Data Structures for Efficient Inference and Optimization

in Expressive Continuous Domains

Scott Sanner

NICTA

Ehsan Abbasnejad
Zahra Zamani

Karina Valdivia Delgado
Leliane Nunes de Barros

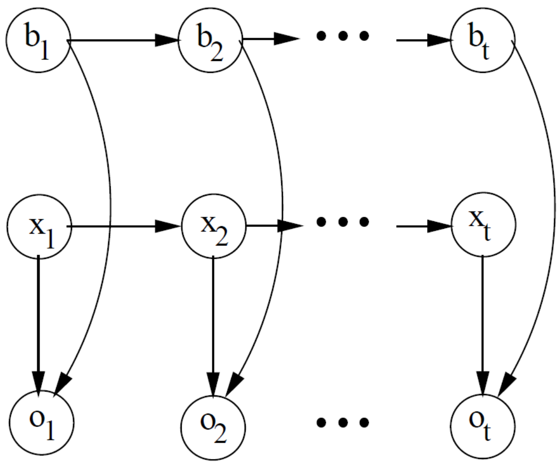
Cheng Fang

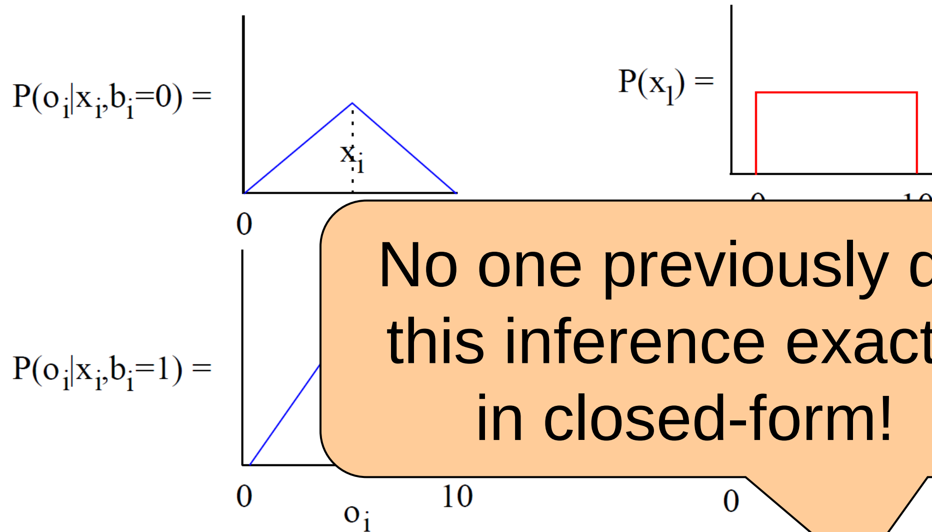
THE AUSTRALIAN NATIONAL UNIVERSITY

Universidade de São Paulo

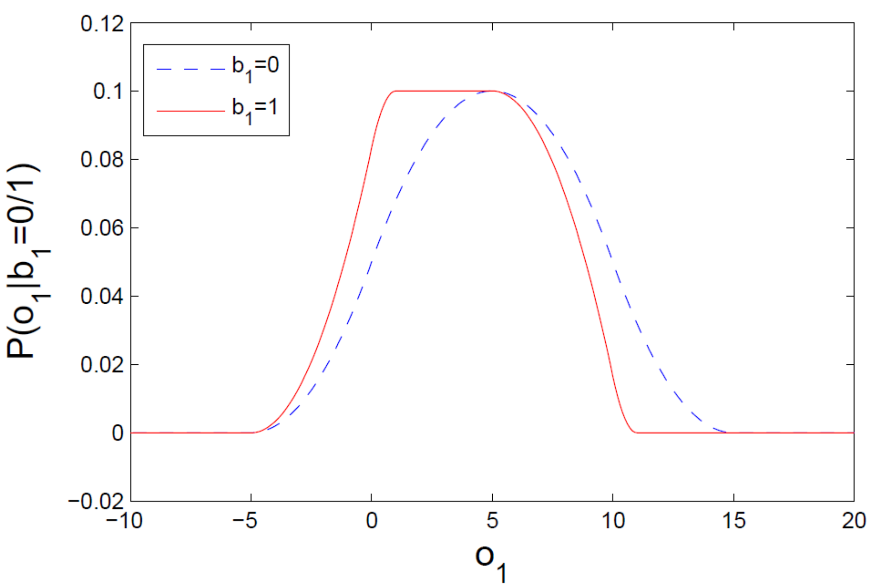
Massachusetts Institute of Technology

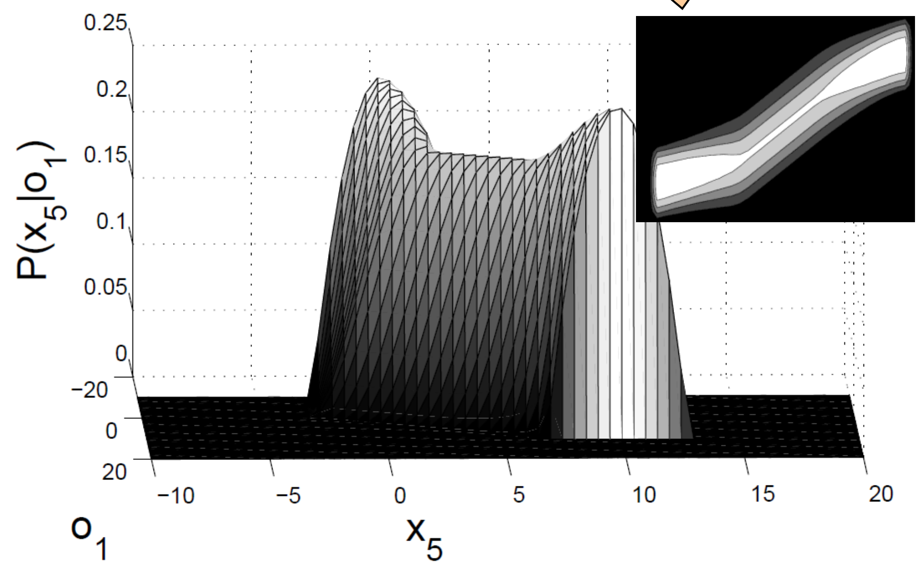
Discrete & Continuous HMMs





No one previously did this inference exactly in closed-form!





Exact Closed-form Continuous Inference

- Fully Gaussian
  - Most inference including conditional
- Fully Uniform
  - 1D, n-D hyperrectangular cases
  - General Uniform
- Piecewise, Asymmetrical, Multimodal
  - Exact (conditional) inference possible in closed-form?

Yes, but not a solution you can write on 1 sheet of paper

What has everyone been missing?

Symbolic representations and operations on piecewise functions

Piecewise Functions (Cases)

$$z = f(x,y) = \begin{cases} (x > 3) \wedge (y \cdot \; x) : & x + y \\ (x \cdot \; 3) \vee (y > x) : & x^2 + xy^3 \end{cases}$$

Partition

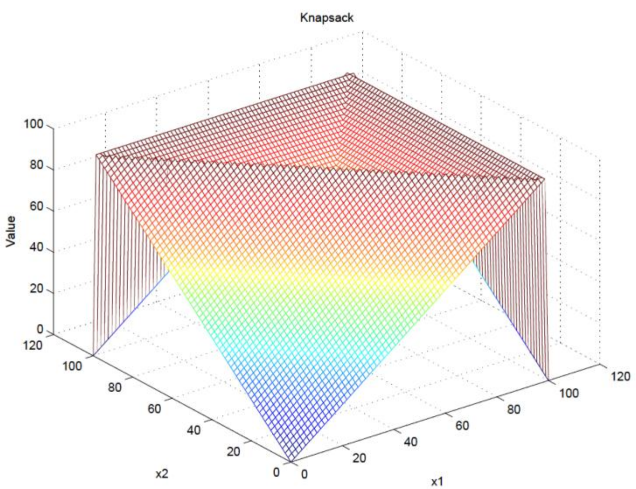
Constraint

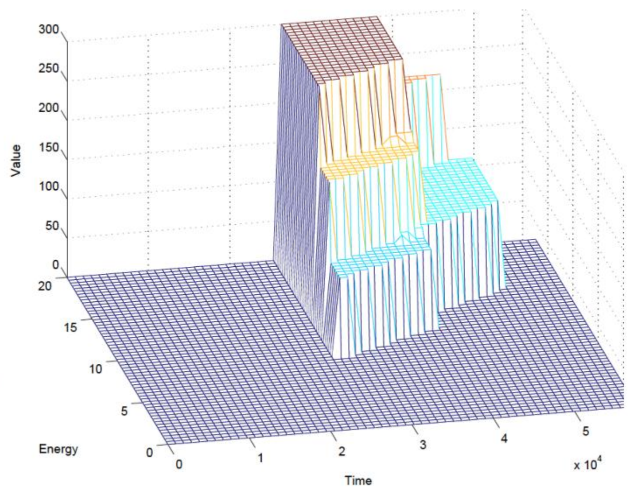
Value

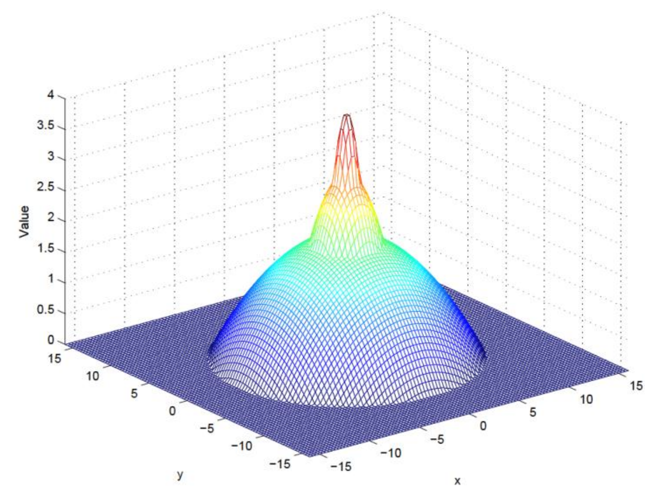
Linear constraints and value

Linear constraints, constant value

Quadratic constraints and value







Formal Problem Statement

- General continuous graphical models represented by piecewise functions (cases)

$$f = \begin{cases} \phi_1 : & f_1 \\ \vdots & \vdots \\ \phi_k : & f_k \end{cases}$$

- Exact closed-form solution inferred via the following piecewise calculus:
  - $f_1 \oplus f_2$, $f_1 \otimes f_2$
  - $\max(f_1, f_2)$, $\min(f_1, f_2)$
  - $\int_x f(x)$
  - $\max_x f(x)$, $\min_x f(x)$

Question: how do we perform these operations in closed-form?

Polynomial Case Operations: ⊕, ⊗

$$\begin{cases} \phi_1 : & f_1 \\ \phi_2 : & f_2 \end{cases} \oplus \begin{cases} \psi_1 : & g_1 \\ \psi_2 : & g_2 \end{cases} = \qquad ?$$

Polynomial Case Operations: ⊕, ⊗

$$\begin{cases} \phi_1 : & f_1 \\ \phi_2 : & f_2 \end{cases} \oplus \begin{cases} \psi_1 : & g_1 \\ \psi_2 : & g_2 \end{cases} = \begin{cases} \phi_1 \wedge \psi_1 : & f_1 + g_1 \\ \phi_1 \wedge \psi_2 : & f_1 + g_2 \\ \phi_2 \wedge \psi_1 : & f_2 + g_1 \\ \phi_2 \wedge \psi_2 : & f_2 + g_2 \end{cases}$$

- Similarly for ⊗
  - Polynomials closed under +, *
- What about max?
  - Max of polynomials is not a polynomial ☹

Polynomial Case Operations: max

$$\max\left(\begin{cases} \phi_1 : & f_1 \\ \phi_2 : & f_2 \end{cases}, \begin{cases} \psi_1 : & g_1 \\ \psi_2 : & g_2 \end{cases} \right) =$$

?

