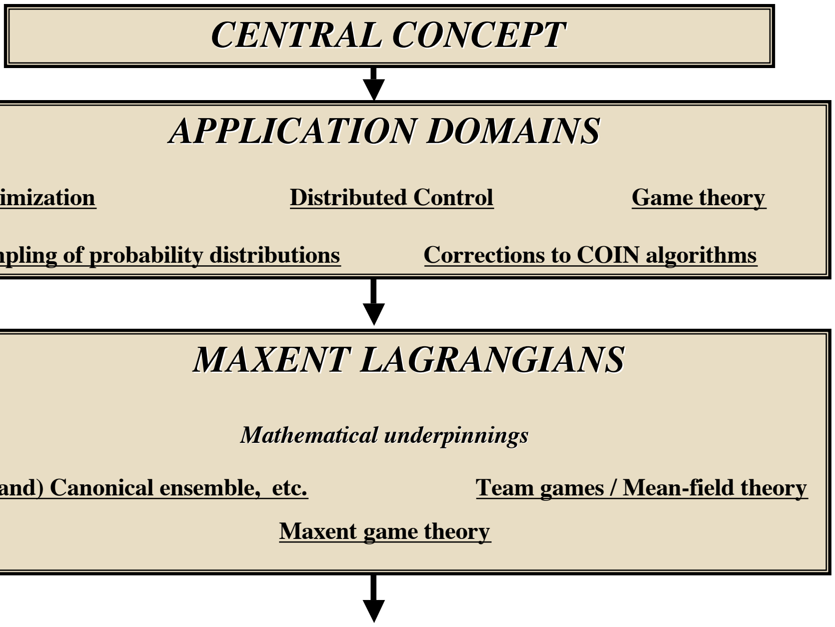
PRODUCT DISTRIBUTION LAGRANGIANS

David H. Wolpert

NASA Ames Research Center

http://ic.arc.nasa.gov/~dhw/

August, 2003



CENTRAL CONCEPT

APPLICATION DOMAINS

Optimization **Distributed Control** **Game theory**

Sampling of probability distributions **Corrections to COIN algorithms**

MAXENT LAGRANGIANS

Mathematical underpinnings

(Grand) Canonical ensemble, etc. **Team games / Mean-field theory**

Maxent game theory

ADAPTIVELY SOLVING FOR PROD. DIST.'S

Brouwer updating **Gradient descent** **Importance sampling**

Adaptive Metropolis-Hastings **Sim. annealing / Prod. Distributions**

POTPOURRI

Coordinate transforms **Continuous spaces** **Constrained optimization**

Black box optimization **Time-extended systems** **Block look-ahead**

Entropy bounds **(un)supervised mach. learning** **Observing agents**

CENTRAL CONCEPT

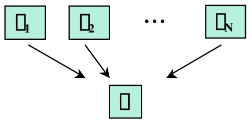
- **A space z □ □**

  z can be *anything*:
  uncountable, symbolic, time-extended, states of human beings, states of computers, mixtures of any of these, etc.

- **N Spaces {x_i □ □$_i$} :**

  x □ □ are the joint elements, $x_{(i)}$ is all {$x_{j \neq i}$}

1 ⇒ ***Need a rule □(x) = z to match any sample x □ □ with a z □ □:***



(Need not be invertible)

This is a *semi-coordinate* system

- **Any distribution P(x) induces a P(z):**

$$\boxed{P(z) = P(\zeta(x)=z) = \int dx\, P(x)\,\delta(\zeta(x)-z)}$$

- **But we don't have P(x); we have N distributions $q_i(x_i \in x_i)$.**

2 ⇒ ***Need a rule*** $\{q_i(x_i)\} \rightarrow P(x)$ ***to get P(z)***

- **For simplicity, choose the product distribution rule:**

$$P(x) = \prod_i q_i(x_i)$$

$$\boxed{\begin{array}{c} q_1(x_1) \times q_2(x_2) \times \ldots \qquad \ldots \times q_N(x_N) \\ \downarrow \\ P(x) \\ \downarrow \\ P(z) = P(\zeta(x) = z) \end{array}}$$

3 ⇒ ***Need a rule to set*** $q \equiv (q_1, q_2 \ldots, q_N)$

I) **Each q_i directly optimizes its own criterion.**

II) **q induces an optimal P(z). E.g.,**

i) Best approximate a provided $P^*(z)$

ii) Best approximate a sample of $P^*(z)$

So each optimal q_i is the vector minimizing the *Lagrangian*

$$L_i(q_i, q_{(i)})$$

subject to q_i being a probability

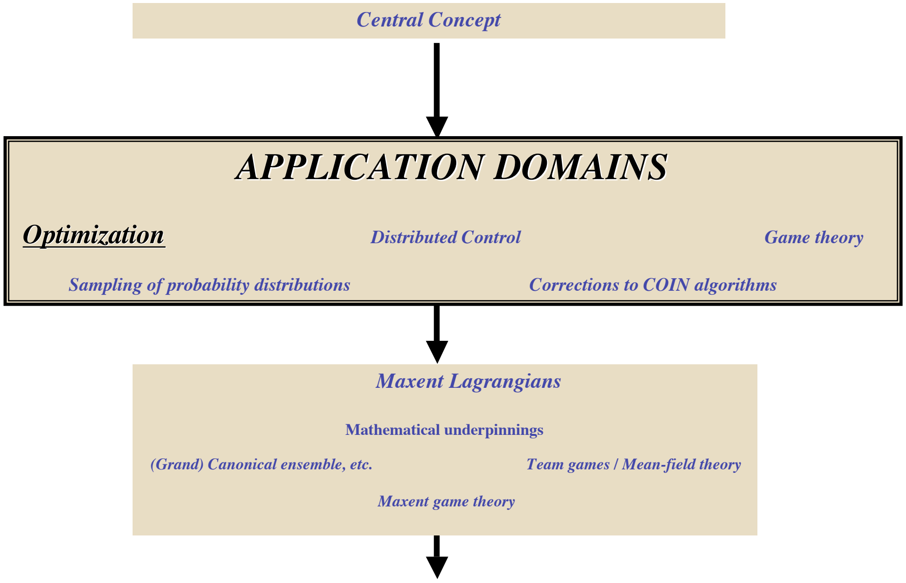
- **q_i may depend on $q_{(i)}$ — but x_i and $x_{(i)}$ are independent**
- **More semi-coordinates allows more accurate approximation**

TAKE-HOME MESSAGE:

Whenever you encounter a distribution P(z) that is difficult to deal with, try expanding it as a product distribution

$\prod_i q_i(x_i)$

with associated Lagrangians.



Central Concept

APPLICATION DOMAINS

Optimization *Distributed Control* *Game theory*

Sampling of probability distributions *Corrections to COIN algorithms*

Maxent Lagrangians

Mathematical underpinnings

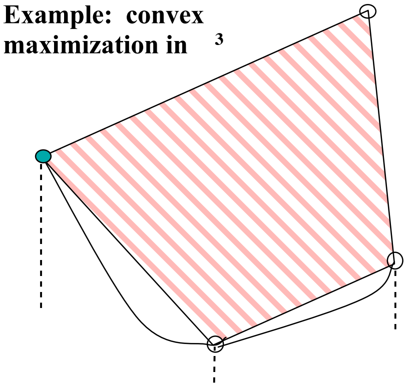
(Grand) Canonical ensemble, etc. *Team games / Mean-field theory*

Maxent game theory

OPTIMIZATION

- *Core issue:* how to use information at one point to choose a next sample point.
- NP hard is when such information is useless.

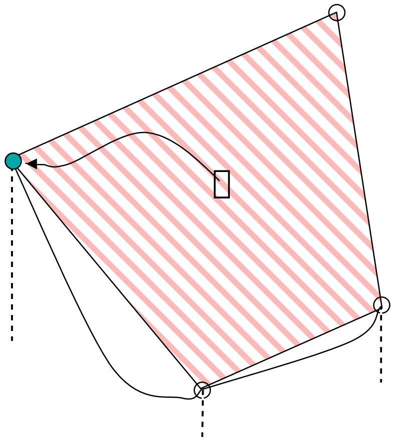
Example: convex maximization in ^{3}



- Why optimization (and therefore control, high-dimension integration, etc.) can be hard

---

- Best case is continuous domains, where smoothness can be exploited — if you aren't trapped in a vertex
- So: Distort problem so solution is off the border, and then weaken the distortion.

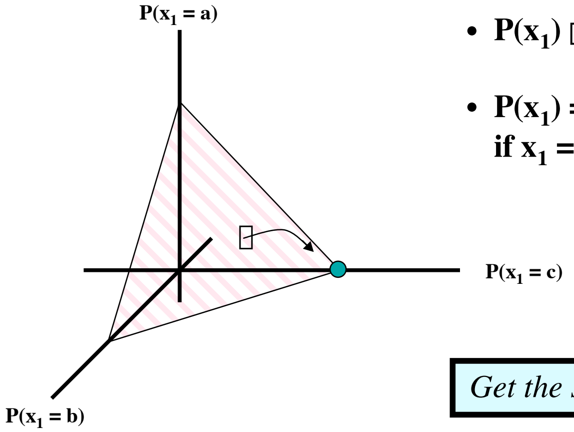


□ — distorted problem solution

● — original problem solution

- Example: Interior point methods

- Can do this for discrete domains by using a probability distribution as the continuous variable



- P(x_1) □ ^{3}
- P(x_1) = □(x_1 - (1, 0, 0)) if x_1 = c exactly

Get the solution off the border

1) For each successive distorted problem, exploit smoothness to search over P(x)'s
   - Gradient descent, Newton's method . . . even simulated annealing.

