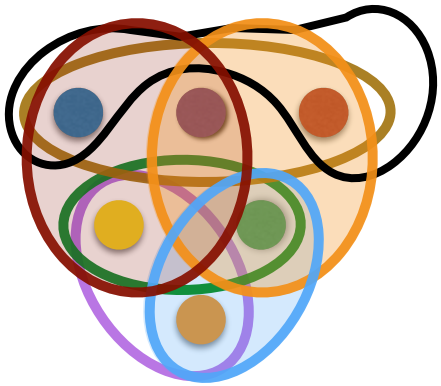
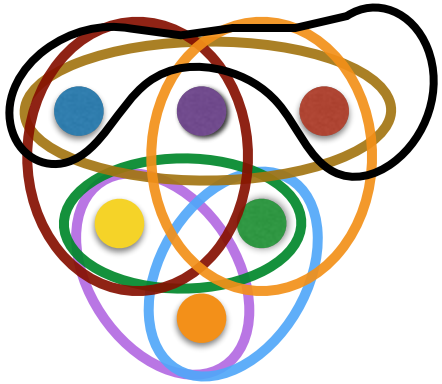
Set cover, linear programming and randomized rounding

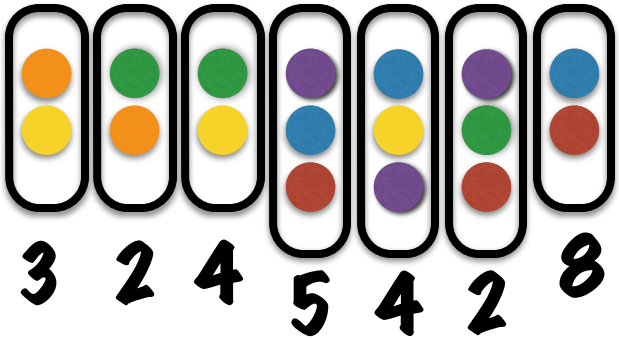


The set cover problem

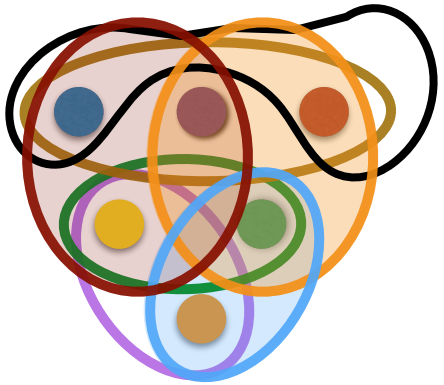


Elements

Subsets with costs



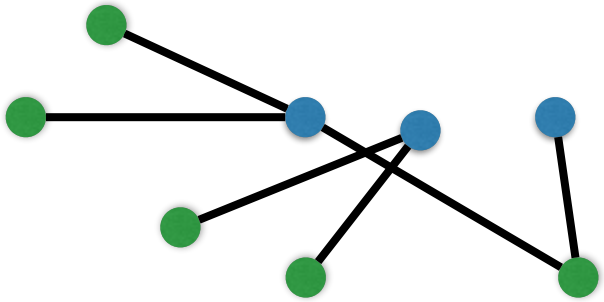
Choose subsets
Cover elements
at min cost



Cost=4+2+2=8

Does this ring a bell?

Vertex cover



Cover all edges with the fewest vertices

edges = elements

vertices = sets

Integer program for Set cover

Variable for subset S:

$$x_S = 1$$

iff S in cover

Constraint for element i:

$$\sum_{S:e \in S} x_S \geq 1$$

Objective:

