chebyshev inequality

Theorem

let X *a random variable with expectation* μ_X *and standard deviation* σ_X

then for all $t > 0$

$$\Pr[|X - \mu_X| \geq t\sigma_X] \leq \frac{1}{t^2}$$

Proof.

- notice that

$$\mathbb{P}[|X - \mu_X| \geq t\sigma_X] = \mathbb{P}\left[(X - \mu_X)^2 \geq t^2\sigma_X^2\right]$$

- the random variable $Y = (X - \mu_X)^2$ has expectation σ_X^2
- apply the Markov inequality on Y

$\square$

Chernoff bounds

Theorem

let $X_1, \ldots, X_n$ *independent Poisson trials, i.e.,* $\mathbb{P}[X_i = 1] = p_i$ *and* $\mathbb{P}[X_i = 0] = 1 - p_i$

define $X = \sum_i X_i$, *so* $\mu = \mathbb{E}[X] = \sum_i \mathbb{E}[X_i] = \sum_i p_i$

for any $\delta > 0$

$$\mathbb{P}[X > (1+\delta)\mu] \leq e^{-\frac{\delta^2\mu}{3}}$$

and

$$\mathbb{P}[X < (1-\delta)\mu] \leq e^{-\frac{\delta^2\mu}{2}}$$

a simple example of using the Chernoff bound

- consider n coin flips; define $X_i = 1$ if i-th coin flip is `H` and 0 if `T`
- the total number of `H`'s is $X = \sum_i X_i$, and expectation is $\mu = E[X] = n/2$
- pick $\delta = \frac{2c\sqrt{n}}{n}$, so $(1-\delta)\mu = (1 - \frac{2c\sqrt{n}}{n})\frac{n}{2} = \frac{n}{2} - c\sqrt{n}$
- for that δ the r.h.s. of the Chernoff bound will be $e^{-\frac{\delta^2\mu}{2}} = e^{-\frac{4c^2 \cdot n \cdot n}{n^2 \cdot 2 \cdot 2}} = e^{-c^2}$

  which drops very fast with c
- the Chernoff bound gives

$$\mathbb{P}\left[X < \frac{n}{2} - c\sqrt{n}\right] = \mathbb{P}[X < (1-\delta)\mu] \leq e^{-\frac{\delta^2\mu}{2}} = e^{-c^2}$$

  and similarly $\mathbb{P}\left[X > \frac{n}{2} + c\sqrt{n}\right] \leq e^{-\frac{\delta^2\mu}{3}} = e^{-2c^2/3}$
- so, probability that number of `H`'s is outside the range $\left[\frac{n}{2} - c\sqrt{n}, \frac{n}{2} + c\sqrt{n}\right]$ is very small

final notes

- there is a extensive amount of work dedicated to concentration inequalities
  - distribution of the random variable
  - additive vs. multiplicative deviations
  - independent vs. dependent settings
- useful in randomized algorithms, large-scale data analysis, machine learning theory

what we want to achieve in a nutshell

- given a set of n points $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in $\mathbb{R}^d$ we want to construct a mapping $f : \mathbb{R}^d \to \mathbb{R}^k$, with $k \ll d$, so that

$$\|\mathbf{x}_i - \mathbf{x}_j\| \approx \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|, \quad \text{for all } \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$$

- the emphasis here is on the "for all"
  - notice that methods like PCA preserve distances "on expectation"
- it is also important that we aim for $k \ll d$
  - the result implies that we can embed a set of points in a much lower dimensional space without loosing much information
- the mapping f will be linear, and will be constructed using random projections
- this result has many applications in machine learning and theoretical computer science

the Johnson-Lindenstrauss lemma, precise formulation

Theorem (Johnson-Lindenstrauss, 1984)

for any $0 < \epsilon < 1$ *and any integer* n, *let*

$$k \geq \frac{4\ln(n)}{\epsilon^2/2 - \epsilon^3/3}$$

then for any set of points $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ *in* $\mathbb{R}^d$ *there exists a mapping* $f : \mathbb{R}^d \to \mathbb{R}^k$, *such that for all* $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$:

$$(1-\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 \leq (1+\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2$$

some observations about the Johnson-Lindenstrauss lemma

- result is about existence of a mapping
  - however, such a mapping can be constructed very easily
- ϵ is a parameter that controls the quality of distance preservation
- dimensionality k of lower-dimensional space grows with $\mathcal{O}(\epsilon^{-2})$
  - this is intuitive, as smaller distortion requires larger dimension of the host space
- dimensionality k of lower-dimensional space grows with $\ln(n)$ (number of points)
- d can be as large as n

