Structural Estimation of Directional Dynamic Games With Multiple Equilibria

Fedor Iskhakov, Australian National University
Dennis Kristensen, University College London
John Rust, Georgetown University
Bertel Schjerning, University of Copenhagen

Econometric Society Australasian Meetings
December 5, 2024

Estimation of stochastic dynamic games

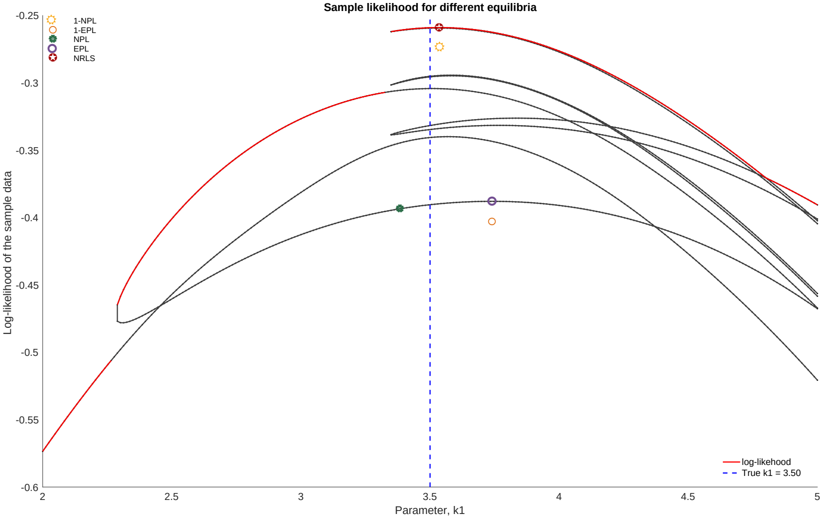
1. Several decision makers (*players*)
2. Maximize discounted expected lifetime utility
3. Anticipate consequences of their current actions
4. Anticipate actions by other players in current and future periods (*strategic interaction*)
5. Operate in a stochastic environment (*state of the game*) which evolution depends on the collective actions of the players

- Estimate structural parameters of these models
- Focus on multiplicity of equilibria in the model and across markets in the data
- Applications in empirical IO, but also family economics, structural labor, public economics, etc.

Estimation of stochastic dynamic games is hard

- Finding an equilibrium = solving a large system of non-linear equations
  Computing even a single equilibrium is not a trivial task
- Existing literature does not fully explore, and usually assumes away multiplicity of solutions in the theoretical model
  Numerical algorithm inadvertently becomes equilibrium selection mechanism
- Standard assumption in the existing literature is that a single equilibrium is played in the data
- Practical methods rely on convergence of iterative algorithms or smoothness in the constrains of the optimization problem
  Existing methods perform poorly when multiplicity is present

Equilibrium correspondence and discontinuous likelihood



Our contribution

Nested Recursive Lexicographical Search (NRLS) estimator

- Propose robust maximum likelihood estimator for a subclass of stochastic dynamic games, directional dynamic games (DDG)
- Fully robust to multiplicity of equilibria
- Relax single-equilibrium-in-data assumption
- Nested MLE estimator: model solved for each trial value of parameters
- Employ algorithm from integer programming to maximize likelihood function over the finite set of equilibria
  The algorithm is computationally more efficient in larger samples, while delivering exact MLE in all samples
- Provide Monte Carlo evidence of computational feasibility
- Compare to a battery of existing estimators: CCP/PML, NPL, EPL and MPEC

Markov Perfect Equilibria

- Discrete-time infinite-horizon dynamic stochastic games with discrete states and actions
- MPE is a pair of strategy profiles and value functions such that

$$\begin{aligned} V &= \Psi^V(V, P, \theta) && \text{(Bellman equations)} \\ P &= \Psi^P(V, P, \theta) && \text{(CCPs = mutual best responces)} \end{aligned}$$

- $\Psi = \left(\Psi^V, \Psi^P\right)$ gives the structure of the model
- Denote the set of all equilibria in the model as

$$\mathcal{E}(\Psi, \theta) = \left\{ (P, V) \,\middle|\, \begin{array}{l} V = \Psi^V(V, P, \theta) \\ P = \Psi^P(V, P, \theta) \end{array} \right\}$$

- Plan: full solution MLE estimator with NFXP structure: solve for all MPE equilibria for each trial value of θ

Maximum Likelihood Estimation

- Data from M independent markets from T periods

$$Z = \left\{ \bar{a}^{mt}, \bar{x}^{mt} \right\}_{m \in \mathcal{M}, t \in \mathcal{T}}$$

- Assume that only one equilibrium is played in the data (we relax this assumption later $\rightarrow$ grouped fixed effects)
- For a given θ denote the choice probabilities for player i at time t and market m as $P_i(a_i^{mt} | x^{mt}; \theta)$

$$\Big(P(\theta), V(\theta)\Big) \in \mathcal{E}(\Psi, \theta) : \quad P(\theta) = \Big\{ P_i(a_i^{mt} | x^{mt}; \theta) \Big\}_{i,m,t}$$

- MLE estimator $\hat{\theta}^{ML}$ is given by

$$\hat{\theta}^{ML} = \arg\max_{\theta} \left[\max_{(P(\theta), V(\theta) \in \mathcal{E}(\Psi, \theta)} \frac{1}{M} \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \log P_i(\bar{a}_i{}^{mt} | \bar{x}^{mt}; \theta) \right]$$

MLE via Constrained Optimization Approach

- Idea: use discretized values of P and V as *variables*
- Augmented log-likelihood function is

$$\mathcal{L}(Z, P, \theta) = \frac{1}{M} \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \log P_i(\bar{a}_i{}^{mt} | \bar{x}^{mt}; \theta)$$

- The constrained optimization formulation of the ML estimation problem is

$$\max_{\theta, P, V} \mathcal{L}(Z, P, \theta) \text{ subject to } \begin{cases} V = \Psi^V(V, P, \theta) \\ P = \Psi^P(V, P, \theta) \end{cases}$$

- Math programming with equilibrium constraints (MPEC)
- Does not rely *as much* on the structure of the problem
- Much bigger computational problem
- Implements the same MLE estimator (*when it works*)

Su (2013); Egesdal, Lai and Su (2015)

Estimation methods for dynamic stochastic games

- Two step (CCP) estimators
  - Fast, do not impose equilibrium constraints, finite sample bias
  1. Estimate CCP $\rightarrow \hat{P}$
  2. Method of moments • Minimal distance • Pseudo likelihood

  Hotz, Miller (1993); Altug, Miller (1998); Pakes, Ostrovsky, and Berry (2007); Pesendorfer, Schmidt-Dengler (2008)

- Nested pseudo-likelihood (NPL)
  - Recursive two step pseudo-likelihood
  - Bridges the gap between efficiency and tractability
  - Unstable under multiplicity

  Aguirregabiria, Mira (2007); Aguirregabiria, Marcoux (2021)

- Efficient pseudo-likelihood (EPL)
  - Incorporates Newton step in the NPL operator
  - More robust to the stability and multiplicity of equilibria

  Dearing, Blevins (2024), ReStud (forthcoming)

