Polynomial-time Computation of Exact Correlated Equilibrium in Compact Games

Albert Xin Jiang Kevin Leyton-Brown
Department of Computer Science
University of British Columbia

June 7, 2011

Correlated Equilibrium

- correlated equilibrium (CE) [Aumann, 1974; Aumann, 1987]
  - generalization of Nash equilibrium

Correlated Equilibrium

- correlated equilibrium (CE) [Aumann, 1974; Aumann, 1987]
  - generalization of Nash equilibrium
  - natural learning dynamics converge to CE

Correlated Equilibrium

- correlated equilibrium (CE) [Aumann, 1974; Aumann, 1987]
  - generalization of Nash equilibrium
  - natural learning dynamics converge to CE
  - tractable to compute: LP
    - polynomial in the size of the normal form

Compact Game Representations

Compact representations are necessary for large games with structured utility functions

- symmetric games / anonymous games
- graphical games [Kearns, Littman & Singh, 2001]
- action-graph games [Jiang, Leyton-Brown & Bhat, 2011]

Compact Game Representations

Compact representations are necessary for large games with structured utility functions

- symmetric games / anonymous games
- graphical games [Kearns, Littman & Singh, 2001]
- action-graph games [Jiang, Leyton-Brown & Bhat, 2011]

Computation of solution concepts

Compact Game Representations

Compact representations are necessary for large games with structured utility functions

- symmetric games / anonymous games
- graphical games [Kearns, Littman & Singh, 2001]
- action-graph games [Jiang, Leyton-Brown & Bhat, 2011]

Computation of solution concepts

- NASH: PPAD-complete $\rightarrow$ PPAD-complete [Daskalakis *et al.*, 2006]
- Pure Nash: P $\rightarrow$ NP-complete [Gottlob *et al.*, 2005]

Compact Game Representations

Compact representations are necessary for large games with structured utility functions

- symmetric games / anonymous games
- graphical games [Kearns, Littman & Singh, 2001]
- action-graph games [Jiang, Leyton-Brown & Bhat, 2011]

Computation of solution concepts

- NASH: PPAD-complete → PPAD-complete [Daskalakis *et al.*, 2006]
- Pure Nash: P → NP-complete [Gottlob *et al.*, 2005]
- CE: P → ?

Computing CE for Compact Game Representations

- Papadimitriou & Roughgarden [2008]: polynomial-time algorithm for computing a CE

Computing CE for Compact Game Representations

- Papadimitriou & Roughgarden [2008]: polynomial-time algorithm for computing a CE
- Stein, Parrilo & Ozdaglar [2010]:
  - this algorithm can fail to find a correct solution
  - a slightly modified version computes approximate CE in time polynomial in $\log \frac{1}{\epsilon}$ and representation size

Computing CE for Compact Game Representations

- Papadimitriou & Roughgarden [2008]: polynomial-time algorithm for computing a CE
- Stein, Parrilo & Ozdaglar [2010]:
  - this algorithm can fail to find a correct solution
  - a slightly modified version computes approximate CE in time polynomial in $\log \frac{1}{\epsilon}$ and representation size
  - a class of similar algorithms also cannot find an exact solution

Computing CE for Compact Game Representations

- Papadimitriou & Roughgarden [2008]: polynomial-time algorithm for computing a CE
- Stein, Parrilo & Ozdaglar [2010]:
  - this algorithm can fail to find a correct solution
  - a slightly modified version computes approximate CE in time polynomial in $\log \frac{1}{\epsilon}$ and representation size
  - a class of similar algorithms also cannot find an exact solution
- Our contribution:
  - a variant of Papadimitriou & Roughgarden's algorithm that computes an exact CE in polynomial time
  - conceptually simpler

Computing CE for Compact Game Representations

- Papadimitriou & Roughgarden [2008]: polynomial-time algorithm for computing a CE
- Stein, Parrilo & Ozdaglar [2010]:
  - this algorithm can fail to find a correct solution
  - a slightly modified version computes approximate CE in time polynomial in $\log \frac{1}{\epsilon}$ and representation size
  - a class of similar algorithms also cannot find an exact solution
- Our contribution:
  - a variant of Papadimitriou & Roughgarden's algorithm that computes an exact CE in polynomial time
  - conceptually simpler
  - new attractive property: outputs CE with polynomial-sized support