Polynomial Case Operations: max

$$\max\left(\begin{cases}\phi_1 : & f_1 \\ \phi_2 : & f_2\end{cases}, \begin{cases}\psi_1 : & g_1 \\ \psi_2 : & g_2\end{cases}\right) = \begin{cases}\phi_1 \wedge \psi_1 \wedge f_1 > g_1 : & f_1 \\ \phi_1 \wedge \psi_1 \wedge f_1 \cdot g_1 : & g_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 > g_2 : & f_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 \cdot g_2 : & g_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 > g_1 : & f_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 \cdot g_1 : & g_1 \\ \phi_2 \wedge \psi_2 \wedge f_2 > g_2 : & f_2 \\ \phi_2 \wedge \psi_2 \wedge f_2 \cdot g_2 : & g_2\end{cases}$$

- Still a piecewise polynomial!

Size blowup?
We'll get to that…

Integration: $\int_x$

- $\int_x$ closed for polynomials
  - But how to compute for case?

$$\int_x \begin{cases} \phi_1 : & f_1 \\ \vdots & \vdots \\ \phi_k : & f_k \end{cases} dx = \int_x \sum_{i=1}^{k} [\phi_i] \cdot f_i \, dx$$

$$= \sum_i \int_x [\phi_i] \cdot f_i \, dx$$

  - Just integrate case partitions, ⊕ results!

Partition Integral

1. Determine integration bounds

$$\int_x [\phi_1] \cdot f_1 \, dx$$

$$\phi_1 := [x > -1] \wedge [x > y - 1] \wedge [x \cdot \quad z] \wedge [x \cdot \quad y + 1] \wedge [y > 0]$$

$$f_1 := x^2 - xy$$

$$LB := \begin{cases} y - 1 > -1 : & y - 1 \\ y - 1 \cdot \quad -1 : & -1 \end{cases} \qquad UB := \begin{cases} z < y + 1 : & z \\ z \geq y + 1 : & y + 1 \end{cases}$$

$$[\phi_{cons}] \int_{x = LB}^{UB} f_1 \, dx$$

UB and LB are symbolic!

How to evaluate?

What constraints here?

- independent of x
- pairwise UB > LB

Definite Integral Evaluation

- How to evaluate integral bounds?

$$\int_{x=LB}^{UB} x^2 - xy \; = \; \frac{1}{3}x^3 - \frac{1}{2}x^2 y \Big|_{LB}^{UB}$$

$$LB := \begin{cases} y - 1 > -1 : & y - 1 \\ y - 1 \cdot \; -1 : & -1 \end{cases} \qquad UB := \begin{cases} z < y + 1 : & z \\ z \geq y + 1 : & y + 1 \end{cases}$$

- Can do polynomial operations on cases!

$$f_1 \Big|_{LB}^{UB} = \left[\frac{1}{3} UB \quad UB \quad UB \ominus \frac{1}{2} UB \quad UB \quad (y) \right]$$

$$\ominus \left[\frac{1}{3} LB \quad LB \quad LB \ominus \frac{1}{2} LB \quad LB \quad (y) \right]$$

Symbolically, exactly evaluated!

Exact Graphical Model Inference!

(directed and undirected)

- Can do general probabilistic inference

$$p(x_2|x_1) = \frac{\int_{x_3} \cdots \int_{x_n} \bigotimes_{i=1}^{k} case_i \; dx_n \cdots dx_3}{\int_{x_2} \cdots \int_{x_n} \bigotimes_{i=1}^{k} case_i \; dx_n \cdots dx_2}$$

- Or an exact expectation of *any* polynomial

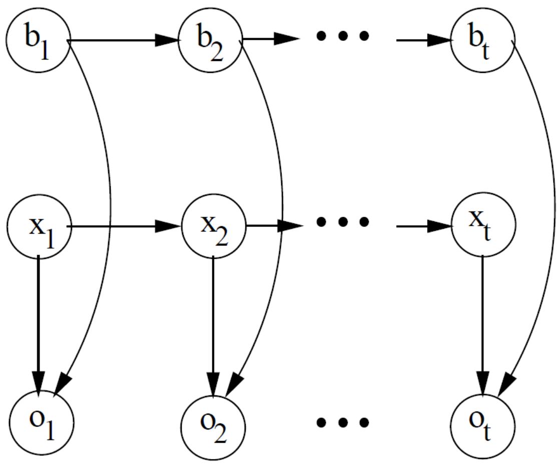
$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{o})}[poly(\mathbf{x})|\mathbf{o}] = \int_{\mathbf{x}} p(\mathbf{x}|\mathbf{o}) poly(\mathbf{x}) d\mathbf{x}$$

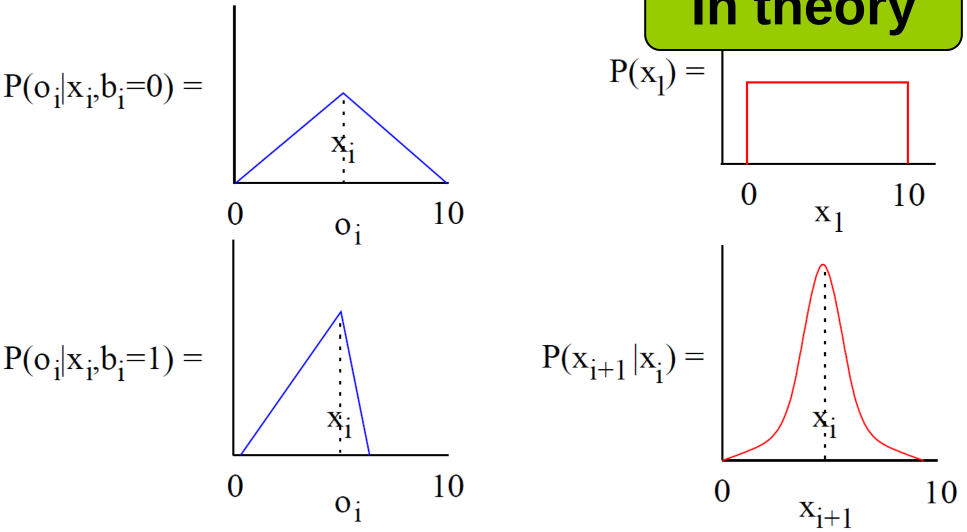
  – *poly*: mean, variance, skew, curtosis, …, x^2+y^2+xy

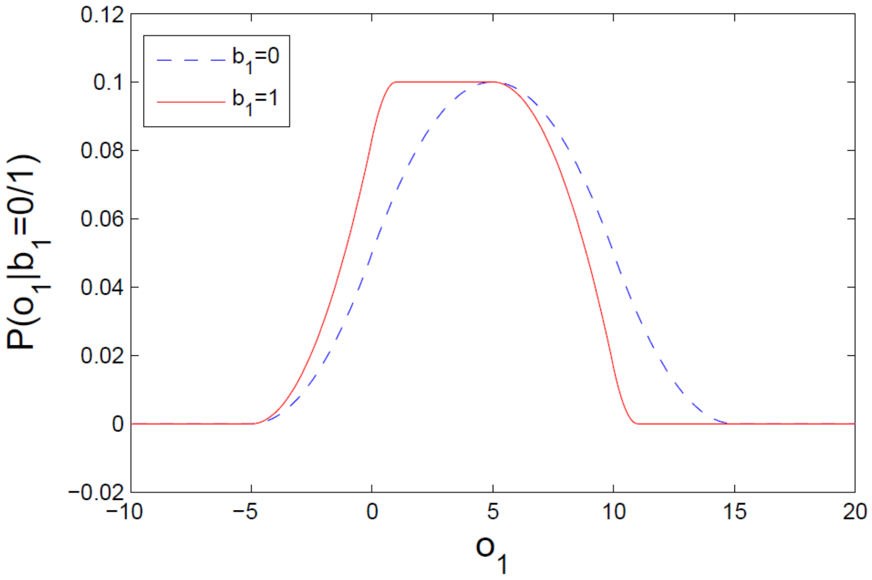
All computed by
Symbolic Variable Elimination (SVE)

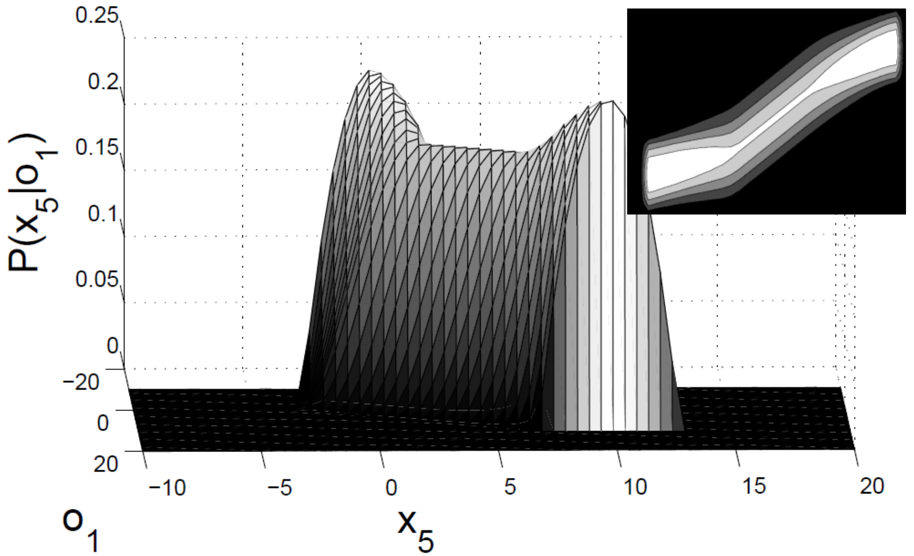
Voila: Closed-form Exact Inference via SVE!

In theory









An Expressive Conjugate Prior for Bayesian Inference

- General Bayesian Inference

$$p(\vec{\theta}|D_{n+1}) \propto p(d_{n+1}|\vec{\theta})p(\vec{\theta}|D_n)$$

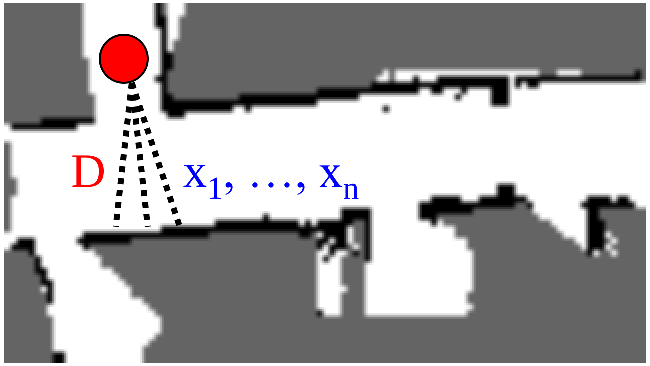
case

case

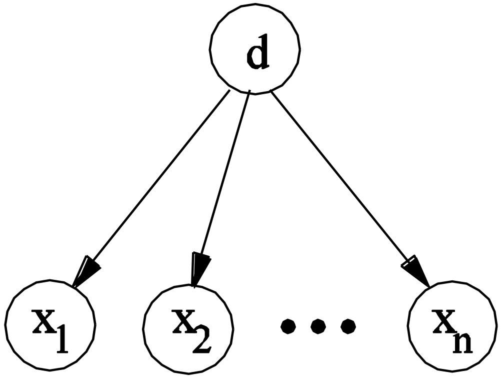
case

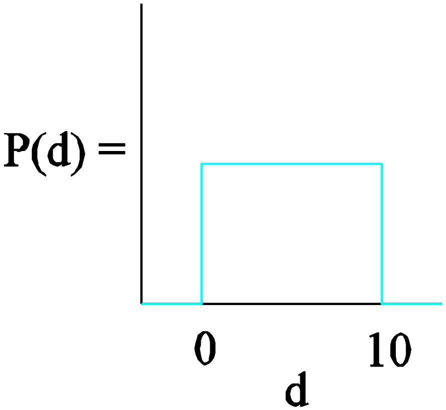
- Prior & likelihood for computational convenience?
  - No, choose as appropriate for your problem!