$$\min \sum_S c_S x_S$$

Linear programming relaxation

$$\min \sum_{\mathbf{S}} \mathbf{c_S x_S}$$

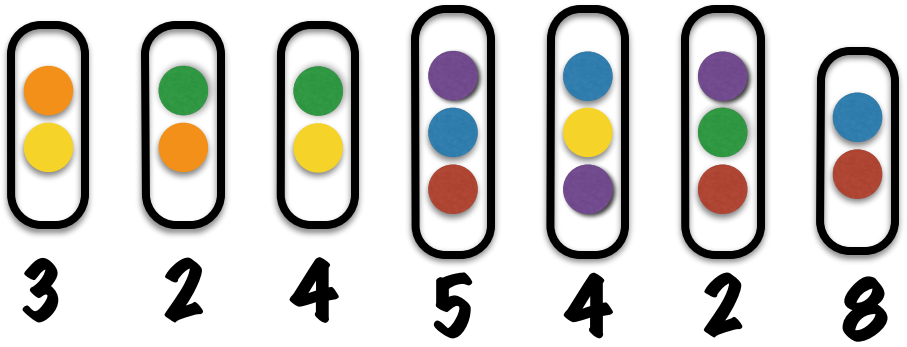
$$\textbf{such that}$$

$$\begin{cases} \sum_{S:e\in S} x_S \geq 1 & \forall e \\ 0 \leq x_S \leq 1 & \forall S \end{cases}$$

Rounding

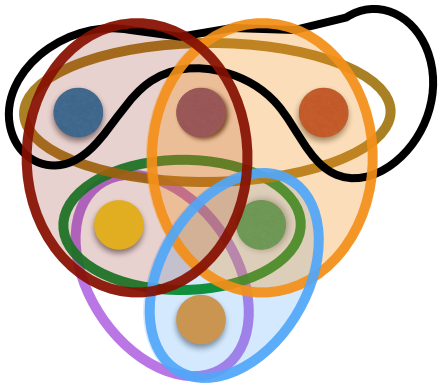
Like vertex cover:
round to 1 iff $x_u \geq 1/2$

Fails

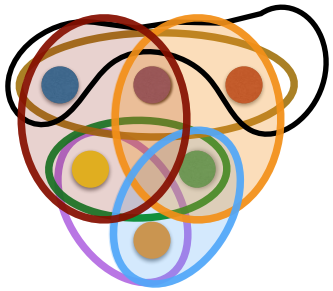


$$x_S : \frac{1}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{2}, \frac{1}{3}$$

Set cover, linear programming and randomized rounding



Set cover, linear programming and randomized rounding



Linear programming relaxation

$$\min \sum_S \mathbf{c_S x_S}$$

$$\textbf{such that}$$

$$\begin{cases} \sum_{S:e \in S} x_S \geq 1 & \forall e \\ 0 \leq x_S \leq 1 & \forall S \end{cases}$$

How do we round the LP solution?

Randomized Rounding: An algorithm

$x_i = .9$: should probably go to 1

$x_i = .1$: should probably go to 0

Cannot fix a threshold

Idea: randomized rounding

$x_i = .8 \implies$ round to 1 w.p. 80%

New rounding algorithm

For each set S
with probability x_S
put S in the cover

Analysis

Is it efficient?
Is the output a cover?
How good is it?

Analysis

Is it efficient?

Yes

Analysis

Is the output a cover?

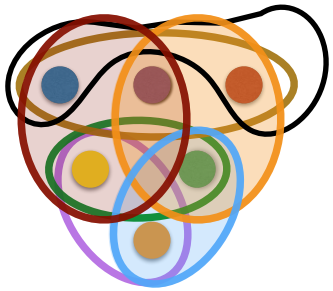
Maybe, maybe not

Analysis

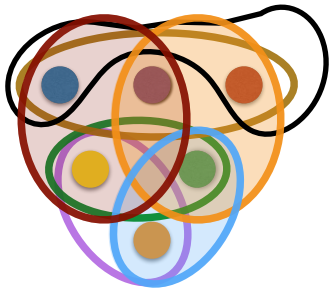
How good is it?

It depends...

Set cover, linear programming and randomized rounding



Set cover, linear programming and randomized rounding



How good is it?

