FROM SMITH'S INVISIBLE HAND TO DISTRIBUTED OPTIMIZATION AND CONTROL

David H. Wolpert (Santa Fe Institute)

with

Brendan Tracey (Deep Mind), Stefan Bieniawski (Boeing),

Dev Rajnarayan (Atomic Machines)

SANTA FE INSTITUTE

COMPLEXITY SCIENCE HUB VIENNA

ARIZONA STATE UNIVERSITY

The Abdus Salam International Centre for Theoretical Physics

ROADMAP

- ”**The invisible hand**”, ”**the market is the computer**”, etc.

- This concept can be adapted for distributed control of MAS:
  - Have each agent run a reinforcement algorithm (emulating individual humans in an economy)
  - Design reward functions of each agent so a Nash equilibrium optimizes behavior of entire system.

ROADMAP

- These distributed control techniques can also be used for distributed optimization.

- The *cross-entropy method*, *genetic algorithms*, *simulated annealing*, etc. are just special cases.

Relax requirement that the implementation be distributed

- Then can exploit a formal correspondence between optimization and machine learning to improve these distributed optimization algorithms,
  - Results in (better than) state of the art performance.

THE GOLDEN RULE FOR AGENTS IN AN ECONOMY

DO NOT:

Find a value of a variable x
that optimizes a function G(x).

INSTEAD:

Find a distribution q(x)
that optimizes expected G

PHRASED DIFFERENTLY – RUN MCO

- Monte Carlo Optimization (MCO) is a set of transform techniques:
  Maps an optimization problem *over x*
  into an optimization problem *over q(x).*

- Solves for that optimal q(x) from given data set

- To invert q(x) → x, just sample q(x).

MCO EXAMPLES FOR SINGLE AGENTS

- Example 1: Genetic algorithm (GA)
- Example 2: Simulated annealing (SA)
  - Produce $q(x)$ from data at iteration t, D^t, by minimizing

    $$E_q(G \mid D^t) - T_{t+1} S(q(x))$$

    where $S(.)$ is Shannon entropy.
  - Sample this q
  - Add those samples to D^t to produce D^{t+1}
  - Repeat

WHAT IS DISTRIBUTED OPTIMIZATION?

1) A set of N agents: Joint move $x = (x_1, x_2, ..., x_N)$

2) Since they are distributed, their joint probability is a product distribution:

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

- Same definition of distributed agents as in (iterated) noncooperative game theory.

3) This suggests each agent modifies q(x) to optimize $\mathbb{E}_q(G)$, rather than try to directly optimize x ... a type of MCO!

4) An iterated exact potential ("team") game. ("Invisible hand")

MCO EXAMPLES FOR MULTIPLE AGENTS

- Example 3: Probability Collectives (PC)
  - Define $q^*(x) := \text{argmin} \, [E_q \, (G \mid D^t) \; - \; T_{t+1} S(q(x))]$
  - SA (tries to) construct q^*

MCO EXAMPLES

- Example 3: Probability Collectives (PC)
  - Define $q^*(x) := \text{argmin } [E_q (G \mid D^t) - T_{t+1} S(q(x))]$
  - Instead, try to find product distribution $q_{\Theta^{t+1}}$ that minimizes

$$[E_{q_{\Theta^{t+1}}}(G \mid D^t) - T_{t+1} S(q_{\Theta^{t+1}}(x))]$$

$$-$$

$$[E_{q_{\Theta^*}}(G \mid D^t) - T_{t+1} S(q_{\Theta^*}(x))]$$

  - That would be the product distribution that is "closest" to SA's goal distribution, q^*

MCO EXAMPLES

- Example 3: Probability Collectives (PC)
  - Define $q^*(x) := \text{argmin}\ [E_q (G \mid D^t) - T_{t+1} S(q(x))]$
  - Instead, try to find product distribution $q_{\Theta^{t+1}}$ that minimizes

$$[E_{q_{\Theta^{t+1}}}(G \mid D^t) - T_{t+1} S(q_{\Theta^{t+1}}(x))]$$

$$-$$

$$[E_{q_{\Theta^*}}(G \mid D^t) - T_{t+1} S(q_{\Theta^*}(x))]$$

  - The solution, for coordinate i, is the marginal of the Boltzmann distribution:

$$q_{\theta_i^{t+1}}(x_i) \propto \exp[-T_{t+1} G(x)]_i$$

  - A mean field approximation:

$$q_{\theta_i^{t+1}}(x_i) \propto \exp[-T_{t+1} E(G \mid x_i)]$$

[redacted]

PROBLEM...