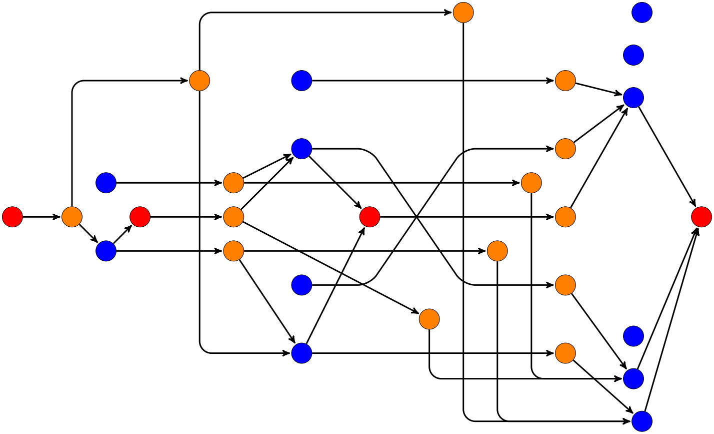
Overview of NRLS

Full solution nested fixed point MLE estimator
with computational enhancements to ensure tractability

- Robust and *computationally feasible*MLE estimator for directional dynamic games (DDG)
- Rely of full solution algorithm that provably computes all MPE under certain regularity conditions
- Employ discrete programming method (BnB) to maximize likelihood function over the finite set of equilibria
- Use non-parametric likelihood to refine BnB algorithm

- Fully robust to multiplicity of MPE
- Relax single-equilibrium-in-data assumption

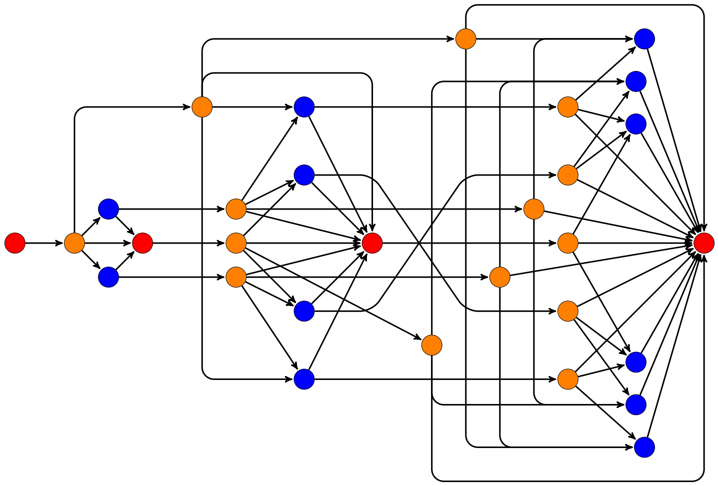
Strategy-specific partial order on game state space



Non-zero transition probabilities corresponding to any strategy profile σ induce a partial order on the state space

Strategy independent partial order on the state space

Coarsest common refinement of partial orders induced by all strategies

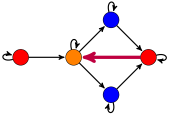


All possible transitions under any strategy profile

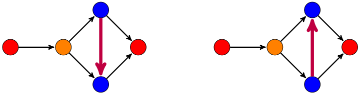
Definition of the Dynamic Directional Games

Finite state Markovian stochastic game is a DDG if it holds:

1. Every feasible strategy σ satisfies the no loop condition.



2. Every pair of feasible Markovian strategies σ and σ' induce consistent partial orders on the state space.

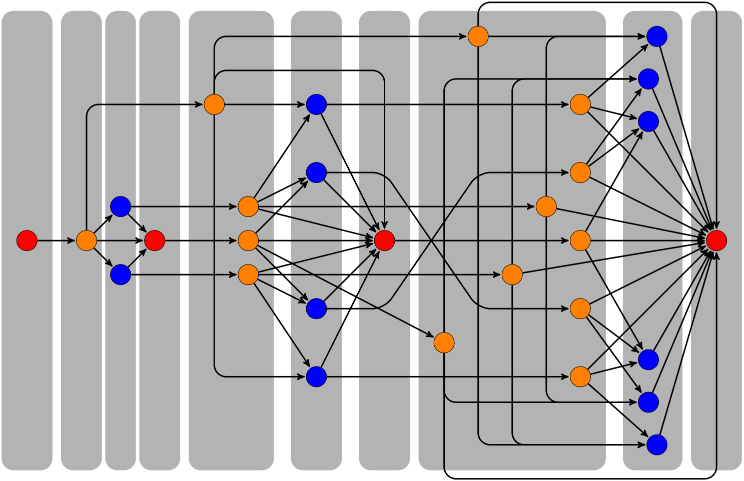


In this case the strategy independent partial order is given by a directional acyclig graph (DAG) with self loops

Iskhakov, Rust and Schjerning (2016) *Review of Economic Studies*

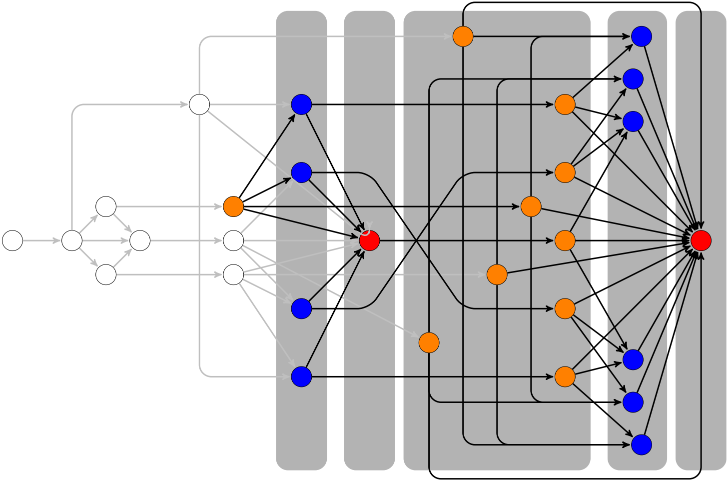
Total order on the set of stages

After running a topoligical sort algorithm on the DAG



Subgames of DDG and continuation strategies

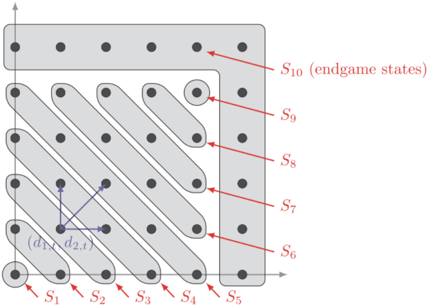
Only solution in continuation strategies is requires in each stage

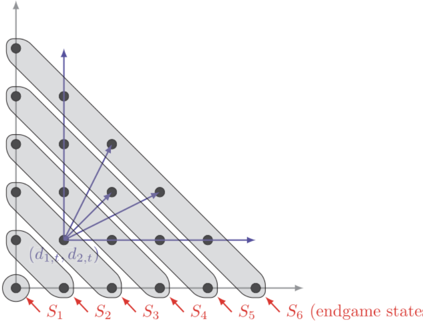


Stage recursion algorith = generalization of backward induction

Examples of Directional Dynamic Games

Many games have state dynamic evolutions described by a DAGs

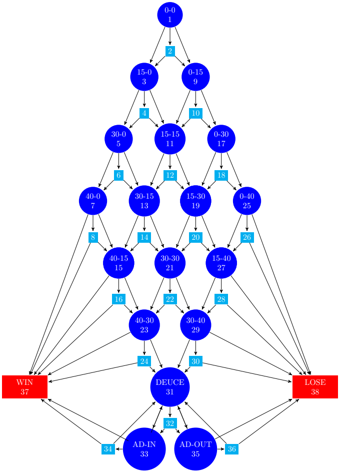




Judd, Schmedders, Yeltekin (2012), *IER*
"Optimal rules for patent researchers"