Gradient descent to optimize <u>categorical</u> variables subject to <u>categorical</u> constraints

2) Example: To minimize G(z), find the P(x) minimizing

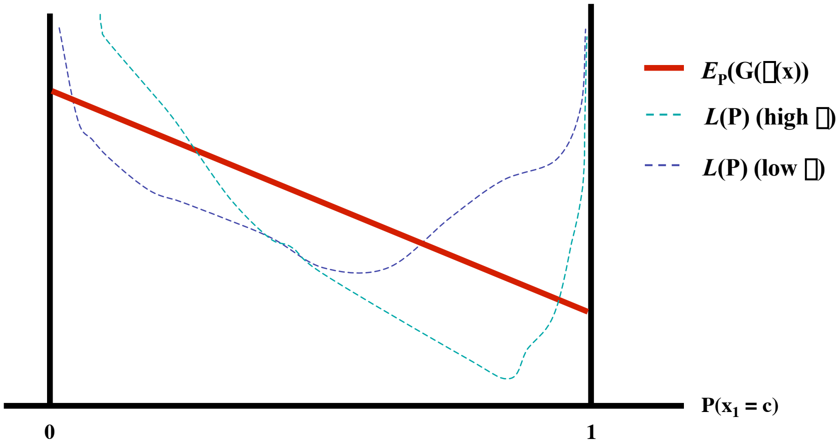
$$L(\mathrm{P}) \;=\; \square E_{\mathrm{P}}(\mathrm{G}(\square(\mathrm{x}))) \;-\; S(\mathrm{P})$$

- *S*(P) has infinite derivative at the simplex border
- Larger □ = less distortion — ***anneal***

E_P(G) = □dx G(□(x))P(x) is linear in P(x). Therefore,

If -S(P) is convex, so is L(P)

So L(P) has a unique minimum, off the border



Example: Take S(P) to be the Shannon entropy,

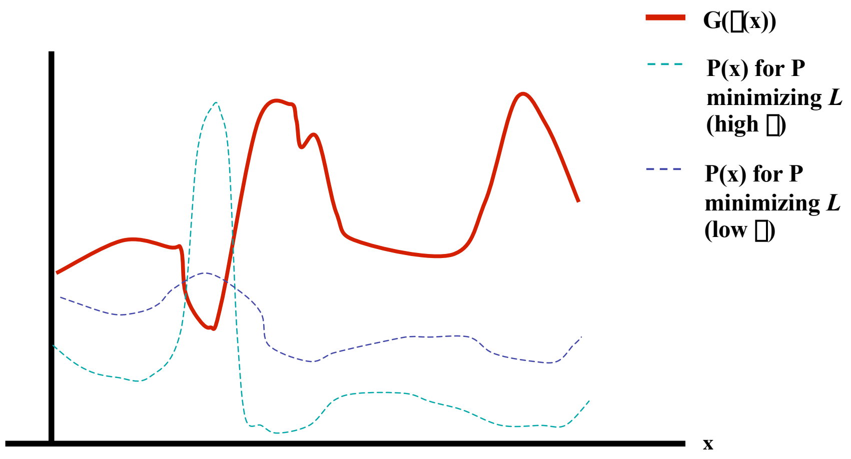
S(P) = - □dx P(x) ln[P(x)]

- **As required, -S(P) is convex, with infinite derivative at the simplex border**

- **L(P) is minimized by the *Boltzmann distribution*,**

P(x) *exp*(-□ G(x))

As □□ , P(x) becomes a delta function about the x minimizing G(x)



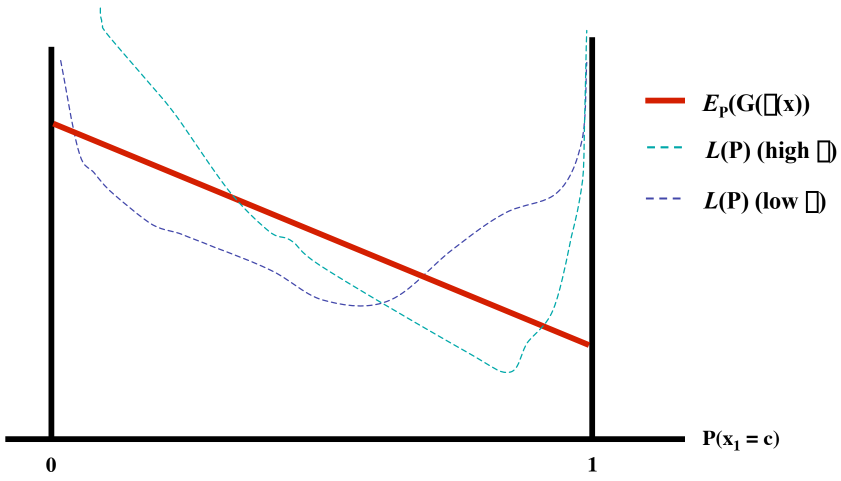
Simulated annealing:

1) At each □, perform an associated Metropolis-Hastings random walk
2) That walk eventually gives a random sample of $P_{□}(x)$
3) When you think it has, increase □, and repeat

So when you get to high □, your sample is likely to be close to argmin G[(□(x))]

. . . **inefficient**

Alternative: Use gradient descent (for example) to find P(x) at each □ :



P(x) lives in a *huge* space. How parameterize it?

With a distributed parameterization, parameters can be estimated separately from each other. So optimization

i) can be *parallelized*,
ii) can be used for *distributed control*,

So . . .

Use a product distribution: $\boxed{P(x) = q(x) = \prod_i q_i(x_i)}$

***Downside*:**

- **L(q) = □E_q(G(□(x))) - S(q)**
  = □dx G(□(x)) □$_i$q$_i$(x) - S(q)

- ***L* is linear in P — but *multilinear* in the q_i**

- **So even for convex S(q), L(q) need not be convex:**

  At any □, L(q) can have multiple minima

- **Even for entropic S,**

  At any □, q(x) can have multiple peaks

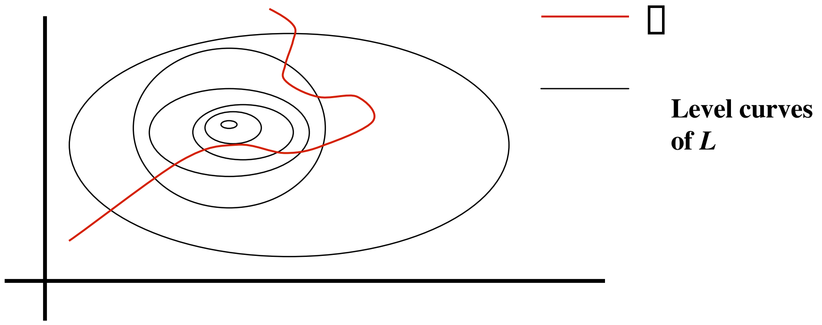
(just like multiple Nash equilibria . . .)

***Intuition*:**

***L* convex over □$^+$, the simplex of all distributions**

***L not* convex over □, the submanifold through □ $^+$ of all product distributions**

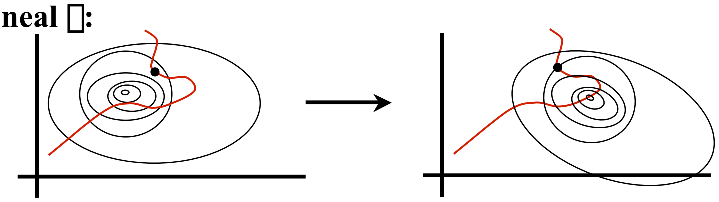
□$^+$:



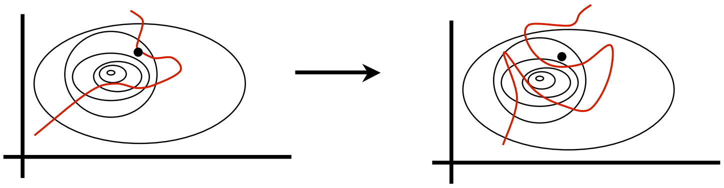
Solutions:

