Learning from Constraints

Bridging perception and symbolic reasoning

Marco Gori
University of Siena (Italy)

WU - 6/1/2012

Acknowledgments

- Stefano Melacci
- Marco Maggini
- Michelangelo Diligenti
- Claudio Saccà
- Leonardo Rigutini
- Salvatore Frandina
- Marcello Sanguineti
- Giorgio Gnecco

Outline

- Environment and constraints
- Learning from (given) constraints
- Case studies
- Developmental agents

Environment and constraints

WU - 6/1/2012

Labeled examples are constraints ...

classic learning from examples

$$x \in \mathcal{X}$$

$$\epsilon - \theta_j \parallel y_j(x) - f_j(x) \parallel_p \geq 0$$

examples can be sets

$$\mathcal{E}_L = \{(\mathcal{X}_i, y_i) \in 2^{\mathcal{X}} \times \mathcal{Y}, i = 1, \ldots, m\}$$

as this becomes a box, the pair is a proposition

Diagnosis and prognosis in medicine

Pima Indian Diabetes Dataset

$$(MASS \geq 30) \wedge (PLASMA \geq 126) \Rightarrow positive$$

$$(MASS \leq 25) \wedge (PLASMA \leq 100) \Rightarrow negative$$

body mass index
blood glucose

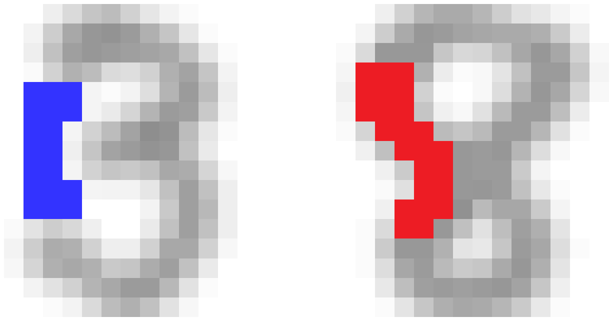
Wisconsin Breast Cancer Prognosis

$$(SIZE \geq 4) \wedge (NODES \geq 5) \quad \Rightarrow \quad recurrent$$

$$(SIZE \leq 1.9) \wedge (NODES = 0) \quad \Rightarrow \quad non\ recurrent$$

diameter of the tumor
number of metastasized lymph nodes

Handwritten char discrimination



GRAYLEVEL >220 in blue region $\Rightarrow$ 3

GRAYLEVEL <160 in red region $\Rightarrow$ $8.$

blue region: a selection of (blue) coordinates out of the 256=16*16

red region: a selection of (red) coordinates out of the 256=16*16

Text categorization

$$\text{graphic} \wedge \text{pixel} \wedge \text{bitmap} \Rightarrow \text{comp.graphics} \vee \text{comp.sys.ibm.pc.hardware}$$

keywords: input level

categories: decision level

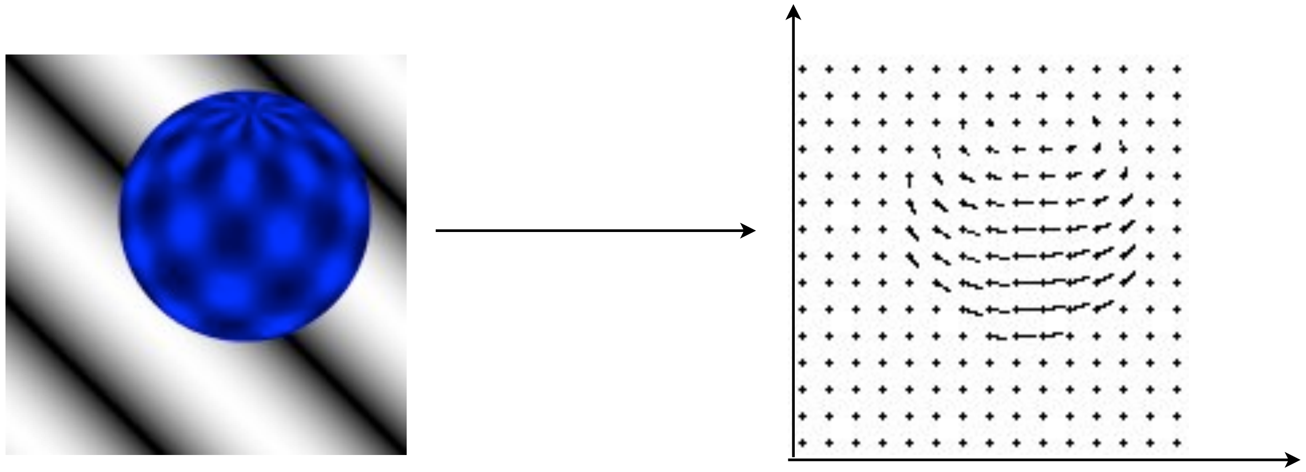
$i. \quad \forall x \forall y \;\; x \bowtie \; y \Leftrightarrow a(x) = a(y)$ ⟵ docs of the same author

$ii. \quad \forall x \; c_1(x) \wedge c_2(x) \Rightarrow c_3(x)$

$iii \quad \forall x \; c_3(x) \Rightarrow c_4(x).$

categories: decision level

Optical flow



$E(x, y, t)$ *is constant*

$u = \dot{x}, \quad v = \dot{y}$

$$\forall t \; \forall x \; \forall y \quad \frac{\partial E}{\partial x} u + \frac{\partial E}{\partial y} v + \frac{\partial E}{\partial t} = 0$$

(Logic) constraints

... given in terms of real numbers ...

$$\forall x \quad a(x) \wedge b(x) \Rightarrow c(x)$$

$$\neg(a(x) \wedge b(x)) \vee c(x))$$

$$\neg\neg(\neg(a(x) \wedge b(x)) \wedge c(x))$$

$$\neg(a(x) \wedge b(x) \wedge \neg c(x))$$

p-norm

$$1- \quad f_a(x) \cdot f_b(x) \cdot (1 - f_c(x)) = 1$$

$$f_a(x) f_b(x)(1 - f_c(x)) = 0$$

(Logic) constraints (con’t)

$$\forall x \quad a(x) \wedge b(x) \Rightarrow c(x)$$

$$\neg(a(x) \wedge b(x) \wedge \neg c(x))$$

Gödel T-norm

$$1 - min\,\{f_a(x), f_b(x), 1 - f_c(x)\} = 1$$

$$min\,\{f_a(x), f_b(x), 1 - f_c(x)\} = 0$$

Equivalence

$$\begin{aligned}\check{\phi}_1(f,y) &= \epsilon - |y-f| \geq 0 \qquad f \in \check{\mathcal{F}}_1 \\ \check{\phi}_2(f,y) &= \epsilon^2 - (y-f)^2 \geq 0 \qquad f \in \check{\mathcal{F}}_2\end{aligned}$$

the same admissible functional space! $\check{\mathcal{F}}_1 = \check{\mathcal{F}}_2$

$$\check{\phi}_1 \sim \check{\phi}_2 \Leftrightarrow \check{\mathcal{F}}_1 = \check{\mathcal{F}}_2$$