Dube, Hitsch, Chintagunta (2010), *Marketing Science*
"Tipping and concentration in markets with indirect network effects"

Tennis is a Directional Dynamic Game

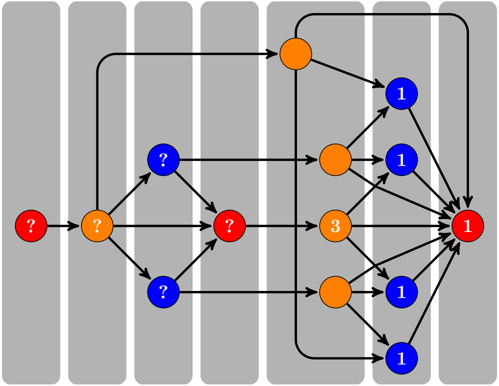


Anderson, Rosen, Rust, Wong (2024) *JPE* (forthcoming)
"Disequilibrium Play in Tennis"

Multiplicity of stage equibiria

Number of equilibria in the higher stages depends on the selected equilibria

- State recursion proceeds conditional on equilibrium selection rule
- Multiplicity of stage equilibria $\Leftrightarrow$ multiplicity
- Can systematically combine different stage equilibria



Recursive Lexicographic Search Algorithm

Building blocks of RLS algorithm:

1. State recursion algorithm solves the game conditional on equilibrium selection rule (ESR)
2. RLS algorithm efficiently cycles through all feasible ESRs

Challenge:

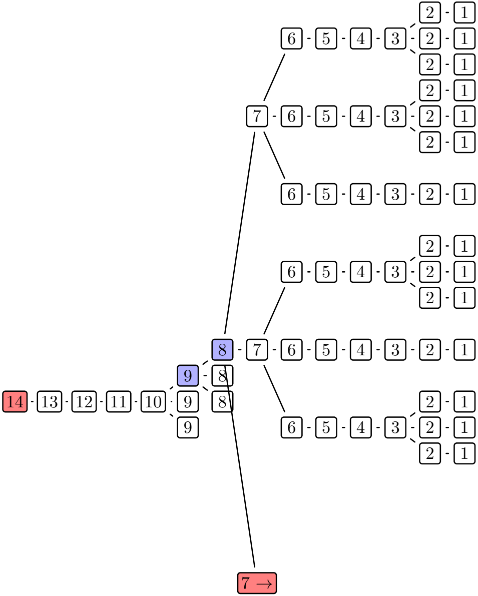
- Choice of a particular MPE for any stage game at any stage
- may alter the set and even the number of stage equilibria at earlier stages

Solution: RLS = depth-first tree traversal (illustration coming)

- Root of the tree is one of the absorbing states
- Levels of the tree correspond to the state points
- Branching happens when stages have multiple equilibria
- MPE of the game is given by a path from root to a leaf

RLS as tree traversal

- Levels of the tree are points in the state space
- Root is the absorbing state
- Leafs correspond to the apex
- MPE = path through the tree from root to leaf
- RLS algorithm = depth-first tree traversal



Nested Recursive Lexicographical Search (NRLS)

- Data from M independent markets from T periods $Z = \{\bar{a}^{mt}, \bar{x}^{mt}\}_{m \in \{1,\ldots,M\}, t \in \{1,\ldots,T\}}$
- Let the set of all MPE equilibria be $\mathcal{E} = \{1, \ldots, K(\theta)\}$

1. Outer loop
   Maximization of the likelihood function w.r.t. to structural parameters θ
   $$\theta^{ML} = \arg\max_{\theta \in \Theta} \mathcal{L}(Z, \theta)$$
2. Inner loop
   Maximization of the likelihood function w.r.t. equilibrium selection
   $$\mathcal{L}(Z, \theta) = \max_{k \in \{1,\ldots,K(\theta)\}} \mathcal{L}(Z, \theta, P_\theta^k)$$

Max of a function on a discrete set organized into RLS tree

Likelihood over the state space

- Given equilibrium k choice probabilities $P_\theta^k(a|x)$ likelihood is

$$\mathcal{L}(Z,\theta,P_\theta^k) = \frac{1}{M}\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{i=1}^{N} \log P_i^k(\bar{a}_i{}^{mt}|\bar{x}^{mt};\theta)$$

- Let ι index points in the state space, $\iota \in \{1,\dots,S\}$
  $\iota = 1$ initial point, $\iota = S$ the terminal (absorbing) state
- Denote n_ι the number of observations in state x_ι and $n_\iota^{a_i}$ the number of observations of player i taking action a_i at x_ι

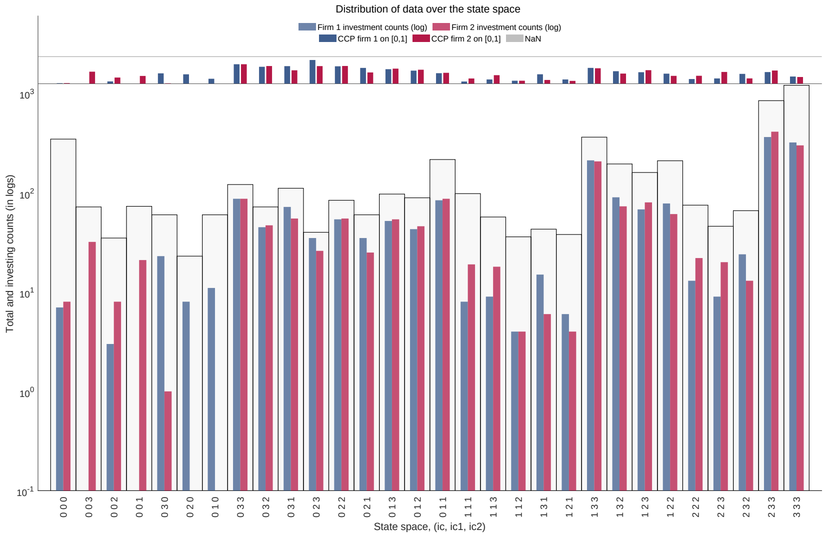
$$n_\iota = \sum_{m=1}^{M}\sum_{t=1}^{T} \mathbb{1}\{\bar{x}^{mt} = x_\iota\} \qquad n_\iota^{a_i} = \sum_{m=1}^{M}\sum_{t=1}^{T} \mathbb{1}\{\bar{a}_i{}^{mt} = a_i, \bar{x}^{mt} = x_\iota\}$$

- Then equilibrium-specific likelihood can be computed as

$$\mathcal{L}(Z,\theta,P_\theta^k) = \frac{1}{M}\sum_{\iota=1}^{S}\sum_{i=1}^{N}\sum_{a} n_\iota^{a_i} \log P_i^k(a|x_\iota;\theta)$$

Data distribution over the state space

1000 markets, 5 time periods, init at apex of the pyramid



Branch and bound (BnB) method

Old method for solving integer programming problems

1. Form a tree of subdivisions of the set of admissible plans
   $\Longrightarrow$ RLS tree
2. Specify a bounding function representing the best attainable objective on a given subset (branch)
   $\Longrightarrow$ Partial likelihood function from subset of states $\mathcal{S}$

$$\mathcal{L}^{\text{part}}(Z^{\mathcal{S}}, \theta, V_{\theta}^{k}) = \sum_{\iota \in \mathcal{S}} \sum_{i=1}^{N} \sum_{a} n_{\iota}^{a_i} \log P_i^k(a|x_\iota; \theta)$$