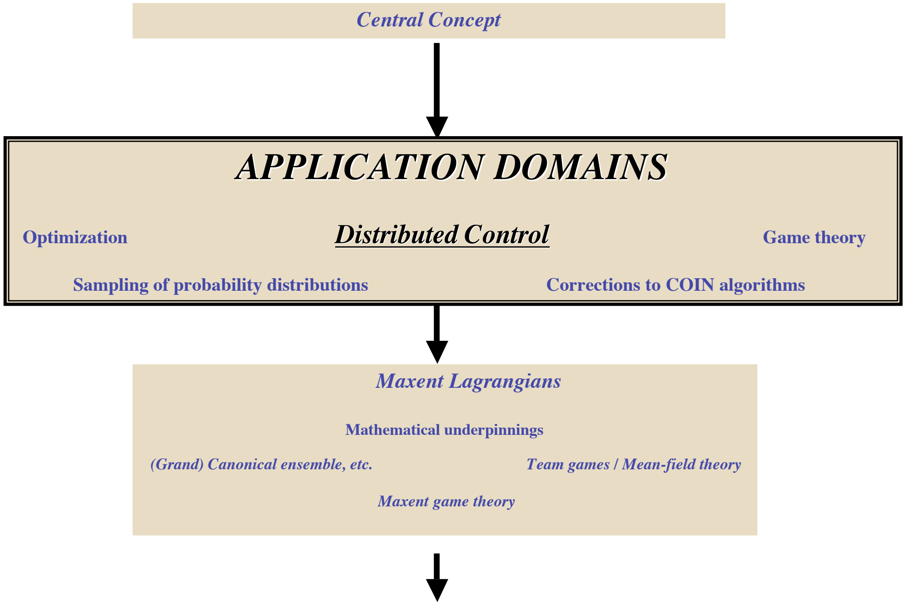
1) If $S(q) = □_i S_i(q_i)$, then for fixed q(i), $L_i(q)$ = is convex in q_i

2) Anneal □:



3) Change coordinates:





Central Concept

APPLICATION DOMAINS

Optimization

Distributed Control

Game theory

Sampling of probability distributions

Corrections to COIN algorithms

Maxent Lagrangians

Mathematical underpinnings

(Grand) Canonical ensemble, etc.

Team games / Mean-field theory

Maxent game theory

DISTRIBUTED CONTROL
(Multi-agent systems)

1) The challenge:

   i) $z = (z_1, z_2)$

   ii) G a function of both z_i

   iii) Can only control z_1 . . .

2) So choose z_1 to maximize $E(G \mid z_1)$, i.e.,

$$\int dz_2\, G(z_1, z_2) P(z_2 \mid z_1)$$

3) Want each control variable □ z_1 set autonomously

1) "Just" optimization;
   Basis of conventional control theory

2) For our desired distributed solution, use a product distribution approach instead of control theory?

3) Two major problems:

   i) In naive prod. distribution optimization you set all q_i - here you can't set q_2.

   ii) P(z) is explicitly *not* a product distribution.

Solution:

Puppet master moves sticks q_i, which move strings $P(z_2 \mid z_1)$, which move puppet, *expected G*

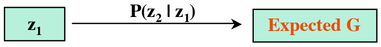
Formally,

1) x = the control variables, z_1

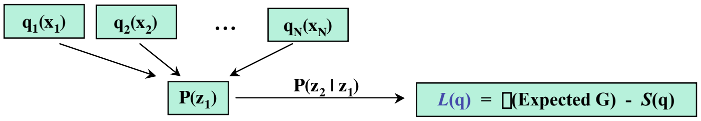
$$\text{So } E_q(G \mid x) = \int dx\, q(x)\, E(G \mid z_1 = x)$$
$$= \int dx_i\, q(x) \int d(z_2) G(z_1, z_2) P(z_2 \mid z_1 = x)$$
$$= \int dx_i \prod_i q_i(x_i) \int d(z_2)\, G(z_1, z_2) P(z_2 \mid z_1 = x)$$

2) Get off the border: $L(q) = \beta E_q(G \mid x) - S(q)$

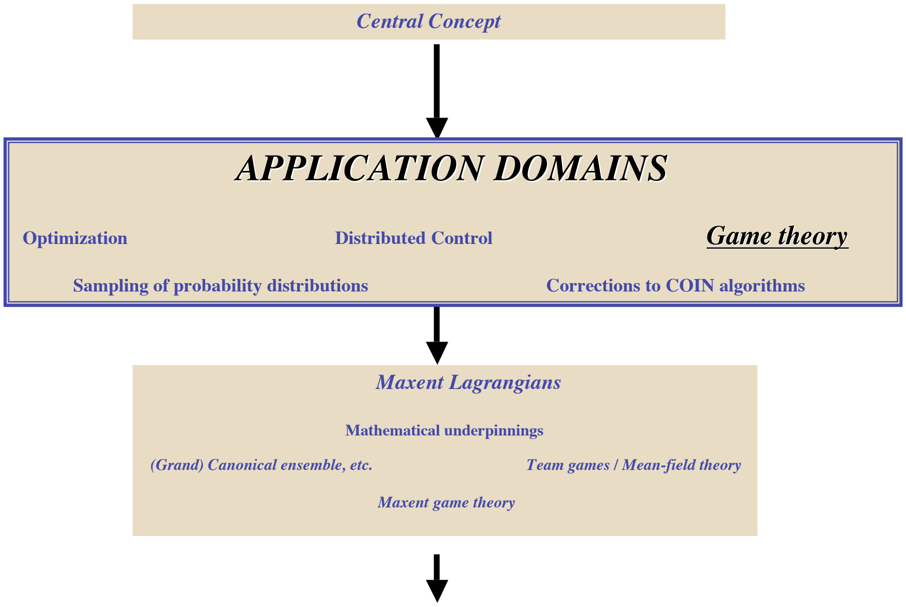
Overview:



z_1 minimizing Expected G hard to find.
So use a product distribution, and get off the border:



Find q(x) minimizing $L(q)$ (easy), and then raise β.



Central Concept

APPLICATION DOMAINS

Optimization

Distributed Control

Game theory

Sampling of probability distributions

Corrections to COIN algorithms

Maxent Lagrangians

Mathematical underpinnings

(Grand) Canonical ensemble, etc.

Team games / Mean-field theory

Maxent game theory

NONCOOPERATIVE GAME THEORY

1) A set of N *players*, each choosing a *pure strategy*, z_i □ □$_i$

2) A set of N *payoff functions* $h_i(z)$

3) z is a *Nash equilibrium* iff for all players i,

   for all z'_i, $h_i(z'_i, z(i)) \leq h_i(z_i, z(i))$

Example: Prisoner's dilemma payoff table ($h_1(z)$, $h_2(z)$)

	Player 1's move	
Player 2's move:	(2, 2)	(10, 0)
	(0, 10)	(7, 7)

- ***Problem***:

  Some games have no Nash equilbrium

- ***Solution***:

  i) Players take *mixed strategies* $P_i(z_i)$;

  ii) $\prod_i P_i(z_i)$ a Nash equilibrium iff for all players i, no change to $P_i(z_i)$ will increase $\int dz\ h_i(z) \prod_i P_i(z_i)$

. . . gee, a product distribution . . .

- **Nash used Brouwer's fixed point theorem to prove always exists a mixed strategy Nash equilibrium**

. . . gee, "Brouwer" is the name of a rule for setting product distributions . . .

- ***Unresolved problems***:

  1) Finding Nash equilibria is a (hard) multi-criteria optimization problem

  2) In real world, never at a Nash equilibrium, due to limited computational power, if nothing else.

Bounded rationality

- **Attempts to date to solve (2) are just more elaborate models of (human) players**

  - Underlying problem is arbitrariness of the models.

Alternative:

1) For now, take □ = □, and define $g_i(x) \equiv -h_i(□(x))$

2) At Nash equilibrium, each q_i minimizes

$$\boxed{L_i(q) \; = \; E_{q_i}(g_i \mid q_{(i)})}$$

$$= \; □dx \; g_i(x) \; □_j \; q_j(x_j)$$

3) Allow broader class of Lagrangians.
E.g., each q_i minimizes

$$\boxed{L_i(q) \; = \; □E_{q_i}(g_i \mid q_{(i)}) \; - \; S(q)}$$

4) □ <   is bounded rationality

1) *S*(q) can be set from first principles (e.g., using information theory)

2) *S*(q) can be set to enforce a particular model of rationality

3) Can also set the model of rationality by replacing the g_i term in L_i . E.g.,

$$\boxed{-g_i(x) \; = \; h_i(□(x)) \; - \; [h_i(□(x))]^2}$$

penalizes q_i for which the r.v. $h_i(□(x))$ has large variance.