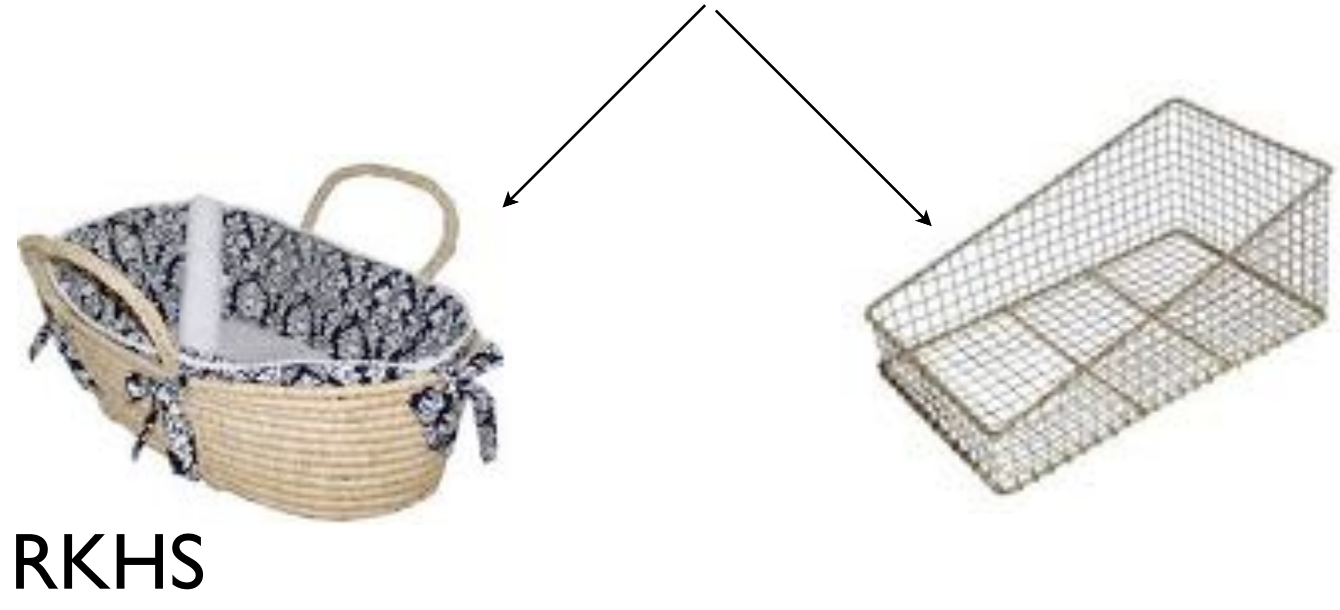
$$\underset{\substack{\uparrow \\ \text{quotient set}}}{\mathcal{F}/\sim} = \{\phi \in \mathcal{F}:\ \phi \sim \underset{\substack{\uparrow \\ \text{representer}}}{[\phi]}\}$$

Learning from (given) constraints

Learning: "the simplest solution" compatible with the constraints

Ambient space

Where do I look for the solution?



$$\mathcal{X} \subset I\!R^d \quad f = [f_1, \ldots, f_n]' \quad f_j : \mathcal{X} \rightarrow I\!R$$

$$\forall j \in I\!N_n : \quad f_j \in W^{k,p}$$

$$\mathcal{X}^\star$$

Search in Sobolev spaces:
it is related to the topic of learning kernels!

Semi-norm in Sobolev spaces

... parsimony principle in learning ...

$$P = \sum_{|\alpha|<m} a_\alpha D_x^\alpha = \sum_{|\alpha|<m} a_\alpha \left(\frac{\partial}{\partial x_1} + \ldots + \frac{\partial}{\partial x_d}\right)^\alpha$$

∞

$a_\alpha \in C^\infty$

under proper boundary conditions ...

$$P = \sum_{h=0}^{m} a_h \sum_{|\alpha|=h} \frac{h!}{\alpha!} \left(\frac{\partial}{\partial x}\right)^\alpha$$

$$P^\star = \sum_{h=0}^{m} (-1)^h a_h \sum_{|\alpha|=h} \frac{h!}{\alpha!} \left(\frac{\partial}{\partial x}\right)^\alpha$$

Given P and $\gamma_i > 0, \ldots, i = 1, \ldots, n$

$$E(f) = \| f \|_{P,\gamma} = \sum_{j=1}^{n} \gamma_j < Pf_j, Pf_j > = \sum_{j=1}^{n} \gamma_j < f_j, P^\star P f_j > = \sum_{j=1}^{n} \gamma_j < f_j, L f_j >$$

Parsimony Principle

Occam’s razor, lex parsimoniae

$\mathcal{F}_\phi$ admissible w.r.t the collection of constraints $\mathcal{C}_\phi$

$$f^\star = argmin_{f \in \mathcal{F}_\phi} \parallel f \parallel_{P,\gamma}$$

strictly (hard)

partially (soft)

check of a “new” constraint

$$\forall x \quad \phi(x, f^\star(x), Df^\star(x)) = 0 \ ?$$

Representation (hard constraints)

from constrained variational calculus

$$\forall x \in \mathcal{X}_i \subset X : \phi_i(x, f(x)) = 0, \; i \in \mathbb{N}_m \qquad \frac{D(\phi_1, \ldots, \phi_m)}{D(f_1, \ldots, f_m)} \neq 0$$

$$\mathcal{L}(f) = \| f \|^2_{P,\gamma} + \sum_{i=1}^{m} \int_{\mathcal{X}} \lambda_i(x) \cdot \phi_i(x, f(x)) dx$$

Lagrangian approach

$$Lf(x) + \sum_{i=1}^{m} \lambda_i(x) \cdot \nabla_f \phi_i(x, f(x)) = 0$$

Euler-Lagrange equations

$$Lg = \delta$$

Green function

$$\omega_i(\cdot) = -\lambda_i(\cdot) \nabla_f \phi_i(\cdot, f^\star(\cdot))$$

reaction of the constraint

support constraints

$$f^\star(\cdot) = \sum_{i=1}^{m} g(\cdot) \otimes \omega_i(f^\star(\cdot))$$

Fredholm eq. (II kind)

"merging of two ideas ..."