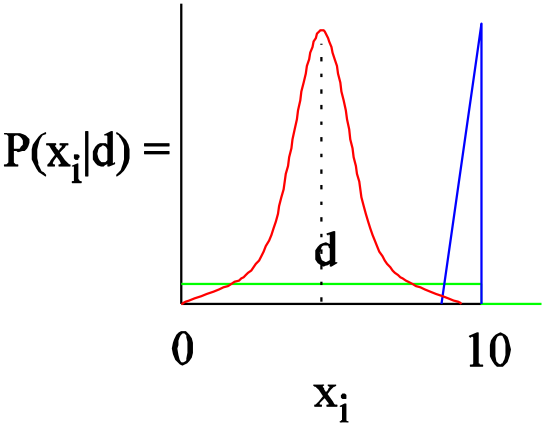
Bayesian Robotics



- D: true distance to wall
- $x_1, \ldots, x_n$: measurements
- want: $E[D \mid x_1, \ldots, x_n]$

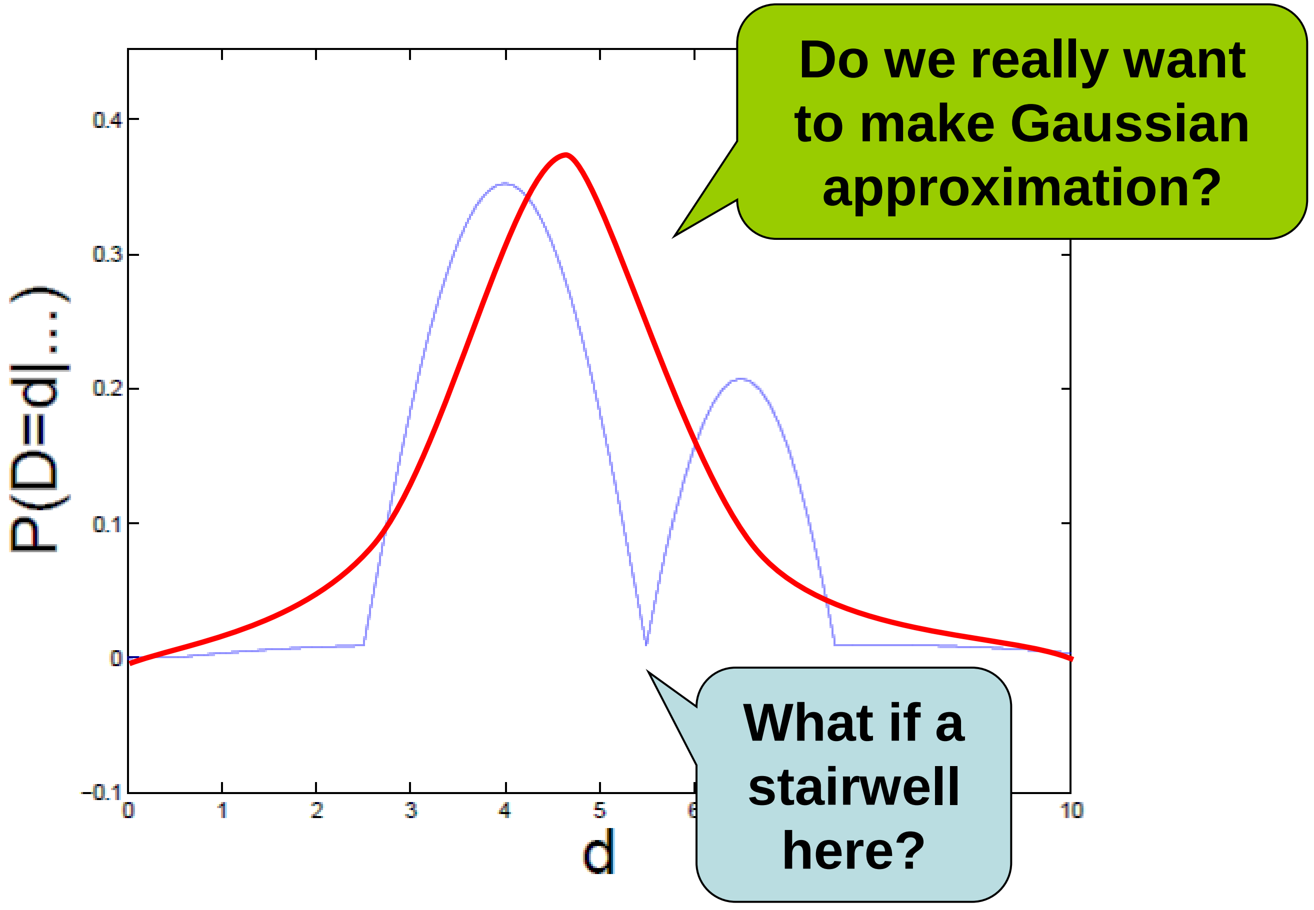






Bayesian Robotics: Results

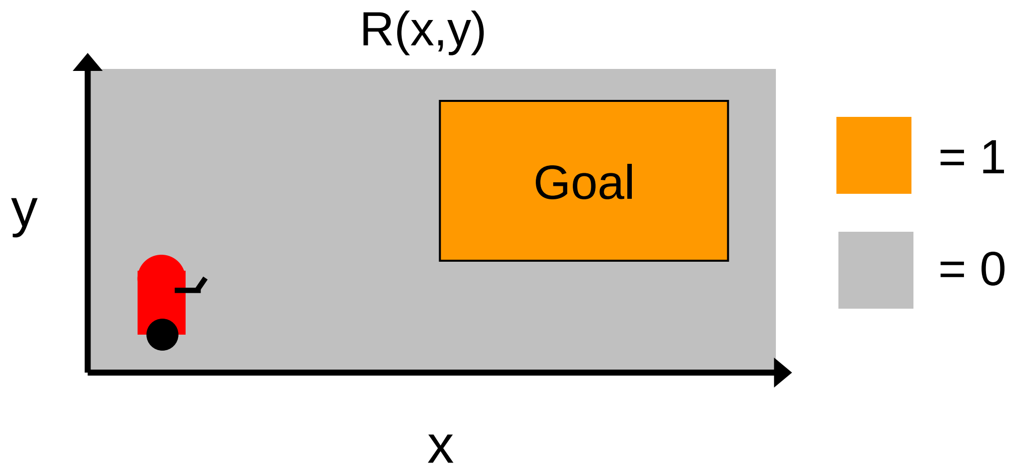
- Example posterior given measurements {3,5,8}:



Symbolic Sequential Decision Optimization?

Continuous State MDPs

- 2-D Navigation
- State: (x,y) $\in \mathbb{R}^2$
- Actions:
  - move-x-2
    - x' = x + 2
    - y' = y
  - move-y-2
    - x' = x
    - y' = y + 2
- Reward:
  - R(x,y) = I[(x > 5) ^ (x < 10) ^ (y > 2) ^ (y < 5)]

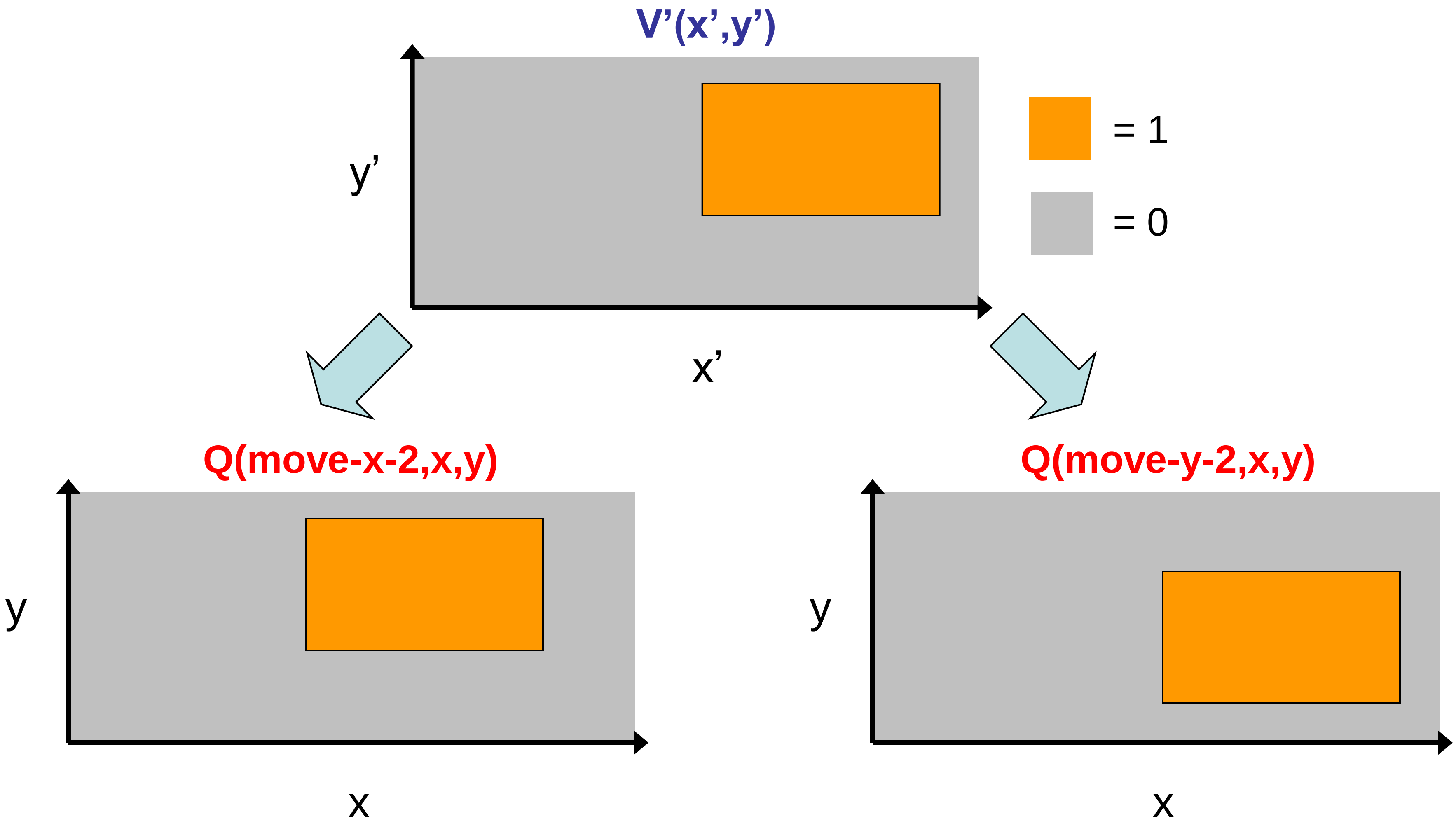


Feng *et al* (UAI-04) Assumptions:

1. Continuous transitions are deterministic and linear
2. Discrete transitions can be stochastic
3. Reward is piecewise rectilinear convex

