Cpts 515. 12/4/2020

Knapsack:

Given: $n$ Boxes, each with

$w_i, v_i > 0$. and $W > 0$.

($w_i$: weight, $v_i$: value.)

Goal: to find a subset $I$ s.t.

$\sum_{i \in I} v_i$ is maxmd with

$\sum_{i \in I} w_i \leq W$

dominate relation: $S_1 \geq S_2$

$S_1$ has less weight but more value.

Let List = $S, S', S'', \ldots$ be a list of subsets of boxes. Then

List + {$Box_k$} to denote the following.

$S + \{Box_k\}, S' + \{Box_k\}, S'' + \{Box_k\}, \ldots$

ALG: (Dynamic Programing).

For each $1 \le k \le n$

if $k = 1$

$List_1 := S_\emptyset, \{Box_1\}$.

if $k > 1$,

$List_k := List_{k-1}$ followed by

$List_{k-1} + \{Box_k\}$,

eliminate all "dominated" subsets in $List_k$;

eliminate all subsets with weight $> W$;

Return the max value subset in $List_n$.

(This is almost Brute-force).

Running time: ~~$O(n \cdot V, W)$~~ $O(n \cdot \min(V, W))$

This time is exponential (size of input ≠ value of input).

you have $n$ rounds

each round, we create $List_k$ which has at most $\min(W, V)$ number of subsets (after eliminating process)

where $V = \sum_{i=1}^{n} v_i$

Why? after eliminating process, each pair of subsets of Boxes, ~~by card~~ to $S_1$ and $S_2$, in the list $List_k$, they can't have the same total value; can't have the same total weight.

Then, I run the previous dynamic prograing alg on $v_i'$ will result in a sdutn $S$ which is a subset of boxes with total ~~weight~~ value

$$\sum_{i \in S} v_i' \xrightarrow{\text{approx back}} \sum_{i \in S} v_i$$

opt sdutn $S^*$:

$$\sum_{i \in S^*} v_i' \xrightarrow{\text{approx back}} \sum_{i \in S^*} v_i$$

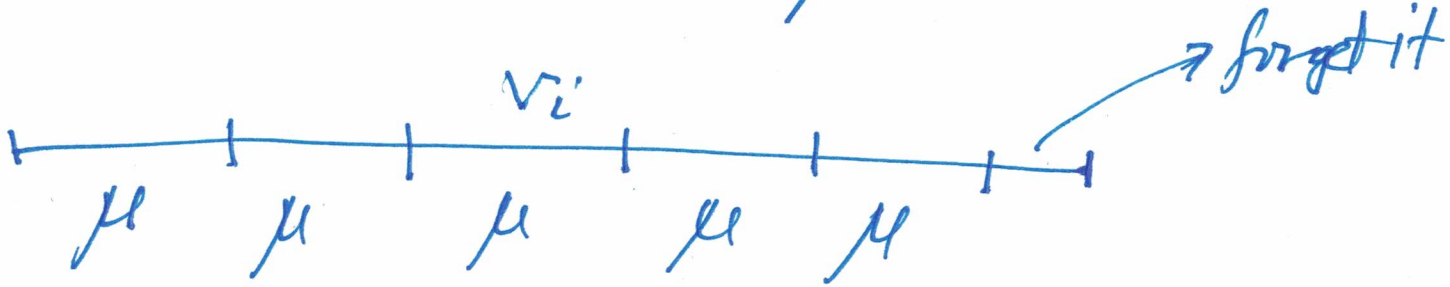
} what is the error value?

Observation: $\sum_{i \in S} v_i' = \sum_{i \in S^*} v_i'$. (Recall $S$ is optimal wrt the approx. values $v_i'$).

"Rounding".

The exp-time in last page happens when W, V are much larger than n. In this case, we can take approximation on values:

each $v_i$ $\xrightarrow{\text{approx}}$ $\left\lfloor \frac{v_i}{\mu} \right\rfloor = v_i'$




$v_i' = 5$

$$\left| \sum_{i \in S} v_i - \sum_{i \in S^*} v_i \right| \leq n \cdot \mu$$

ERROR

each box creates at most $\mu$ error.

We need only to set $n\mu = \varepsilon \cdot \sum_{i=1}^{n} v_i = \varepsilon \cdot V$

(This is the way you get $\mu$ from the given $\varepsilon$).

text error corrected.

(not $\cdot \max_i v_i$).

Then, $\text{Error} \leq \varepsilon \cdot V$ or

$$\frac{\text{Error}}{V} \leq \varepsilon .$$

Running time is (using $v_i'$):

$$O(n \cdot \min(W, V')) \leq O(n \cdot V')$$

where

$$V' = \sum_{i=1}^{n} v_i' = \sum_{i=1}^{n} \left\lfloor \frac{v_i}{\varepsilon \cdot V/n} \right\rfloor$$

textbook error corrected.

$$\leq \sum_{i=1}^{n} \frac{v_i}{\varepsilon \cdot V/n} = \frac{n}{\varepsilon} .$$

So, Running time is $O(n \cdot \frac{n}{\varepsilon})$ which is polynomial on $n$ and $\frac{1}{\varepsilon}$. This is called full poly-approx.

