20241007 Softmax Mashup

$$g(q,k,v)_j = \sum_i \frac{\exp(q\cdot k_i)\cdot v_{ij}}{\sum_z \exp(q\cdot k_z)}$$

$$h(q,k)_i = \frac{\exp(q\cdot k_i)}{\sum_z \exp(q\cdot k_z)} = \frac{\exp\left(-\frac{\|q-k_i\|_2^2}{2}\right)}{\sum_z \exp\left(-\frac{\|q-k_z\|_2^2}{2}\right)}$$

$k : \mathbb{R}^{N\times d}$

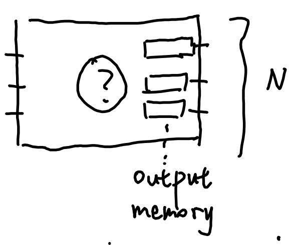
$v : \mathbb{R}^{N\times d}$

$$f(q,x)_j = \sum_i h(q, R(xK))_i \cdot (xV)_{ij}$$

20241007 Combined Input & Output Queueing

Output Queueing : When input packet arrives at the switch, it is immediately queued to the output port.

This can only be achived (for worst case) if output memory is faster than input memory by N times.



Knock-out Scheme : if output memory is only faster by k times, only k packets are considered for each output queue.

Idea: use input queueing to emulate output queuing results.

If output memory is 2x faster than input memory, the <u>packet scheduling policy</u> is restricted.

20241007 Quality of Service

1. throughput
2. latency
3. jitter

Packet Scheduling Policy :

1. $\underbrace{\text{work-conserving}}_{\text{don't idle if the queue is not empty}}$, v.s. not

20241007 Packet Scheduling Policy : Deficit Round Robin

Idea : each flow is assigned a quantum, i.e. a flow occupies multiple positions in round-robin.

each flow has a deficit counter

Algorithm :

1. rotating among the flows
2. action on a flow f : add a quantum to the flow

20241007 Packet Scheduling Policy : Weight Fair Queuing

Generalized Processor Sharing : compute the finish time for each packet using infinitely fast round robin.

Among all HOL packets, select the one with earliest GPS finishing time.

20241007 Why uniformity condition holds for $d=1$?

$$q_{i,1}(d)$$

$$q_{i-1}(d) \geq q_i(d) + q_i(d+1)$$

$$\mathbb{P}\{X_i = \{j\}\} = p_S(i,1)\cdot q_{i-1}(1) = q_i(1)$$

$$\mathbb{P}\{X_i = \{i\}\} = \sum_d p_R(i,d)\cdot\mu_{i-1}(d) = \sum_d \binom{i-1}{d} q_{i-1}(d)\cdot(1-p_A(\cdot,\cdot)-p_S(\cdot,\cdot))$$

$$= \sum_d \binom{i-1}{d}(q_{i-1}(d) - q_i(d+1) - q_i(d))$$

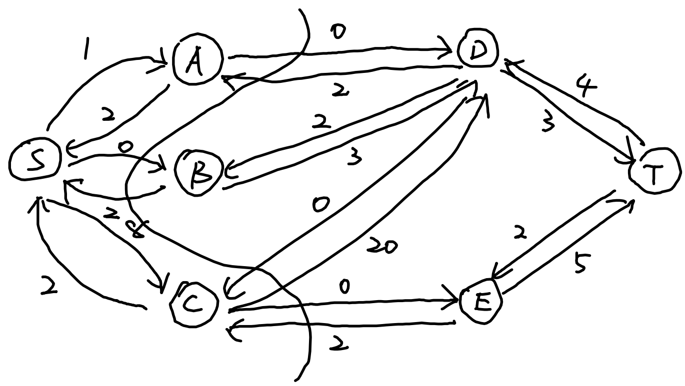
$$= \overset{1}{\cancel{\sum_d \binom{i-1}{d} q_{i-1}(d)}} - \sum_{d=2}^{i}\binom{i-1}{d-1} q_i(d) - \sum_{d=1}^{i-1}\binom{i-1}{d} q_i(d)$$

$$= 1 - \binom{i-1}{i-1} q_i(i) - \binom{i-1}{1} q_i(1) - \sum_{d=2}^{i-1}\binom{i}{d} q_i(d)$$