Outline

Outline

1. Problem Formulation
2. Papadimitriou and Roughgarden's Algorithm
3. Algorithm for Exact Correlated Equilibrium

CE

- simultaneous-move game
  - n players
  - player p's pure strategy $s_p \in S_p$
  - pure strategy profile $s \in S = \prod_{p=1}^{n} S_p$
  - utility for p under pure strategy profile s is integer u_s^p

CE

- simultaneous-move game
  - n players
  - player p's pure strategy $s_p \in S_p$
  - pure strategy profile $s \in S = \prod_{p=1}^{n} S_p$
  - utility for p under pure strategy profile s is integer u_s^p
- a CE is a distribution x over S:
  - a trusted intermediary draws a strategy profile s from this distribution
  - announce to each player p (privately) her own component s_p
  - p will have no incentive to choose another strategy, assuming others follow suggestions

LP formulation

- incentive constraints: for all players p and all $i, j \in S_p$:

$$\sum_{s \in S_{-p}} [u^p_{is} - u^p_{js}] x_{is} \geq 0$$

write as

$$Ux \geq 0.$$

LP formulation

- incentive constraints: for all players p and all $i, j \in S_p$:

$$\sum_{s \in S_{-p}} [u^p_{is} - u^p_{js}] x_{is} \geq 0$$

  write as

$$Ux \geq 0.$$

- x is a distribution: $x \geq 0$, $\sum_s x_s = 1$

LP formulation

- incentive constraints: for all players p and all $i, j \in S_p$:

$$\sum_{s \in S_{-p}} [u^p_{is} - u^p_{js}] x_{is} \geq 0$$

write as

$$Ux \geq 0.$$

- x is a distribution: $x \geq 0$, $\sum_s x_s = 1$
- m^n variables, nm^2 constraints

Outline

Unbounded LP and Infeasible Dual

- consider the linear program (P):

$$\max \sum_s x_s$$

$$Ux \geq 0, \qquad x \geq 0$$

Unbounded LP and Infeasible Dual

- consider the linear program (P):

$$\max \sum_s x_s$$
$$Ux \geq 0, \qquad x \geq 0$$

- interested in nonzero solution

Unbounded LP and Infeasible Dual

- consider the linear program (P):

$$\max \sum_s x_s$$

$$Ux \geq 0, \qquad x \geq 0$$

- interested in nonzero solution
- its dual (D)

$$U^T y \leq -1$$

$$y \geq 0$$

has nm^2 variables, about m^n constraints

Ellipsoid Against Hope

Ellipsoid Against Hope

- run the ellipsoid algorithm on (D), with the following Product Separation Oracle:
    - given a vector $y^{(i)} \geq 0$, compute product distribution $x^{(i)}$ such that $x^{(i)} U^T y^{(i)} = 0$.
    - output cutting plane $[x^{(i)} U^T] y \leq -1$.

Ellipsoid Against Hope

- run the ellipsoid algorithm on (D), with the following Product Separation Oracle:
  - given a vector $y^{(i)} \geq 0$, compute product distribution $x^{(i)}$ such that $x^{(i)} U^T y^{(i)} = 0$.
  - output cutting plane $[x^{(i)} U^T] y \leq -1$.
  - $[x^{(i)} U^T]$ are differences of expected utilities under product distributions

Ellipsoid Against Hope

- run the ellipsoid algorithm on (D), with the following Product Separation Oracle:
  - given a vector $y^{(i)} \geq 0$, compute product distribution $x^{(i)}$ such that $x^{(i)} U^T y^{(i)} = 0$.
  - output cutting plane $[x^{(i)} U^T] y \leq -1$.
  - $[x^{(i)} U^T]$ are differences of expected utilities under product distributions
    - Assumption: $\exists$ a poly-time algorithm for expected utilities under product distributions

Ellipsoid Against Hope

- run the ellipsoid algorithm on (D), with the following Product Separation Oracle:
  - given a vector $y^{(i)} \geq 0$, compute product distribution $x^{(i)}$ such that $x^{(i)}U^T y^{(i)} = 0$.
  - output cutting plane $[x^{(i)}U^T]y \leq -1$.
  - $[x^{(i)}U^T]$ are differences of expected utilities under product distributions
    - Assumption: $\exists$ a poly-time algorithm for expected utilities under product distributions
- The ellipsoid algorithm will stop after a polynomial number of steps and determine that the program is infeasible.