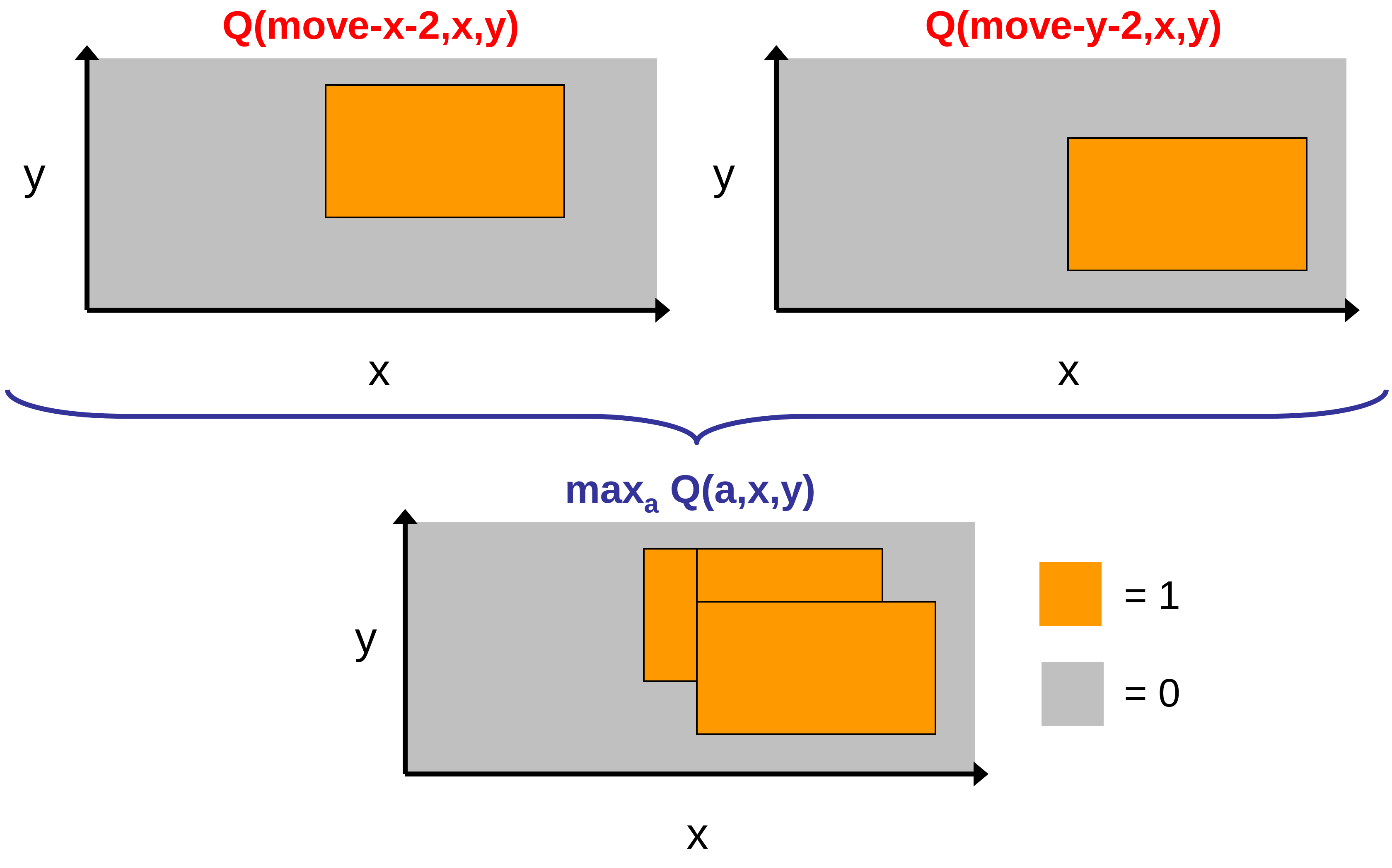
Exact Solutions to DC-MDPs: Regression

- Continuous regression is just translation of “pieces”



Exact Solutions to DC-MDPs: Maximization

- Q-value maximization yields piecewise rectilinear solution



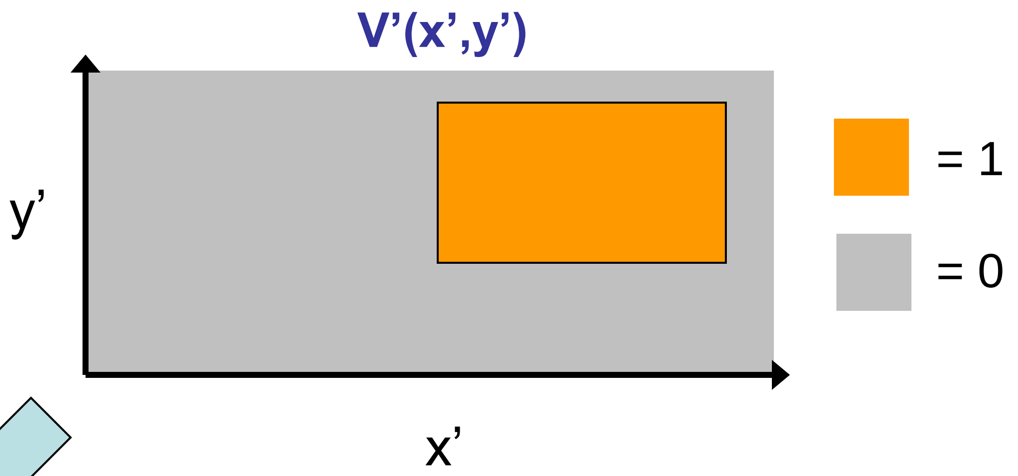
Previous Work Limitations I

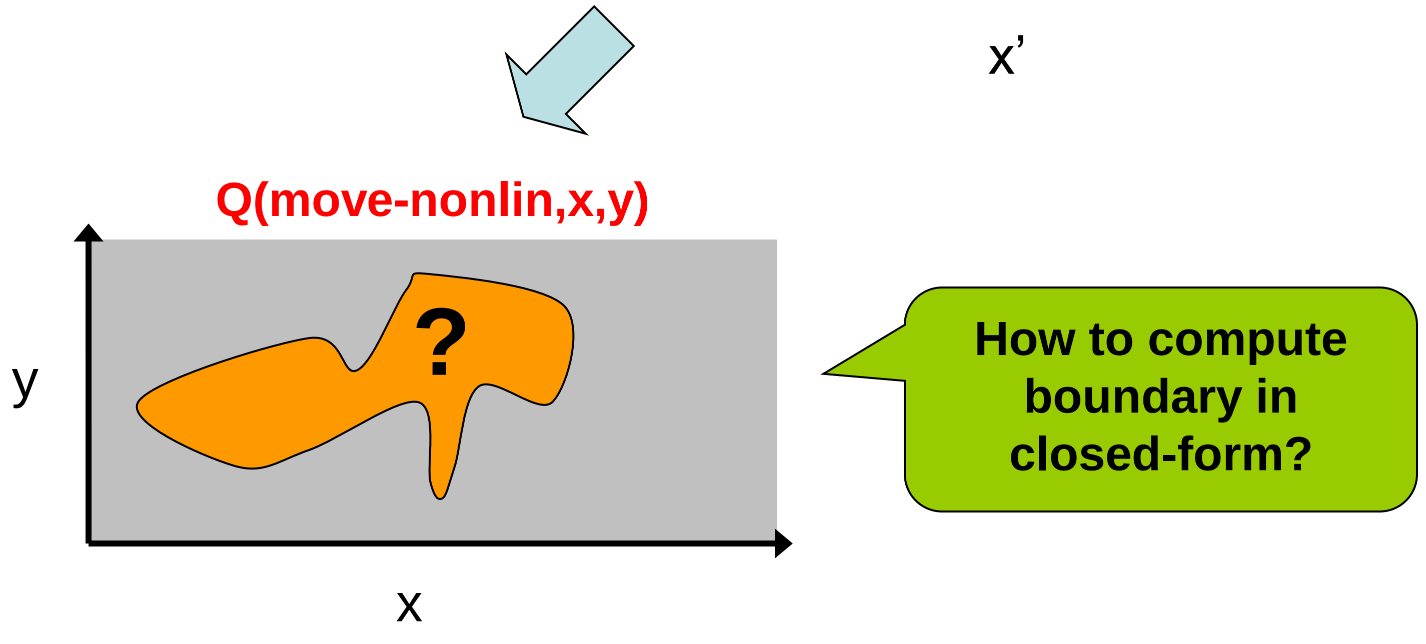
- Exact regression when transitions nonlinear?

Action move-nonlin:

– $x' = x^3y + y^2$

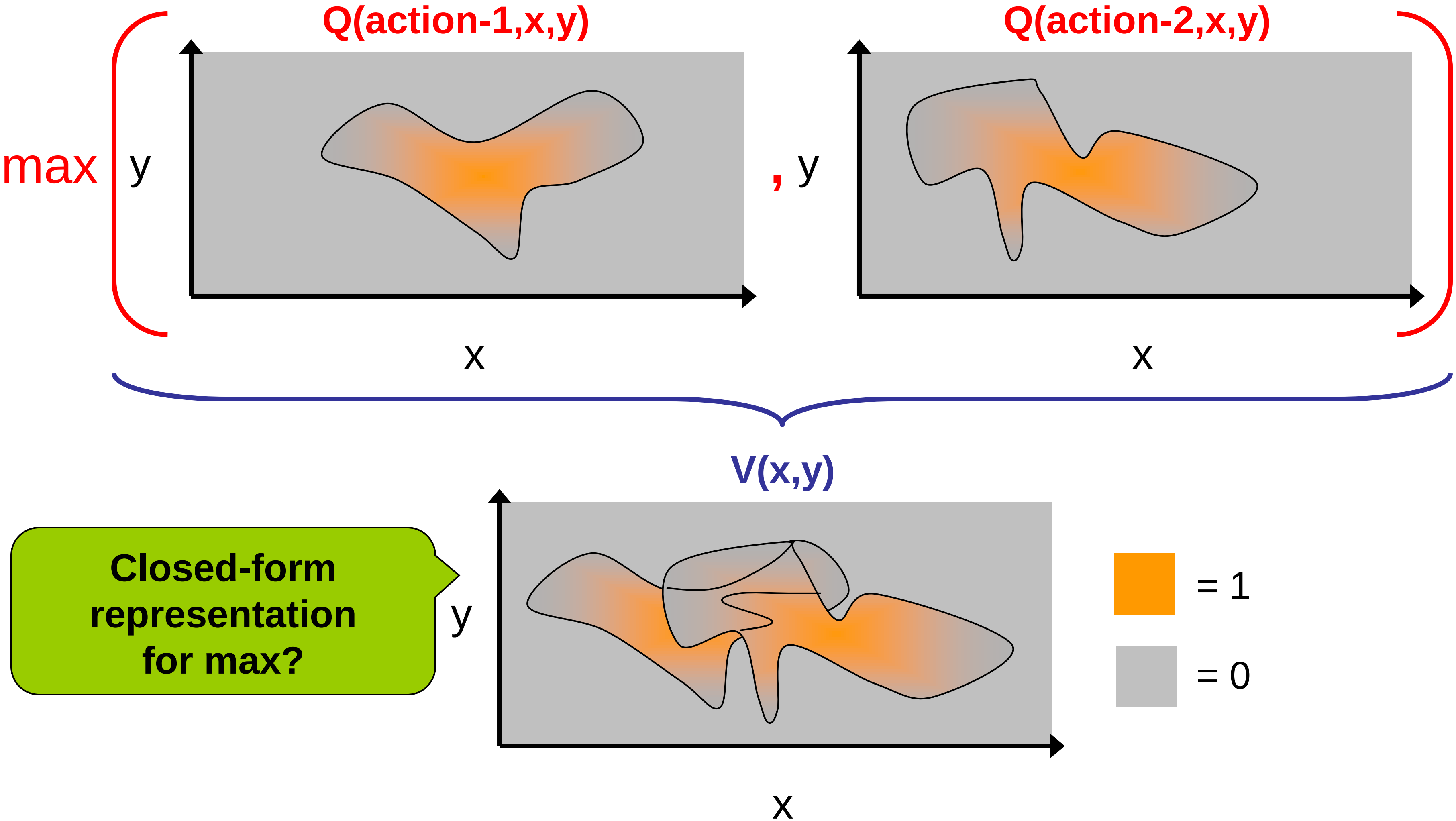
– $y' = y * \log(x^2y)$





Previous Work Limitations II

- max(.,.) when reward/value arbitrary piecewise?



Continuous Actions?

If we can solve this, can solve **multivariate inventory control** – closed-form policy unknown for 50+ years!