$$= q_i(1)$$

2024 1008 Test Practice

Draw residual network



max flow is 6

min cut is SAC BDET

Short answers

1. poly-time is polynomial to the # of input bits.
   pseude poly-time is polynomial to input numbers, e.g. the value of a max-flow.
2. PTAS is poly(input bits) when approximation error ε is fixed.
   FPTAS is poly (input bits, $1/\varepsilon$), which is stricter.
3. Join two matchings together, we will get even cycles and alternating paths.
   If current matching is not maximum, there should be an augmenting path.
4. A set where any linear combination of two points is also in the set.
5. Conceptually, you can view $G_f(\Delta)$ as Δ-rounded G_f, where each edge e is scaled by $\lfloor \frac{e}{\Delta} \rfloor$. Therefore, they are different concepts.
6. Yes, Farka's lemma states equivalence between the existence of λ and feasibility
7. Yes, we can compute is with dynamic programming.

20241008 Test Practice

LP Approximation of Knapsack

1. for the original problem we compute max(O)

$$O = \{ \sum_i v_i x_i : \sum_i s_i x_i \le B \wedge x_i \in \{0,1\} \}$$

for approximation: max(S)

$$S = \{ \sum_i v_i x_i : \sum_i s_i x_i \le B \wedge 0 \le x_i \le 1 \}$$

since $O \subseteq S$ $\max(O) \le \max(S)$

$$\Delta \le \sum_i v_i x_i^*$$

2. Lagrange Multiplier:

$$\max_x \min_{\lambda,\mu,\nu} \sum_i v_i x_i + \lambda (B - \sum_i s_i x_i) \cancel{+ \sum_i \mu_i x_i} + \sum_i \nu_i (1 - x_i)$$

Dual LP:

$$\min_{\lambda,\nu} \lambda B + \sum_i \nu_i \quad \Big\} \quad \sum_i (v_i - \lambda s_i - \nu_i) x_i$$

$$v_i - \lambda s_i - \nu_i \le 0 \ (\forall i)$$

$$\lambda \ge 0$$

$$\nu_i \ge 0 \ (\forall i)$$

3. For other solutions, if there exists a variable x_i $(i \le k)$ such that $x_i \ne 1$, then set $x_i = \frac{B - \sum_{j \ne i} s_j x_j}{s_i}$ gives a better solution when $\sum_i s_i x_i < B$. which means the best solution must have $\sum_i s_i x_i = B$.

If a pair of (x_i, x_j) has $i \le k$ but $x_i \ne 1$, then there exists a $x_j \ne 0$ $(i \ge k+1)$, adjust $x_i = 1$ and

$$x_j = \frac{B - \sum_{k \ne j,i} s_k x_k - s_i}{s_j}.$$

There value at least doesn't decrease.

If $x_{k+1} < (B - \sum_{i \le k} s_i)/s_{k+1}$, then there exists a $x_j \ne 0$ $(j > k+1)$, analogously, adjust x_{k+1} to $(B - \sum_{i \le k} s_i)/s_{k+1}$ makes an at least doesn't worse solution.

Therefore, to be optimal, the solution must take the given form.

4. Take the first k items is also a feasible solution of the original knapsack problem.

Therefore, $\sum_{1 \le i \le k} x_i^* v_i \le \Delta$.

Now we only have to proof $(B - \sum_{i \le k} s_i)/s_{k+1} \cdot v_{k+1} \le \Delta$.

We know $B - \sum_{i \le k} s_i \le s_{k+1}$, or else k can be k+1.

Therefore it suffices to show $\Delta \ge v_{k+1}$.

Without loss of generality, we suppose $s_i \le B$, then this is true.

20241008 Longest Increasing Subseq

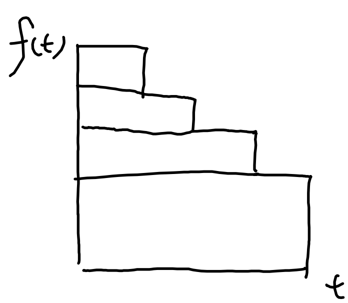
$T[i, v]$ = maximum length of subseq considering $a[1 \dots i]$ where $a[\cdot] \le v$

$$T[i, v] = \begin{cases} T[i-1, v] & \text{if } v < a[i] \\ \max\{T[i-1, v], T[i-1, a[i]]+1\} & \text{if } v \ge a[i] \end{cases}$$

$$T[0, v] = \begin{cases} 0 & \text{if } a[0] > v \\ 1 & \text{o.w.} \end{cases}$$

20241009 Staircase Query Problem

Given a staircase function, determine t s.t. $\int_0^t f(x)\,dx = V$.