Ellipsoid Against Hope (cont'd)

- Consider the linear program (D'):

$$[XU^T]y \leq -1, \qquad y \geq 0$$

whose constraints are the generated cutting planes.

Ellipsoid Against Hope (cont'd)

- Consider the linear program (D'):

$$[XU^T]y \leq -1, \qquad y \geq 0$$

  whose constraints are the generated cutting planes.
- If we apply the same ellipsoid method, with a separation oracle that returns the cut $x^{(i)}U^T y \leq -1$ given query $y^{(i)}$, it would go through the same sequence of queries $y^{(i)}$ and return infeasible.
- Therefore (D') is infeasible (presuming that numerical problems do not arise).

Ellipsoid Against Hope (cont'd)

- Infeasibility of (D') implies that its dual program (P'):

$$[UX^T]\alpha \geq 0, \qquad \alpha \geq 0$$

  is unbounded.
- (P') has polynomial size.

Ellipsoid Against Hope (cont'd)

- Infeasibility of (D') implies that its dual program (P'):

$$[UX^T]\alpha \geq 0, \qquad \alpha \geq 0$$

  is unbounded.
- (P') has polynomial size.
- compute a nonzero solution α
- $X^T\alpha$ is a nonzero solution of (P), therefore can be scaled to be a CE $\Rightarrow$ mixture of polynomial number of product distributions

Ellipsoid Against Hope (cont'd)

- Infeasibility of (D') implies that its dual program (P'):

$$[UX^T]\alpha \geq 0, \qquad \alpha \geq 0$$

  is unbounded.
- (P') has polynomial size.
- compute a nonzero solution α
- $X^T\alpha$ is a nonzero solution of (P), therefore can be scaled to be a CE $\Rightarrow$ mixture of polynomial number of product distributions

$$\begin{array}{ccc} (P) & \rightarrow & (D) \\ \uparrow & & \downarrow \\ (P') & \leftarrow & (D') \end{array}$$

Numerical Precision Issues

- a run of the ellipsoid method requires as inputs
  - initial ball with radius R
  - volume lower bound v
- correct values of R and v depend on the max encoding size of a constraint of the LP

Numerical Precision Issues

- a run of the ellipsoid method requires as inputs
  - initial ball with radius R
  - volume lower bound v
- correct values of R and v depend on the max encoding size of a constraint of the LP
- a constraint of (D'), e.g. $x^{(i)}U^Ty \leq -1$, may require more bits than any of the constraints of (D)
  - infeasibility of (D') is not guaranteed

Numerical Precision Issues

- a run of the ellipsoid method requires as inputs
  - initial ball with radius R
  - volume lower bound v
- correct values of R and v depend on the max encoding size of a constraint of the LP
- a constraint of (D'), e.g. $x^{(i)}U^T y \leq -1$, may require more bits than any of the constraints of (D)
  - infeasibility of (D') is not guaranteed
- Papadimitriou and Roughgarden [2008] proposed a method to overcome this issue
- Stein, Parrilo & Ozdaglar [2010] showed that it is insufficient to compute an exact CE.

Numerical Precision Issues

- a run of the ellipsoid method requires as inputs
  - initial ball with radius R
  - volume lower bound v
- correct values of R and v depend on the max encoding size of a constraint of the LP
- a constraint of (D'), e.g. $x^{(i)}U^T y \leq -1$, may require more bits than any of the constraints of (D)
  - infeasibility of (D') is not guaranteed
- Papadimitriou and Roughgarden [2008] proposed a method to overcome this issue
- Stein, Parrilo & Ozdaglar [2010] showed that it is insufficient to compute an exact CE.
  - any algorithm that outputs a mixture of product distributions with symmetry-preserving property would fail to find an exact CE.