It depends...

$$\mathbf{Value(Output)} = \sum_{S \textbf{ in cover}} c_S$$

How good is it on average?

$$\mathbf{E}[\sum_{\mathbf{S}} \mathbf{1(S \text{ in cover})c_S}] = ?$$

Linearity of expectation

$$\mathbf{E}[\mathbf{A} + \mathbf{B}] = \mathbf{E}[\mathbf{A}] + \mathbf{E}[\mathbf{B}]$$

$$\mathbf{E}[\textstyle\sum_{\mathbf{S}} \mathbf{1}(\mathbf{S\ in\ cover})\mathbf{c_S}] =$$
$$\textstyle\sum_{\mathbf{S}} \mathbf{E}[\mathbf{1}(\mathbf{S\ in\ cover})\mathbf{c_S}]$$

$$\mathbf{E}[\lambda \mathbf{X}] = \lambda \mathbf{E}[\mathbf{X}]$$

$$\sum_{\mathbf{S}} \mathbf{E}[\mathbf{1}(\mathbf{S} \textbf{ in cover}) \mathbf{c_S}] = \sum_{\mathbf{S}} \mathbf{E}[\mathbf{1}(\mathbf{S} \textbf{ in cover})] \mathbf{c_S}$$

$$\mathbf{E}[\mathbf{1}(\mathbf{S\ in\ cover})] = \Pr[\mathbf{S\ in\ cover})]$$

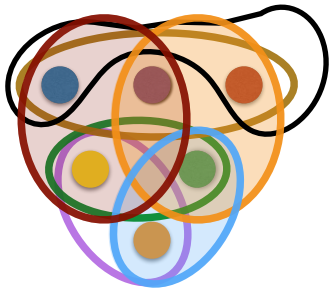
$$\Pr[\mathbf{S \text{ in cover}})] = \mathbf{x_S}$$

Together

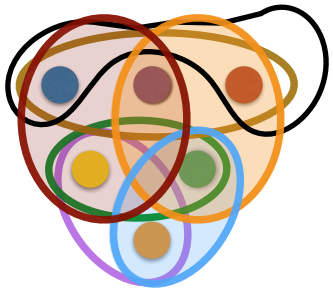
$$\mathbf{E[Value(Output)] = \sum_S x_S c_S}$$

Value of the linear program!

Set cover, linear programming and randomized rounding



Set cover, linear programming and randomized rounding



Is the output a cover?

Maybe, maybe not

Number of elements covered:

$$\sum_{\mathrm{e}} 1(\mathrm{e\ covered})$$

On average:

$$\sum_{\mathrm{e}} \Pr[\mathrm{e\ covered}]$$

Consider an element e.
With what probability
is it covered by output?

$$\Pr[\textbf{e covered}] =$$
$$\Pr[\textbf{there exists } S \textbf{ in output:}$$
$$\textbf{e} \in \textbf{S}]$$

$$\Pr[\textbf{there exists } S \textbf{ in output: } \mathbf{e \in S}] =$$
$$\mathbf{1} - \Pr[\textbf{for all } S \textbf{ containing } e\textbf{: S not in output}]$$

Independence

If independence:

$$\Pr[\textbf{A and B}] = \Pr[\textbf{A}] \times \Pr[\textbf{B}]$$

$$\Pr[\textbf{for all } S \textbf{ containing } e\textbf{:} \\ \textbf{S not in output}] = \\ \prod_{\textbf{S}:\textbf{e}\in\textbf{S}} \Pr[\textbf{S not in output}]$$

$$\Pr[\mathbf{S \text{ not in output}}] = \mathbf{1 - x_S}$$

Together

$$\Pr[\mathbf{e} \textbf{ covered}] = 1 - \prod_{\mathbf{S}:\mathbf{e}\in\mathbf{S}}(1 - \mathbf{x_S})$$

Algebra

$$\mathbf{X} = \mathbf{e}^{\ln \mathbf{X}}$$

$$\ln(\mathbf{XY}) = \ln \mathbf{X} + \ln \mathbf{Y}$$

$$\prod_{\mathbf{S}:\mathbf{e}\in\mathbf{S}}(\mathbf{1} - \mathbf{x_S}) = \mathbf{e}^{\sum_{\mathbf{S}:\mathbf{e}\in\mathbf{S}} \ln(\mathbf{1}-\mathbf{x_S})}$$