However, some of these rectangles may grow.

20241010 Post-training Compression

Problem: Model of different sizes are all trained from scratch

Idea: Some attention heads are not important. Remove them to get smaller models. Estimate importance with small data. (calibration data)

Q: Domain Shift of Training Data

Q: Serve with SLoRA? How does the router models change when the biggest model is changed?

Fact: with finetuning, last layers matter less.
without ..., first layers matter less.

20241010 Simplex Algorithm

Observation: for a linear programming problem, the optimal solution is either too or at a "corner" (intersection of d constraints) (if there exists d linearly independent constraints)

Idea: start from a corner;
move along edges s.t. the value of LP increases;

Compute initial corner: construct a new problem $\begin{pmatrix} A & \begin{matrix}-1\\ \vdots \\ -1\end{matrix} \\ -I & 0 \end{pmatrix}\begin{pmatrix} x \\ \hline z \end{pmatrix} \le \begin{pmatrix} b \\ 0 \end{pmatrix}$

Decide the constraint to remove from $(Ax)_I \le b_I$

① Pick index i, if adjusting $(Ax)_I \le b_I - \text{onehot}_i \cdot \varepsilon$ makes $c^T x$ larger, then constraint i can be removed.

$c^T (A_I)^{-1} \text{onehot}_i < 0$ ← condition

② Pick index j, take $I' = I \setminus \{i\} \cup \{j\}$ s.t. $c^T A$

20241015 Walzer Code

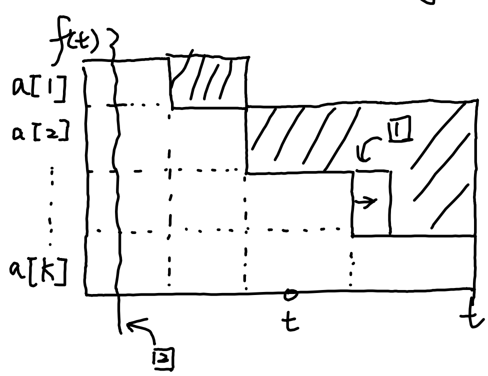
$e(x) = \{h_1(x), \dots, h_k(x)\} \subseteq [n]$

$(h_\cdot)$ independent and fully random

l-peelability : min degree of buckets is at most l.

Idea : embed code and input to two lines

20241016 Staircase Query Cont.



Queries : $F(t) = \int_0^t f(t)\,dt$

① Given t, compute $F(t)$

② Give $F(t)$, compute t

Updates :

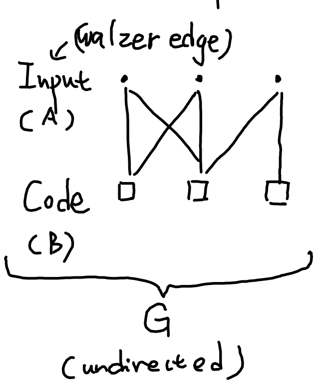
$\boxed{1}$ Increase a give stair $a[i]$

$\boxed{2}$ Crop each stair by the same value

Solution : augment a binary search tree

① complete the stairs in to a rectangle, store the added area

20241018 Spatial Coupling



Equivalent Descriptions for Peelability (d-peelability) :

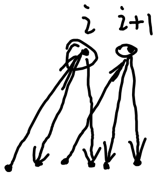
$\exists$ orientation $\vec{G}$ $\begin{cases} ① \text{ acyclic} \\ ② \ \forall v \in A : \text{in degree is exactly } 1 \\ ③ \ \forall u \in B : \text{out degree is at most } d \end{cases}$

2024 1021 Streaming

$X_{i,d}$: the packet selected by i^{th} input at d^{th} throw

Previously, $X_{i,j}$ & $X_{i,k}$ are i.i.d. (for Walzer's Construction)

Now, let's suppose $\Pr\{X_{i,j+1} > X_{i,j}\} = 1$



Idea: instead of proving $\Pr\{\exists$ acyclic orientation$\}$

prove a certain orientation

instantiate a 1-peelable graph w.h.p.