Continuous Actions

- Inventory control
  - Reorder based on stock, future demand
  - Action: $a(\vec{\Delta});\ \vec{\Delta} \in \mathbb{R}^{|a|}$

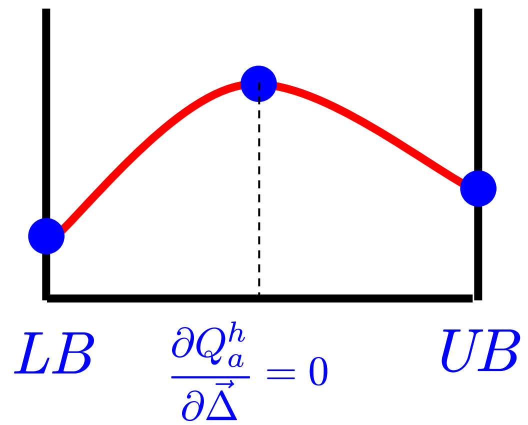
- Need $\max_{\Delta}$ in Bellman backup

$$V_{h+1} = \max_{a \in A} \max_{\vec{\Delta}} Q_a^{h+1}(\vec{\Delta})$$

- $\max_x$ case(x) similar to $\int_x$ case(x)
  - Track maximizing Δ substitutions to recover π

Max-out Case Operation

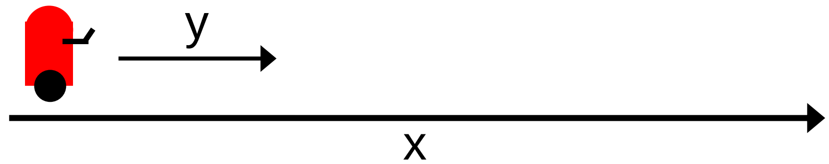
- Like $\int_x$ case(x), reduce to single partition max
  - In a *single* case partition …*max* w.r.t. critical points
    - LB, UB
    - Derivative is zero (Der0)
    - max(case(x/LB),
      case(x/UB),
      case(x/Der0))

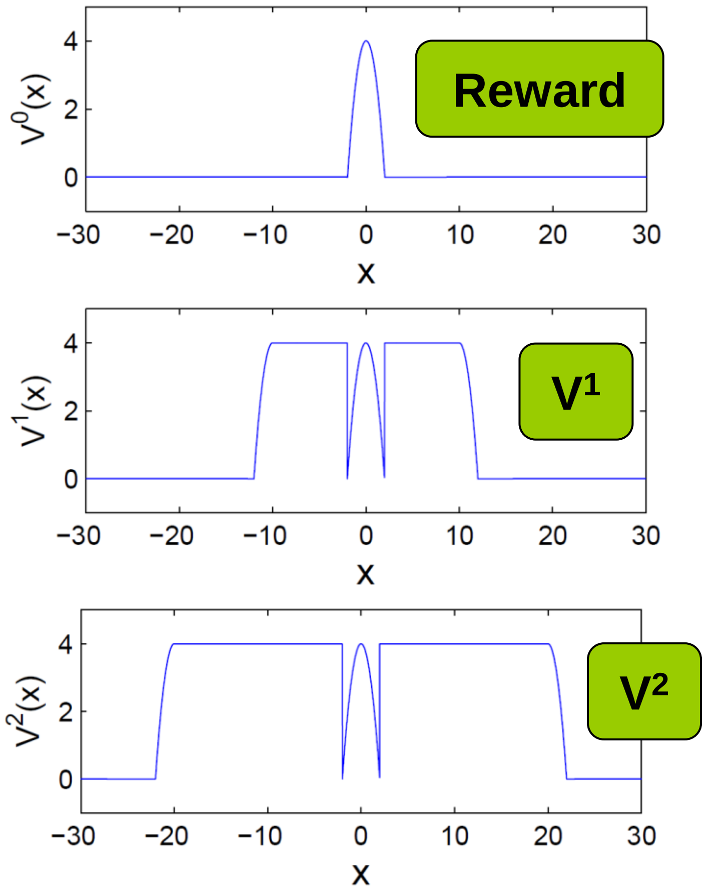


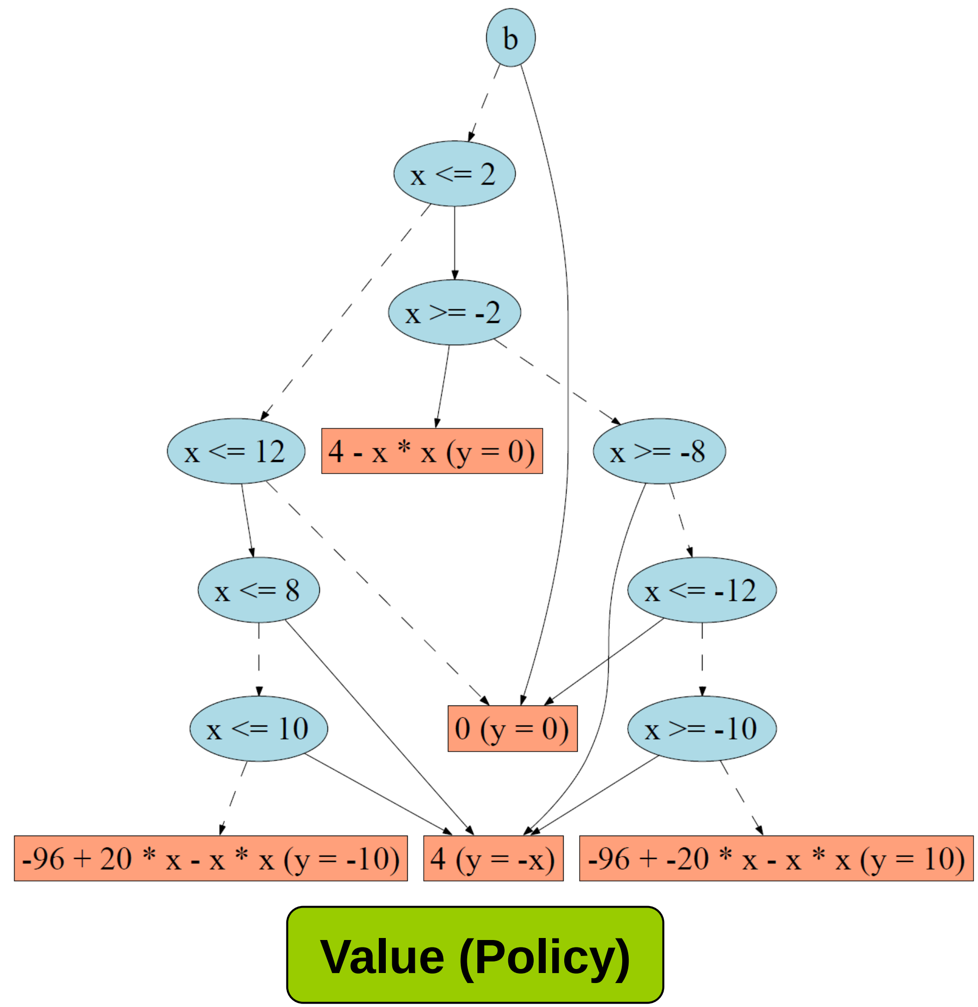
See UAI 2011, AAAI 2012 papers for more details

  - Can even track substitutions through max to recover function of maximizing assignments!

Illustrative Value and Policy







Sequential Control Summary

- Continuous state, action, observation (PO)MDPs
  - Discrete action MDPs **UAI-11**
  - Continuous action MDPs (incl. exact policy) **AAAI-12b**
  - Continuous observation POMDPs **NIPS-12**
  - Extensions to general continuous noise **In progress**

Symbolic Constrained Optimization

$\max_x$ case(x) = Constrained Optimization!

- Conditional constraints
  - E.g., if $(x > y)$ then $(y < z)$
  - 0-1 MILP, MIQP equivalent

- Factored / sparse constraints
  - Constraints may be sparse!

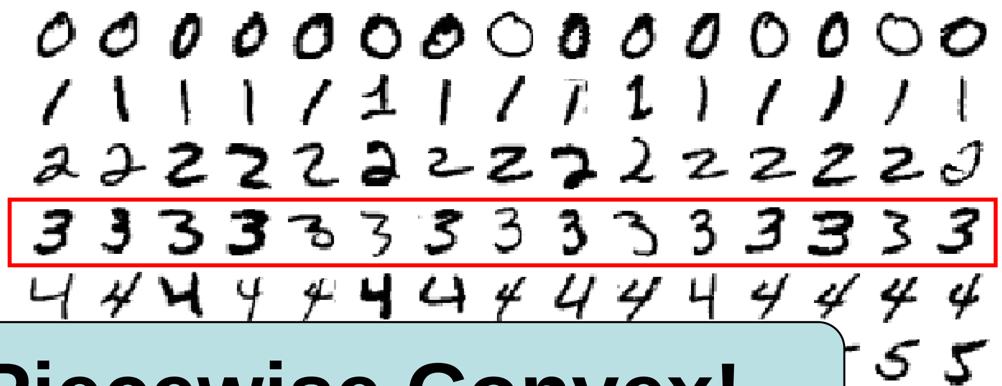
    $x_1 > x_2, \quad x_2 > x_3, \quad \ldots, \quad x_{n-1} > x_n$
  - Dynamic programming for continuous optimization!

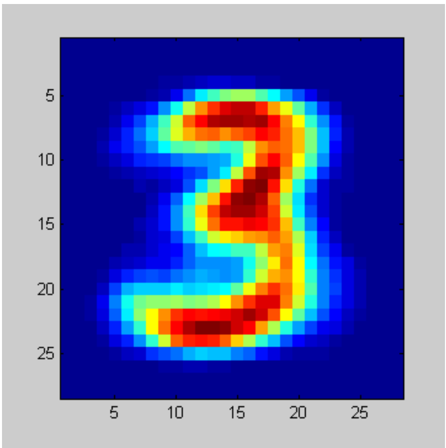
- Parameterized optimization
  - $f(y) = \max_x f(x,y)$
  - Maximum value, substitution as a **function of y**

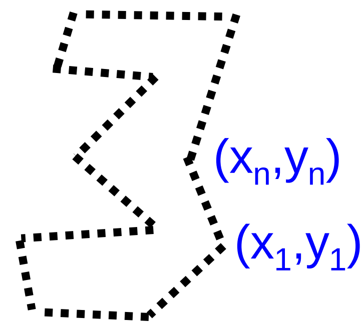
Symbolic Machine Learning

Geometric Models: Piecewise Regression

- How to learn geometric models of objects?





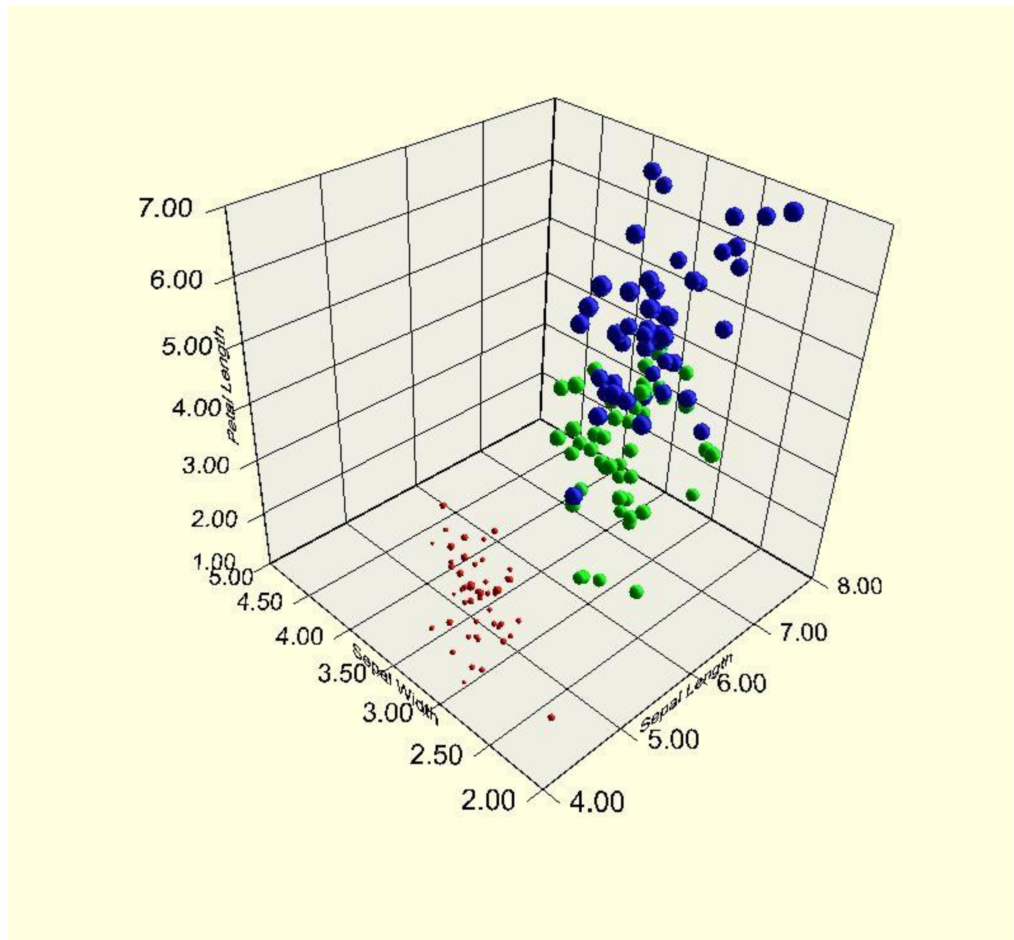


Piecewise Convex!

$$\arg\min_{x_1,y_1,\ldots,x_n,y_n} \sum_{(x,y)\to z^* \in Images} \left| z^* - \begin{cases} x \geq x_1 \wedge y \geq y_1 \wedge \cdots : & 1 \\ x \cdot \;\; x_1 \wedge y \cdot \;\; y_1 \wedge \cdots : & 0 \end{cases} \right|$$

Optimal Clustering?