Lagrange multipliers and probability density

Unilateral constraints

hard constraints

$$\forall x \in \mathcal{X}_i \subset X : \phi_i(x, f(x)) = 0, \; i \in I\!N_m$$

$$\mathcal{L}(f) = \| f \|^2_{P,\gamma} + \sum_{i=1}^{m} \int_{\mathcal{X}} \lambda_i(x) \check{\phi}_i(x, f(x)) dx$$

soft constraints

$$\mathcal{L}(f) = \| f \|^2_{P,\gamma} + C \sum_{i=1}^{m} \int_{\mathcal{X}} p_i(x) \check{\phi}_i(x, f(x)) dx$$

Two remarkable "examples"

Optical flow *(Horn and Schunck (1981))*

$$\frac{\partial E}{\partial x}u + \frac{\partial E}{\partial y}v + \frac{\partial E}{\partial t} = 0$$

$$\int_X \int_X \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \right] dxdy$$

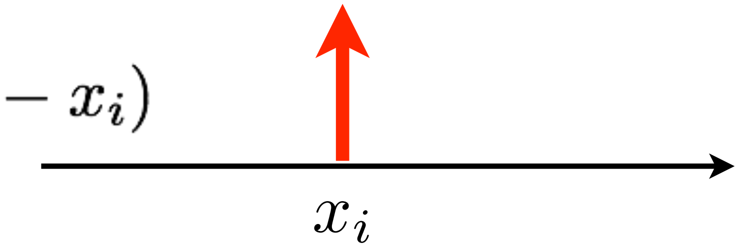
Learning from examples *(Poggio and Girosi (1989))*

$$E(f) = C \sum_{i=1}^{m} V(x_i, f(x_i)) + \frac{1}{2} < Pf, Pf >$$

Where do kernel machines come from ...

$$Lf^\star + C\sum_{i=1}^{m} V_f'(x_i, f^\star(x_i))\delta(x - x_i) = 0$$

$$\omega_i(f^\star(x)) \quad = \quad -C \cdot V_f'(x_i, f^\star(x_i)) \cdot \delta(x - x_i)$$



reaction of the constraint

$$f^\star(x) = \sum_{i=1}^{m} \alpha_i g(x, x_i)$$

finite convolution

When kernels arise from regularization operators ...

$Lg = \delta$ Green function / “plain kernel”

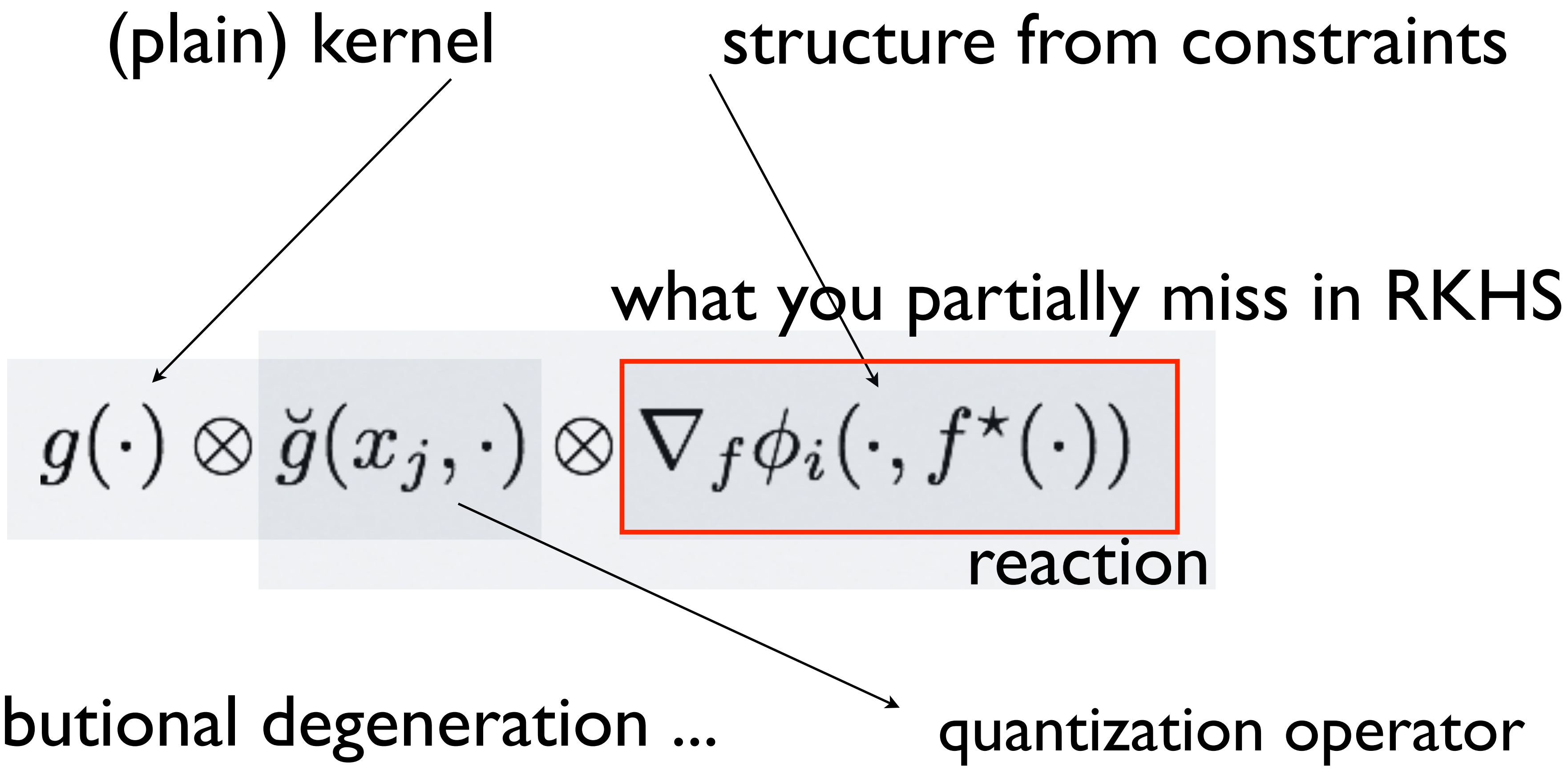
$L = d^4/dx^4$ $g(x) = |x|^3$

$L = (\sigma^2 I - \nabla^2)^n$ Sobolev spline kernel

$L = \sum_{\kappa=0}^{\infty} (-1)^\kappa \frac{\sigma^{2\kappa}}{\kappa! 2^\kappa} \nabla^{2\kappa}$ Gaussian kernel

Polynomial kernels don’t come from regularization operators!