  (but not larger, as n points lie always on a $(n-1)$-dimensional hyperplane)
- thus, the reduction in dimension can be exponentially large (from n to $\mathcal{O}(\ln(n))$)
  - this can be very useful for algorithms that are exponential in d
  - after a random projection such algorithms become polynomial in d
- on the other hand, the lemma is useful only if the dimension d is large enough
  - in particular, it should be $d = \omega(\ln(n))$

random projections

1-dimensional space

- let $\mathbf{z}$ be a random vector drawn from the uniform distribution on the unit sphere in $\mathbb{R}^d$
- the function $\pi_{\mathbf{z}} : \mathbb{R}^d \to \mathbb{R}$ with $\pi_{\mathbf{z}}(\mathbf{x}) = \langle \mathbf{z}, \mathbf{x} \rangle = \mathbf{z}^T \mathbf{x}$, for $\mathbf{x} \in \mathbb{R}^d$ is a random projection of $\mathbf{x}$ on the 1-dimensional space

k-dimensional space

- let $\mathbf{z}_1, \ldots, \mathbf{z}_k$ be k random vectors drawn from the uniform distribution on the unit sphere in $\mathbb{R}^d$
- let $\mathcal{Z}$ be the subspace spanned by $\{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$, and let $\mathbf{Z}$ be a $k \times d$ matrix whose rows are the vectors $\mathbf{z}_1, \ldots, \mathbf{z}_k$
- the function $\pi_{\mathcal{Z}} : \mathbb{R}^d \to \mathbb{R}^k$ with $\pi_{\mathcal{Z}}(\mathbf{x}) = (\langle \mathbf{z}_1, \mathbf{x} \rangle, \ldots, \langle \mathbf{z}_k, \mathbf{x} \rangle) = \mathbf{Z}\,\mathbf{x}$, for $\mathbf{x} \in \mathbb{R}^d$ is a random projection of $\mathbf{x}$ on the k-dimensional space

expected length of random projections

Proposition (1-dimensional random projection)

let $\mathbf{x}$ *be a fixed vector in* $\mathbb{R}^d$ *with* $\|\mathbf{x}\| = 1$*; let* $\pi_{\mathbf{z}}(\mathbf{x})$ *be its projection on a random vector* $\mathbf{z}$ *sampled from the uniform distribution on the unit sphere in* $\mathbb{R}^d$*; then*

$$\mathbb{E}_{\mathbf{z}}\left[|\pi_{\mathbf{z}}(\mathbf{x})|^2\right] = 1/d$$

- (random projection shrinks vector lengths by $\sqrt{1/d}$)

Proof.

- by a symmetry argument we can assume that $\mathbf{x} = \mathbf{e}_1 = (1, 0 \ldots, 0)$
- then
$$\mathbb{E}_{\mathbf{z}}\left[|\pi_{\mathbf{z}}(\mathbf{x})|^2\right] = \mathbb{E}_{\mathbf{z}}\left[|\pi_{\mathbf{z}}(\mathbf{e}_1)|^2\right] = \mathbb{E}_{\mathbf{z}}\left[|\langle \mathbf{z}, \mathbf{e}_1 \rangle|^2\right] = \mathbb{E}_{\mathbf{z}}\left[|z_1|^2\right]$$
- the random vector $\mathbf{z}$ has unit length
$$1 = \|\mathbf{z}\|^2 = \mathbb{E}_{\mathbf{z}}\left[\|\mathbf{z}\|^2\right] = \mathbb{E}_{\mathbf{z}}\left[\sum_i |z_i|^2\right] = \sum_i \mathbb{E}_{\mathbf{z}}\left[|z_i|^2\right]$$
- by symmetry again
$$\mathbb{E}_{\mathbf{z}}\left[|z_i|^2\right] = 1/d \quad \text{for all } i = 1, \ldots, d$$
- we can conclude
$$\mathbb{E}_{\mathbf{z}}\left[|\pi_{\mathbf{z}}(\mathbf{x})|^2\right] = \mathbb{E}_{\mathbf{z}}\left[|z_1|^2\right] = 1/d$$