- Use min to make any point “snap to” nearest center
  - Minimize sum of min distances
- Piecewise convex!
- No latent variables



What has everyone been missing?

Symbolic algebra / calculus?

No, they weren't Computer Scientists.

"case" representation blows up… need a data structure: XADD.

BDD / ADDs

Quick Introduction

Function Representation (Tables)

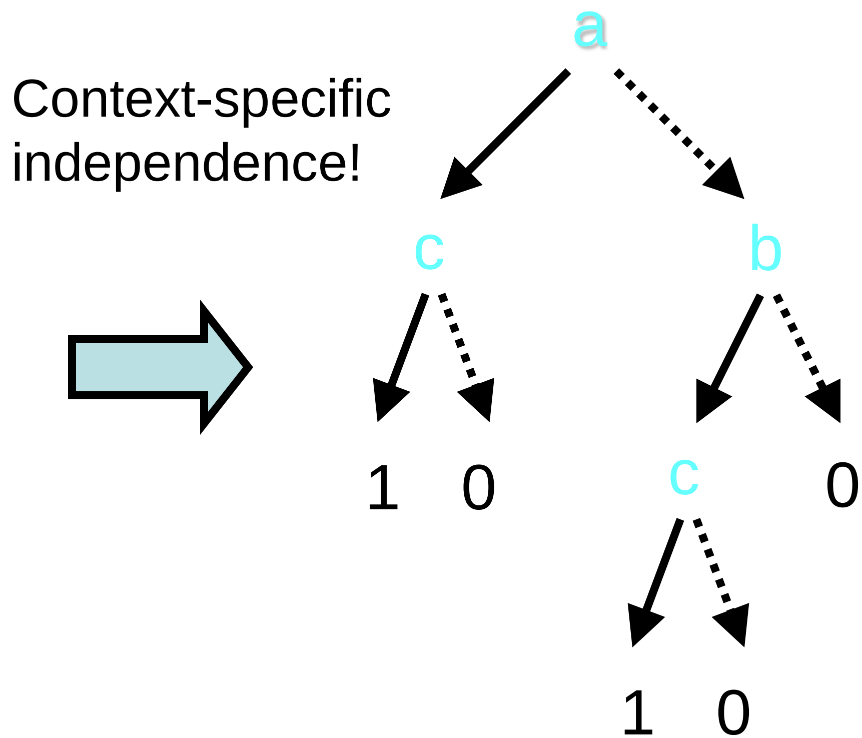
- How to represent functions: $B^n \rightarrow R$?
- How about a fully enumerated table…
- …OK, but can we be more compact?

a	b	c	F(a,b,c)
0	0	0	0.00
0	0	1	0.00
0	1	0	0.00
0	1	1	1.00
1	0	0	0.00
1	0	1	1.00
1	1	0	0.00
1	1	1	1.00

Function Representation (Trees)

- How about a tree? Sure, can simplify.

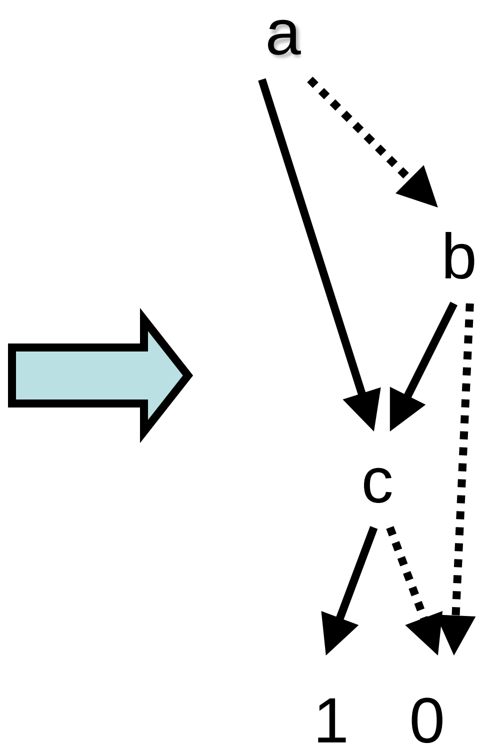
a	b	c	F(a,b,c)
0	0	0	0.00
0	0	1	0.00
0	1	0	0.00
0	1	1	1.00
1	0	0	0.00
1	0	1	1.00
1	1	0	0.00
1	1	1	1.00



Function Representation (ADDs)

- Why not a directed acyclic graph (DAG)?

a	b	c	F(a,b,c)
0	0	0	0.00
0	0	1	0.00
0	1	0	0.00
0	1	1	1.00
1	0	0	0.00
1	0	1	1.00
1	1	0	0.00
1	1	1	1.00

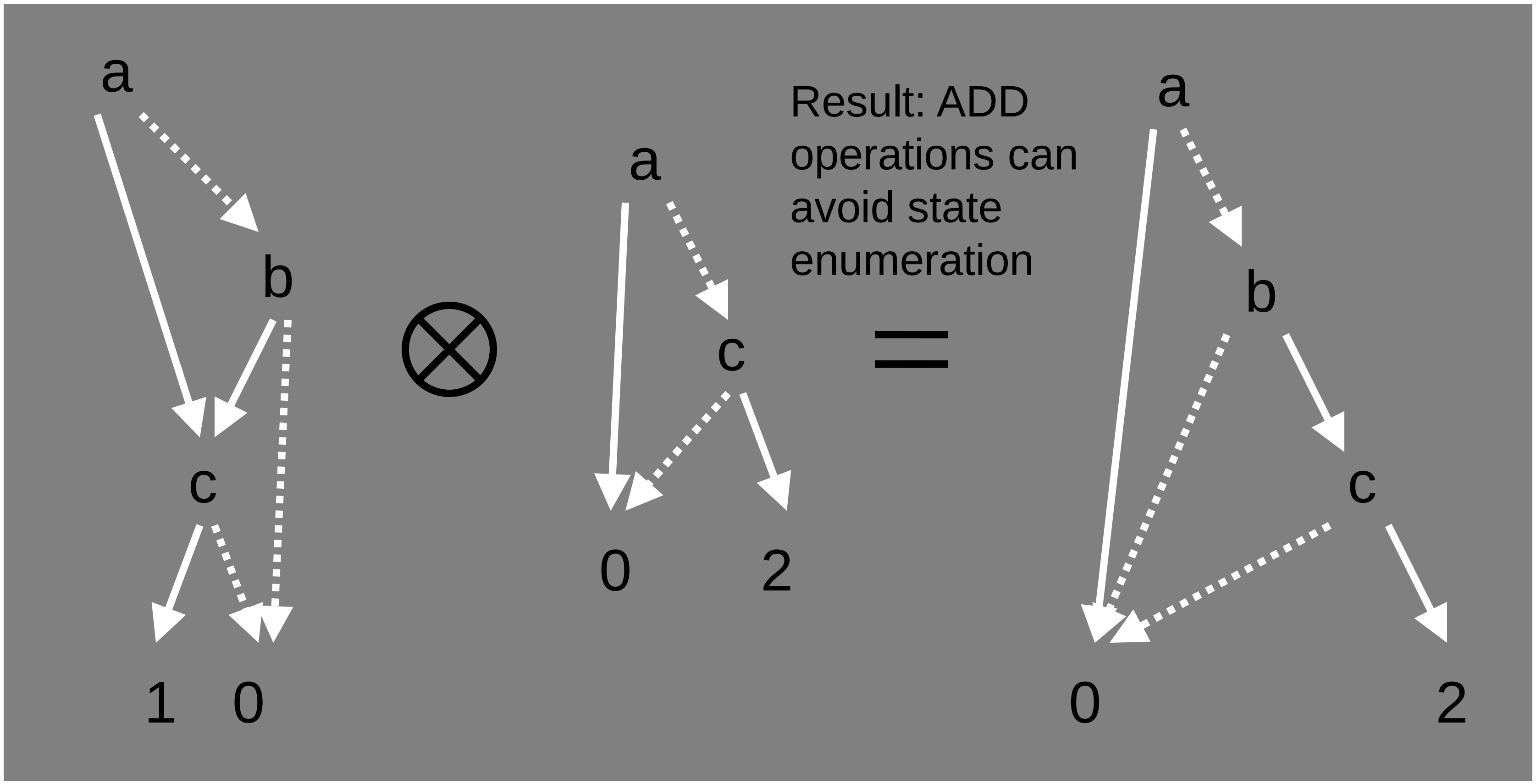


Algebraic Decision Diagram (ADD)

Think of BDDs as {0,1} subset of ADD range

Binary Operations (ADDs)

- Why do we order variable tests?
- Enables us to do efficient binary operations…



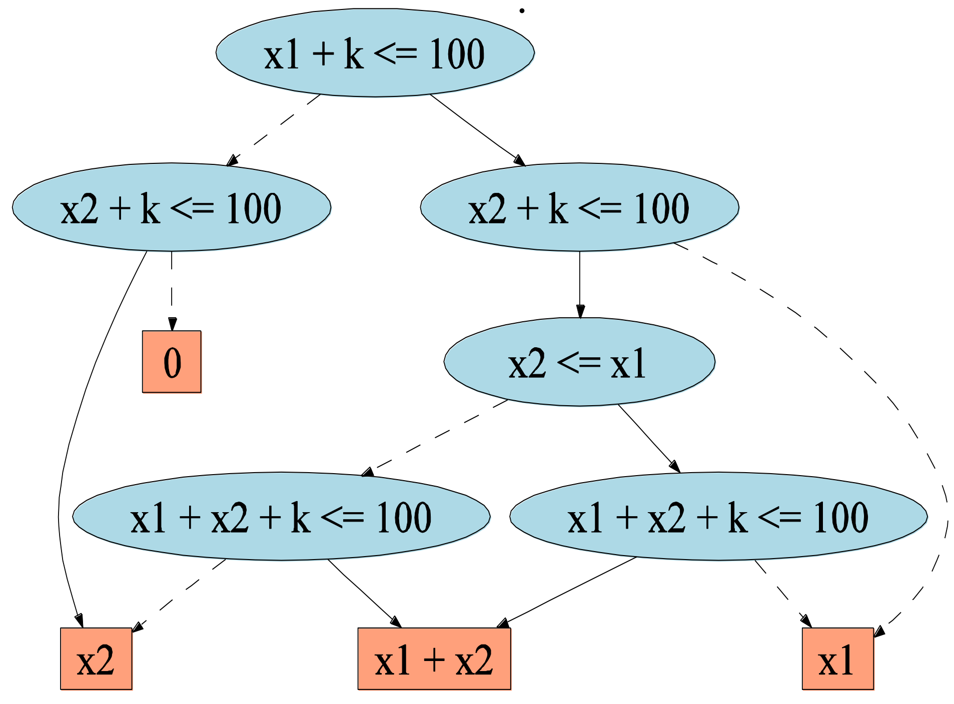
Case $\rightarrow$ XADD

XADD = continuous variable e**x**tension
of **a**lgebraic **d**ecision **d**iagram

- ➔ compact, minimal case representation
- ➔ efficient case operations

Case → XADD

$$V = \begin{cases} x_1 + k > 100 \wedge x_2 + k > 100 : & 0 \\ x_1 + k > 100 \wedge x_2 + k \leq 100 : & x_2 \\ x_1 + k \leq 100 \wedge x_2 + k > 100 : & x_1 \\ x_1 + x_2 + k > 100 \wedge x_1 + k \leq 100 \wedge x_2 + k \leq 100 \wedge x_2 > x_1 : & x_2 \\ x_1 + x_2 + k > 100 \wedge x_1 + k \leq 100 \wedge x_2 + k \leq 100 \wedge x_2 \leq x_1 : & x_1 \\ x_1 + x_2 + k \leq 100 : & x_1 + x_2 \\ \vdots & \vdots \end{cases}$$

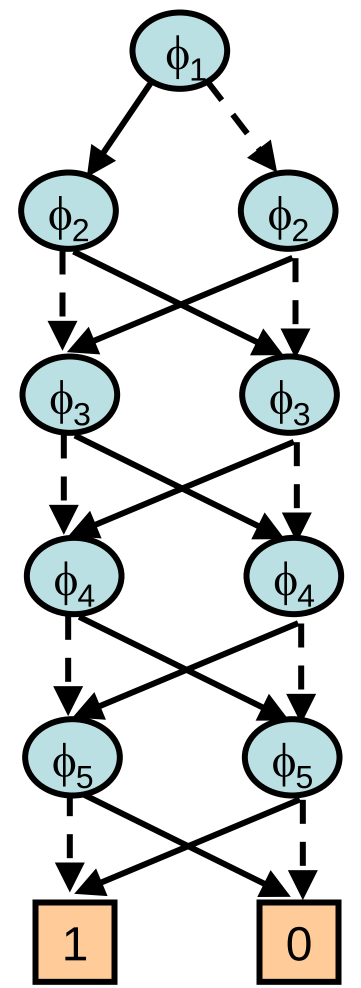


*With non-trivial extensions over ADD, can reduce to a minimal canonical form!