New Kernels (from prior knowledge)



(plain) kernel

structure from constraints

what you partially miss in RKHS

$$g(\cdot) \otimes \breve{g}(x_j, \cdot) \otimes \boxed{\nabla_f \phi_i(\cdot, f^\star(\cdot))}$$

reaction

quantization operator

... distributional degeneration ...

$$\breve{g}(x_j, x) = \delta(x - x_j)$$

$$\begin{aligned} f^\star(\cdot) &= -\sum_{i=1}^{m}\sum_{j=1}^{\ell} \lambda_j \cdot g(\cdot) \otimes \delta(\cdot - x_j) \otimes \nabla_f \phi_i(\cdot, f^\star(\cdot)) \\ &= -\sum_{i=1}^{m}\sum_{j=1}^{\ell} \lambda_j \cdot g(\cdot - x_j) \otimes \nabla_f \phi_i(\cdot, f^\star(\cdot)) \end{aligned}$$

Putting things in context by unsupervised data

i. more accurate description of reality
ii. simplified math and the birth of $\forall_p$

Algorithmic issues

kernel-based representation

$$g(\cdot) \otimes \breve{g}(x_j, \cdot) \otimes \nabla_f \phi_i(\cdot, f^\star(\cdot))$$

kernel machines math &
algorithmic apparatus

fixed-point iteration algorithms

$$\omega_i(f^\star) = -\lambda_i(\cdot)\nabla_f \phi_i(\cdot, f^\star(\cdot))$$

$$f^\star = \sum_{i=1}^{m} g \otimes \omega_i(f^\star)$$

$$\forall \mathcal{X}_i \quad \phi_i(x, f^\star(x)) = 0$$

Reduction to kernels

direct computation of the constraint reaction

- Learning from examples
- Learning from sets (propositions) - box kernels
- Linear constraints
- Quadratic constraints (Fredholm linear equation)
- Poly constraints via auxiliary functions & quadratic constraints

Case studies

Where one faces the problem of determining the constraint reactions ...

WU - 6/1/2012

Preliminary “walk-through”

- diagnosis, prognosis in medicine (Pima Indian Diabetes Dataset, Wisconsin Breast Cancer Prognosis)
- handwritten chars (USPST)
- tagging (Flickr)
- asset allocation in finance

Multi-intervals

Back to kernels (under some hyp)!

$$\phi_i(x, f(x)) := max\left\{0, 1 - y_i f(x_i)\right\} \cdot c_{\mathcal{X}_i}(x)$$

$$\omega_i = -\lambda \nabla \phi(x, f(x)) \propto c_{\mathcal{X}_i}(x)$$

sign consistency   uniform weight reaction

$$g \otimes \boxed{c_{\mathcal{X}_i}} \leftarrow \text{constraint reaction}$$

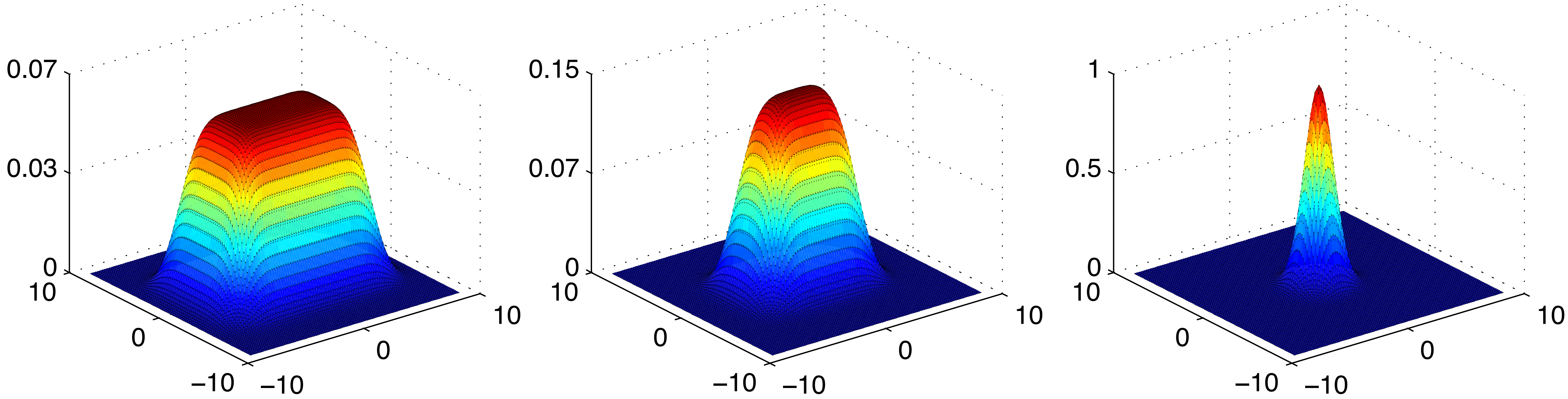
box kernels!

the case of soft-constraints ...

$g \otimes c\chi_i$

Box kernels

Gaussian (plain kernel) $\Longrightarrow$ $\prod_{i=1}^{d} \frac{(\sqrt{2\pi}\sigma)}{2}(erfc(\frac{x^i - b^i_j}{\sqrt{2}\sigma}) - erfc(\frac{x^i - a^i_j}{\sqrt{2}\sigma}))$