Suppose i fails, then all $X_{i,(\cdot)}$ are dirty.

Idea: make i and $i+1$ not overlap.

Idea: embed inputs into the codeword where it is decoded.

In current scheme, if we drop a codeword with ε prob., alternatively mask input w.p. ε has worse implication to later packets.

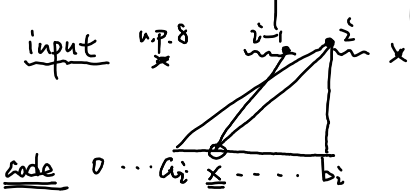
wrong, for a (codeword may not be covered by input

2024 1021 Recipe, Cont.

Suppose the switch has changing information, how to gather changing information?

2024 1021 Streaming, Input mask δ define $x \vee y = 1-(1-x)(1-y)$

Given input dropping by rate δ, for given i, compute



$\Pr\{\underline{x}$ is contaminated for some reason$\} = \mathbb{E}\{\gamma(\underline{x})\}$

$= \delta \cdot \Pr\{\bigvee_{j=1}^{i-1} (j \text{ covers } \underline{x})\} \vee \bigvee_{j=i+1}^{\infty} \Pr\{j \text{ covers } \underline{x}\} \vee \varepsilon$

$= \lambda(\underline{x}) \leftarrow \gamma(x)$ is a Bernoulli r.v. for each x ①

sort z by $X_{i,z}$ desc.

get new $X_{i,\cdot}$

(w.l.o.g.)

$\Pr\{\underline{i}$ cannot decode before $\{\underline{i+1}, \dots\}\} = \delta$ ②

$= \Pr\{\forall \underline{x} : \underline{i} \text{ covers } \underline{x} \Rightarrow \underline{x} \text{ is contaminated}\}$

$= \mathbb{E}\{\prod_{z=1}^{d} \gamma(X_{i,z})\} = \mathbb{E}\{\mathbb{E}\{\prod_{z=1}^{d-1}\gamma(X_{i,z}) \mid X_{i,d}\}\cdot\gamma(X_{i,d})\}$

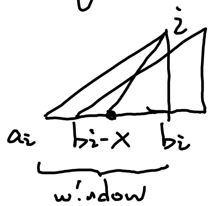
$\underline{i}$: packet i

$\underline{x}$: codeword x

20241021 Streaming, Input mask δ, Examples

Distribution 1 : Sample Unif i.i.d. $X_{i,\cdot}$ (count 1 for repeated in implementation)

$q(i, b_i - x)$ = the probability of i select $b_i - x$ $\quad x \in \{0, \cdots, b_i - a_i + 1\}$ (o/w zero)



not p.m.f, so write q linearility $\quad X_{i,k}$ has distrib.

$$= \underbrace{1 - (1-\beta^{-1})^d}_{} \leftarrow q$$

$a_i = \lfloor \alpha i \rfloor$

$b_i = \lfloor \alpha i \rfloor + \beta - 1$

$$\lambda(b_i - x) = \left(\delta - (1-q)^{\lceil \frac{\beta}{\alpha} \rceil}\delta\right) \vee \left(1 - (1-\varepsilon)(1-q)^{\lceil \frac{\beta}{\alpha} \rceil}\right)$$

$$= 1 - (1-\varepsilon)(1 - \delta + (1-q)^{\lceil \frac{\beta}{\alpha} \rceil}\delta)(1-q)^{\lceil \frac{\beta}{\alpha} \rceil}$$

$$= \underbrace{1 - (1-\varepsilon)(1-q)^{\lceil \frac{\beta}{\alpha} \rceil}}_{\zeta} + \delta \underbrace{(1 - (1-q)^{\lceil \frac{\beta}{\alpha} \rceil})(1-\varepsilon)(1-q)^{\lceil \frac{\beta}{\alpha} \rceil}}_{\xi}$$

$$\lambda(x) = \zeta + \xi\delta \quad x \in \{a_i, \cdots, b_i\} \quad ①$$

$$\delta = \sum_{g=1}^{d} \frac{\beta^g g^{d-g}}{\beta^d} \binom{d}{g} (\zeta + \xi\delta)^g$$

20241024 Gradient Descent

L-smooth functions :

