Fictitious Play beats Simplex for fractional packing and covering

Christos Koufogiannakis and Neal E. Young
University of California, Riverside

June 28, 2007

fractional packing and covering

Linear programming with non-negative coefficents.

Equivalent to solving a zero-sum matrix game A with non-negative coefficients:

Theorem (von Neumann's Min-Max Theorem 1928)

$$\min_x \; \max_i A_i x \;=\; \max_{\hat{x}} \; \min_j A_j^{\mathsf{T}} \hat{x}$$

x: mixed strategy for min (column) player

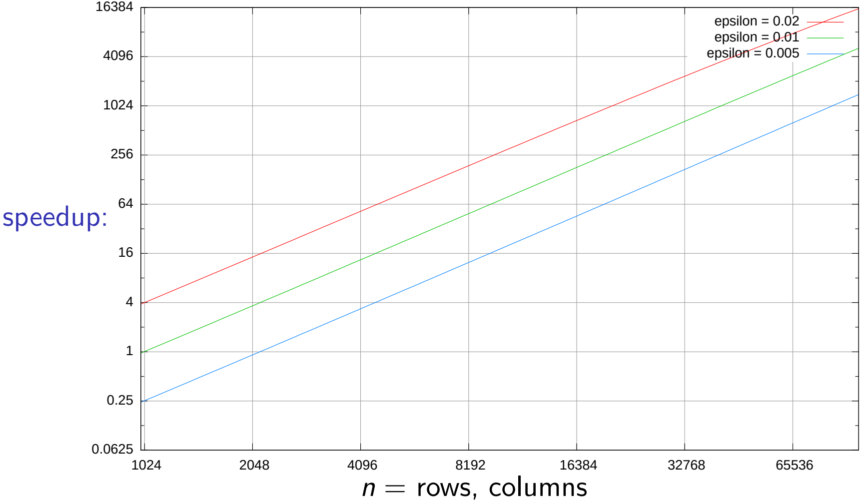
$\hat{x}$: mixed strategy for max (row) player

i: row, j: column

- How to compute $(1 \pm \varepsilon)$-optimal x and $\hat{x}$ quickly?
- Simplex algorithm: $\Omega(n^3)$ time for dense $n \times n$ matrix.

This talk: $O(n^2 + n\log(n)/\varepsilon^2)$ time.

practical performance versus simplex



playing a zero-sum game

- x = mixed strategy for min
- $A_i x$ = payoff if max plays row i against mixed strategy x

			min			
	x :	.5	0	.5	Ax	
		1	0	0	.5	
max :	A :	1	1	0	.5	
		0	1	1	.5	$\leftarrow$ max gets ≤ 5

Min plays $x = (.5, 0, .5)$, max gets at most .5 $\Rightarrow$ game val $\leq .5$.

playing a zero-sum game

- $x =$ mixed strategy for min
- $A_i x =$ payoff if max plays row i against mixed strategy x
- $\hat{x} =$ mixed strategy for max
- $A_j^{\mathsf{T}}\hat{x} =$ payoff if min plays column j against mixed strategy $\hat{x}$

min

$$
\begin{array}{ccccc}
 & \hat{x} & & & \\
 & .2 & 1 & 0 & 0 \\
\text{max}: & .4 & A: \; 1 & 1 & 0 \\
 & .4 & 0 & 1 & 1 \\
 & & A^{\mathsf{T}}\hat{x}: \; .6 & .8 & .4 \\
 & & & & \uparrow
\end{array}
$$

Max plays $\hat{x} = (.2, .4, .4)$, min pays at least $.4 \Rightarrow$ game val $\geq .4$.

playing a zero-sum game

- x = mixed strategy for min
- $A_i x$ = payoff if max plays row i against mixed strategy x
- $\hat{x}$ = mixed strategy for max
- $A_j^\mathsf{T} \hat{x}$ = payoff if min plays column j against mixed strategy $\hat{x}$

			min		
	$\hat{x}$	x : .5	0	.5	Ax
	.2	1	0	0	.5
max :	.4	A : 1	1	0	.5
	.4	0	1	1	.5
		$A^\mathsf{T}\hat{x}$: .6	.8	.4	

Min plays $x = (.5, 0, .5)$, max gets at most .5 $\Rightarrow$ game val $\leq .5$.