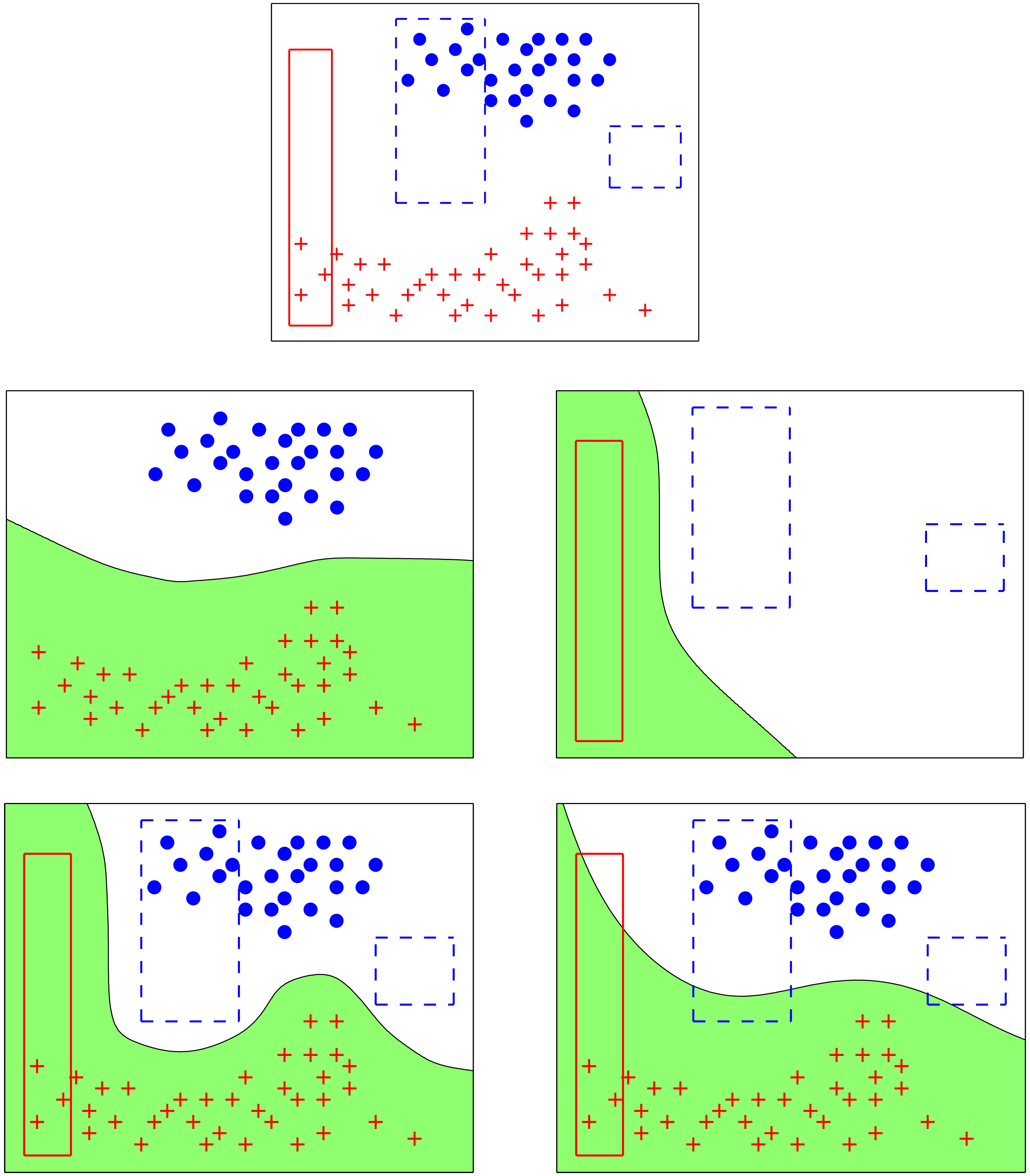


degeneration to the Gaussian

$[-6,-4] \times [6,4]$ $[-3,-2] \times [3,2]$ $(0,0)$

plain kernel (Gaussian) + multi-interval knowledge = box kernel

response to the “rectangular impulse”