4) Alternativley, replacing g_i with

$$\boxed{g_i(x) \; = \; □_j \; f_{i,j}(x)}$$

is equivalent to having player i try to optimize several payoff functions at once

1) If S(q) has infinite derivatives at □'s border, the optimal q for □< is off that border — and usually easier to find

2) If in addition $S(q) = \square_i \square dx_i S_i(q_i(x_i))$ and S_i is bounded below, minimizing $L_i(q)$ is conventional (full rationality) game theory — just with a new payoff function,

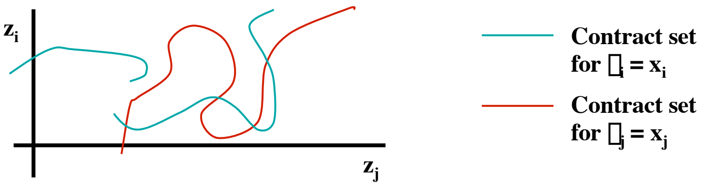
$$\boxed{f_i(x, q) = \square g_i(x) - S_i(q_i(x_i)) / q_i(x_i)}$$

So $- S_i(q_i(x_i)) / \square q_i(x_i)$ is a preference ordering for (the difficulty of) the computation of $q_i(x_i)$

1) If □ ≠ □, every $x_i \square \square_i$ delineates a set of binding contracts among the players — a set of z — that coordinate i "offers":

$$\boxed{\square_i = x_i \;\square\; z \;\square\; \square_{x(i)} \square(x_i, x_{(i)})}$$

2) The contract finally accepted — the value of z — is the intersection of the contract sets offered by all players


z_i
z_j
Contract set for $\square_i = x_i$
Contract set for $\square_j = x_j$

In addition, if □ ≠ □, the strategies of the players are no longer independent:

$$\boxed{P(z_i, z_j) = \square dx \; \square_k q_k(x_k) \; \square(\square_i(x) - z_i) \; \square(\square_j(x) - z_j)}$$

- **So player i's strategy choice affects the strategy choice of player j**

1) **Stochastic dependence, but not necessarily Bayes-optimality (as in correlated equilibria)**

2) **If z is interpreted as the final joint action in a multi-stage game, this gives Stackelberg games, signalling, etc.**

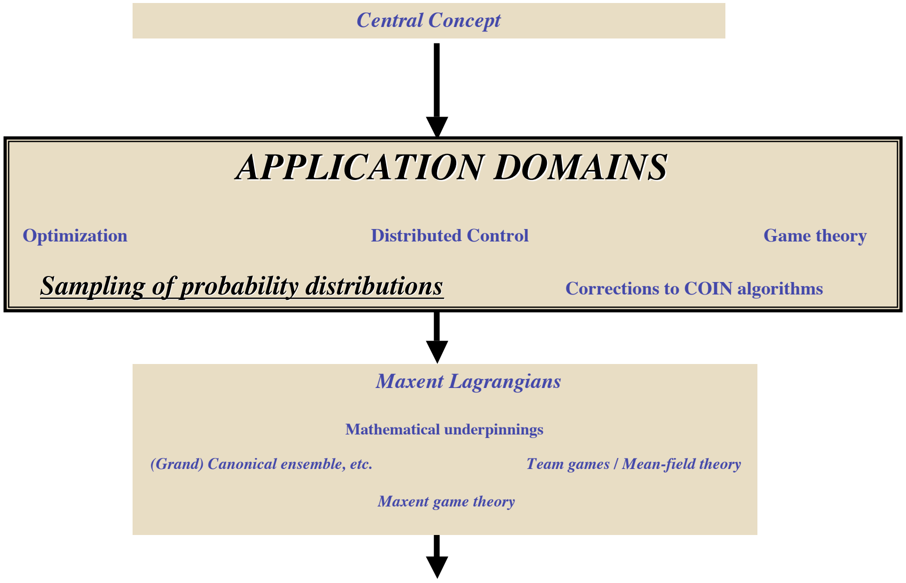
1) **In a *team game*, all g_i are the same function, the *world utility*, G**

   E.g., $G(x) = \square_i \, h_i(\square(x))$

2) **For *S*(P) concave with infinite derivative at □'s border, $L(P) = \square E_P(G(\square(x))) - S(P)$ is a convex surface with a single global minimum:**

   - **One and only one solution**
   - **The solution is easy to find**

3) **This optimal G is not a product distribution in general, i.e., it couples the players, regardless of whether □ = □**



Central Concept

APPLICATION DOMAINS

Optimization

Distributed Control

Game theory

Sampling of probability distributions

Corrections to COIN algorithms

Maxent Lagrangians

Mathematical underpinnings

(Grand) Canonical ensemble, etc.

Team games / Mean-field theory

Maxent game theory

SAMPLING PROBABILITY DISTRIBUTIONS

- **Say you want to evaluate a high-dimensional integral**

$$\boxed{\int dz\ f(z)\ p(z)}$$

where p(z) is a probability distribution

- **A very common problem, e.g., in Bayesian analysis, materials science, physics, chemistry, etc.**
- **In Monte Carlo algorithms, one does this by repeatedly sampling p(z), and averaging the associated values of f(z)**
- **But how do you sample p(z)?**

1) Perform a guided random walk through □

- **i) *Metropolis Hastings* (MH) algorithm — the basis of simulated annealing**
- **ii) Only exactly correct asymptotically**

2) Approximate p(z) with a product distribution q and sample q directly

- **i) No wait for asymptotia**
- **ii) There are two primary approximation error measures: *forward KL* and *backward KL***
- **iii) They give different Lagrangians, and so different algorithms for estimating optimal q**
- **iv) Associated integration errors may be correctable with *importance sampling***

Hybrid combinations of (1) and (2):

I) MH uses a distribution R to set the walk's initial z

II) MH uses a *proposal distribution* Q after that:

- **i) Q gives the "exploration" point z^e found from the current point z^t**
- **ii) z^t becomes z^e always if $p(z^e) > p(z^t)$**
- **iii) else z^t becomes z^e with probability $p(z^e)Q(z^t) / p(z^t)Q(z^e)$**

Either R and/or Q can be set to
the q found via either inverse KL and/or forward KL

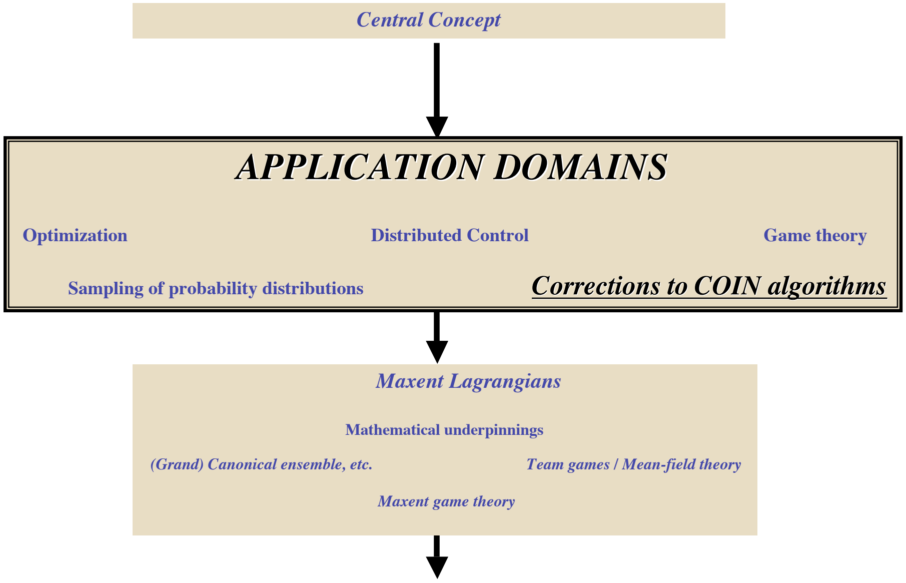
Hybrid combinations of (1) and (2):

I) MH's walk gives a sample D of p;
D can be used to estimate the q that best approximates p

- **Can be used for either the approximation error of inverse KL q or of forward KL q**
- **Can then sample from this q (not the same as re-sampling from D)**

II) In *adaptive MH*, (I) is done repeatedly;

- **Each time the new q is used to modify Q**
- **Crucial that the modification is Markovian**



Central Concept

APPLICATION DOMAINS

Optimization

Distributed Control

Game theory

Sampling of probability distributions

Corrections to COIN algorithms

Maxent Lagrangians

Mathematical underpinnings

(Grand) Canonical ensemble, etc.

Team games / Mean-field theory

Maxent game theory

CORRECTIONS TO COIN ALGORITHMS

1) In optimization and sampling, calculating the optimal $\{q_i\}$ usually intractable.

   The $\{q_i\}$ must be set adaptively

2) In control, often don’t even know what to calculate (can’t accurately model the system) . . .

   Agents — the $\{q_i\}$ — must be set adaptively

3) Control should be robust against failures/noise, and if distributed have few communication requirements . . .