$\square$

expected length of random projections

Proposition (k-dimensional random projection)

let $\mathbf{x}$ *be a fixed vector in* $\mathbb{R}^d$ *with* $\|\mathbf{x}\| = 1$*; let* $\pi_{\mathcal{Z}}(\mathbf{x})$ *be its projection on a* k*-dimensional random space* $\mathcal{Z}$*; then*

$$\mathbb{E}_{\mathcal{Z}}\left[\|\pi_{\mathcal{Z}}(\mathbf{x})\|^2\right] = k/d$$

- (random projection on k-dimensional space shrinks vector lengths by $\sqrt{k/d}$)

Proof.

- consider the rotation $\mathbf{R}$ that maps $\mathcal{Z}$ to the k-dimensional space $\mathcal{E}_k$ spanned by the k basis vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$, that is, $\mathcal{E}_k = \mathbf{R}\mathcal{Z}$; also define $\mathbf{y} = \mathbf{R}\,\mathbf{x}$
- since a rotation does not change the vector norms we have

$$\mathbb{E}_{\mathcal{Z}}\left[\|\pi_{\mathcal{Z}}(\mathbf{x})\|^2\right] = \mathbb{E}_{\mathbf{R}\mathcal{Z}}\left[\|\pi_{\mathbf{R}\mathcal{Z}}(\mathbf{R}\mathbf{x})\|^2\right] = \mathbb{E}_{\mathbf{R}\mathcal{Z}}\left[\|\pi_{\mathcal{E}_k}(\mathbf{y})\|^2\right]$$

- length of a fixed unit vector projected on a random subspace $=$
  $=$ length of a random unit vector projected on a fixed subspace
- therefore, we have

$$\mathbb{E}_{\mathbf{R}\mathcal{Z}}\left[\|\pi_{\mathcal{E}_k}(\mathbf{y})\|^2\right] = \mathbb{E}_{\mathbf{y}}\left[\|\pi_{\mathcal{E}_k}(\mathbf{y})\|^2\right] = \mathbb{E}_{\mathbf{y}}\left[\sum_{i=1}^{k} |\langle \mathbf{e}_i, \mathbf{y}\rangle|^2\right] = \sum_{i=1}^{k} \mathbb{E}_{\mathbf{y}}\left[|y_i|^2\right] = k/d$$

$\square$

so far

- we have shown that the expected length of a unit vector on a random k-dimensional subspace is $\sqrt{k/d}$
- we also want to show that the distribution is sharply concentrated around its mean

concentration properties of random projections

Proposition

let $L = \|\pi_{\mathcal{Z}}(\mathbf{x})\|^2$, *be the squared length of a random projection of unit vector* $\mathbf{x}$

then $\mathbb{E}[L] = k/d$, *and*

for $\beta > 1$

$$\mathbb{P}\left[L \geq \beta \frac{k}{d}\right] \leq \exp\left(\frac{k}{2}(1 - \beta + \ln \beta)\right)$$

for $\beta < 1$

$$\mathbb{P}\left[L \leq \beta \frac{k}{d}\right] \leq \exp\left(\frac{k}{2}(1 - \beta + \ln \beta)\right)$$

- (the probability that $\|\pi_{\mathcal{Z}}(\mathbf{x})\|^2$ deviates by more than a factor β from its expectation is exponentially small)

implication of concentration inequalities

Proposition

let $0 < \epsilon < 1$ *and* $k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \ln(n)$

let $\mathbf{x}$ *be any vector, and* $L = \|\pi_{\mathcal{Z}}(\mathbf{x})\|^2$

define $\mu = k/d\|\mathbf{x}\|^2$, *so* $\mathbb{E}[L] = \mu$

then

$$\mathbb{P}[L \geq (1+\epsilon)\mu] \leq \frac{1}{n^2} \quad \textit{and} \quad \mathbb{P}[L \leq (1-\epsilon)\mu] \leq \frac{1}{n^2}$$

Proof.