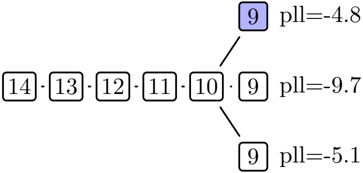
   where $Z^{\mathcal{S}} = \{(a, x) : x \in \mathcal{S} \subset \{1, \dots, S\}\}$ denotes data on $\mathcal{S}$
   - Monotonic decreasing in cardinality of $\mathcal{S}$
     (declines as more data is added)
   - Equals to full log-likelihood on full state space when $Z^{\mathcal{S}} = Z$
3. Dismiss the subsets of the plans where the bound is below the current best attained value of the objective

Ailsa Land and Alison Doig (Harcourt), 1960 *Econometrica*

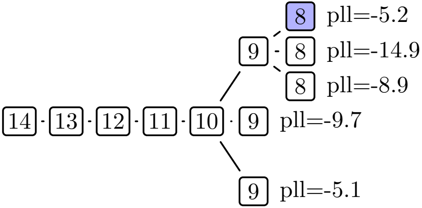
BnB on RLS tree, step 1

$\boxed{14} \cdot \boxed{13} \cdot \boxed{12} \cdot \boxed{11} \cdot \boxed{10}$ Partial loglikelihood = -3.2