Algebra

$$\ln(\mathbf{1} - \mathbf{X}) \leq -\mathbf{X}$$

$$\mathbf{e}^{\sum_{\mathbf{S}:\mathbf{e}\in\mathbf{S}} \ln(\mathbf{1}-\mathbf{x_S})} \leq \mathbf{e}^{-\sum_{\mathbf{S}:\mathbf{e}\in\mathbf{S}} \mathbf{x_S}}$$

Use LP constraint

$$\sum_{S:e \in S} x_S \geq 1$$

$$e^{-\sum_{S:e \in S} x_S} \leq e^{-1}$$

Combining

$$\Pr[\text{e covered}] \geq 1 - 1/e$$

Average number of elements covered:

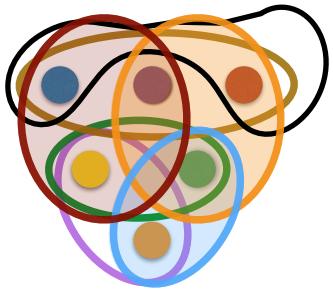
$$\#(\text{elements})(1 - 1/e)$$

$$1 - 1/e = 0.63...$$

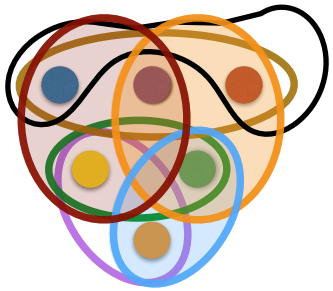
Recap

Randomized rounding gives
collection of sets
with average cost
at most OPT
and covering on average
63% of the elements.

Set cover, linear programming and randomized rounding



Set cover, linear programming and randomized rounding



Getting a set cover

Idea: repeat!

Randomized rounding algorithm

n=#elements
Repeat ln(n)+3 times
  For each S
    Put S in cover w.pr. x(S)
    (if not there already)

Note: $e^3 = 20.0\ldots$

Cost

In expectation:
at most
(ln(n) +3)OPT

Correctness

$$\mathbf{Pr[cover]} = \mathbf{1} - \mathbf{Pr[not\ cover]}$$

$$\mathbf{Pr[not\ cover]} =$$
$$\Pr[\exists\ \mathbf{element\ not\ covered}] \leq$$
$$\sum_{\mathbf{e}} \Pr[\mathbf{e\ not\ covered}]$$

For one element e and for one iteration

$$\mathbf{Pr}[\mathrm{e\ not\ covered}] < \mathbf{1/e}$$

For one element e and for all iterations together

$$\mathbf{Pr}[\mathbf{e\ not\ covered}] < (\mathbf{1/e})^{\ln(\mathbf{n})+\mathbf{3}} = \frac{\mathbf{1}}{\mathbf{e^3 n}}$$

$$\sum_{\mathbf{e}} \mathbf{Pr[e\ not\ covered]} < \mathbf{n}\frac{\mathbf{1}}{\mathbf{e^3 n}} = \frac{\mathbf{1}}{\mathbf{e^3}} < \mathbf{0.05}$$

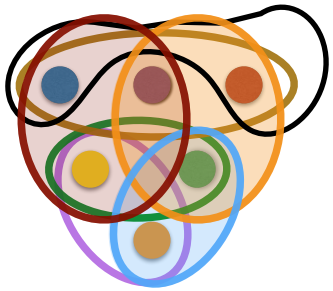
So:

$$\mathbf{Pr[cover]} > \mathbf{0.95}$$

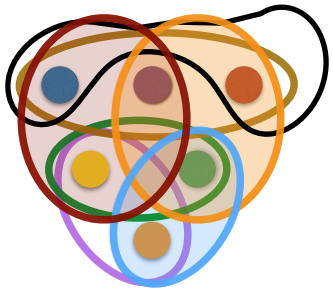
Result

Iterated randomized rounding gives
collection of sets
that is a set cover with
probability 95% and
with average cost
at most (ln(n)+3) OPT.

Set cover, linear programming and randomized rounding



Set cover, linear programming and randomized rounding



Result so far

Iterated randomized rounding gives collection of sets that is a set cover with probability 95% and with average cost at most (ln(n)+3) OPT.