points only

boxes only

changing the regularization parameter

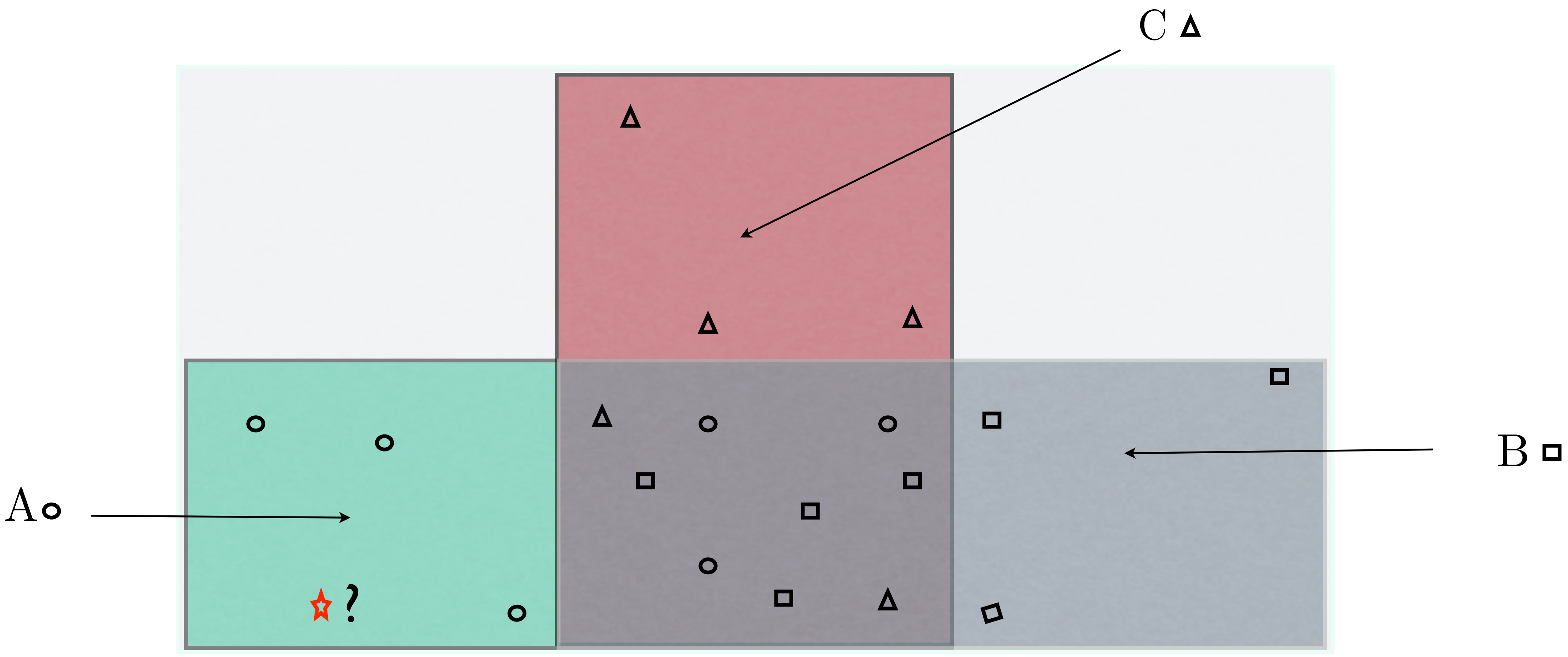
WU - 6/1/2012

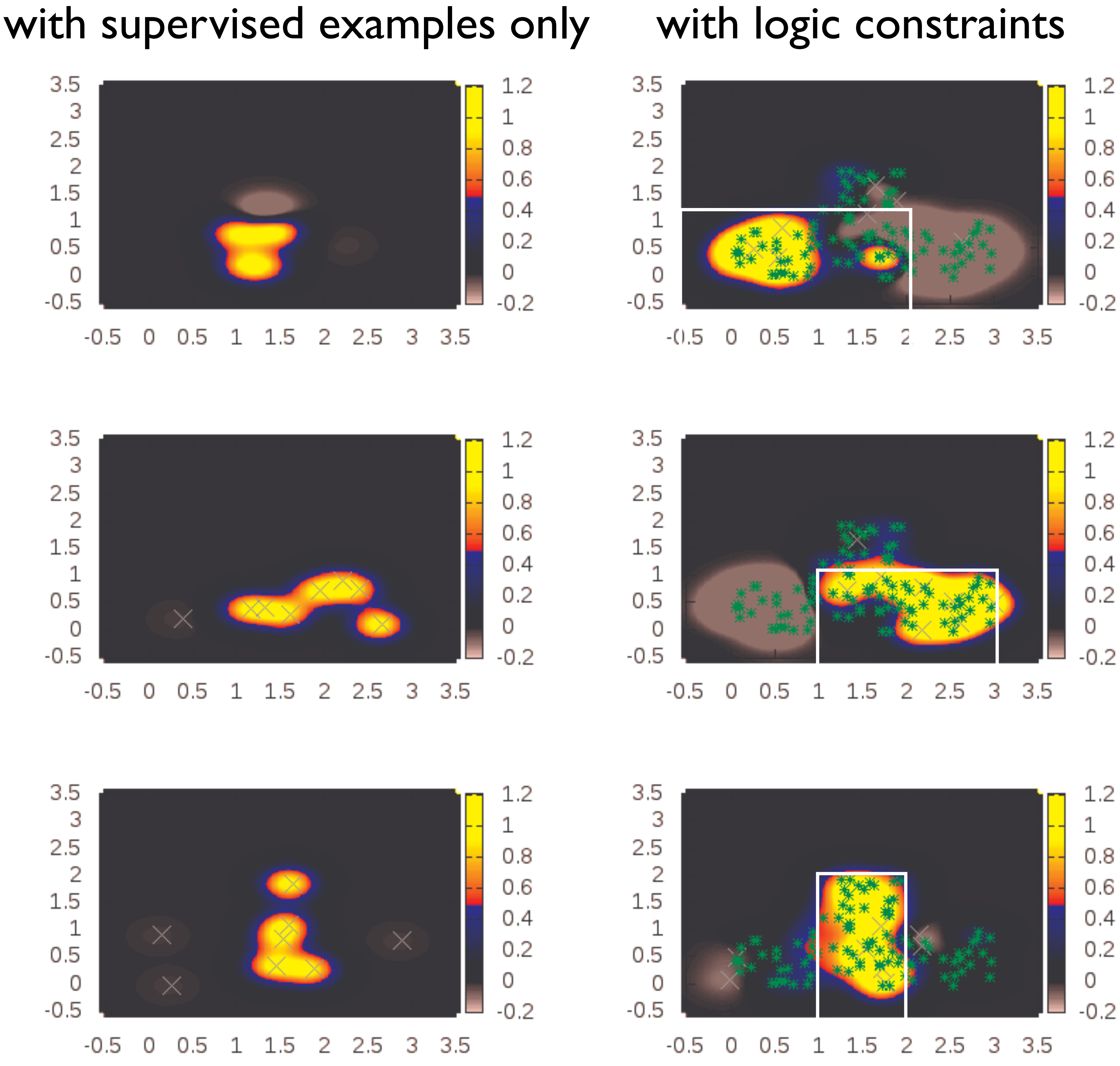
Perceptual and logic constraints ...

Back to kernels (soft-constraints)

$$A = \{(x_1, x_2) \in R^2 : \quad 0 \leq x_1 < 2, \quad 0 \leq x_2 \leq 1\}$$
$$B = \{(x_1, x_2) \in R^2 : \quad 1 \leq x_1 < 3, \quad 0 \leq x_2 \leq 1\}$$
$$C = \{(x_1, x_2) \in R^2 : \quad 1 \leq x_1 < 2, \quad 0 \leq x_2 \leq 2\}$$