   The $\{q_i\}$ *must* be set adaptively

- A *collective* is

  i) A set of agents $\{i\}$, each of which

  ii) tries to make the move x_i that maximizes an associated *private utility* function $g_i(x)$,

  iii) together with a *world utility* $G(x)$ measuring the performance of the overall system

- The probability distribution across G values is set by

  i) how “aligned” each g_i is with G; does replacing $(x_i, x_{(i)})$ □ $(x'_i, x_{(i)})$ improve g_i iff it improve G?

  ii) the size of the “signal” of the change in g_i under $(x_i, x_{(i)})$ □ $(x'_i, x_{(i)})$ in comparison to the “noise” of the change under $(x_i, x_{(i)})$ □ $(x_i, x'_{(i)})$

- **In COllective INtellience (COIN) experiments, at each iteration the simplest common machine learning algorithm was used by each i to choose x_i:**

  i) For each x_i □ $□_i$, estimate $g_i(x_i, x_{(i)})$ by averaging the g_i values in previous iterations in which $□_i = x_i$

  ii) To trade off "exploration vs. exploitation", choose among the x_i according to a Boltzmann distribution over those estimated g_i values

Product distribution theory provides an alternative perspective:

Rather than "trying to maximize g_i" by "trading off exploration and exploitation", the algorithms "try to find a bounded rational equilibrium"

1) **Previous work based on a set of mathematical premises expected to hold for any learning algorithm**
2) **Using those can solve for the g_i of a particular form that are aligned with G and have best signal / noise:**

   $$AU_i(x) \equiv G(x) - □dx'_i\, f(x'_i)\, G(x'_i, x_{(i)})$$

   for a distribution f(.)
3) **Usually arbitrarily chose f(.) to be uniform**

Product distribution theory says what f(.) should be

- uniform is not correct

1) Computer experiments compared g_i = AU_i and the team game g_i = G

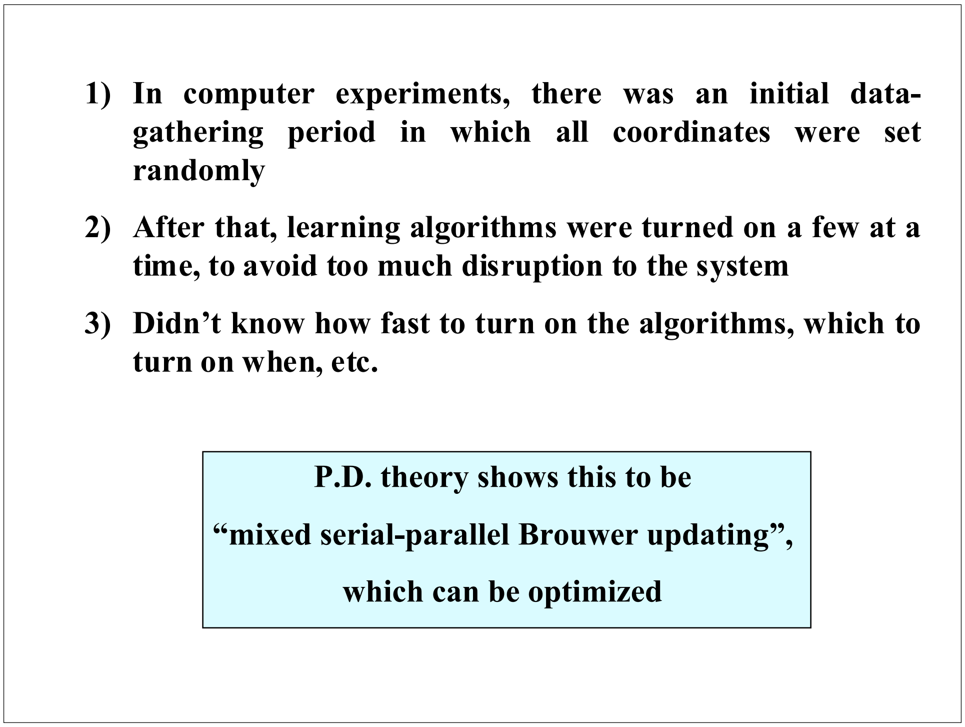
   It was found that when they shared the same temperature, for some temperature ranges the team game outperformed AU

2) No understanding of how to avoid this without modifying AU’s temperature

P.D. theory shows that this phenomenon is due to a biased estimator of the Boltzmann exponentials

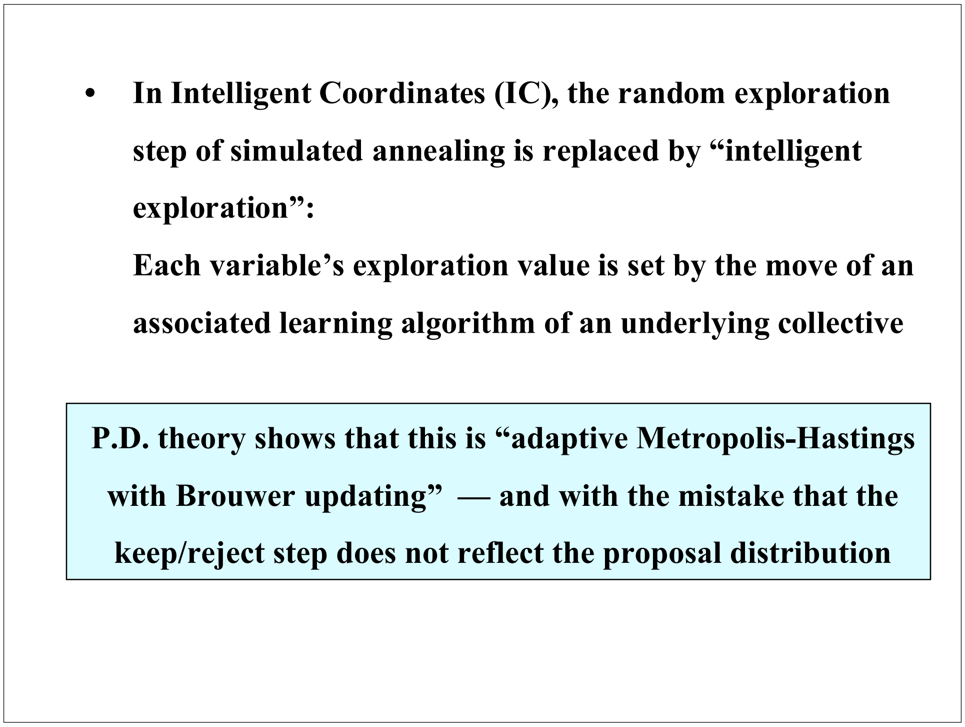
1) A problem with AU_i is that it requires evaluating G for counter-factual x_i values
2) A partial solution is to approximate $f(x_i)$ = □(x_i, CL_i) for some “clamping parameter” CL_i.
3) This defines the private utility WLU_i
4) Didn’t know how to choose CL_i in practice (intuition usually used)

P.D. theory says what CL_i should be to best approximate the correct AU



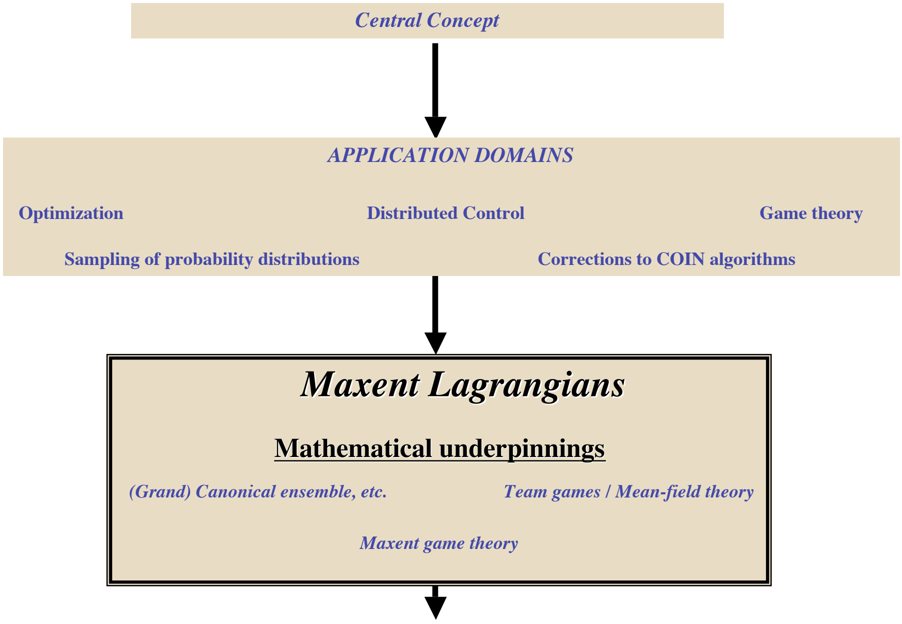
1) In computer experiments, there was an initial data-gathering period in which all coordinates were set randomly
2) After that, learning algorithms were turned on a few at a time, to avoid too much disruption to the system
3) Didn't know how fast to turn on the algorithms, which to turn on when, etc.

P.D. theory shows this to be
"mixed serial-parallel Brouwer updating",
which can be optimized



- In Intelligent Coordinates (IC), the random exploration step of simulated annealing is replaced by "intelligent exploration":
  Each variable's exploration value is set by the move of an associated learning algorithm of an underlying collective

P.D. theory shows that this is "adaptive Metropolis-Hastings with Brouwer updating" — and with the mistake that the keep/reject step does not reflect the proposal distribution



Central Concept

APPLICATION DOMAINS

Optimization

Distributed Control

Game theory

Sampling of probability distributions

Corrections to COIN algorithms

Maxent Lagrangians

Mathematical underpinnings

(Grand) Canonical ensemble, etc.