$f : \mathbb{R}^n \to \mathbb{R}$, differentiable

$\|\nabla f(x) - \nabla f(y)\| \le L \cdot \|x - y\|$

e.g. $f(x) = e^x$ is not L-smooth

L-Lipshitz functions :

$|f(x) - f(y)| \le L \cdot \|x - y\|$

Theorem : f is L-Lipshitz & convex $\Rightarrow \left(\frac{\sum_{t=1}^{T} f(x^t)}{T}\right) - f(x^*) \le \frac{\eta L^2}{2} + \frac{\|x^0 - x^*\|}{2\eta T}$

if $\eta = \frac{\|x^0 - x^*\|^2}{L\sqrt{T}}$

Collary : $f\left(\frac{\sum_{t=1}^{T} x^t}{T}\right) - f(x^*) \le \frac{\eta L^2}{2} + \frac{\|x^0 - x^*\|^2}{2\eta T}$

Analyze $\|x^{t+1} - x^*\|_2^2 = \|x^t - \eta \cdot \nabla f(x^t) - x^*\|_2^2$

$$= \|x^t - x^*\|_2^2 - 2\eta \nabla f(x^t)^T (x^t - x^*) + \|\eta \cdot \nabla f(x^t)\|_2^2$$

$$\le \|x^t - x^*\|_2^2 - 2\eta f(x^t) + 2\eta f(x^*) + \eta^2 L^2$$

20241024 Gradient Descent

$$f(x^t) - f(x^*) \le \frac{\|x^t - x^*\|_2^2 - \|x^{t+1} - x^*\|_2^2 + \eta^2 L^2}{2\eta}$$ f is convex & L-lipschitz

$$\sum_{t=0}^{T-1} \left(f(x^t) - f(x^*)\right) \le \frac{\|x^0 - x^*\|_2^2 - \|x^t - x^*\|_2^2 + \eta^2 L^2}{2\eta}$$

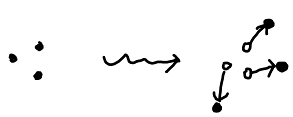
$$\frac{\sum_{t=0}^{T-1} f(x^t)}{T} - f(x^*) \le \frac{\|x^0 - x^*\|_2^2}{2\eta T} + \frac{\eta L^2}{2}$$

20241024 Stochastic Gradient Descent

Don't have gradient, but have $g(x)$ $\mathbb{E}[g(x)] = \nabla f(x)$ & $\|g(x)\| \le L$

Then $x^{t+1} \leftarrow x^t - \eta g^t$ is still have $\frac{\sum_{t=0}^{T-1} f(x^t)}{T} - f(x^*) \le \frac{\|x^0 - x^*\|_2^2}{2\eta T} + \frac{\eta L^2}{2}$

20241025 Nearest Neighbor Search via Diffusion



$$\min_z \|f(z) - q\|_2^2$$

f is something smooth but weird, then generally HNSW good

Can we modify f to a sequence of $\{f^{(i)}\}$ s

make things look easier? (with $\lim_{i \to \infty} f^{(i)} = f$, roughly)

Idea: co-iterate $f^{(i)}$ & $z^{(i)}$ s.t.

① if $z^{(i)}$ is near optimal for $f^{(i)}$, it is near optimal for $f^{(i+1)}$

② each iteration of $z^{(i)} \rightsquigarrow z^{(i+1)}$ makes it a better solution to $f^{(i+1)}$

$\forall q$: ① & ②, this can be tough

Issue: $z^{(i)}$ is discrete $f^{(i)} \to f^{(i+1)}$ may lose ①.

$z^{(i)} \to z^{(i+1)}$ may not let $f^{(i+1)}$ go down.

20241025 Preliminary : LSH for L_2 Vector Search

$$h(x) = \left\lfloor \frac{a^Tx+b}{w} \right\rfloor \qquad a \sim \mathcal{N}(0, I)$$
$$b \sim \mathrm{Unif}(0, w)$$

$$\mathbb{P}\{h(x) = h(y)\} = \mathbb{P}\{h(0) = h(y-x)\}$$

Proof: $h(x) - h(y) = 0 \iff \exists n: \ a^Tx + b \in [nw, (n+1)w)$
$\qquad a^Ty + b \in [nw, (n+1)w)$