Not guaranteed!

Not guaranteed!

Q: What if you want the output to always be a set cover?

Desired result:
algorithm that gives collection of sets that is a set cover and with average cost at most O(ln(n)) OPT.

Obvious solution:

Replace
"repeat ln(n)+3 times"
by
"repeat until you have a set cover."

Equivalent algorithms

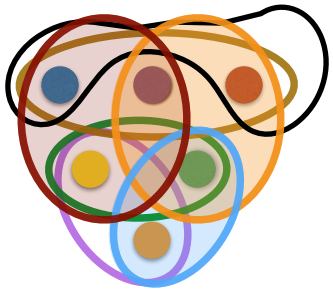
Repeat
 For each S
  Put S in cover w.pr. x(S)
  (if not there yet)
Until you have a set cover

Repeat
 Choose S w.pr. $x_S / \sum_{S'} x(S')$
 Put S in cover
 (if not there yet)
Until you have a set cover

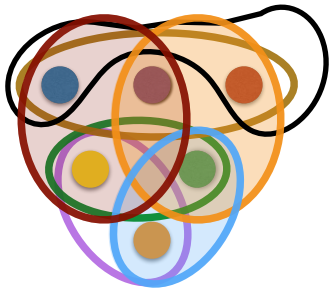
Sample-and-iterate algorithm

Repeat
  Choose S w.pr. $x_S / \sum_{S'} x(S')$
  Put S in cover
  (if not there yet)
Until you have a set cover

Set cover, linear programming and randomized rounding



Set cover, linear programming and randomized rounding



Sample-and-iterate algorithm

Repeat
  Choose S w.pr. $x_S / \sum_{S'} x(S')$
  Put S in cover
  (if not there yet)
Until you have a set cover

Repeat
  Choose S w.pr. $x_S / \sum_{S' x(S')}$
  Put S in cover
  (if not there yet)
Until you have a set cover

Analysis

1. Expected cost of output

T = #iterations (stopping time)
C_t = cost of set chosen at iteration t

Cost of output: $\sum_{t=1}^{T} C_t$

Repeat

Choose S w.pr. $x_S / \sum_{S' x(S')}$

Put S in cover

(if not there yet)

Until you have a set cover

Cost of output: $\sum_{t=1}^{T} C_t$

Expected cost of output: $E[\sum_{t=1}^{T} C_t]$

NB: cannot exchange E[] and summation here!

Wald's equation

T stopping time

X_t random variable

If X_t bounded from above:

$$\mathbf{E}[\mathbf{X_t}] \leq \mu$$

then $\mathbf{E}[\sum_{\mathbf{t} \leq \mathbf{T}} \mathbf{X_t}] \leq \mu \mathbf{E}[\mathbf{T}]$

NB: can now exchange E[] and summation!

Expected cost of set chosen at iteration t:

$$\mathbf{E}[\mathbf{C_t}] = \sum_{\mathbf{S}} \Pr[\mathbf{S \text{ chosen}}]\mathbf{c_S} = \frac{\sum_{\mathbf{S}} \mathbf{c_S x_S}}{\sum_{\mathbf{S}} \mathbf{x_S}} = \mu$$

Expected cost of output: $\mathbf{E}[\mathbf{T}]\mu = \mathbf{E}[\mathbf{T}] \frac{\sum_{\mathbf{S}} \mathbf{c_S x_S}}{\sum_{\mathbf{S}} \mathbf{x_S}}$

Next problem: What is E[T]?