Team games / Mean-field theory

Maxent game theory

MATHEMATICAL FOUNDATIONS

1) We want to formalize how "surprised" you are if you observe a value *s* generated from a distribution P(*s*)

2) We want the surprise at seeing the IID pair (*s*, *s*′) to equal the sum of the surprises for *s* and for *s'*

3) This means surprise(*s*) = -ln[P(*s*)]

4) So expected surprise is the *Shannon entropy*

$$\boxed{S(\mathrm{p}) \equiv -\sum_s \mathrm{P}(s)\ln[\mathrm{P}(s)]}$$

- Shannon entropy is concave over P
- *Information* in P is what's left over after surprise: -*S*(P)

Maxent: Given only constraints $\{E(\mathrm{g}_i) = 0\}$, choose minimal information P consistent with those constraints

1) **We want to formalize "how far apart" P_1 and P_2 are**

2) **Generate m unordered data D by IID sampling P_1, then misassigning to each $d_i \in D$ the probability $P_2(d_i)$**

3) **So you assign to all of D the *likelihood* $\prod_{i \leq m} P_2(d_i)\, C(D)$ where C(D) is the multinomial counting factor**

4) **Take log of this and divide by m, to get "likelihood rate". As $m \to \infty$, with $S(P \,||\, P') \equiv -\sum_s P(s)\ln[P'(s)]$, the rate is the *Kullback-Leibler* distance**

$$\boxed{KL(P_1 \,||\, P_2) \equiv S(P_1 \,||\, P_2) - S(P_1 \,||\, P_1)}$$

- **$KL(P_1 \,||\, P_2)$ is never negative, and equals 0 iff $P_1 = P_2$**

- **We want to minimize a smooth function $f(s \in \mathbb{R}^n)$ subject to K constraints $\{g_i(s) = 0\}$**

- **Define** $\boxed{L(f, \{g_i\})(s) \equiv f(s) + \sum_i \lambda_i g_i(s)}$

- ***L* is the *Lagrangian*, and the $\{\lambda_i\}$ the Lagrange parameters**

- **Set the partial derivatives of *L* with respect to both s and the Lagrange parameters to 0. Voila.**

Example: Each $g_i(s)$ forces a different subset of s's components to sum to 1, i.e., to be a probability distribution.

- **Convex f enforces non-negativity.**

Brouwer's fixed point theorem **:**

- **Let f(s) be a smooth map from V, into V, where V is a bounded convex connected subset of ^{n}**
- **Then there exists s such that s = f(s)**

1) Both □ and □$^{+}$ are bonded convex connected subsets of ^{n} So any smooth map over them has a fixed point

2) In particular, if the Lagrange minimization problem gives q = f(q) for a smooth f(.), then the problem has a solution

- **q □ f(q) is a *Brouwer update* of q**

Problem: How to express arbitrary P(z) with a prod. dist.?

Solution:

□ = □ won't work . . . so introduce more semi-coordinates

Example:

1) i) z = (z_1, z_2)

ii) |□$_i$| possible values of each z_i

2) i) Have □$_1$ = □$_1$ — the value of x_1 tells you z_1

ii) Have an extra □$_i$ for each possible value of z_1; x_{z_1} says what value z_2 has when □$_1$ = z_1

Formally,

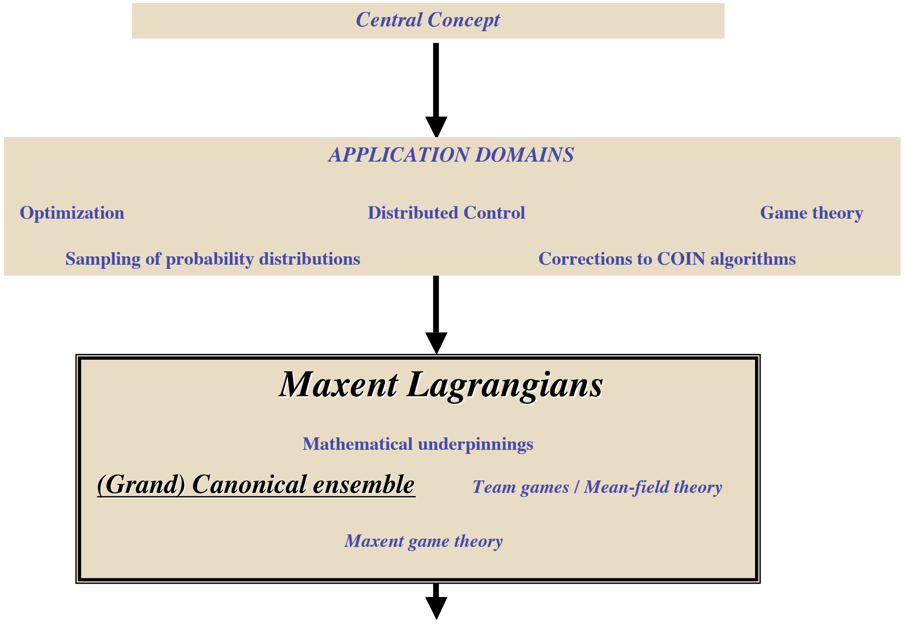
$$z_1 = \square_1(x_1, x_{z'_1}, x_{z''_1}, \ldots, x_{|\square_1|+1}) = x_1$$
$$z_2 = \square_2(x_1, x_{z'_1}, x_{z''_1}, \ldots, x_{|\square_1|+1}) = x_{z_1} = x_{x_1}$$

So

$$P(z_1) = P(x_1) = q_1(x_1)$$

$$P(z_2 \mid z_1) = P(x_{z_1} = z_2 \mid x_1 = z_1) = q_{z_1}(x_{z_1})$$

***Representation theorem*: For any P(z), there exists a coordinate system □(.) and product distribution q such that q induces P**



Central Concept

APPLICATION DOMAINS

Optimization

Distributed Control

Game theory

Sampling of probability distributions

Corrections to COIN algorithms

Maxent Lagrangians

Mathematical underpinnings

(Grand) Canonical ensemble

Team games / Mean-field theory

Maxent game theory

(GRAND) CANONICAL ENSEMBLE

1) Consider the Lagrangian $L_i(q) = \square E_{q_i}(g_i \mid q_{(i)}) - S(q)$ where S is Shannon entropy

2) This L_i minimizes $KL(q \parallel p^{\square g_i})$, where $p^{\square g_i}$ is the exact *Canonical ensemble*

3) Its optimizing q_i is

$$q_i^{g_i}(x_i) \quad e^{\square\square[g_i]_{i,q^{g_i}}(x_i)}$$

where as before $[g_i]_{i,q}(x_i)$ is expected g_i conditioned on x_i, when other coordinates are distributed according to $q_{(i)}$

Each "particle" i coupled to its own distinct "heat bath", i.e., a mean field approximation

---

1) Now have each $g_i(x) = G(x) + \square_i h_i(x)$, where the $\{h_i\}$ are all integer-valued functions

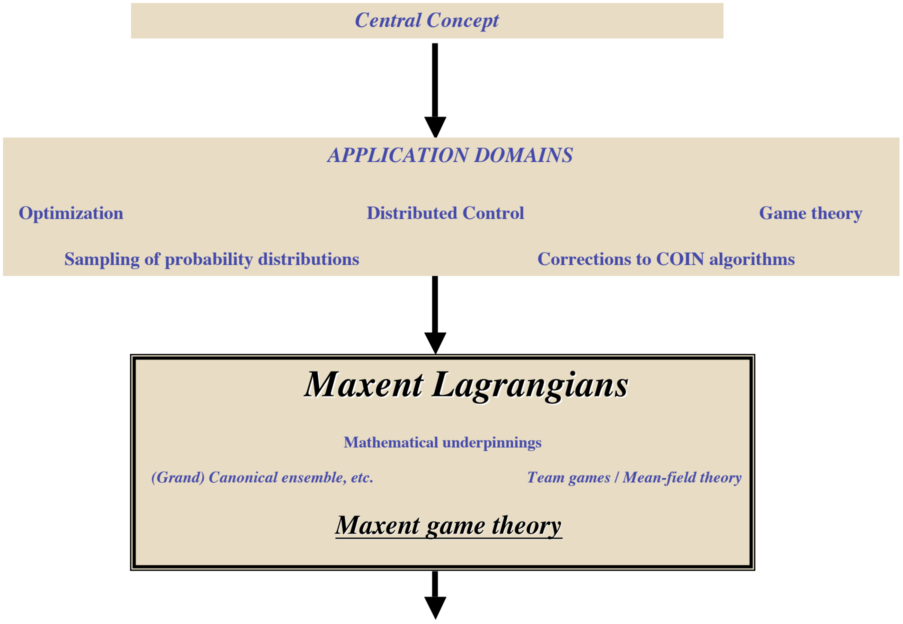
2) Then the *L*-minimizing P is the *Grand canonical ensemble*, and the minimizing q is a mean field approximation to it

- x_i encodes the state of all particles of type i
- $h_i(x)$ is the chemical potential of particles of type i multiplied by their number — which is allowed to vary

- If we minimize $KL(p^{\square g_i} \parallel q)$ instead, we get the marginal,

$$q_i(x_i) = p^{\square g_i}(x_i)$$

- Unlike $q_i^{g_i}$, this *inverse KL* q is independent of q(i)
- Can calculate it through importance sampling



Central Concept

APPLICATION DOMAINS

Optimization **Distributed Control** **Game theory**

Sampling of probability distributions **Corrections to COIN algorithms**

Maxent Lagrangians

Mathematical underpinnings

(Grand) Canonical ensemble, etc. ***Team games / Mean-field theory***

Maxent game theory

MAXENT GAME THEORY

Consider bounded rational game theory with Lagrangians $L_i(q) \;=\; \square E_{q_i}(g_i \mid q_{(i)}) - S(q)$ where S is Shannon entropy.