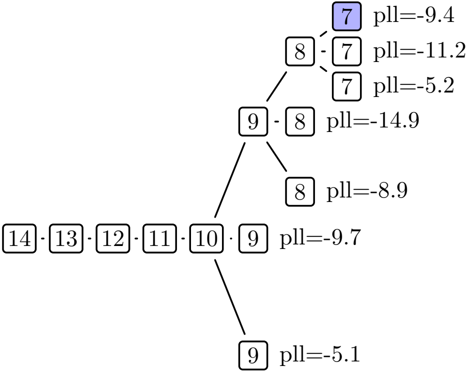
BnB on RLS tree, step 2



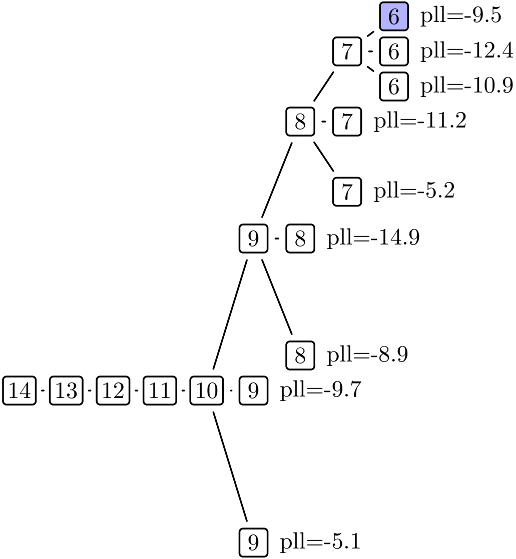
BnB on RLS tree, step 3



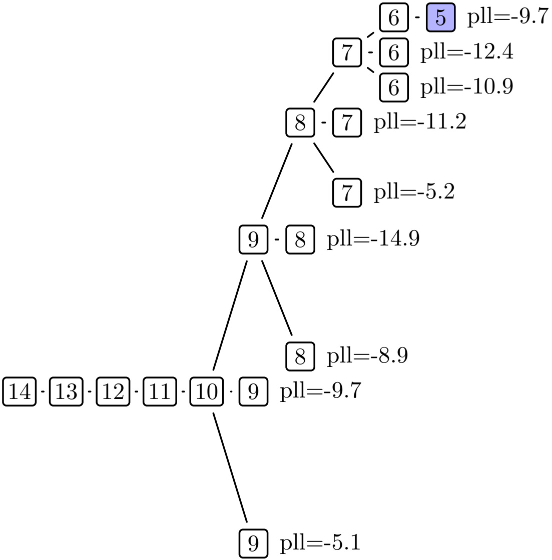
BnB on RLS tree, step 4



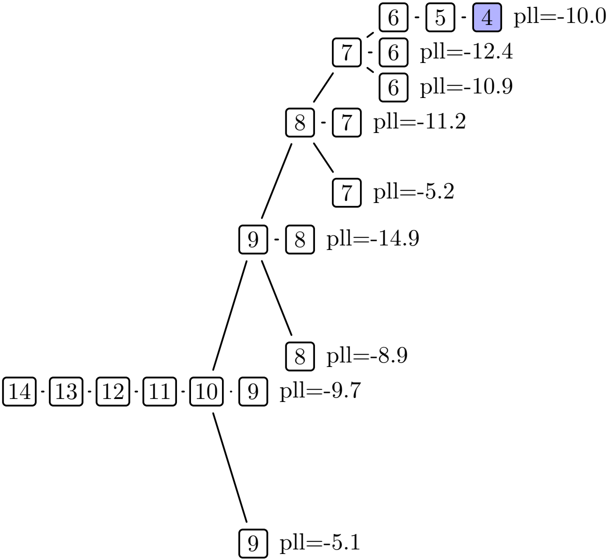
BnB on RLS tree, step 5



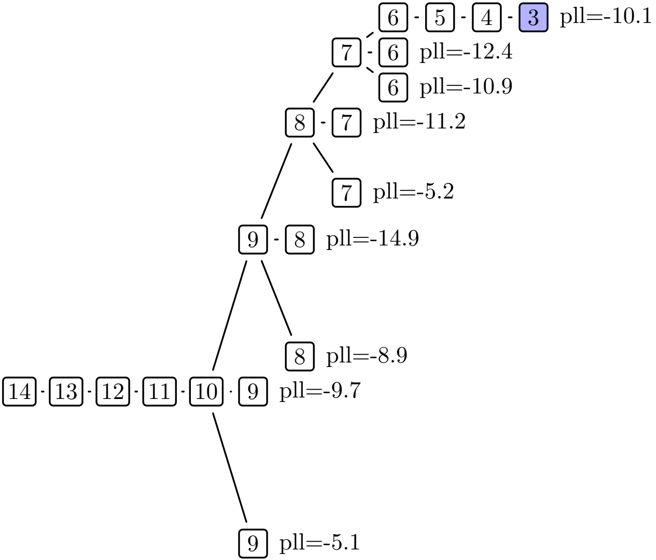
BnB on RLS tree, step 6



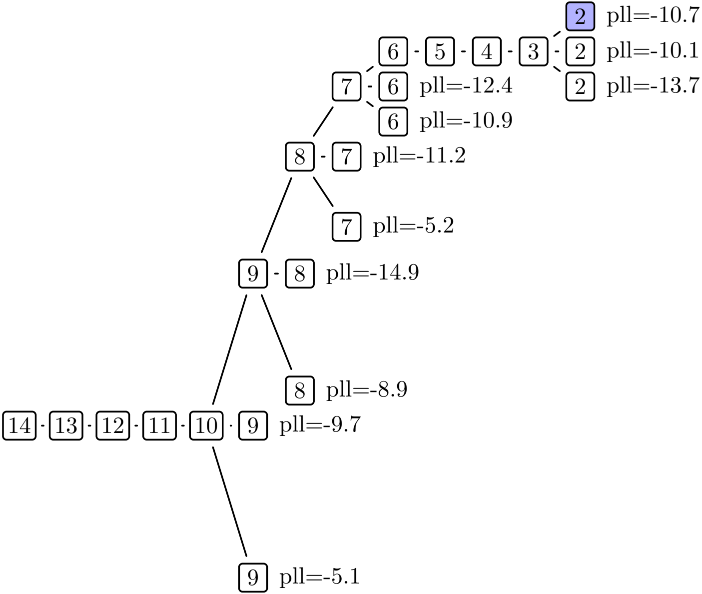
BnB on RLS tree, step 7



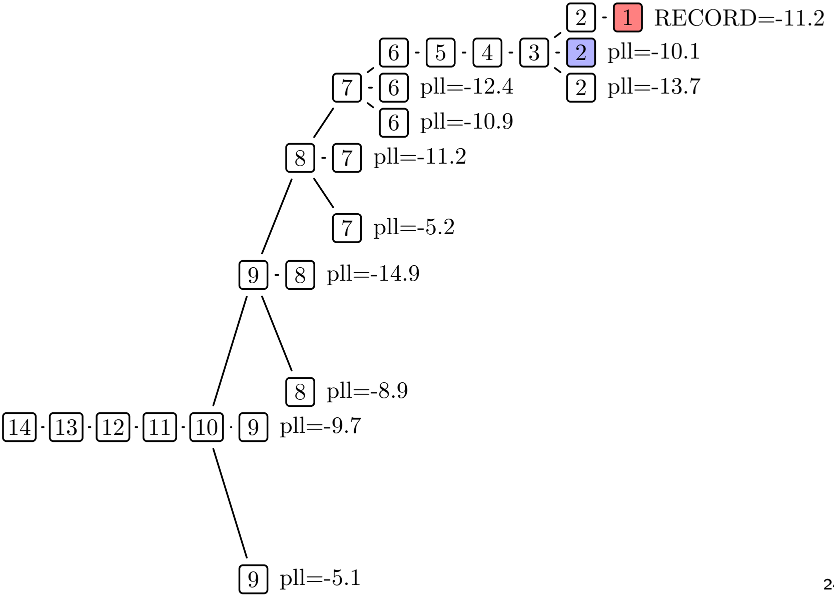
BnB on RLS tree, step 8



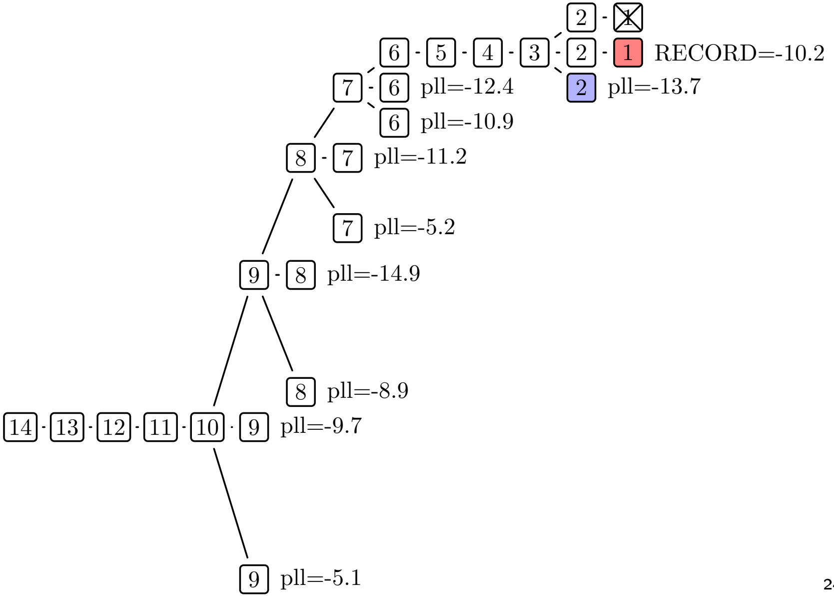
BnB on RLS tree, step 9



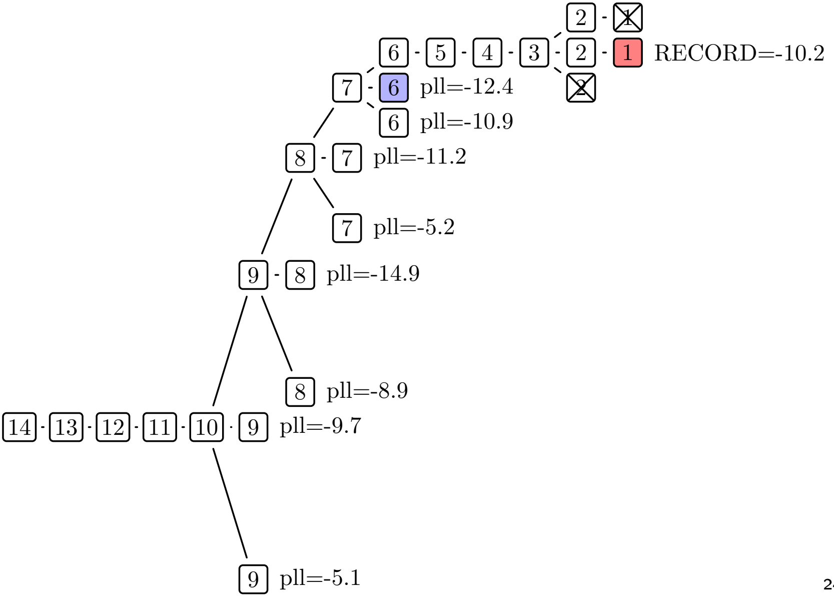
BnB on RLS tree, step 10



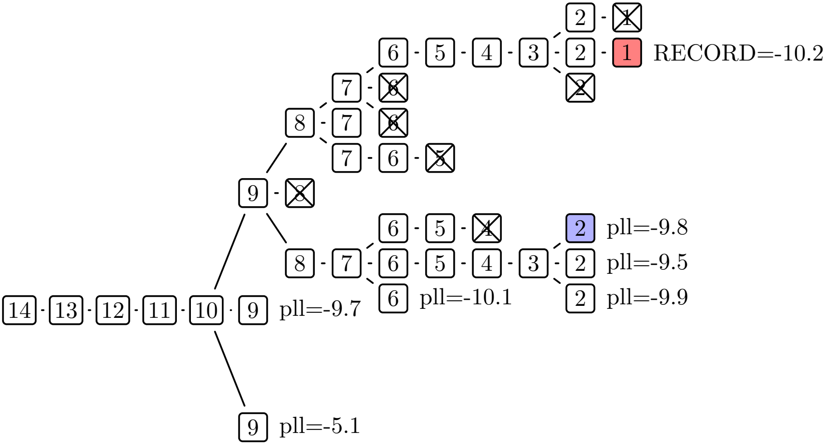
BnB on RLS tree, step 11



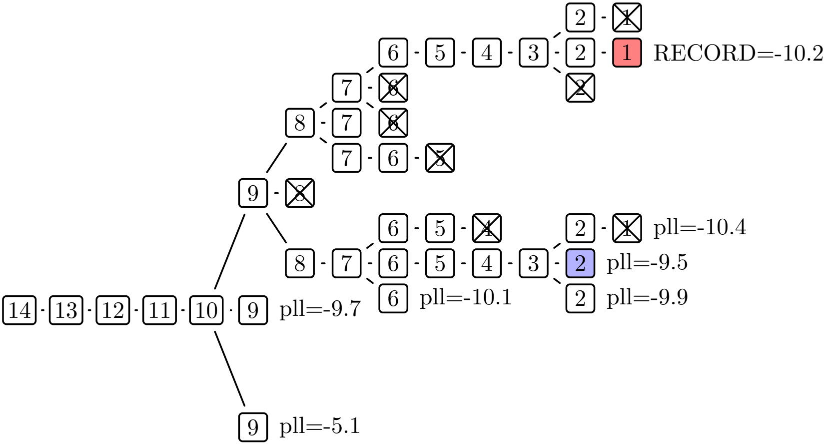
BnB on RLS tree, step 12



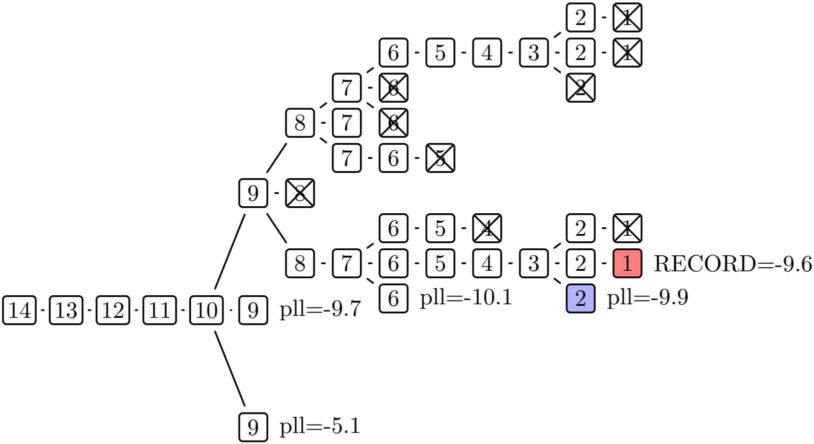
BnB on RLS tree, step 28



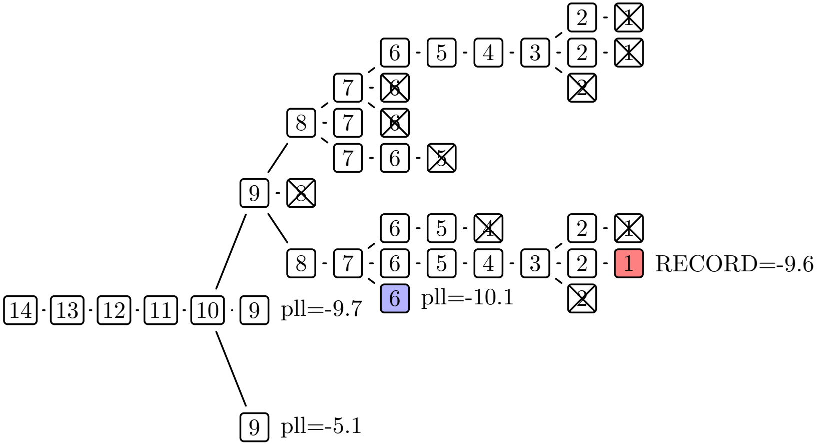
BnB on RLS tree, step 29



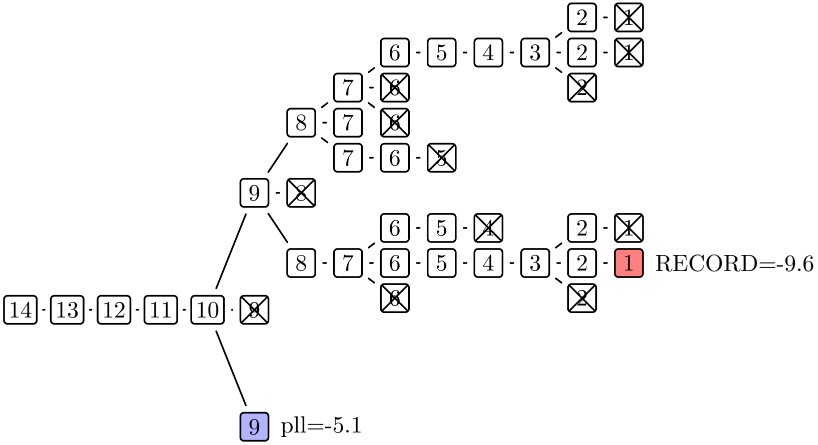