- Example 3: Probability Collectives (PC)
  - The solution, for coordinate i, is the marginal of the Boltzmann distribution:

$$q_{\theta_i^{t+1}}(x_i) \propto \exp[-T_{t+1} G(x)]_i$$

  - A mean field approximation:

$$q_{\theta_i^{t+1}}(x_i) \propto \exp[-T_{t+1} E(G \mid x_i)]$$

- Must estimate $E(G \mid x_i)]$ from data - how?
  - *Hack*: Just histogram historical data set.
- But older data points in dataset produced using different distribution than recent data points – how address that?
  - *Hack*: Data-aging, i.e., exponentially weight data points in the histogram

BETTER – MCO AND MACHINE LEARNING

1) Want θ minimizing

$$\int dxdG\ P(G \mid x)q_\theta(x)G$$

- **Hard**. E.g., gradient descent would require evaluating a gradient – which is another difficult integral

2) Importance sample:

$$\int dxdG\, h_x(x)P(G \mid x)\frac{q_\theta(x)G}{h_x(x)} \simeq \sum_{j=1}^{N} \frac{G^j q_\theta(x^j)}{N h_{x^j}(x^j)}$$

where $h_x(x)$ is the distribution used to create the sample x

3) Find θ minimizing RHS

- **Easier**. E.g., estimating gradient is just calculating a sum

SUPERVISED MACHINE LEARNING

1) Conditional distribution $P(Y \mid X)$. Loss function $L : Y \times Y \to R$.

2) Want function $f_\theta(x)$ that minimizes associated expected loss,

   i.e., want θ that minimizes

$$E_\theta(L) \equiv \int dx\, dy\; P(x) P(y \mid x) L(y, f_\theta(x))$$

3) "Training set" D : N samples of $P(x)\, P(y \mid x)$, $\{(x^j, y^j) : j = 1, \ldots N\}$

   i.e., a set of N functions $\{\theta \to L(y^j, f_\theta(x^j)) : j = 1, \ldots N\}$

MCO = Machine Learning (!)

$$\sum_{j=1}^{N} \frac{G^j q_\theta(x^j)}{Nh(x^j)} \ vs. \ \sum_{j=1}^{N} \frac{L(y^j, f_\theta(x^j))}{N}$$

MCO	**Machine Learning**
$\boldsymbol{x}$	$\boldsymbol{x}$
$\boldsymbol{G}$	$\boldsymbol{y}$
$\boldsymbol{h_x(x)}$	$\boldsymbol{P(x)}$
$\frac{G^j q_\theta(x^j)}{N h_{x^j}(x^j)}$	$L(y, f_\theta(x))$

IMPLICATIONS OF THE EQUALITY

MCO problem	**Machine Learning solution**
How shrink bias of $q_{\theta^*}(x)$*?*	*Expand model class*
How shrink variance of $q_{\theta}(x)$*?*	*Bag / regularize*
How set temperature?	*Cross-validation*
How set proposal dist., h(x)?	*Active Learning*
How weight samples?	*Boosting*
How combine q's?	*Stacking*
More expressive q's?	*Kernel machines*

IMPLICATIONS OF THE EQUALITY

MCO problem	**Machine Learning solution**
How shrink bias of $q_\theta(x)$?	*Expand model class*
How shrink variance of $q_\theta(x)$?	**Bag / regularize**
How set temperature?	*Cross-validation*
How set proposal dist., $h(x)$?	*Active Learning*
How weight samples?	*Boosting*
How combine q's?	*Stacking*
More expressive q's?	*Kernel machines*

IMPLICATIONS OF THE EQUALITY

MCO problem	**Machine Learning solution**
How shrink variance of $q_\theta(x)$?	**Bag / regularize**

In the context of MCO, we can regularize with entropy of q_θ

- That gives us

$$\int dx dG\, h_x(x) P(G \mid x) \frac{q_\theta(x) G}{h_x(x)} \simeq \sum_{j=1}^{N} \frac{G^j q_\theta(x^j)}{N h_{x^j}(x^j)} + TS(q_\theta)$$

- Just like PC – but with distribution that generated x^j used!