given $b \in [nw - \min\{a^Tx, a^Ty\}, (n+1)w - \max\{a^Tx, a^Ty\})$

just compute the length of

$$\bigcup_{n=-\infty}^{+\infty} [0, w) \cap [nw - \min\{a^Tx, a^Ty\}, (n+1)w - \max\{a^Tx, a^Ty\})$$

nw-min $\quad$ (n+1)w-max

$[\quad [\quad)[\quad)\quad)$

nw $\quad$ (n+1)w

sps. $\max\{a^Tx, a^Ty\} - \min\{a^Tx, a^Ty\} \geq w$, then the length $= 0$

o/w

it is the same as

$$\bigcup_{n=-\infty}^{\infty} [nw, (n+1)w) \cap [-\min\{a^Tx, a^Ty\}, -\max\{a^Tx, a^Ty\} + w)$$

$w - |a^Tx - a^Ty|$ the length $\quad$ Therefore,

$$\mathbb{P}\{h(x) = h(y) \mid a\} = \mathbb{P}\{h(0) = h(x-y) \mid a\}$$

$$\mathbb{P}\{h(x) = h(0)\} = \mathbb{P}\{h(y) = h(0)\} \text{ iff } \|x\|_2 = \|y\|_2$$

Proof: it is easy to verify $a^Tx \sim \mathcal{N}(0, x^Tx)$ and
$a^Ty \sim \mathcal{N}(0, y^Ty)$

$\max\{0, -\gamma\}, \min\{w-\gamma, w\}$

so the direction does not matter.

$$\mathbb{P}\{h(x) = h(y)\} = \mathbb{P}\left\{\left\lfloor \gamma + \frac{b}{w} \right\rfloor = 0\right\} \qquad \gamma \sim \mathcal{N}\left(0, \frac{\|x-y\|_2^2}{w^2}\right)$$

$\frac{b}{w} \in \mathrm{Unif}(0, 1)$

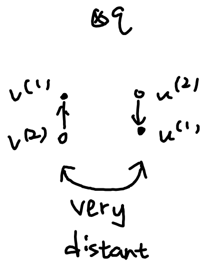
$$= \int_0^1 (1-z)\mathcal{N}(z; 0, r^2)\, dz + \int_{-1}^0 (1+z)\mathcal{N}(z; 0, r^2)\, dz$$
$$= 2\int_0^1 (1-z)\mathcal{N}(z; 0, r^2)\, dz$$

where $r = \frac{\|x-y\|_2}{w}$

so $\mathbb{P}\{h(x) = h(y)\}$ is big if $\|x-y\|_2$ is small

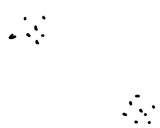
20241026 Optimize from Nearby Values

One problem: how to adjust this?



Fact: k-NN of q might be very distant from other k-NNs.

HNSW also fails on dataset with several clusters.



20241026 Gradient Descend

Give η: $f(x^t - \eta \cdot \nabla f(x^t)) > f(x^t) - \frac{\eta}{2} \|\nabla f(x^t)\|^2$

$\eta > \frac{1}{L}$ $\qquad$ $\eta \leftarrow 2\eta$

$\eta \leftarrow \frac{\eta}{2}$

$$f(x^t - \eta \cdot \nabla f(x^t)) \le f(x^t) - \frac{\eta}{2} \|\nabla f(x^t)\|^2$$

① $\eta > \frac{1}{L}$, then $f(\dots) \le f(x^t) - \frac{1}{4L} \|\nabla f(x^t)\|^2$

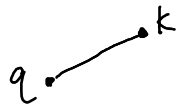
② $\eta \le \frac{1}{L} \wedge \eta > \frac{1}{2L}$, then

$$|f(y) - f(x) - \nabla f(x)^T (y - x)| \le \frac{L}{2} \|x - y\|^2$$

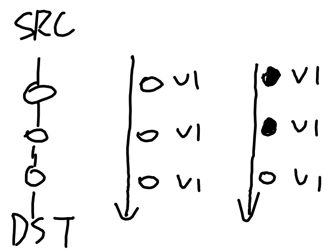
20241030 LSH "AND" Query (DB-LSH)

Observation: if $\|q-k\| \le r$, then for a random projection

$\|w^T q - w^T k\| \le r\|w\|$



20241030



Glauber Dynamics

20241030 Solve Triangle Detection $\leftrightarrow$ BMM

Given a graph $G(V,E)$, A is an adjacency matrix, no self-loops.