BnB on RLS tree, step 30



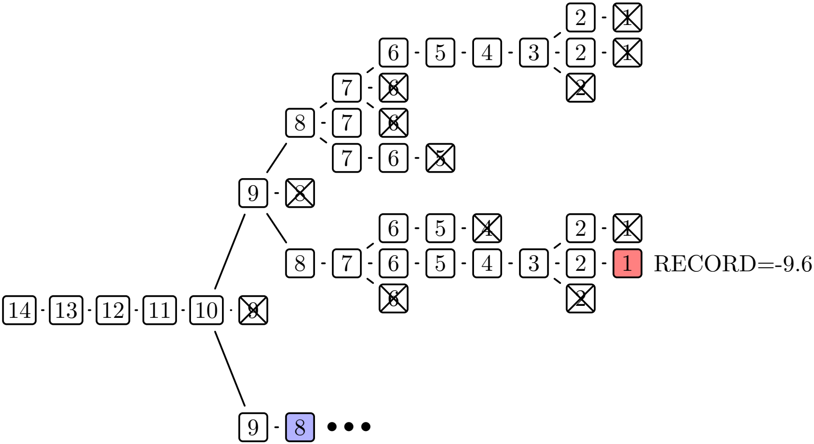
BnB on RLS tree, step 31



BnB on RLS tree, step 33



BnB on RLS tree



After the whole tree is traversed the exact maximum is found!

Non-parametric likelihood bounding

- Idea: augment the partial likelihood bounding function with non-parametric likelihood computed on the "rest of the data"
- Parametric choice probabilities $P_i^k(a|x_\iota;\theta) \to$ freq $n_\iota^{a_i}/n_\iota$

$$\mathcal{L}^{\text{non-par}}(Z^{\mathcal{S}}) = \sum_{\iota\in\mathcal{S}} \sum_{i=1}^{N} \sum_{a} n_\iota^{a_i} \log(n_\iota^{a_i}/n_\iota)$$

- $\mathcal{L}^{\text{non-par}}(Z^{\mathcal{S}})$ depends only on the counts from the data!
- Not hard to show algebraically that for any $Z^{\mathcal{S}}$ ($\approx$ Gibbs ineq)

$$\mathcal{L}^{\text{non-par}}(Z^{\mathcal{S}}) > L^{\text{part}}(Z^{\mathcal{S}}, \theta, P_\theta^k) \quad \forall S$$