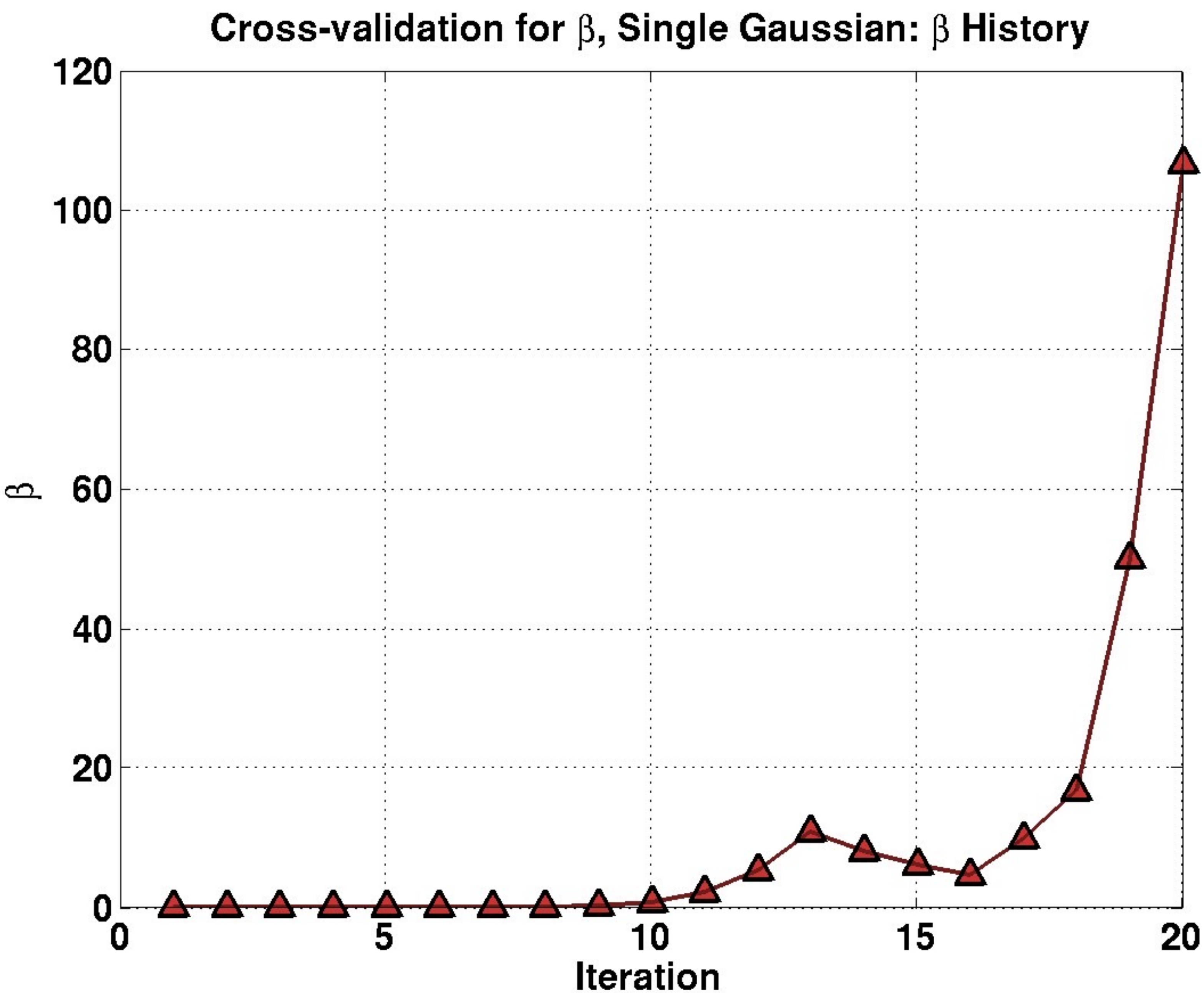
MCO = Machine Learning

MCO problem	**Machine Learning solution**
How shrink bias of $q_\theta(x)$*?*	*Expand model class*
How shrink variance of $q_\theta(x)$*?*	*Bag / regularize*
How set temperature T in PC?	**Cross-validation**
How set proposal dist., h(x)?	*Active Learning*
How weight samples?	*Boosting*
How combine q's?	*Stacking*
More expressive q's?	*Kernel machines*