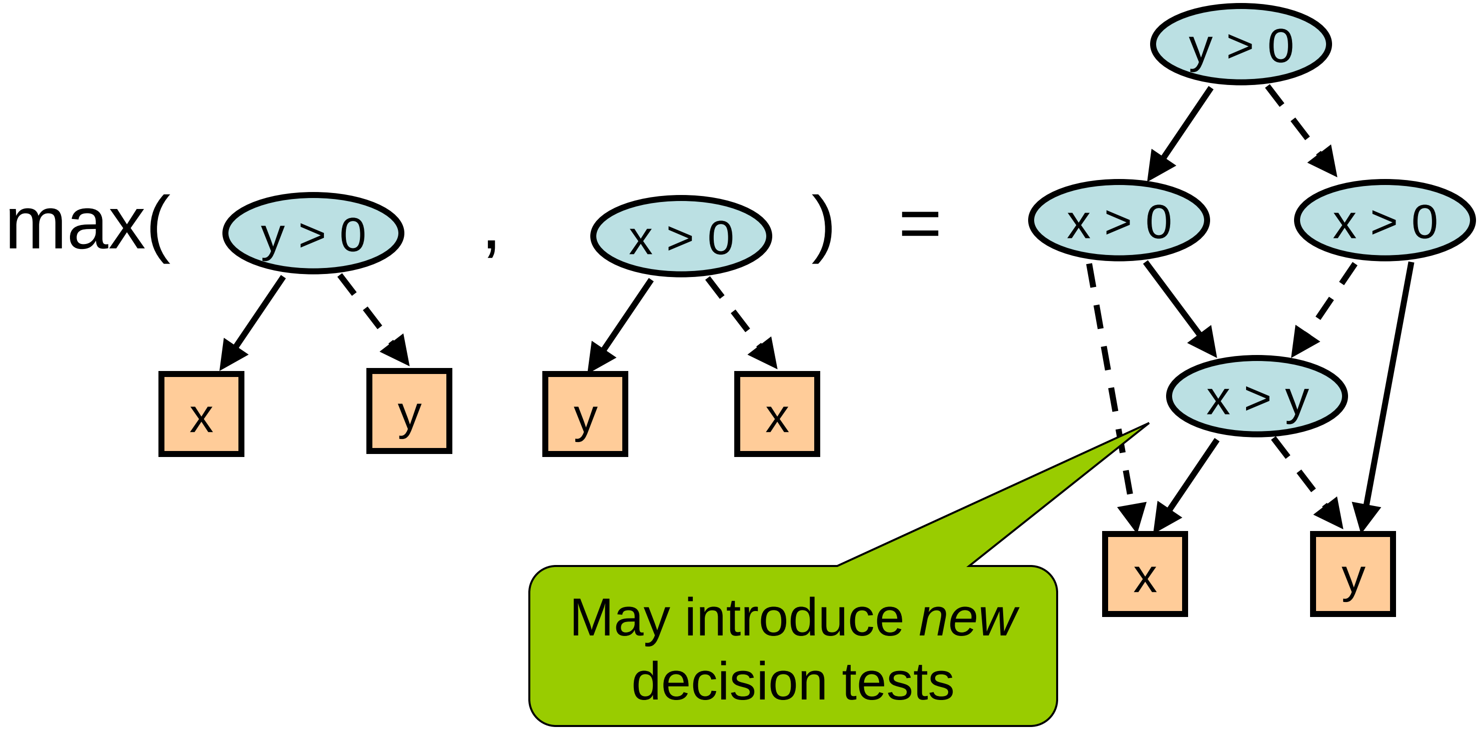
Compactness of (X)ADDs

- Linear in number of decisions ϕ_i

- Case version has exponential number of partitions!



XADD Maximization



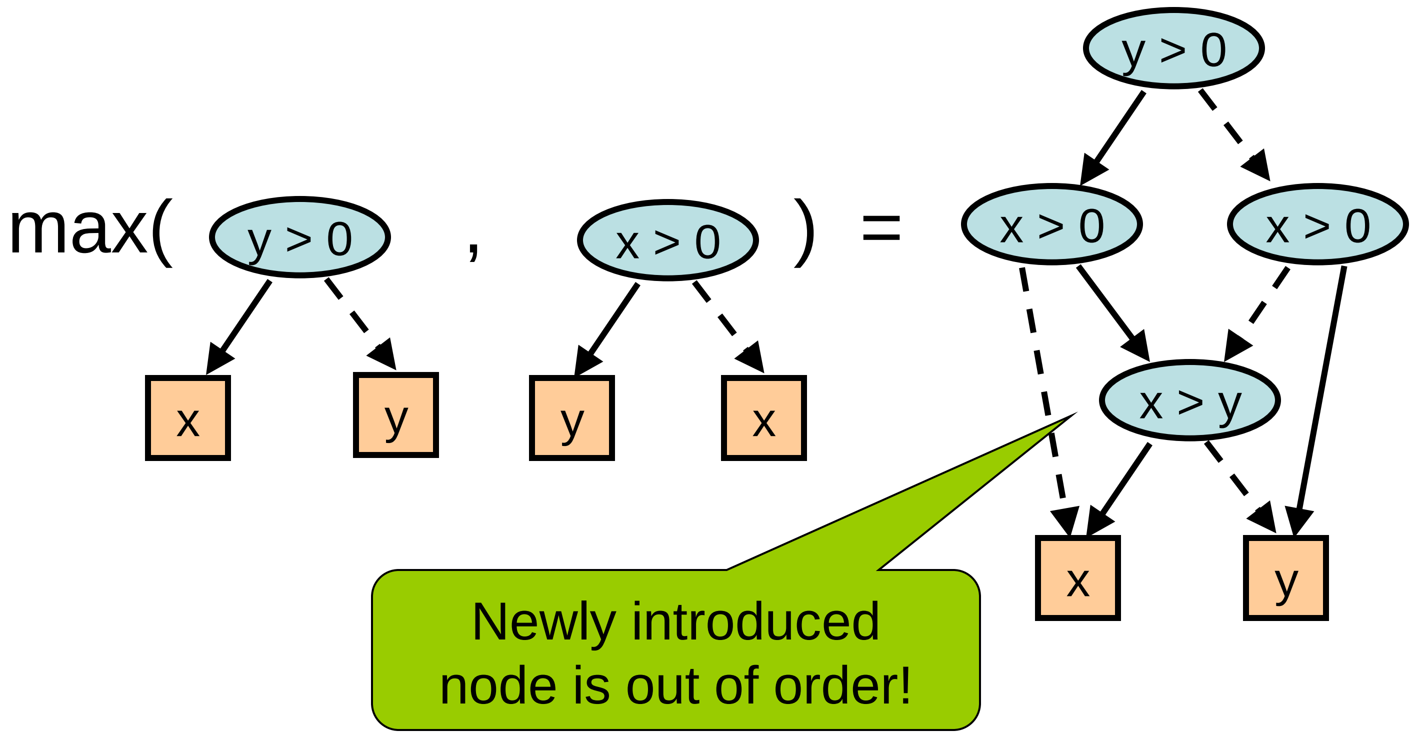
Operations exploit structure: O(|f||g|)

Maintaining XADD Orderings

- Max may get decisions out of order

Decision ordering (root→leaf)