Max plays $\hat{x} = (.2, .4, .4)$, min pays at least .4 $\Rightarrow$ game val $\geq .4$.

mixed strategies via fictitious play (Brown, Robinson 1951)

Repeated play. In each round each player plays single pure strategy, chosen by considering only opponent's past plays.

- x_j = #times column j played so far.
- $\hat{x}_i$ = #times row i played so far. ... *note* $|x| = |\hat{x}| \neq 1$

e.g. in 21'st round...

		min	
	x : 8	1	11
	1	0	0
max :	1	1	0
	0	1	1

Robinson's update rule ($x/|x|$, $\hat{x}/|\hat{x}|$ converge to optimal):

- Max plays best row against x.
- Min plays best col against $\hat{x}$.

mixed strategies via fictitious play (Brown, Robinson 1951)

Repeated play. In each round each player plays single pure strategy, chosen by considering only opponent's past plays.

- x_j = #times column j played so far.
- $\hat{x}_i$ = #times row i played so far. ... *note* $|x| = |\hat{x}| \neq 1$

e.g. in 21'st round...

		min		
	x : 8	1	11	Ax
	1	0	0	8
max :	1	1	0	9
$\rightarrow$	0	1	1	12 $\leftarrow$ max plays best row against x

Robinson's update rule ($x/|x|$, $\hat{x}/|\hat{x}|$ converge to optimal):

- Max plays best row against x.
- Min plays best col against $\hat{x}$.

mixed strategies via fictitious play (Brown, Robinson 1951)

Repeated play. In each round each player plays single pure strategy, chosen by considering only opponent's past plays.

- x_j = #times column j played so far.
- $\hat{x}_i$ = #times row i played so far. ... *note* $|x| = |\hat{x}| \neq 1$

e.g. in 21'st round...

	$\hat{x}$		min	↓
	1	1	0	0
max :	10	1	1	0
	9	0	1	1
		$A^\mathsf{T}\hat{x}$: 11	19	9

↑ min plays best col against $\hat{x}$

Robinson's update rule ($x/|x|$, $\hat{x}/|\hat{x}|$ converge to optimal):

- Max plays best row against x.
- Min plays best col against $\hat{x}$.

mixed strategies via fictitious play (Brown, Robinson 1951)

Repeated play. In each round each player plays single pure strategy, chosen by considering only opponent's past plays.

- x_j = #times column j played so far.
- $\hat{x}_i$ = #times row i played so far. ... *note* $|x| = |\hat{x}| \neq 1$

e.g. in 21'st round...

	$\hat{x}$	x : 8	1 (min)	11 ($\downarrow$)	Ax	
	1	1	0	0	8	
max :	10	1	1	0	9	
$\rightarrow$	9	0	1	1	12	$\leftarrow$ max plays best row against x
	$A^{\mathsf{T}}\hat{x}$:	11	19	9		

$\uparrow$ min plays best col against $\hat{x}$

Robinson's update rule ($x/|x|$, $\hat{x}/|\hat{x}|$ converge to optimal):

- Max plays best row against x.
- Min plays best col against $\hat{x}$.

algorithm = smoothed fictitious play

random play from exp. distribution (a la Grigoriadis/Khachiyan 1995, expert advice)

e.g. in round 201: $\varepsilon = .1$

	min				
	x : 80	10	110	Ax	p
	1	0	0	80	e^8
max :	1	1	0	90	e^9
	0	1	1	120	e^{12}

- max plays random row i from distribution $p/|p|$
  where $p_i = \exp(\ \varepsilon A_i x)$ – *concentrated on best columns against* x

algorithm = smoothed fictitious play

random play from exp. distribution (a la Grigoriadis/Khachiyan 1995, expert advice)

e.g. in round 201: min $\varepsilon = .1$

	$\hat{x}$			
	10	1	0	0
max :	100	1	1	0
	90	0	1	1
		$A^{\mathsf{T}}\hat{x}$: 110	190	90
		$\hat{p}$: e^{-11}	e^{-19}	e^{-9}