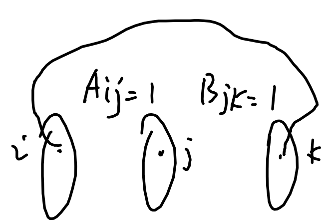
If there exists a triangle Δ, then for some $u,v \in E$

$(A \times A)_{u,v} \neq 0$. $\leftarrow$ This means $\sum_k (A)_{u,k}(A)_{k,v} \neq 0$

$\exists k: (A)_{u,k}(A)_{k,v} \neq 0$

<u>Claim</u>: reduction from BMM to triangle detection is surprising because BMM outputs n^2 bits and triangle detection returns only 1 bit.

BMM: Boolean Matmul (and-or matmul)



$A, B \in \{0,1\}^{n\times n}$

$(A\cdot B)_{ij} \neq 0$ iff $\exists k: (A)_{ik}\cdot(B)_{kj} = 1$

$\forall (i,k) \in S: (AB)_{i,k} = 0$

$\exists (i,k) \in S: (AB)_{i,k} \neq 0$

If we can detect triangles in time $O(n^{3-\varepsilon})$ then given $A, B \in \{0,1\}^{n\times n}$, $S \subseteq \{1,\ldots n\}^2$. We can find one $\{i,j\} \in S$ with $(AB)_{i,j} \neq 0$ in $O(n^{3-\varepsilon}\log n)$ time.

20241030 Solve BMM w/ Triangle Detection

BMM(A , B)

split A and B to dxd matrices

for $i = 1 \dots n/d$

for $j = 1 \dots n/d$

$c^{ij} =$ dxd zero matrix

$S = \{1, \dots, d\}^2$

for $k = 1 \dots n/d$

$T =$ find positions with $(A^{ik} \cdot B^{kj})_{a,b} \neq 0 \quad (a,b) \in S$

$C^{i,j}_{a,b} = 1 \quad \forall (a,b) \in T$

$S = S \setminus T$

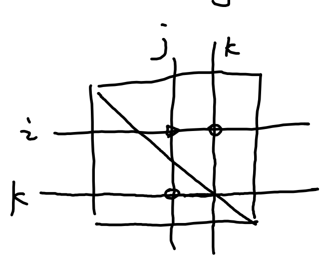
return C

Run time : $\left(\frac{n}{d}\right)^3 d^{3-\varepsilon} + \left(\frac{n}{d}\right)^2 \cdot d^{3-\varepsilon} \cdot \log d \cdot d^2$

$= n^3 d^{-\varepsilon} + n^2 d^{3-\varepsilon} \cdot \log d \qquad$ take $d = n^{\frac{1}{3}}$

$= O(n^{3-\frac{1}{3}\varepsilon} \log n)$

20241101 Triangle Detection & QR (6515 Problem Set 7)



20241103 Linear Algebra

$\nexists S \subseteq \mathbb{I}_n : |S| \geq n+2 \wedge (\forall x, y \in S : x^T y < 0)$

$\mathbb{I}_n$: some n-dimensional inner product space

Lemma : Suppose $x^T z < 0 \wedge y^T z < 0$

Define : x', y', x_z, y_z s.t. $x = x' + \frac{z z^T}{z^T z} x = x' + x_z$

$$y = y' + \frac{z z^T}{z^T z} y = y' + y_z$$

$$x^T y = (x' + x_z)^T (y' + y_z)$$

$$= x' \cdot y' + \cancel{x' \cdot y_z} \cancel{+ y' \cdot x_z} + x_z \cdot y_z$$

Therefore , $x' \cdot y' \leq x \cdot y$

By contradiction :

Suppose : $\exists S : |S| \geq n+2 \wedge (\forall x, y \in S : x \cdot y < 0)$

Let $S = \{x_1, \cdots, x_{n+2}\}$

Use previous lemma , get $S' = \{x_1', \cdots, x_{n+1}'\}$

This reduces the problem to n-1 dimensional inner product space, as clearly x_{n+2} is linearly independent from $x_1', \cdots, x_{n+1}'$.

By induction , $\exists x, y, z \in \mathbb{R} : xy < 0, yz < 0, xz < 0$

However , this is not possible.