$$A \wedge B \implies C$$
$$A \vee B \vee C$$





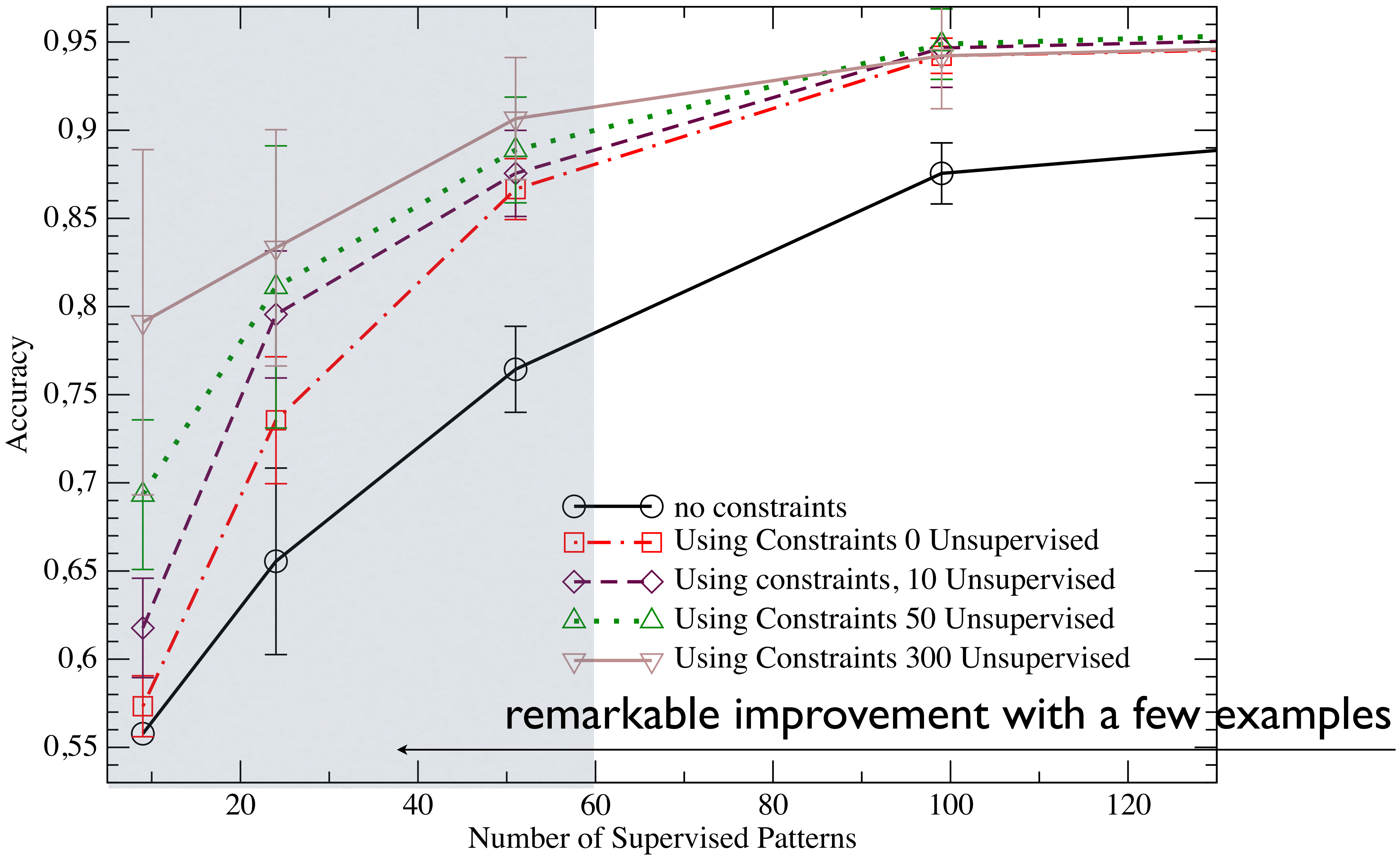
with supervised examples only

with logic constraints

(a)

(b)

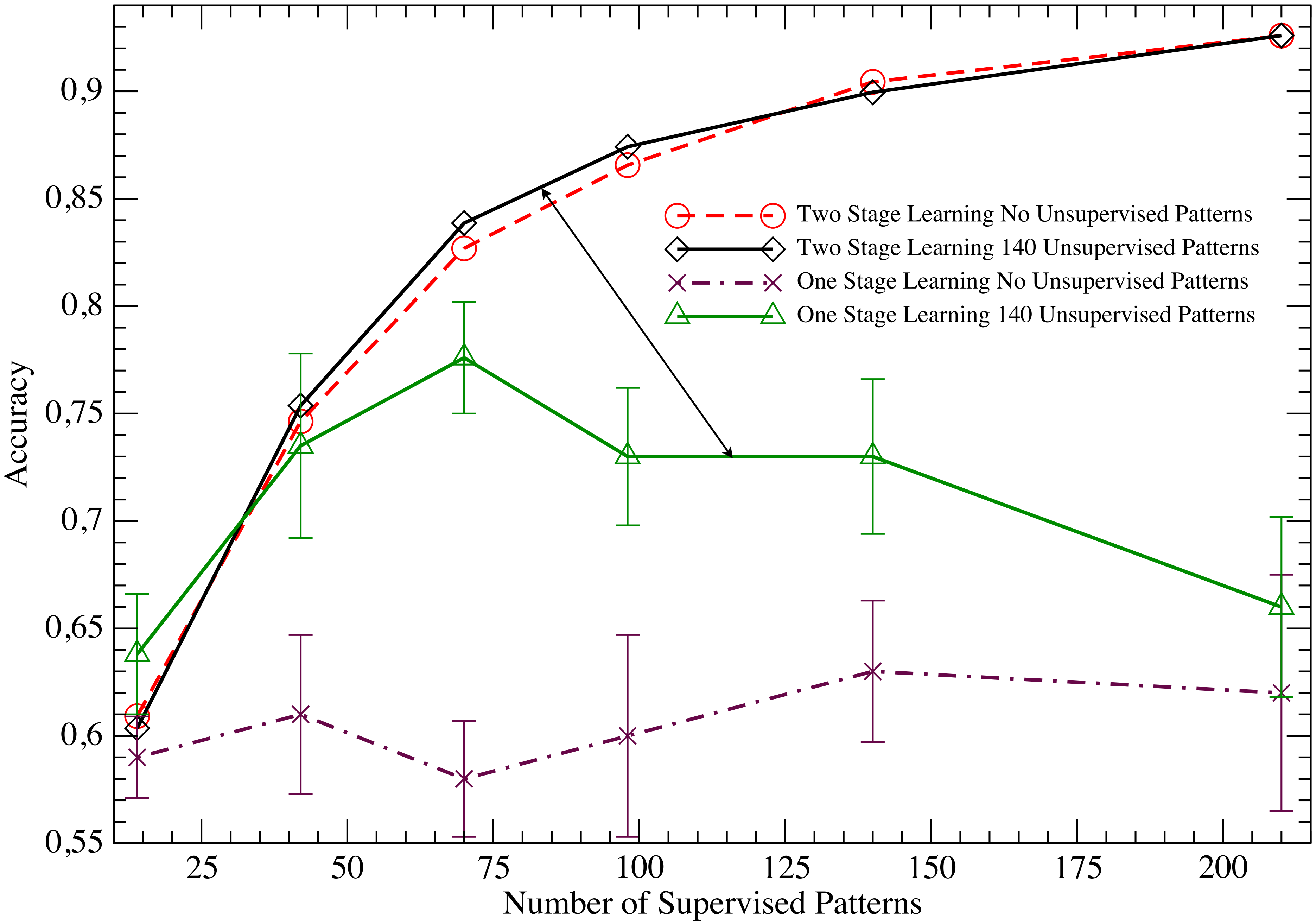
The effect of forcing logic constraints



remarkable improvement with a few examples

WU - 6/1/2012

Two-stages ...