Repeat
  Choose S w.pr. $x_S / \sum_{S' x(S')}$
  Put S in cover
  (if not there yet)
Until you have a set cover

More notations

2. Expected number of iterations

n_t = #elts not yet covered after t iterations

$$n_0 = n, n_T = 0, n_{T-1} \geq 1$$

Wald's equation for dependent decrements

Consider a random decreasing sequence

$$n_0, n_1, \ldots, n_T,$$

where T is a stopping time. If

$$E[n_t - n_{t+1} | n_t] \geq f(n_t),$$

where f is an increasing function, then

$$E[T] \leq 1 + E[\int_{n_{T-1}}^{n_0} \frac{1}{f(z)dz}].$$

To bound E[T], analyze
#elts not yet covered after t iterations

Analyze n_t

Analyze $E[n_t - n_{t+1} | n_t]$ **: elts covered in next iteration**

Fix an element e **among the** n_t

$$\Pr[\text{e covered in next iteration}] =$$
$$\Pr[\text{S chosen contains e}] =$$
$$\sum_{S:e \in S} x_S / \sum_{S'} x_{S'} \geq$$
$$1 / \sum_{S'} x_{S'}$$

Sum over e

$$E[n_t - n_{t+1} | n_t] \geq f(n_t)$$

$$\mathbf{E}[\mathbf{n_t} - \mathbf{n_{t+1}} | \mathbf{n_t}] \geq \mathbf{n_t} / \sum_{\mathbf{S}'} \mathbf{x_{S'}}$$

Wald's equation for dependent decrements

Consider a random decreasing sequence

$$\mathbf{n_0}, \mathbf{n_1}, \ldots, \mathbf{n_T},$$

where T is a stopping time. If

$$\mathbf{E}[\mathbf{n_t} - \mathbf{n_{t+1}} | \mathbf{n_t}] \geq \mathbf{f}(\mathbf{n_t}),$$

where f is an increasing function, then

$$\mathbf{E}[\mathbf{T}] \leq 1 + \mathbf{E}[\int_{\mathbf{n_{T-1}}}^{\mathbf{n_0}} \frac{\mathbf{1}}{\mathbf{f}(\mathbf{z})\mathbf{dz}}].$$

Application:

Stopping time $\mathbf{E}[\mathbf{T}] \leq 1 + \int_1^{\mathbf{n}} \frac{\sum_{\mathbf{S}} \mathbf{x_S}}{\mathbf{z}} \mathbf{dz} = 1 + \sum_{\mathbf{S}} \mathbf{x_S} \ln(\mathbf{n})$

Together

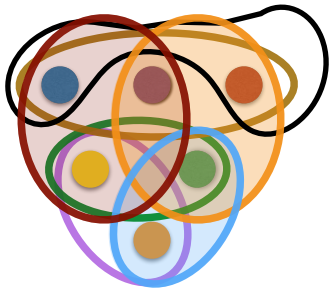
Expected cost of output:

$$(1 + (\sum_S x_S) \ln(n)) \frac{\sum_S c_S x_S}{\sum_S x_S} \leq (1 + \ln(n))\ OPT$$

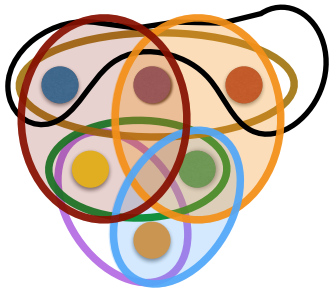
Result:

the algorithm gives
a collection of sets that is a set cover
with average cost at most (1+ln(n)) OPT.

Set cover, linear programming and randomized rounding



Set cover, linear programming and randomized rounding



Result

The sample-and-iterate algorithm gives a collection of sets that is a set cover with average cost at most (1+ln(n)) OPT.

A more efficient algorithm

Linear programming takes
polynomial time
but
is often slower than
combinatorial algorithms

Greedy

Repeat
  Choose S maximizing $\#(\text{new elts covered})/c_S$
   Put S in cover
Until you have a set cover

Result:

Greedy also gives
a collection of sets
that is a set cover
and with
cost at most (1+ln(n)) OPT.

Can we do better?

No:
It is NP-hard to obtain (in polynomial time) a better-than-ln(n) approximation for set cover

Uri Feige

Vašek Chvátal

László Lovász

David Johnson

Neal Young

What have we learned?

- Famous problem: set cover
- Concept: Randomization
- Algorithmic technique: Randomized rounding
- Analysis tool: Linearity of expectation
- Laying out an analysis: slow & steady, orderly - like hiking up a mountain

Set cover, linear programming and randomized rounding