20241103 Data Streaming

Association Rule Mining :

Given an index set I, a table $\tau : \{1, \dots, n\} \to 2^I$

Problem : count the frequency of each set

Solution :

How do we say "set" in TT?
I mix annotations but it is bad.

Define $A_i = \{x : \tau(x) \ni i\}$

Use min-hash to estimate A_i, and estimate $\left|\bigcup_{i \in J} A_i\right|$ using sketch of each A_i.

The worst thing of TT is you cannot do union effectively.

You have to define $\{x : X\} \to \mathrm{Prop}_1(x) \to T_1$

and $\{x : X\} \to \mathrm{Prop}_2(x) \to T_2$

then $\{x : X\} \to \mathrm{Prop}_1(x) \sqcup \mathrm{Prop}_2(x) \to T_1$ union T_2

We also cannot do cardinality ... btw.

Sketching L^p norm :

$$L^p(x) = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

20241107 Elephant Detection

① w/ count-min sketch :

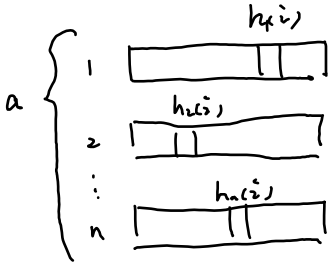
get count-min sketch of item i (f_i, say, i is Nth item)

if $f_i \geq \theta N$ then

compare f_i to heap top (there is a min-heap)

remove heap top if it is smaller than θN

② Charikar-Chen Sketch (Tug-of-War)



Given item $i \in \mathcal{I}$

$h_\cdot : \mathcal{I} \to \{1, \dots, m\}$

$g_\cdot : \mathcal{I} \to \{+, -\}$

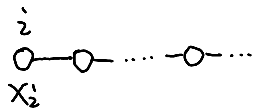
$g_n(i) = +$, then $a[h_n(i)]$ += 1

$g_n(i) = -$, then $a[h_n(i)]$ -= 1

Retrieve median of $\{a[h_\cdot(i)] \cdot g_\cdot(i)\}$

20241107 Uniformity in RECIPE

Goal: reduce uniformity to weaker condition on any switch

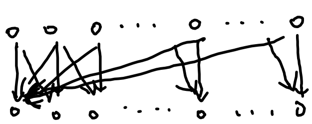


$\{ \{\dots\}, \{\dots\} \} \xrightarrow{\text{replace}} \{\{n+1\}\}$ $\quad \mathbb{P}\{X_{n+1} = \{n+1\}\}$

$\xrightarrow{\text{skip}} \{ \{\dots\}, \{\dots\}, \}$ $\quad \mathbb{P}\{X_{n+1} = X_n \mid |X_{n+1}| = d\}$

$\xrightarrow{+} \{ \{\dots, n+1\}, \{\dots, n+1\}, \}$ $\quad \mathbb{P}\{X_{n+1} = X_n \cup \{n+1\} \mid |X_{n+1}| = d\}$

Decision is only based on current degree d.



20241108 Exam

Topic: Simplex Algorithm

Gradient Descent

Fine Grained Complexity

20241108 Decidability

Halting Problem:

Input: algrithm A and input I

Output: $A(I)$ is going to stop

Theorem: There is no algorithm for halting problem.

Proof: If there exists algorithm $\mathcal{H}(A, I)$ for this,

$\mathcal{H}'(A)$:

$h = \mathcal{H}(A, A)$

if h then loop forever

else return

$\mathcal{H}'(\mathcal{H}')$ halts, then

$h = \mathcal{H}(\mathcal{H}', \mathcal{H}')$ is true,

but then $\mathcal{H}'$ will loop forever,

If $\mathcal{H}'(\mathcal{H}')$ does not halt, then

$h = \mathcal{H}(\mathcal{H}', \mathcal{H}')$ is false.

but then $\mathcal{H}'$ halts.

Therefore, $\mathcal{H}$ can only return incorrect value.

Testing Problem:

Input: property P and algorithm A

Output: A satisfies P

Rice Theorem: $\forall P$: P is not trival $\Rightarrow \nexists$ algorithm for testing P

Let A be an algorithm that has property P.

$\mathcal{H}(\mathcal{M}, I)$

$B = A$ iff $\mathcal{M}(I)$ halts

```
def B(x)
    run M(I)
    return A(x)
return Test(B)
```