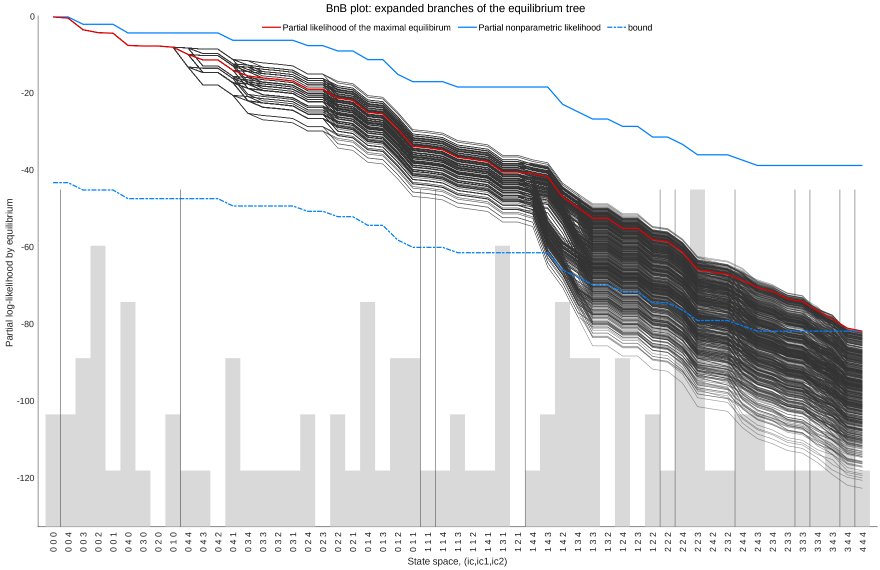
- Therefore partial likelihood can be optimistically extrapolated by empirical likelihood at any step ι of the RLS tree traversal

$$\mathcal{L}^{\text{part}}(Z^{\{S,S-1,\dots,\iota\}}, \theta, P_\theta^k) + \mathcal{L}^{\text{non-par}}(Z^{\{\iota-1,\dots,1\}})$$

- Augmented partial likelihood is more powerful bound for BnB

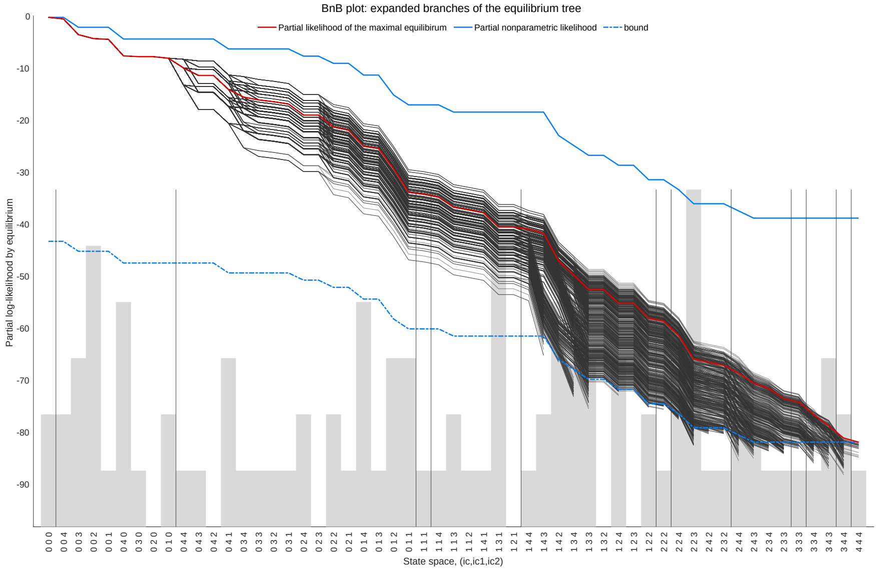
Non-parameteric likelihood bounding

$\iota = S = 14$ (terminal state) on the left, $\iota = 1$ (initial state) on the right



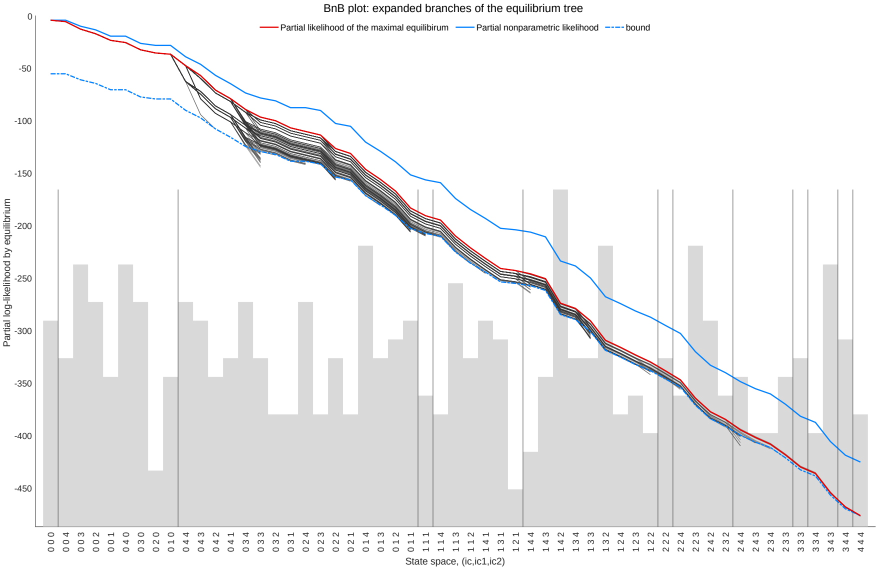
BnB with non-parameteric likelihood bound

Greedy traversal + non-parameteric likelihood bound



BnB with non-parameteric likelihood bound, larger sample

Non-parametric → parametric likelihood as $N \to \infty$ at true $\theta \Rightarrow$ even less computation



BnB refinement with non-parametric likelihood

- For any amount of data the non-parametric likelihood is greater or equal to the parametric likelihood *algebraically*
- BnB augmented with non-parameteric likelihood bound gives sharper Bounding Rules $\rightarrow$ less computation

- Wih more data as $M \rightarrow \infty$
- Non-parametric log-likelihood $\rightarrow$ the true partial likelihood (assuming the model is well-specified and in the neighborhood of the true parameter)
- The width of the band between the blue lines in the plots decreases

  $\rightarrow$ Even sharper Bounding Rules
  $\rightarrow$ Even less computation
  $\rightarrow$ Still delivering exact maximum

MLE for any sample size, but much easier to compute it!

Monte Carlo simulations using leapfrogging model

A

Single equilibrium in the model
One equilibrium in the data

All estimators work well

B

Multiple equilibria in the model
Same equilibrium played the data

C

Multiple equilibria in the model
Multiple equilibria in the data

Most estimators break down
(*work in progress*)

- Bertrand price competition with cost reducing investments
- Number of equilibria ranges from 1 to millions

Iskhakov, Rust and Schjerning (2018) *International Economic Review*

Monte Carlo B, run 2: little multiplicity, unstable

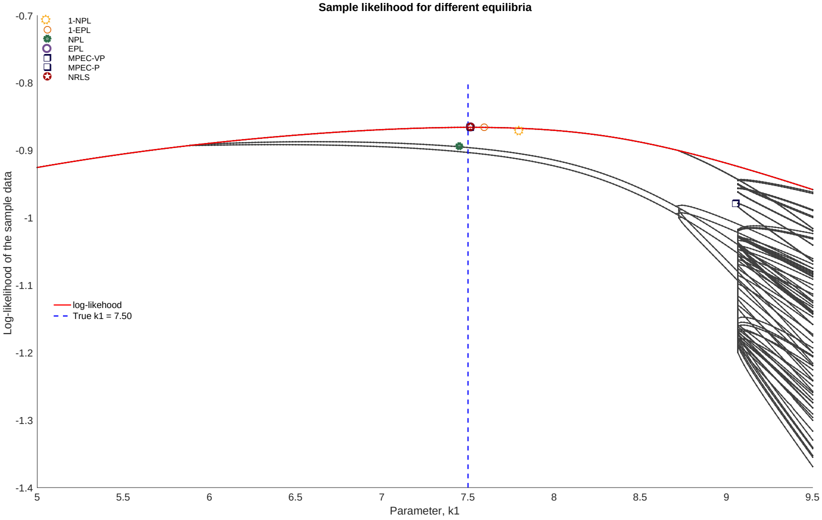
Number of equilibria at true parameter: 3
Number of equilibria in the data: 1, DGP is unstable equilibrium