1) **This Lagrangian arises if each player (chooses its mixed strategy to) maximize its entropy, subject to a provided expected payoff and the other players' mixed strategies.**

2) **Alternatively, it arises if each player maximizes its expected payoff, subject to a provided entropy.**

All mathematical machinery of statistical physics can be applied to bounded rational game theory

1) Want a measure of "how rational" q_i is

- Can't use $E_{q_i}(g_i)$ — it depends on $q_{(i)}$

2) A *rationality function* $R(U, q_i)$ measures how peaked q_i is about $\text{argmin}_{x_i} U(x_i)$ for any function U

i) Rationality is the inverse temperature if q_i is a Boltzmann distribution in U:

$$\boxed{R(U, q_i) = \beta \text{ if } q_i \propto exp\{-\beta U\}}$$

ii) Maximizing entropy subject to a rationality value gives a Boltzmann distribution at that temperature:

Of all q_i such that $R(U, q_i) = \beta'$, the one with maximal entropy is $q_i \propto exp\{-\beta' U\}$

3) We are interested in $U(x_i)$ that measure expected payoff to i if it makes move x_i. So for any function V(x), define

$$\boxed{[V]_{i,q}(x_i) = E_{q_{(i)}}(V(x_i)) \equiv \int dx_{(i)} V(x_i, x_{(i)})\, q_{(i)}(x_{(i)})}$$

4) $\boxed{R([g_i]_{i,q}, q_i)}$ is our measure of "how rational" q_i is.

5) Intuitively, it is the inverse temperature of the distribution over i's expected payoffs when it chooses moves according to q_i.

- **The optimal q, given rationalities $\{\square^*_i\}$, is the minimizer over q and the $\{\square_i\}$ of**

$$\boxed{L(q, \square) \;\equiv\; \square_i \square_i[R([g_i]_{i,q}, q_i) - \square^*_i] \;-\; S(q)}$$

- **At any local minimum of *L*(q, □), for all i,**

$$\boxed{q_i \quad exp\{-\square^*_i[g_i]_{i,q}\}}$$

***Proof*: i) The Lagrange parameter term forces any local minimum to obey $R([g_i]_{i,q}, q_i) = \square^*_i$ for all i.**

ii) The q_i maximizing entropy while obeying $R([g_i]_{i,q}, q_i) = \square^*_i$ is the Boltzmann distribution. QED

The maxent q is the minimal information q that is consistent with specified player rationalities

- **Finding the Nash equilibria of a non-team game is typically viewed as a multi-criteria optimization problem**

- **Finding the bounded rational equilibria is a single-criteria optimization problem:**

  Minimize *L*(q, □)

- **All solutions to this problem are off □'s border, and therefore easy to find**

Example: Rationality is the inverse temperature of that Boltzmann distribution that best fits q_i:

$$\boldsymbol{R}(U, q_i) \;=\; \text{argmin}_\beta \,[\, KL(q_i \,||\, \mathit{exp}\{-\beta U\} \,/\, N(\beta U)\}) \,]$$

***Must establish both requirements of a rationality function are met*:**

1) KL distance is non-negative, equalling zero only if its arguments are equal.

If $q_i = \mathit{exp}\{-\beta' U\} / N(\beta' U)\}$, taking $\beta = \beta'$ gives a KL distance of 0.

So the rationality of this q_i is 0, as required.

2) i) Writing it out,

$$\boldsymbol{R}(U, q_i) \;=\; \text{argmin}_\beta \,[\, \beta E_{q_i}[U(x_i)] \;+\; \ln(N(\beta U) \,]$$

ii) So $E_{q_i}[U(x_i)] \;=\; -\partial_\beta \ln(N(\beta U))|_{\beta = R(U,\, q_i)}$

iii) So all q_i with rationality β^* have the same $E_{q_i}[U(x_i)]$

iv) Therefore of all q_i with rationality β^*, the one with the maximal entropy is the Boltzmann distribution with that inverse temperature. QED

In practice, replacing the rationality constraint term in $L(q, \beta)$ with an expected utility constraint may be easier

The grand canonical ensemble can model bounded rational games in which the number of actors varies.

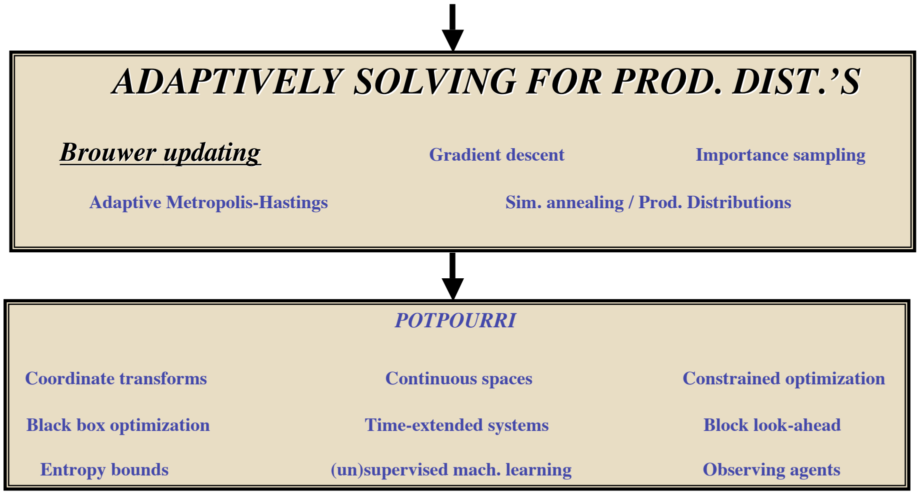
***Intuition*: Actors have “types”, just like particles have properties**

Example 1 (microeconomics):

i) A set of bounded rational companies,
ii) with payoff functions given by market valuations,
iii) each of which must decide how many employees of various types to have.

Example 2 (evolutionary game theory):

i) A set of species,
ii) with payoff functions given by fractions of total resources they consume,
iii) each of which must “decide” how many phenotypes of various types to express.



ADAPTIVELY SOLVING FOR PROD. DIST.’S

Brouwer updating

Gradient descent

Importance sampling

Adaptive Metropolis-Hastings

Sim. annealing / Prod. Distributions

POTPOURRI

Coordinate transforms

Continuous spaces

Constrained optimization

Black box optimization

Time-extended systems

Block look-ahead

Entropy bounds

(un)supervised mach. learning

Observing agents

BROUWER UPDATING

1) To “set q_i adaptively” means iteratively trying to minimize $L(q_i, q_{(i)})$, given partial information about $q_{(i)}$.

2) As an example, consider again the Lagrangian

$$L_i(q) \;=\; \beta E_q(g_i(\zeta(x))) \;-\; S(q)$$

3) Say $S(q) = \sum_i S_i(q_i)$

So S is linear in the coordinates . . .

3) E.g., recall that since q is a product distribution, such linearity holds when S is the entropy,

$$S(q) \;=\; -\int dx\, q(x) \ln[q(x)] \;=\; -\sum_i \int dx_i\, q(x_i) \ln[q(x_i)]$$

4) For any such linear S, L is linear:

$$L(q) \;= \sum_i \Big(\int dx_i\, q_i(x_i)\, [g_i]_{i,q}(x_i) \;-\; S_i(q_i) \Big)$$

where as before, $[g_i]_{i,q}(x_i)$ is expected g_i conditioned on x_i, when other coordinates are distributed according to $q_{(i)}$

5) i) If we sample g_i (x) repeatedly for a particular x_i, we get an estimate of $[g_i]_{i,q}(x_i)$

ii) Say the adaptive algorithm setting q_i can always evaluate the current $S_i(q_i)$

In this situation,

Each q_i can adaptively estimate its contribution to *L*(q)

6) Recall that at the q minimizing the entropic *L*(q),

$$q_i^{g_i}(x_i) \propto e^{-\beta [g_i]_{i,q^{g_i}}(x_i)}$$

Each q_i can adaptively estimate its best-case form

***Parallel Brouwer updating*:**

All coordinates i simultaneously replace

$$q_i(x_i) \leftarrow \frac{e^{-\beta [\hat{g}_i]_{i,q}(x_i)}}{N_{i,q}(\beta [\hat{g}_i]_{i,q})}$$

where $[\hat{g}_i]_{i,q}(.)$ is the estimated $[g_i]_{i,q}$, and $N_{i,q}(.)$ is the associated normalization constant (partition function).