- min plays random column j from distribution $\hat{p}/|\hat{p}|$ where $\hat{p}_j = \exp(-\varepsilon A_j^{\mathsf{T}}\hat{x})$ – *concentrated on best rows against* $\hat{x}$

algorithm = smoothed fictitious play

random play from exp. distribution (a la Grigoriadis/Khachiyan 1995, expert advice)

e.g. in round 201: $\varepsilon = .1$

		min			
$\hat{x}$	x : 80	10	110	Ax	p
10	1	0	0	80	e^8
max : 100	1	1	0	90	e^9
90	0	1	1	120	e^{12}
	$A^{\mathsf{T}}\hat{x}$: 110	190	90		
	$\hat{p}$: e^{-11}	e^{-19}	e^{-9}		

- max plays random row i from distribution $p/|p|$
  where $p_i = \exp(\ \varepsilon A_i x)$ – *concentrated on best columns against* x
- min plays random column j from distribution $\hat{p}/|\hat{p}|$
  where $\hat{p}_j = \exp(-\varepsilon A_j^{\mathsf{T}} \hat{x})$ – *concentrated on best rows against* $\hat{x}$

STOP when $\max_i A_i x \approx \ln(n)/\varepsilon^2$ or $\min_j A_j^{\mathsf{T}} \hat{x} \approx \ln(n)/\varepsilon^2$.

correctness

With high probability, mixed strategies $x/|x|$ for min and $\hat{x}/|\hat{x}|$ for max are $(1 \pm O(\varepsilon))$-optimal.

Proof.

Recall $p_i = \exp(\varepsilon A_i x)$, $\hat{p}_j = \exp(-\varepsilon A_j^{\mathsf{T}} \hat{x})$, min plays from $\hat{p}$, max from p.

By algebra: $$\frac{|p'| \times |\hat{p}'|}{|p| \times |\hat{p}|} \approx 1 + \varepsilon \frac{p^{\mathsf{T}}}{|p|} A \Delta x - \varepsilon \frac{\hat{p}^{\mathsf{T}}}{|\hat{p}|} A^{\mathsf{T}} \Delta \hat{x}.$$

By update rule, $\mathrm{E}[\Delta x] = \frac{\hat{p}}{|\hat{p}|}$ and $\mathrm{E}[\Delta \hat{x}] = \frac{p}{|p|}$

$\Rightarrow$ expectation of r.h.s. equals 1 (i.e., $|p| \times |\hat{p}|$ non-increasing)

$\Rightarrow$ (w.h.p.) $|p| \times |\hat{p}| = n^{O(1)}$

$\Rightarrow$ $\max_i A_i x \leq \min_j A_j^{\mathsf{T}} \hat{x} + O(\ln(n)/\varepsilon)$.

Stopping cond'n and weak duality $\Rightarrow$ $(1 \pm O(\varepsilon))$-optimal. $\square$

implementation in time $O(n^2 + n\log(n)/\varepsilon^2)$

- max plays random i from p, where $p_i = \exp(\ \varepsilon A_i x)$
- min plays random j from $\hat{p}$, where $\hat{p}_j = \exp(-\varepsilon A_j^{\mathsf{T}} \hat{x})$

STOP when $\max_i A_i x \approx \ln(n)/\varepsilon^2$ or $\min_j A_j^{\mathsf{T}} \hat{x} \approx \ln(n)/\varepsilon^2$.

Bottleneck is maintaining p, $\hat{p}$ (i.e., Ax, $A^{\mathsf{T}}\hat{x}$):

$\Delta x :$	$+1$		ΔAx
1	0	0	
1	1	0	$+1$
0	1	1	$+1$

Do work for each increase in a row payoff $A_i x$...

but $A_i x \leq \ln(n)/\varepsilon^2$, so total work $O(n\log(n)/\varepsilon^2)$.

implementation in time $O(n^2 + n\log(n)/\varepsilon^2)$

- max plays random i from p, where $p_i = \exp(\ \varepsilon A_i x)$
- min plays random j from $\hat{p}$, where $\hat{p}_j = \exp(-\varepsilon A_j^\intercal \hat{x})$

STOP when $\max_i A_i x \approx \ln(n)/\varepsilon^2$ or $\min_j A_j^\intercal \hat{x} \approx \ln(n)/\varepsilon^2$.