Outline

Overview of Our Approach

- We replace the Product Separation Oracle with a modified version (Purified Separation Oracle) that generates cuts corresponding to pure strategy profiles.

Overview of Our Approach

- We replace the Product Separation Oracle with a modified version (Purified Separation Oracle) that generates cuts corresponding to pure strategy profiles.
- Now each constraint of (D') is one of the original constraints of (D).
  - any run of ellipsoid method that is valid for (D) is also valid for (D')
  - no longer requiring special mechanism to deal with numerical issues

Overview of Our Approach

- We replace the Product Separation Oracle with a modified version (Purified Separation Oracle) that generates cuts corresponding to pure strategy profiles.
- Now each constraint of (D') is one of the original constraints of (D).
  - any run of ellipsoid method that is valid for (D) is also valid for (D')
  - no longer requiring special mechanism to deal with numerical issues
- A solution of (P') is a mixture of polynomial number of pure-strategy profiles.

Overview of Our Approach

- We replace the Product Separation Oracle with a modified version (Purified Separation Oracle) that generates cuts corresponding to pure strategy profiles.
- Now each constraint of (D') is one of the original constraints of (D).
    - any run of ellipsoid method that is valid for (D) is also valid for (D')
    - no longer requiring special mechanism to deal with numerical issues
- A solution of (P') is a mixture of polynomial number of pure-strategy profiles.

$$\begin{array}{ccc} (P) & \rightarrow & (D) \\ \uparrow & & \downarrow \\ (P') & \leftarrow & (D') \end{array}$$

Purified Separation Oracle: Existence

Lemma

Given any dual vector $y \geq 0$, there exists a pure strategy profile s such that $(U_s)^T y \geq 0$.

Purified Separation Oracle: Existence

Lemma

Given any dual vector $y \geq 0$, there exists a pure strategy profile s such that $(U_s)^T y \geq 0$.

Proof.

- we know there exists a product distribution x such that $xU^T y = 0$.
- $x[U^T y]$ is the expected value of $(U_s)^T y$ under distribution x, which we denote $E_{s \sim x}[(U_s)^T y]$
- there must exist s such that $(U_s)^T y \geq xU^T y = 0$.

Purified Separation Oracle: Existence

Lemma

Given any dual vector $y \geq 0$, there exists a pure strategy profile s such that $(U_s)^T y \geq 0$.

Proof.

- we know there exists a product distribution x such that $xU^T y = 0$.
- $x[U^T y]$ is the expected value of $(U_s)^T y$ under distribution x, which we denote $E_{s\sim x}[(U_s)^T y]$
- there must exist s such that $(U_s)^T y \geq xU^T y = 0$.

- not efficiently constructive

Purified Separation Oracle: Existence

Lemma

Given any dual vector $y \geq 0$, there exists a pure strategy profile s such that $(U_s)^T y \geq 0$.

Proof.

- we know there exists a product distribution x such that $xU^T y = 0$.
- $x[U^T y]$ is the expected value of $(U_s)^T y$ under distribution x, which we denote $E_{s\sim x}[(U_s)^T y]$
- there must exist s such that $(U_s)^T y \geq xU^T y = 0$.

- not efficiently constructive
- sampling from x yields approximate cutting planes

Purified Separation Oracle: Algorithm

Derandomize using the method of conditional probabilities

1. Given $y \geq 0$, compute product distribution x satisfying $xU^T y = 0$, i.e. $E_{s\sim x}[(U_s)^T y] = 0$.

Purified Separation Oracle: Algorithm

Derandomize using the method of conditional probabilities

1. Given $y \geq 0$, compute product distribution x satisfying $xU^T y = 0$, i.e. $E_{s \sim x}[(U_s)^T y] = 0$.
2. For each player p,
   - pick $s_p \in S_p$ such that the conditional expectation

$$E_{s \sim x}[(U_s)^T y | s_1, \ldots, s_p] \geq 0.$$

Purified Separation Oracle: Algorithm

Derandomize using the method of conditional probabilities

1. Given $y \geq 0$, compute product distribution x satisfying $xU^T y = 0$, i.e. $E_{s\sim x}[(U_s)^T y] = 0$.
2. For each player p,
   - pick $s_p \in S_p$ such that the conditional expectation

$$E_{s\sim x}[(U_s)^T y | s_1, \ldots, s_p] \geq 0.$$

- such s_p must exists because $E_{s\sim x}[(U_s)^T y | s_1, \ldots, s_{p-1}] \geq 0$
- each conditional expectation can be efficiently computed using the expected utility algorithm