- by the concentration inequality claimed before, with $\beta = 1 + \epsilon > 1$

$$\begin{aligned}\mathbb{P}[L \geq (1+\epsilon)\mu] &\leq \exp\left(\frac{k}{2}(1-(1+\epsilon)+\ln(1+\epsilon))\right) \\ &\leq \exp\left(\frac{k}{2}\left(-\epsilon+(\epsilon-\epsilon^2/2+\epsilon^3/3)\right)\right) = \exp\left(-\frac{k(\epsilon^2/2-\epsilon^3/3)}{2}\right) \\ &\leq \exp(-2\ln n) = \frac{1}{n^2}\end{aligned}$$

  where the second inequality follows by $\ln(1+x) \leq x - x^2/2 + x^3/3$,
  which holds for all $x \geq 0$
  and the last inequality holds by our choice of $k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1}\ln(n)$

- the case of $\mathbb{P}[L \leq (1-\epsilon)\mu] \leq \frac{1}{n^2}$ can be shown in a similar manner

$\square$

proof of Johnson-Lindenstrauss lemma (1/2)

- consider mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ so that $f(\mathbf{x}) = \sqrt{d/k}\,\pi_{\mathcal{Z}}(\mathbf{x})$
- fix two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ in the set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
- consider the difference of $\mathbf{x}_i$ and $\mathbf{x}_j$ as a new vector $\mathbf{y} = \mathbf{x}_i - \mathbf{x}_j$
- note that $\|f(\mathbf{x}_i) - f(\mathbf{x}_j)\| = \|f(\mathbf{x}_i - \mathbf{x}_j)\| = \|f(\mathbf{y})\|$
- the expected length of $f(\mathbf{y})$ is simply $\mathbb{E}\left[\|f(\mathbf{y})\|^2\right] = \|\mathbf{y}\|^2$ and by the previous proposition

$$\mathbb{P}\left[\|f(\mathbf{y})\|^2 \leq (1-\epsilon)\|\mathbf{y}\|^2\right] \leq \frac{1}{n^2} \quad \text{and} \quad \mathbb{P}\left[\|f(\mathbf{y})\|^2 \geq (1+\epsilon)\|\mathbf{y}\|^2\right] \leq \frac{1}{n^2}$$

so

$$(1-\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 \leq (1+\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \tag{JL}$$

does not hold with probability at most $2/n^2$ (union bound)

proof of Johnson-Lindenstrauss lemma (2/2)

- consider now all $\binom{n}{2}$ pairs of points in $\mathcal{X}$
- (JL) does not hold for at least one pair with probability at most (union bound, again)

$$\frac{(n-1)\,n}{2} \cdot \frac{2}{n^2} = 1 - \frac{1}{n}$$

- it follows that (JL) holds for all pairs with probability at least $\frac{1}{n}$
- probability of existence of a mapping f that satisfies (JL) for all pairs is at least $\frac{1}{n}$
- non zero probability implies that such a mapping exists
  - (an instance of the probabilistic method)

□

how can we find such a random projection?

- we know that a “good” random projection exists with probability at least $1/n$
- we can easily sample a random projection and check if it satisfies the desired property
  - if not, we discard it and repeat
- we will need $\mathcal{O}(n)$ trials to find one with the desired property
- checking one trial requires time $\mathcal{O}(n^2 d)$
- so, we have a randomized algorithm
- total running time $\mathcal{O}(n^3 d)$, on expectation
  - quite expensive, but at least polynomial

how to construct a random projection?

- we need to sample random vectors from the uniform distribution on the unit sphere in $\mathbb{R}^d$
- how to sample one such a vector?

answer:

1. sample each coordinate independently from the normal distribution $\mathcal{N}(0,1)$
2. normalize the vector to unit norm

how to construct a random projection?