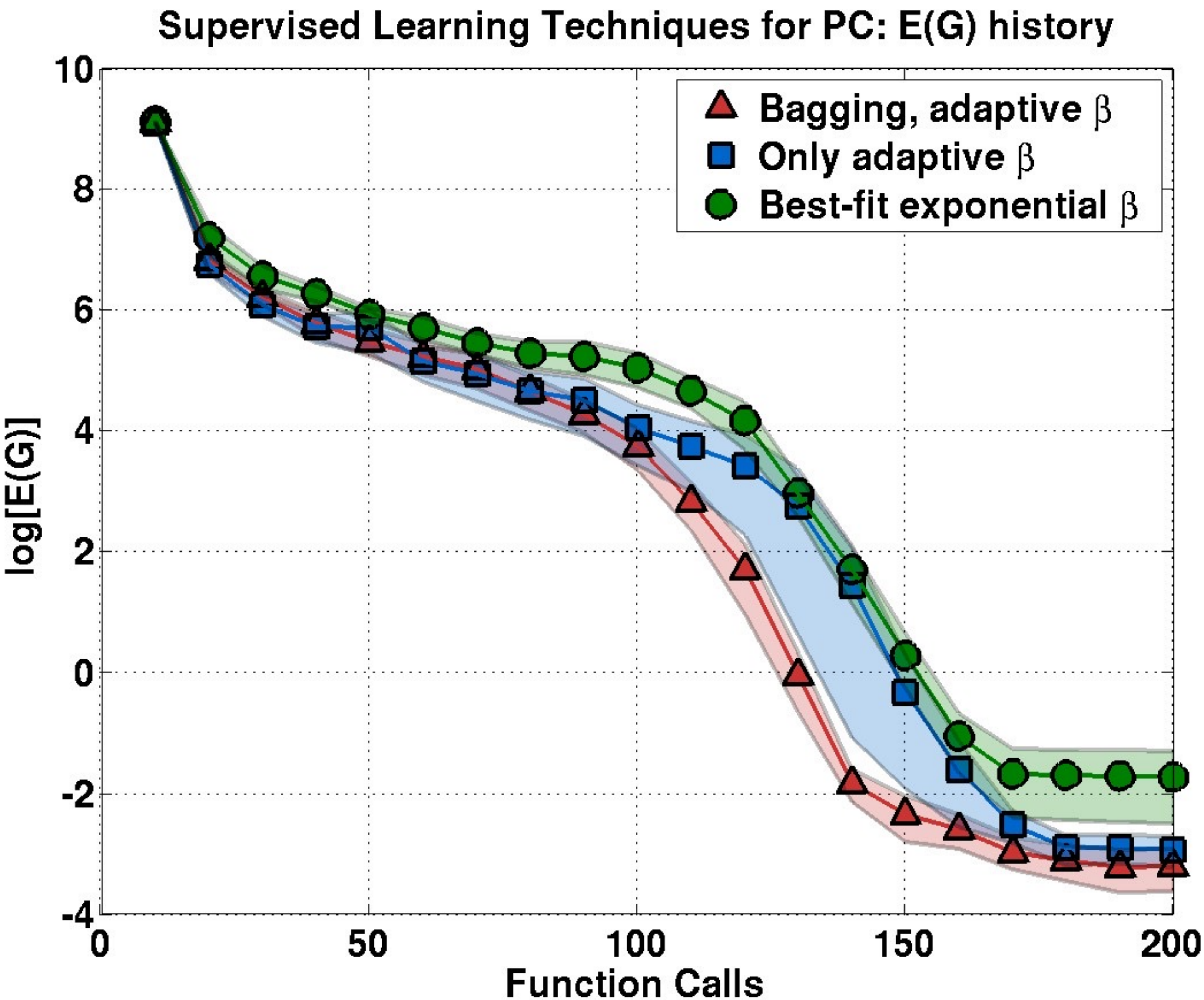
MCO = Machine Learning

MCO problem	Machine Learning solution
How set temperature T in MCO with entropy regularizer and single Gaussian $q_\theta(x)$?	Cross-validation

- No new samples (like would be required in SA)
- Can update T continually – keep changing T to optimize cross-validation, as data set grows
- In other words, ***auto-annealing***, rather than following a pre-fixed annealing schedule



Cross-validation for β, Single Gaussian: β History
β
Iteration
0
20
40
60
80
100
120
0
5
10
15
20