- x > y
- y > 0
- x > 0

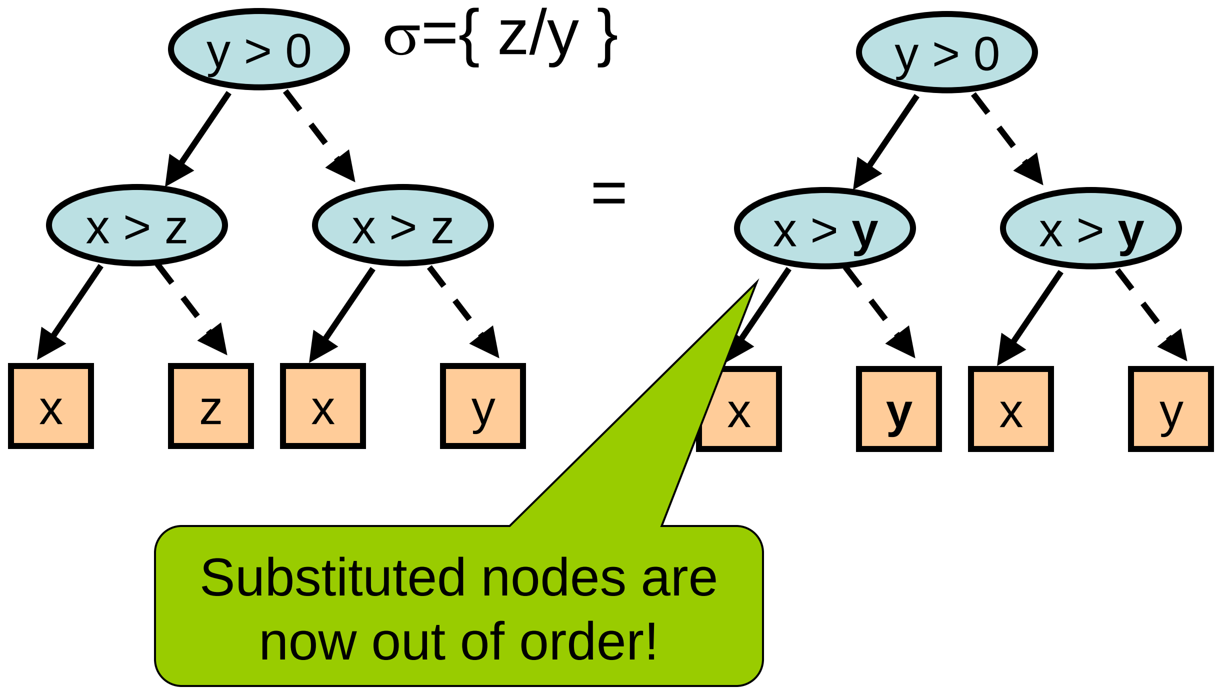


Maintaining XADD Orderings

- Substitution may get decisions out of order

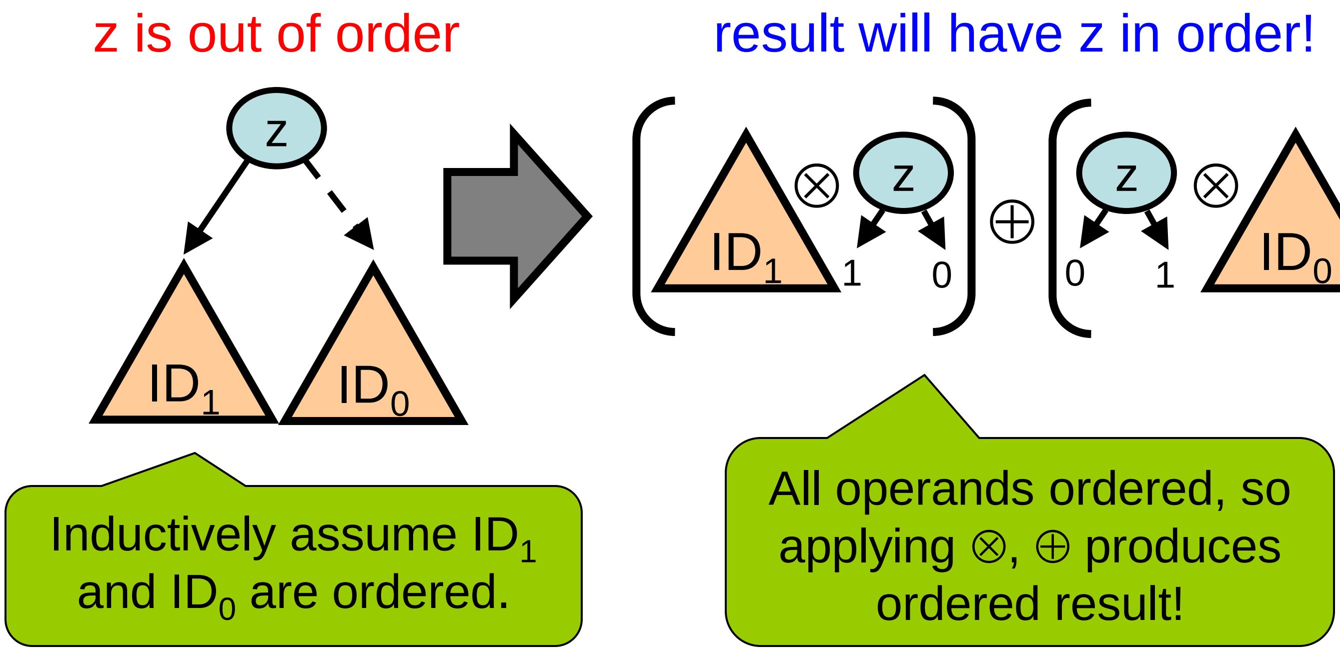
Decision ordering (root→leaf):

- x > y
- y > 0
- x > z

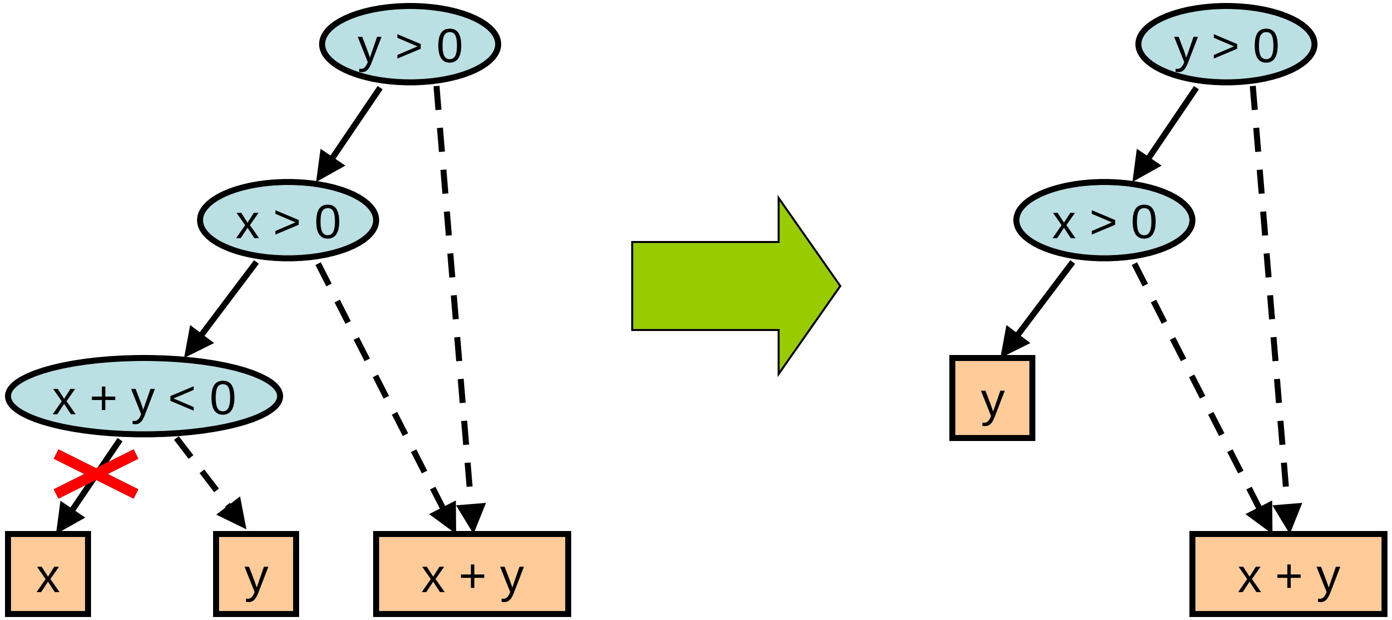


Correcting XADD Ordering

- Obtain *ordered* XADD from *unordered* XADD
  - key idea: binary operations maintain orderings



Maintaining Minimality



Node unreachable – x + y < 0 always false if x > 0 & y > 0

If **linear**, can detect with feasibility checker of LP solver & prune

More subtle prunings as well.

XADD Makes Possible all Previous Inference

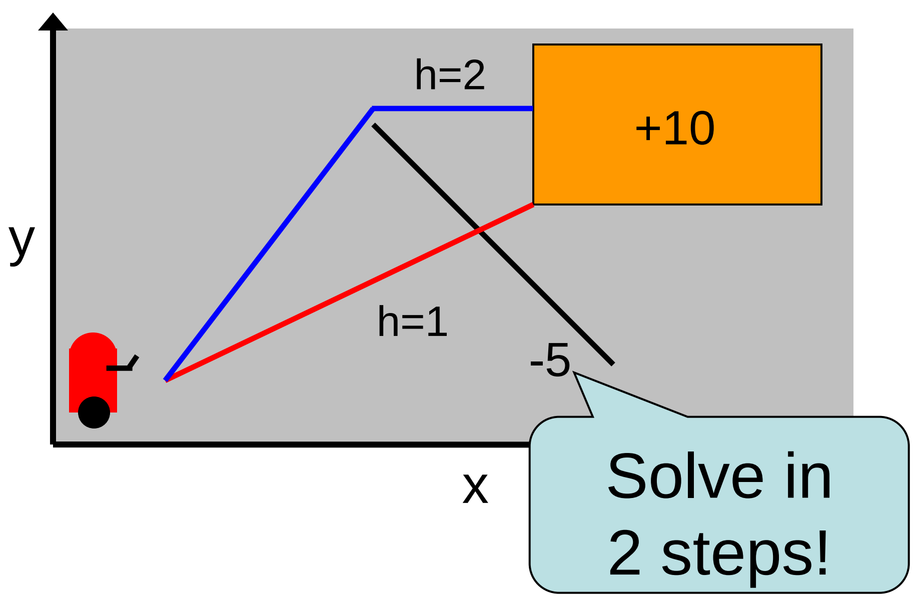
Could not even do a single integral or maximization without it!

Open Problems

Continuous Actions, Nonlinear

- **Robotics**
  - Continuous position, joint angles
  - Represent exactly with polynomials
    - Radius constraints

- **Obstacle Navigation**
  - 2D, 3D, 4D (time)
  - Don't discretize!
    - ~~Grid worlds~~
  - But nonlinear ☹

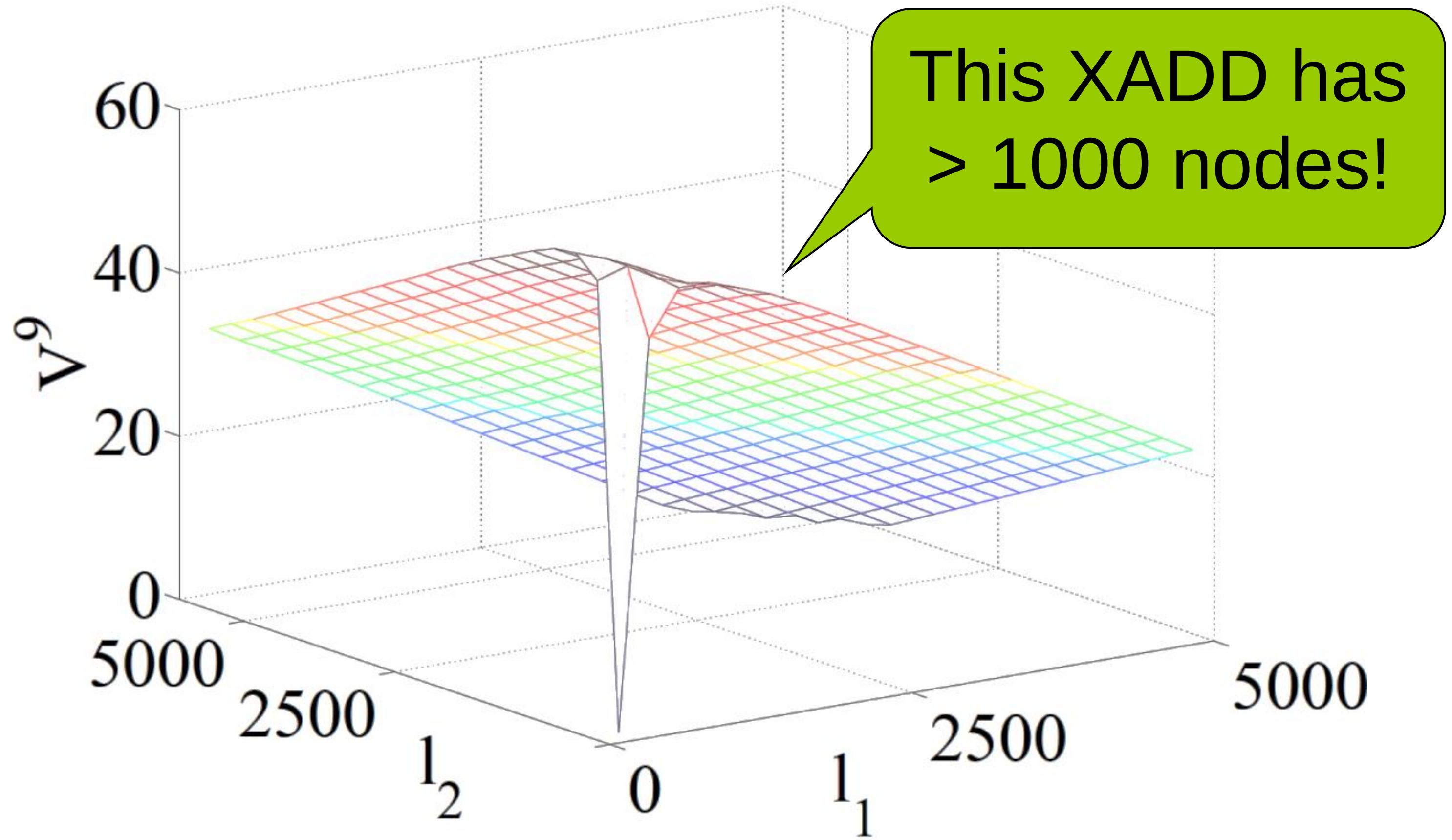


Multilinear, quadratic extensions.

In general: algebraic geometry.

Open Problems

- Bounded (interval) approximation



Recap

- Defined a calculus for piecewise functions
    - $f_1 \oplus f_2$, $f_1 \otimes f_2$
    - $\max(f_1, f_2)$, $\min(f_1, f_2)$
    - $\int_x f(x)$
    - $\max_x f(x)$, $\min_x f(x)$
- **Defined XADD to efficiently compute with cases**
- **Makes possible**
    - **Exact inference in continuous graphical models**
    - **New paradigms for optimization and sequential control**
    - **New formalizations of machine learning problems**

Symbolic Piecewise Calculus + XADD

= Expressive Continuous Inference, Optimization, & Control

Thank you!

Questions?