- an alternative elegant construction was given by Achlioptas (2003) :
- create a $k \times d$ matrix $\mathbf{Z}$, where each entry z_{ij} is sampled independently as follows

$$z_{ij} = \begin{cases} 1 & \text{with probability } 1/6 \\ 0 & \text{with probability } 4/6 \\ -1 & \text{with probability } 1/6 \end{cases}$$

- the matrix $\mathbf{Z}$ is used to define the random projection $f(\mathbf{x}) = \sqrt{3/k}\,\mathbf{Z}\,\mathbf{x}$

simplified version of Johnson-Lindenstrauss lemma

Theorem (Achlioptas, 2003)

for $\epsilon, \beta > 0$*, and integer* n*, take*

$$k \geq \frac{4 + 2\beta}{\epsilon^2/2 - \epsilon^3/3} \ln(n)$$

for any set of points $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ *in* $\mathbb{R}^d$ *construct the mapping* f *as shown in the previous slide;*

then

$$(1 - \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 \leq (1 + \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2$$

holds for all pairs $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$*, with probability at least* $1 - \frac{1}{n^\beta}$

simplified version of Johnson-Lindenstrauss lemma

- notes on the JL version by Achlioptas
  - spherical symmetry is not essential; concentration is more crucial
  - construction succeeds with high probability; one sampled matrix is sufficient (whp)
  - matrix $\mathbf{Z}$ is sparse, matrix-vector multiplication very easy; "database friendly"

embeddings in algorithmic design and machine learning

- the Johnson-Lindenstrauss lemma can be used to speed up many algorithms, simply by reducing the dimensionality of the data
- the topic of embedding is very broad and has many different "flavors"
  - high to low dimensional spaces
  - general metrics to vector spaces
  - graphs to trees
  - graphs to vector spaces

embeddings in algorithmic design and machine learning

- k-means clustering of d-dimensional points :
  - there exist a random projection of the data to dimension $\mathcal{O}(k/\epsilon^2)$ that preserves the k-means clustering solution to factor $2+\epsilon$
- column subset selection problem
  - select k columns of a matrix that give the best rank-k approximation of the matrix
  - random projections can be used to give a good approximation to this problem
- streaming computation
  - compute a “sketch” over a data stream to estimate useful properties of the stream
  - many streaming algorithms rely on random projections

data streams

- a data stream is a massive sequence of data
  - too large to store (on disk, memory, cache, etc.)
- examples:
  - social media (e.g., twitter feed, foursquare checkins)
  - sensor networks (weather, radars, cameras, etc.)
  - network traffic (trajectories, source/destination pairs)
  - satellite data feed
- how to deal with such data? what are the issues?

issues when working with data streams

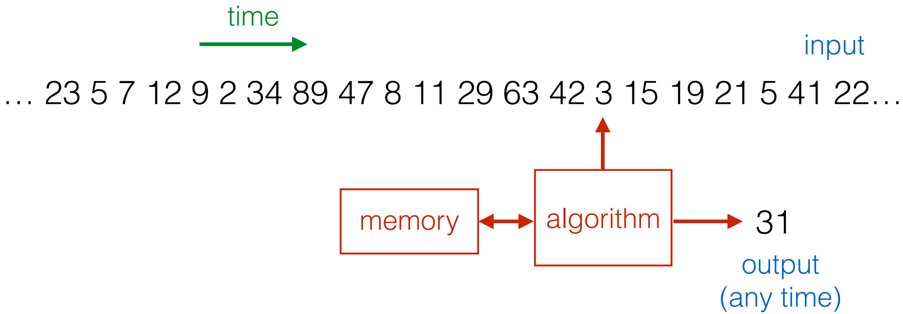
- space
  - data size is very large
  - often not possible to store the whole dataset
  - inspect each data item, make some computations, not possible to store it, never get to inspect it again
  - some times data is stored, but making one single pass takes a lot of time, especially when the data is stored on disk
  - other times, we can afford a small number of passes over the data
- time
  - data "flies by" at a high speed
  - computation time per data item needs to be small

data streams

- data items can be of complex types
  - documents, images, geo-located time-series . . .
- to study basic algorithmic ideas we abstract away application-specific details
- consider the data stream as a sequence of numbers

data-stream model

for simplicity assume that the input is a stream of numbers



data-stream model

- stream: m elements from universe of size n, e.g.,

$$\langle x_1, x_2, \ldots, x_m \rangle = 6, 1, 7, 4, 9, 1, 5, 1, 5, \ldots$$

- goal: compute a function over the elements of the stream, e.g., median, number of distinct elements, quantiles, etc.