- Akin to game theory’s “ficticious play” strategy
- Slow convergence — jumps all over □.
  Can even worsen the approximation in any given update

Serial Brouwer updating :

One coordinate i at a time Brouwer updates

- Guaranteed to decrease L_i if estimate of $[g_i]_{i,q}$ is accurate

Greedy serial Brouwer updating :

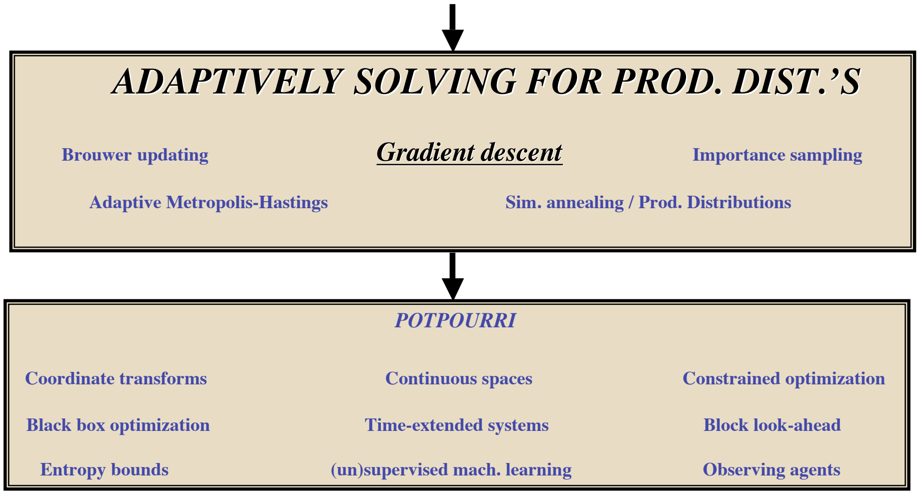
1) The *Lagrangian gap* of coordinate i is how much Li drops if only i updates:

$$\ln[N_{i,q}([g_i]_{i,q})] \;+\; E_{q_i}([g_i]_{i,q}) \;+\; S_i(q_i)$$

2) The coordinate with the largest gap updates

Mixed serial/greedy Brouwer updating :

Optimal COIN "turning on algorithms", i.e., optimal Stackelberg game, i.e., optimal organization chart



ADAPTIVELY SOLVING FOR PROD. DIST.'S

Brouwer updating

Gradient descent

Importance sampling

Adaptive Metropolis-Hastings

Sim. annealing / Prod. Distributions

POTPOURRI

Coordinate transforms

Continuous spaces

Constrained optimization

Black box optimization

Time-extended systems

Block look-ahead

Entropy bounds

(un)supervised mach. learning

Observing agents

GRADIENT DESCENT

1) Say $\boxed{S_i(q_i) = \square_{x_i} S_{i,x_i}(q_i(x_i))}$ (again, like with entropy).

2) Then the $q_i(x_i)$ component of $\square L(q)$, projected onto the space of allowed $q_i(x_i)$, is

$$\boxed{\begin{array}{c} \square[G]_{i,q}(x_i) + S_{i,x_i}(q_i(x_i)) \,/\, q_i(x_i) \\ - \\ \square dx\square_i \big(\square[G]_{i,q}(x\square_i) + S_{i,x_i}(q_i(x\square_i)) \,/\, q_i(x\square_i)\big) \end{array}}$$

- The subtracted term ensures normalization

3) The $S_{i,x_i}(q_i(x_i)) / q_i(x_i)$ values are known by inspection

4) The $\square[G]_{i,q}(x_i)$ terms are estimated as in Brouwer updating

Each q_i can adaptively estimate how it should change under gradient descent over $L(q)$

5) Similarly the Hessian can readily be estimated (for Newton’s method), etc.

1) Consider a team game. Let n_i be the samples of G used by coordinate i to decide how to change under gradient descent

2) The expected quadratic error in that descent step is

$$\boxed{\int dq_{(i)}\, P(q_{(i)})\, [\, \int dn_i\, P(n_i \mid q_{(i)}, g_i)\, \{\nabla L_G(q) - \nabla L_{n_i}(q_i)\}^2\,]}$$

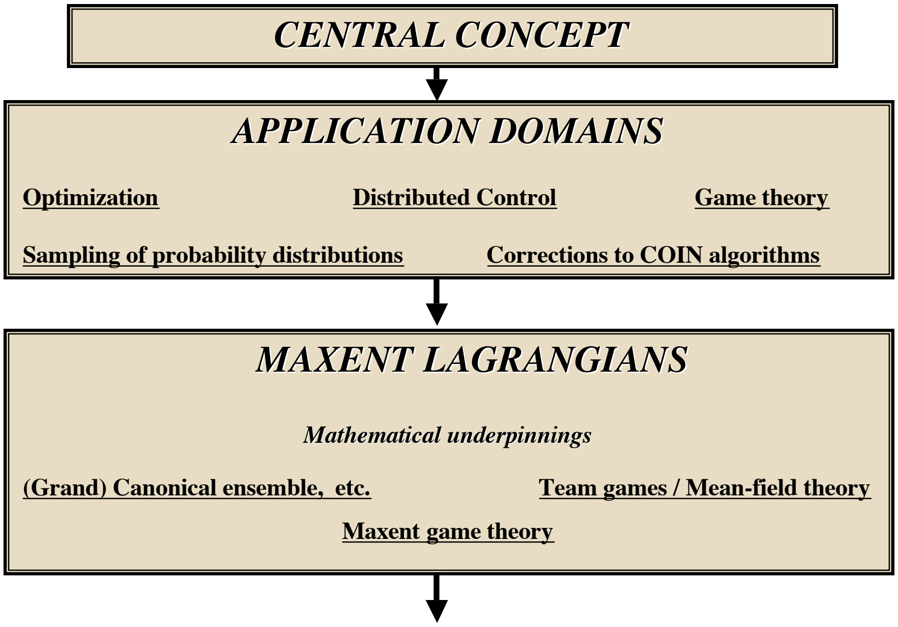
where the gradients are the true gradient of *L* for utility G and the estimated gradient for utility g_i

3) This is just a conventional bias^2 plus variance!

4) Of the g_i guaranteed to be unbiased, the one with the smallest variance is

$$\boxed{G(x) \;-\; \int dx'_i\, G(x'_i, x_{(i)})\, A(x'_i)}$$

where A(.) a distribution, $A(x'_i)$ being proportional to the reciprocal of the number of times x'_i occured in n_i



CENTRAL CONCEPT

APPLICATION DOMAINS

Optimization

Distributed Control

Game theory

Sampling of probability distributions

Corrections to COIN algorithms

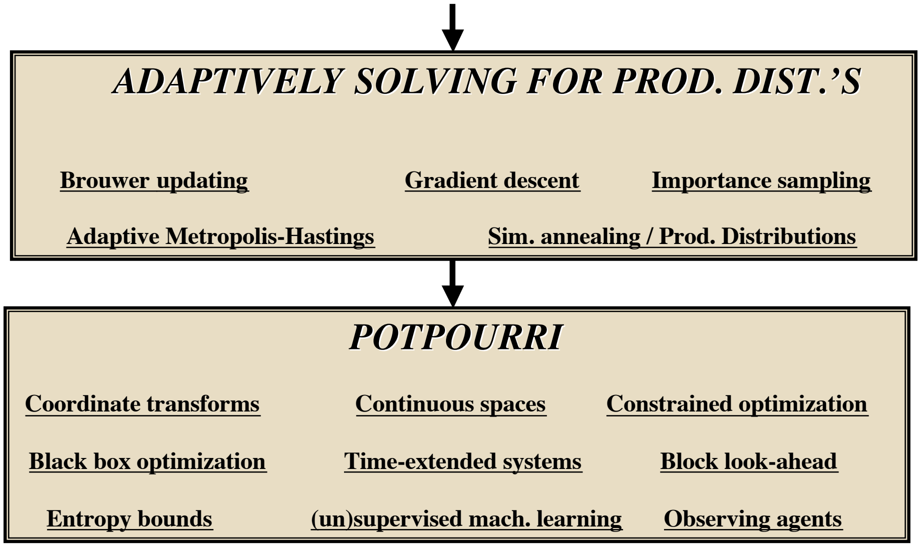
MAXENT LAGRANGIANS

Mathematical underpinnings

(Grand) Canonical ensemble, etc.

Team games / Mean-field theory

Maxent game theory



ADAPTIVELY SOLVING FOR PROD. DIST.'S

Brouwer updating **Gradient descent** **Importance sampling**

Adaptive Metropolis-Hastings **Sim. annealing / Prod. Distributions**

POTPOURRI

Coordinate transforms **Continuous spaces** **Constrained optimization**

Black box optimization **Time-extended systems** **Block look-ahead**

Entropy bounds **(un)supervised mach. learning** **Observing agents**

TAKE-HOME MESSAGE:

Whenever you encounter a distribution P(z) that is difficult to deal with, try expanding it as a product distribution

$$\prod_i q_i(x_i)$$

with associated Lagrangians.