	2step	NPL	EPL	MPEC-VP	MPEC-P	NRLS
True k1=7.5	7.54238	7.39276	7.48044	7.73133	7.63100	7.50176
Bias	0.04238	-0.10724	-0.01956	0.23133	0.13100	0.00176
MCSD	0.17145	0.05608	0.15801	0.72988	0.89874	0.03820
ave log-like	-0.86834	-0.89374	-0.86550	-0.88512	-0.90196	-0.86504
log-like short	-	-765.242	-11.413	-502.121	-920.643	-0.000
KL divergence	0.02271	0.15996	0.00257	0.11452	0.20182	0.00012
$\|\|P-P_0\|\|$	0.09757	0.20709	0.00619	0.03860	0.02504	0.00307
$\|\|\Psi(P)-P\|\|$	0.160102	0.000000	0.000000	0.000000	0.000000	0.000000
$\|\|\Gamma(v)-v\|\|$	1.126738	0.000000	0.000000	0.000000	0.000000	0.000001
N runs of 100	100	18	100	99	98	100

- Equilibrium conditions satisfied (except 2step)
  (as seen from max differences in values and CCP after one iteration of Γ/Ψ)
- NPL (when converges) and EPL estimators approach MLE
- MPEC converges to *wrong* equilibria
  (as seen from KL divergence from the DGP equilibrium, or CCP differences)
- NRLS is right on the target

Likelihood correspondence

Lines are costructed using symmetric KL-divergence



Monte Carlo B, run 3: discontinuous likelihood

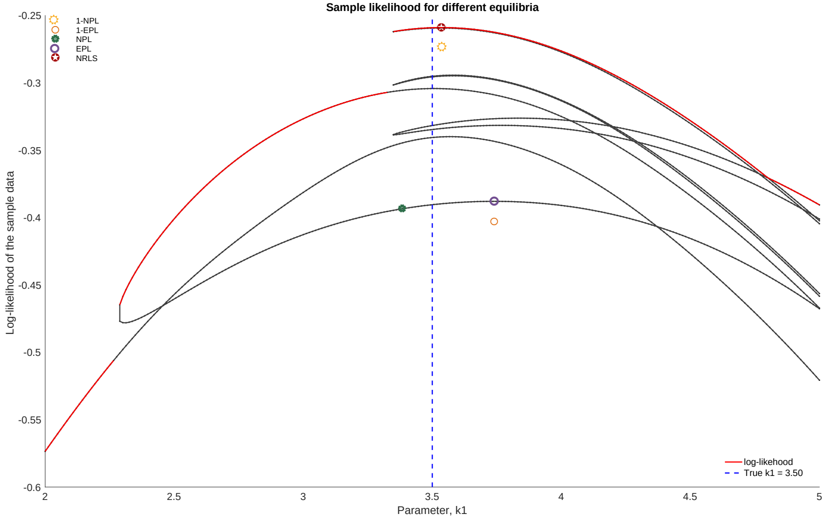
Number of equilibria at true parameter: 9
Number of equilibria in the data: 1
Data generating equilibrium: unstable, near "cliffs"

	2step	NPL	EPL	MPEC-VP	MPEC-P	NRLS
True k1=3.5	3.49739	3.55144	3.64772	3.65943	3.67027	3.50212
Bias	-0.00261	0.05144	0.14772	0.15943	0.17027	0.00212
MCSD	0.13999	0.07133	0.12900	0.12693	0.11583	0.03255
ave log-like	-0.27494	-0.29474	-0.29528	-0.30330	-0.30257	-0.25086
log-like short	-	-219.375	-222.104	-270.999	-267.523	-0.000
KL divergence	0.01512	0.04889	0.04495	0.04102	0.04078	0.00016
$\|\|P - P_0\|\|$	0.62850	0.86124	0.83062	0.66562	0.65879	0.01610
$\|\|\Psi(P) - P\|\|$	0.763764	0.000000	0.000000	0.000000	0.000000	0.000002
$\|\|\Gamma(v) - v\|\|$	0.852850	0.000000	0.000000	0.000000	0.000000	0.000005
N runs of 100	100	100	100	28	27	100

- Poor estimates by all eqb estimators (EPL, NPL and MPEC)
  (constraints are satisfied, yet low likelihood and high KL divergence)
- NRLS is right on the target

Likelihood correspondence

Lines are costructed using symmetric KL-divergence



Monte Carlo B, run 4: massive multiplicity

Number of equilibria at true parameter: 2455
Number of equilibria in the data: 1
Time to enumerate all equilibria (RLS): 10m 39s

	1-NPL	NPL	EPL	NRLS
True k1=3.75	3.70959	3.71272	3.78905	3.74241
Bias	-0.04041	-0.03728	0.03905	-0.00759
MCSD	0.11089	0.06814	0.40716	0.03032
ave log-likelihood	-0.38681557	-0.37348793	-0.45256293	-0.35998461
log-like shortfall	-	-66.529	-467.607	-0.000
KL divergence	Inf	14.07523	12231.59186	0.32429
$\|\|P - P_0\|\|$	0.82204	0.65580	0.79241	0.07454
$\|\|\Psi(P) - P\|\|$	0.963574	0.000000	0.000000	0.000006
$\|\|\Gamma(v) - v\|\|$	7.020899	0.000000	0.000000	0.000008
N runs of 100	100	18	68	100
CPU time	0.159s	11.262s	4.013s	4.731s

- Severe convergence problems for NPL and EPL
- Poor eqb identification (low likelihood and high KL divergence)
- NRLS has comparable CPU time (way faster!)

Monte Carlo C, multiple equilibria in the data

- Assume that the same equilibrium is played in each market over time (large T asymptotics)
- Grouped fixed-effects, groups defined by the equilibria played

1. Joint grouped fixed-effects estimation
   - For each market compute maximum likelihood over all equilibria and "assign" it to the relevant group (estimation+classification)
   - Computationally very demanding: BnB market-by-market, non-parametric refinement has no bite
2. Two-step grouped fixed-effects estimation
   - Step 1: partition the markets based on some observable characteristics (K-means clustering)
   - Step 2: estimate θ allowing different equilibria in different groups
   - Small additional computational cost!

Bonhomme, Manresa (2015); Bonhomme, Lamadon, Manresa (2022)

NRLS estimator for directional dynamic games

Complicated computational task involving maximization over the large finite set of all MPE equilibria $\rightarrow$ branch-and-bound algorithm with refined bounding rule

NRLS nested structure:

1. Each stage game $\rightarrow$ non-linear solver, specific to the model
2. Combining stage game solutions to full game MPEs $\rightarrow$ State Recursion algorithm
3. Solving for all MPE equilibria $\rightarrow$ Recursive Lexicographic Search
4. Structural estimation $\rightarrow$ high-dimensional optimization algorithm

- Implementation of statistically efficient estimator (MLE)
- Using BnB NRLS avoids full enumeration at no cost.
- Computationally trackable, better performance with more data
- Fully robust to multiplicity of equilibria