- constraints:

  1. limited working memory, sublinear in n and m, e.g., $\mathcal{O}(\log n + \log m)$,
  2. access data sequentially
  4. limited number of passes, in some cases only one pass
  4. process each element quickly, e.g., $\mathcal{O}(1)$, $\mathcal{O}(\log n)$, etc.

warm up: computing some simple functions

- assume that a number can be stored in $\mathcal{O}(\log n)$ space
- min and max can be computed with $\mathcal{O}(\log n)$ space
- sum and mean need $\mathcal{O}(\log n + \log m)$ space

$$\mu_X = \mathbb{E}[X] = \mathbb{E}[x_1, \ldots, x_m] = \frac{1}{m}\sum_{i=1}^{m} x_i$$

- variance

$$\mathrm{Var}\left[X\right] = \mathrm{Var}\left[x_1, \ldots, x_m\right] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_X)^2$$

- can be computed in a single pass?

how to tackle massive data streams?

- a general and powerful technique: sampling
- idea:
  1. keep a random sample of the data stream
  2. perform the computation on the sample
  3. extrapolate
- example: compute the median of a data stream
  - the median of the sample is a good estimate of the median on the whole stream

how to tackle massive data streams?

- a general and powerful technique: sampling
- idea:
  1. keep a random sample of the data stream
  2. perform the computation on the sample
  3. extrapolate
- example: compute the median of a data stream
  - the median of the sample is a good estimate of the median on the whole stream
- but . . . how to keep a random sample of a data stream?

reservoir sampling

- problem : take a uniform sample s from a stream of unknown length while keeping in memory a single number

reservoir sampling

- problem : take a uniform sample s from a stream of unknown length while keeping in memory a single number
- algorithm :
  1. initially $s \leftarrow x_1$
  2. on seeing the t-th element, $s \leftarrow x_t$ with probability $1/t$

reservoir sampling

- problem : take a uniform sample s from a stream of unknown length while keeping in memory a single number
- algorithm :
  1. initially $s \leftarrow x_1$
  2. on seeing the t-th element, $s \leftarrow x_t$ with probability $1/t$
- analysis :
  - what is the probability that $s = x_i$ at some time $t \geq i$?

$$
\begin{aligned}
\mathbb{P}[s = x_i] &= \frac{1}{i}\left(1 - \frac{1}{i+1}\right) \cdots \left(1 - \frac{1}{t-1}\right)\left(1 - \frac{1}{t}\right) \\
&= \frac{1}{i}\,\frac{i}{i+1} \cdots \frac{t-2}{t-1}\,\frac{t-1}{t} = \frac{1}{t}
\end{aligned}
$$

- reservoir sampling algorithm uses $\mathcal{O}(\log n)$ bits of memory
- can easily be extended to taking k samples with $\mathcal{O}(k \log n)$ bits of memory

how to tackle massive data streams?

- another powerful technique: sketching
- idea:
  - apply a random projection that maps high-dimensional data to a small "sketch"
  - post-process sketch to estimate quantities of interest

computing frequency moments on data streams

- $X = (x_1, x_2, \ldots, x_m)$ a sequence of elements
  each element x_j has a "key" in the set $N = \{1, \ldots, n\}$
  $m_i = |\{j : x_j = i\}|$ the number of occurrences of key i

computing frequency moments on data streams

- $X = (x_1, x_2, \ldots, x_m)$ a sequence of elements
  each element x_j has a “key” in the set $N = \{1, \ldots, n\}$
  $m_i = |\{j : x_j = i\}|$ the number of occurrences of key i

- define the k-th frequency moment

$$F_k = \sum_{i=1}^{n} m_i^k$$

computing frequency moments on data streams

- $X = (x_1, x_2, \ldots, x_m)$ a sequence of elements
  each element x_j has a "key" in the set $N = \{1, \ldots, n\}$
  $m_i = |\{j : x_j = i\}|$ the number of occurrences of key i
- define the k-th frequency moment

$$F_k = \sum_{i=1}^{n} m_i^k$$

  - F_0 is the number of distinct elements
  - F_1 is the length of the sequence
  - F_2 is the second moment: index of homogeneity, size of self-join, other applications
  - F_∞^* frequency of most frequent element

computing frequency moments on data streams

- we can compute all frequency moments using $O(n \log m)$ memory bits in a straightforward manner
- can be done more efficiently?
- problem studied by Alon, Matias, and Szegedy in their seminal paper
- the idea is to create a sketch that requires small space and provides an estimate of F_k
- sketch can be considered a random projection on the streaming setting

estimating F_2

- recall our problem setting :