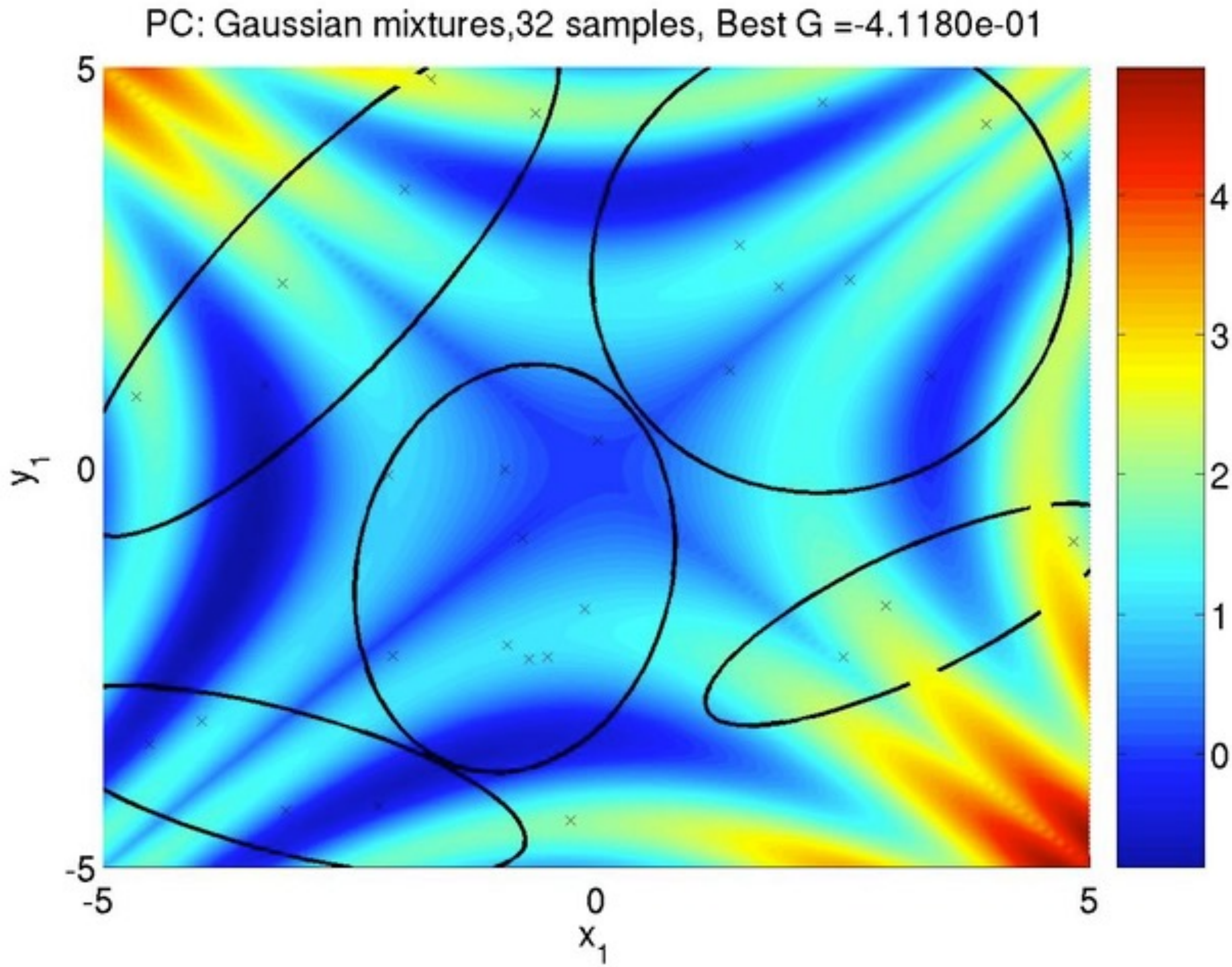


Supervised Learning Techniques for PC: E(G) history
Bagging, adaptive β
Only adaptive β
Best-fit exponential β
log[E(G)]
Function Calls
10
8
6
4
2
0
-2
-4
0
50
100
150
200

MCO = Machine Learning

<u>MCO problem</u>	<u>Machine Learning solution</u>
How set number of components in a mixture model $q_\theta(x)$?	Cross-validation

- No new samples (like would be required in SA)
- Can update number of components continually – keep changing number to optimize cross-validation, as data set grows
- In other words, ***automatically set number of political parties***, with each mixing component a different "party".



PC: Gaussian mixtures,32 samples, Best G =-4.1180e-01
5
0
-5
-5
0
5
y_1
x_1
0
1
2
3
4

FUTURE WORK

- Combine (machine-learning-augmented) MCO with PC, to get

$$\int dx dG \, h_x(x) P(G \mid x) \frac{q_\theta(x) G}{h_x(x)} \simeq \sum_{j=1}^{N} \frac{G^j q_\theta(x^j)}{N h_{x^j}(x^j)} + TS(q_\theta)$$

with a a product distribution $q_\theta(x)$

- Requires each agent to broadcast its sampling distribution to all others after using it