Purified Separation Oracle: Algorithm

Derandomize using the method of conditional probabilities

1. Given $y \geq 0$, compute product distribution x satisfying $xU^T y = 0$, i.e. $E_{s\sim x}[(U_s)^T y] = 0$.
2. For each player p,
   - pick $s_p \in S_p$ such that the conditional expectation

$$E_{s\sim x}[(U_s)^T y | s_1, \ldots, s_p] \geq 0.$$

   - such s_p must exists because $E_{s\sim x}[(U_s)^T y | s_1, \ldots, s_{p-1}] \geq 0$
   - each conditional expectation can be efficiently computed using the expected utility algorithm
3. Output $s = (s_1, \ldots, s_p)$, and cutting plane $(U_s)^T y \leq -1$.

Purified Separation Oracle: Algorithm

Derandomize using the method of conditional probabilities

1. Given $y \geq 0$, compute product distribution x satisfying $xU^T y = 0$, i.e. $E_{s\sim x}[(U_s)^T y] = 0$.
2. For each player p,
   - pick $s_p \in S_p$ such that the conditional expectation

$$E_{s\sim x}[(U_s)^T y | s_1, \ldots, s_p] \geq 0.$$

   - such s_p must exists because $E_{s\sim x}[(U_s)^T y | s_1, \ldots, s_{p-1}] \geq 0$
   - each conditional expectation can be efficiently computed using the expected utility algorithm
3. Output $s = (s_1, \ldots, s_p)$, and cutting plane $(U_s)^T y \leq -1$.

Can return asymmetric cuts even for symmetric games and symmetric y.

Conclusion

- A variant of Ellipsoid Against Hope algorithm that computes an exact CE in polynomial time
  - derandomization of the Product Separation Oracle
  - as a result the algorithm is simplified
  - polynomial-sized support

Conclusion

- A variant of Ellipsoid Against Hope algorithm that computes an exact CE in polynomial time
  - derandomization of the Product Separation Oracle
  - as a result the algorithm is simplified
  - polynomial-sized support
- Directions
  - Apply to other related algorithms, e.g. Huang and Von Stengel [2008] polynomial-time algorithm for extensive-form correlated equilibria.

Conclusion

- A variant of Ellipsoid Against Hope algorithm that computes an exact CE in polynomial time
  - derandomization of the Product Separation Oracle
  - as a result the algorithm is simplified
  - polynomial-sized support
- Directions
  - Apply to other related algorithms, e.g. Huang and Von Stengel [2008] polynomial-time algorithm for extensive-form correlated equilibria.
  - Practical computation of CE

Aumann, R. (1974). Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, *1*(1), 67–96.

Aumann, R. (1987). Correlated equilibrium as an expression of Bayesian rationality. *Econometrica: Journal of the Econometric Society*, 1–18.

Daskalakis, C., Fabrikant, A., & Papadimitriou, C. (2006). The game world is flat: The complexity of Nash equilibria in succinct games. *ICALP: Proceedings of the International Colloquium on Automata, Languages and Programming* (pp. 513–524).

Gottlob, G., Greco, G., & Scarcello, F. (2005). Pure Nash equilibria: Hard and easy games. *Journal of Artificial Intelligence Research*, *24*, 357–406.

Huang, W., & Von Stengel, B. (2008). Computing an extensive-form correlated equilibrium in polynomial time. *WINE: Proceedings of the Workshop on Internet and Network Economics* (pp. 506–513).

Papadimitriou, C., & Roughgarden, T. (2008). Computing correlated equilibria in multi-player games. *Journal of the ACM*, *55*(3), 14.