$\lambda$ = an exp over $(b_1, \ldots, b_k)$

cases: certain symmetric choices:

① . linear combination of $b_i$'s . ✗

② . $\sum_i 2^i \cdot b_i$ (unsigned number coding of k-bits)

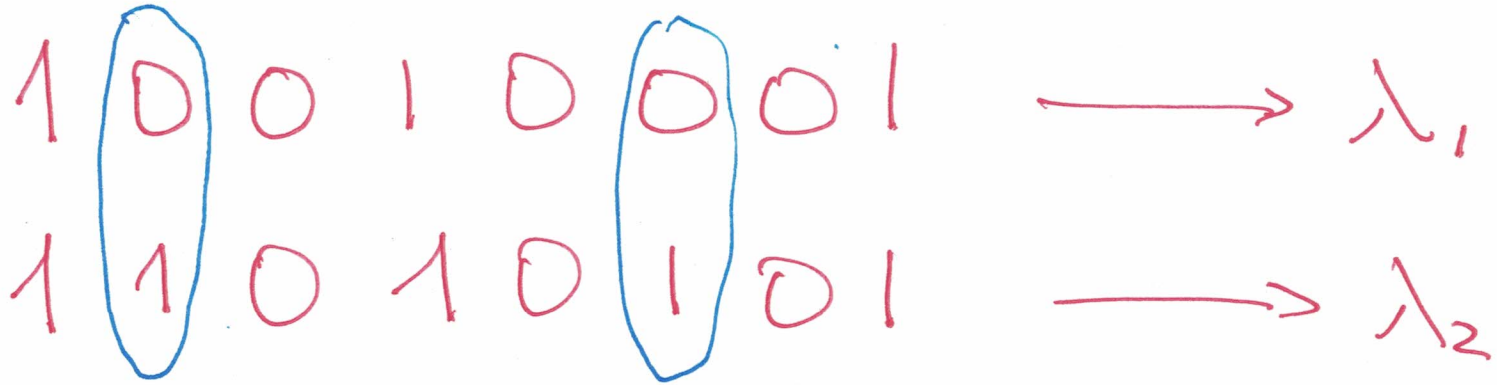
→ 1-1.

bad: can't maintain Haar!

↙

②′: lower sensitivity on $2^k$ term.
larger sensitivity on $2^0$ term.

exade of two bit-array:




Hamig distance is symetric? invarant on pentahi!

1-1 is asymetric? Sensitive to pentahi.

② has new problem: lose 1-1.

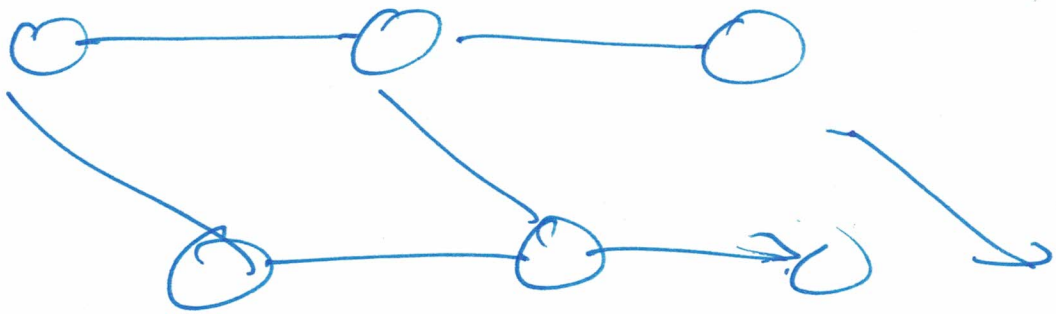
Fix: don't use 2-base.

use transcendental number $e$.

---

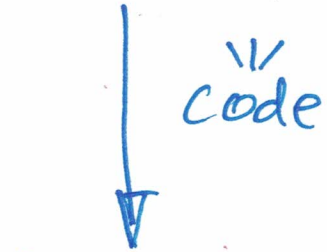
$$\lambda = \frac{1}{k} \ln\left(e^{b_1} + e^{b_1+b_2} + \dots + e^{b_1+\dots b_k}\right).$$

↳ free energy of molecule moving along.



+w. prob.

$b_1, b_2, \ldots\ldots b_k,$ k-bits.




$\lambda$, a real number

s.t. code is 1-1

code is locality sensitive.

(small Hamming dist $\Rightarrow$ small code value difference).