Bottleneck is maintaining p, $\hat{p}$ (i.e., Ax, $A^\intercal \hat{x}$):

$\Delta\hat{x}$			
	1	0	0
	1	1	0
$+1$	0	1	1
$\Delta A^\intercal \hat{x}$:		$+1$	$+1$

Do work for each increase in a row payoff $A_i x$...

or a column payoff $A_j^\intercal \hat{x}$... (?!)

but $A_i x \leq \ln(n)/\varepsilon^2$, so total work $O(n\log(n)/\varepsilon^2)$.

implementation in time $O(n^2 + n\log(n)/\varepsilon^2)$

- max plays random i from p, where $p_i = \exp(\ \varepsilon A_i x)$
- min plays random j from $\hat{p}$, where $\hat{p}_j = \exp(-\varepsilon A_j^{\mathsf{T}} \hat{x})$

STOP when $\max_i A_i x \approx \ln(n)/\varepsilon^2$ or $\min_j A_j^{\mathsf{T}} \hat{x} \approx \ln(n)/\varepsilon^2$.

Bottleneck is maintaining p, $\hat{p}$ (i.e., Ax, $A^{\mathsf{T}}\hat{x}$):

$\Delta\hat{x}$			
	1	0	0
	1	1	0
$+1$	0	1	1
$\Delta A^{\mathsf{T}}\hat{x}$:		$+1$	$+1$

Do work for each increase in a row payoff $A_i x$...
or a column payoff $A_j^{\mathsf{T}}\hat{x}$... (?!)
but $A_i x \le \ln(n)/\varepsilon^2$, so total work $O(n\log(n)/\varepsilon^2)$.
fix: delete column j when $A_j^{\mathsf{T}}\hat{x} \ge \ln(n)/\varepsilon^2$... ($O(n^2)$ time)

generalizing to any non-negative matrix A

- adapt ideas for width-independence (Garg/Könemann 1998)
- random sampling to deal with small A_{ij}
- preprocess matrix — approximately sort within each row & column

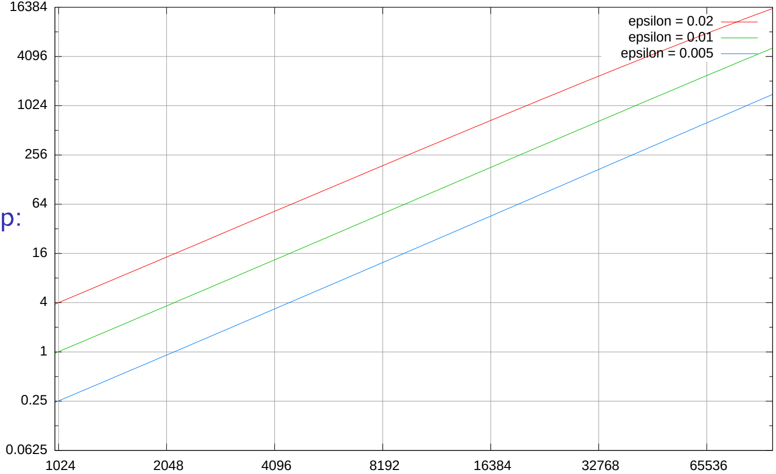
running time for N non-zeros, r rows, c cols:

$$O(N + (r + c)\log(N)/\varepsilon^2).$$

practical performance

- first implementation: $10n^2 + 75n\log(n)/\varepsilon^2$ basic op's
- simplex (GLPK): at least $5n^3$ basic op's for $\varepsilon \leq 0.05$

speedup:



n = rows, columns

conclusion

For dense matrices with thousands of rows and columns, the algorithm finds near-optimal solution much faster than Simplex!

open problems:

- improve Luby & Nisan's parallel algorithm (1993)
- mixed packing/covering problems
- implicitly defined problems (e.g. multicommodity flow)
- dynamic problems