  $X = (x_1, x_2, \ldots, x_m)$ a sequence of elements

  each element x_j has a "key" in the set $N = \{1, \ldots, n\}$

  $m_i = |\{j \,:\, x_j = i\}|$ the number of occurrences of key i

  $F_k = \sum_{i=1}^{n} m_i^k$

estimating F_2

- recall our problem setting :

  $X = (x_1, x_2, \ldots, x_m)$ a sequence of elements

  each element x_j has a "key" in the set $N = \{1, \ldots, n\}$

  $m_i = |\{j \, : \, x_j = i\}|$ the number of occurrences of key i

  $F_k = \sum_{i=1}^{n} m_i^k$

- the sketching algorithm :
  - hash each key $i \in \{1, \ldots, n\}$ to a random $\epsilon_i \in \{-1, +1\}$
  - maintain sketch $Z = \sum_i \epsilon_i m_i$, need only $\mathcal{O}(\log n + \log m)$ space
  - take $X = Z^2$
  - return $Y = \frac{1}{k} \sum_{j=1}^{k} X_j$, i.e., the average of k such estimates $X_1, \ldots, X_k$,

    where $k = \frac{16}{\lambda^2}$, and where λ controls the accuracy of the estimate

expectation of the estimate is correct

$$
\begin{aligned}
\mathbb{E}[X] &= \mathbb{E}\left[Z^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{n} \epsilon_i m_i\right)^2\right] \\
&= \sum_{i=1}^{n} m_i^2\,\mathbb{E}\left[\epsilon_i^2\right] + 2\sum_{i<j} m_i m_j\,\mathbb{E}[\epsilon_i]\,\mathbb{E}[\epsilon_j] \\
&= \sum_{i=1}^{n} m_i^2 = F_2
\end{aligned}
$$

accuracy of the estimate

easy to show

$$\mathbb{E}[X^2] = \sum_{i=1}^{n} m_i^4 + 6 \sum_{i<j} m_i^2 m_j^2$$

accuracy of the estimate

easy to show

$$\mathbb{E}\left[X^2\right] = \sum_{i=1}^{n} m_i^4 + 6 \sum_{i<j} m_i^2 m_j^2$$

which gives

$$\mathrm{Var}\left[X\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 = 4 \sum_{i<j} m_i^2 m_j^2 \leq 2F_2^2$$

accuracy of the estimate

easy to show

$$\mathbb{E}[X^2] = \sum_{i=1}^{n} m_i^4 + 6 \sum_{i<j} m_i^2 m_j^2$$

which gives

$$\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = 4 \sum_{i<j} m_i^2 m_j^2 \leq 2F_2^2$$

and by Chebyshev's inequality

$$\Pr[|Y - F_2| \geq \lambda F_2] \leq \frac{\mathrm{Var}[Y]}{\lambda^2 F_2^2} = \frac{\mathrm{Var}[X]/k}{\lambda^2 F_2^2} \leq \frac{2F_2^2/k}{\lambda^2 F_2^2} = \frac{2}{k\lambda^2} = \frac{1}{8}$$

estimation of F_2

Theorem (Alon, Matias, Szegedy, 1999)

let $X_1, \ldots, X_k$ be AMS sketches, with $k = \frac{16}{\lambda^2}$, and Y be their average $Y = \frac{1}{k}\sum_{j=1}^{k} X_j$

then, Y is an unbiased estimator of F_2, and the quality of the approximation is given by

$$\Pr[|Y - F_2| \geq \lambda F_2] \leq \frac{1}{8}$$

summary and discussion

- random projections have many applications, also in streaming computation
- streaming is a widely-researched topic; it has been studied in the context of
  - estimating number of distinct items in data streams
  - estimating frequencies and finding most frequent items
  - matrix approximations and PCA
  - graph mining