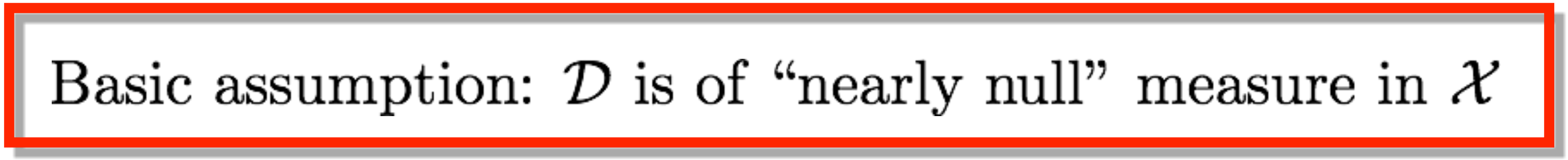
WU - 6/1/2012

Constraint Check and Perceptual Logic

check of a new constraint $\mathcal{C} \models \phi$

$$\forall x \quad \phi(x, f^{\star}(x)) = 0$$

$$\begin{aligned} \| \, \phi(\cdot, f^{\star}(\cdot)) \, \|^2 \quad &= \quad \left(\int_{\mathcal{X}} \phi^2(x, f^{\star}(x) dx \right) \\ &\propto \quad \sum_{x_\kappa \in \mathcal{D}} \phi^2(x_\kappa, f^{\star}(x_\kappa)) \end{aligned}$$



Basic assumption: $\mathcal{D}$ is of "nearly null" measure in $\mathcal{X}$

Facing the intractability coming from formal logic formal

Poly check

$$\begin{array}{rcl} \phi_1(f(x)) & = & f_2(x)f_4(x) - f_3(x) + 1 = 0 \\ \phi_2(f(x)) & = & f_1(x)f_3(x) + f_2^2(x) + 6 = 0 \qquad \Longrightarrow \\ \phi_3(f(x)) & = & f_1^2(x) - f_4(x) = 0 \\ \hline \end{array}$$

$\forall x$ (formal check)

$f_2^2(x) + f_1(x)f_2(x)f_4(x) + f_1(x) + 6 = 0$?

$\phi_1 : \; f_3 = 1 + f_2 f_4$

$$f_1 f_3 = f_1 + f_1 f_2 f_4$$

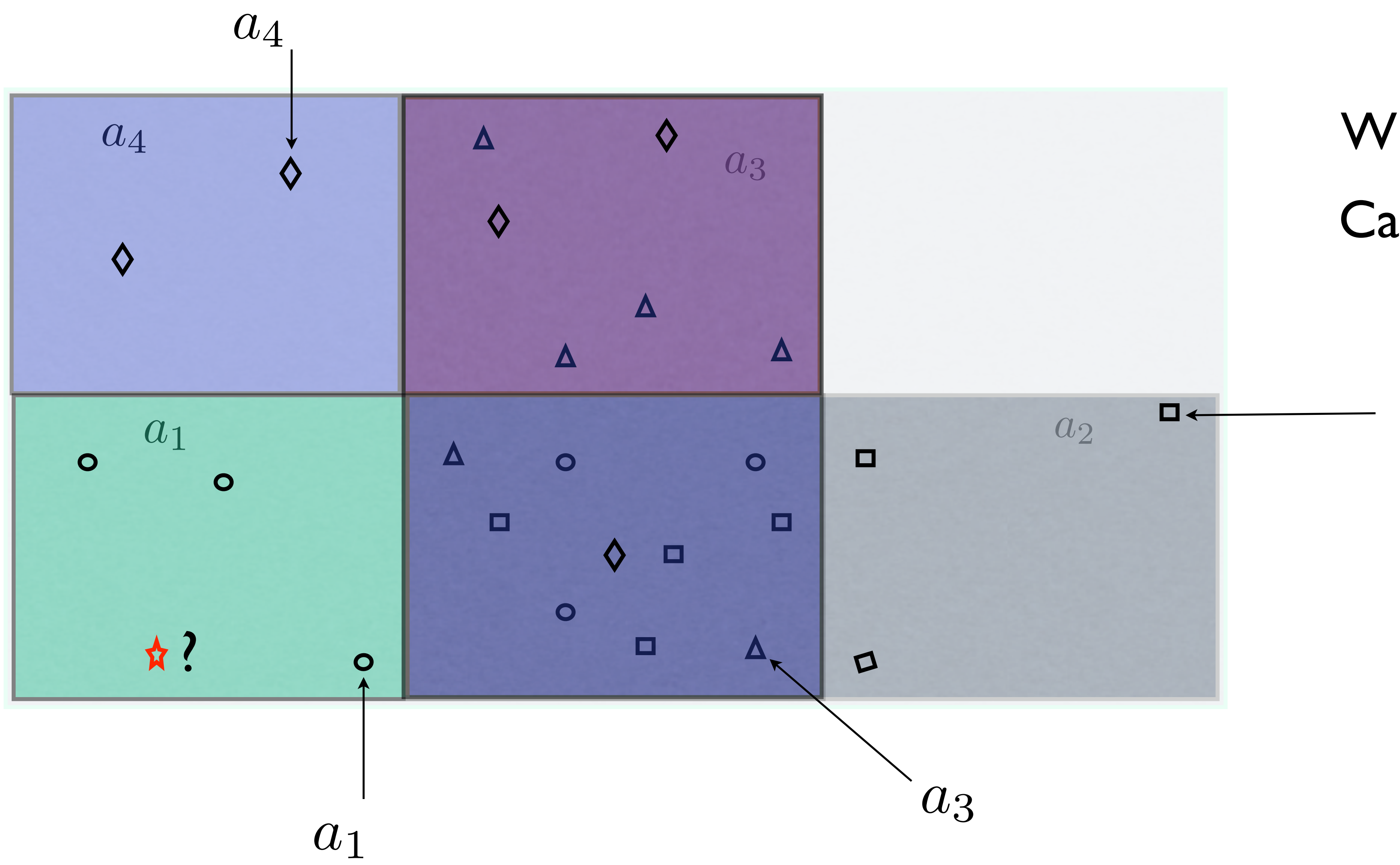
$\phi_2 : \; f_2^2 + f_1 f_2 f_4 + f_1 + 6 = 0$ ok

Learning from points and clauses

$$A = \left\{(x_1, x_2) \in R^2 : \quad 0 \leq x_1 < 2, \quad 0 \leq x_2 \leq 1\right\}$$
$$B = \left\{(x_1, x_2) \in R^2 : \quad 1 \leq x_1 < 3, \quad 0 \leq x_2 \leq 1\right\}$$
$$C = \left\{(x_1, x_2) \in R^2 : \quad 1 \leq x_1 < 2, \quad 0 \leq x_2 \leq 2\right\}$$
$$D = C \cup \left\{(x_1, x_2) \in R^2 : 0 \leq x_1 \leq 1, \ 1 \leq x_2 \leq 2\right\}$$

“Knowledge Base”

$$a_1(x) \ \wedge \ a_2(x) \ \Rightarrow \ a_3(x)$$
$$a_3(x) \ \Rightarrow \ a_4(x)$$
$$a_1(x) \ \vee \ a_2(x) \ \vee a_3(x)$$



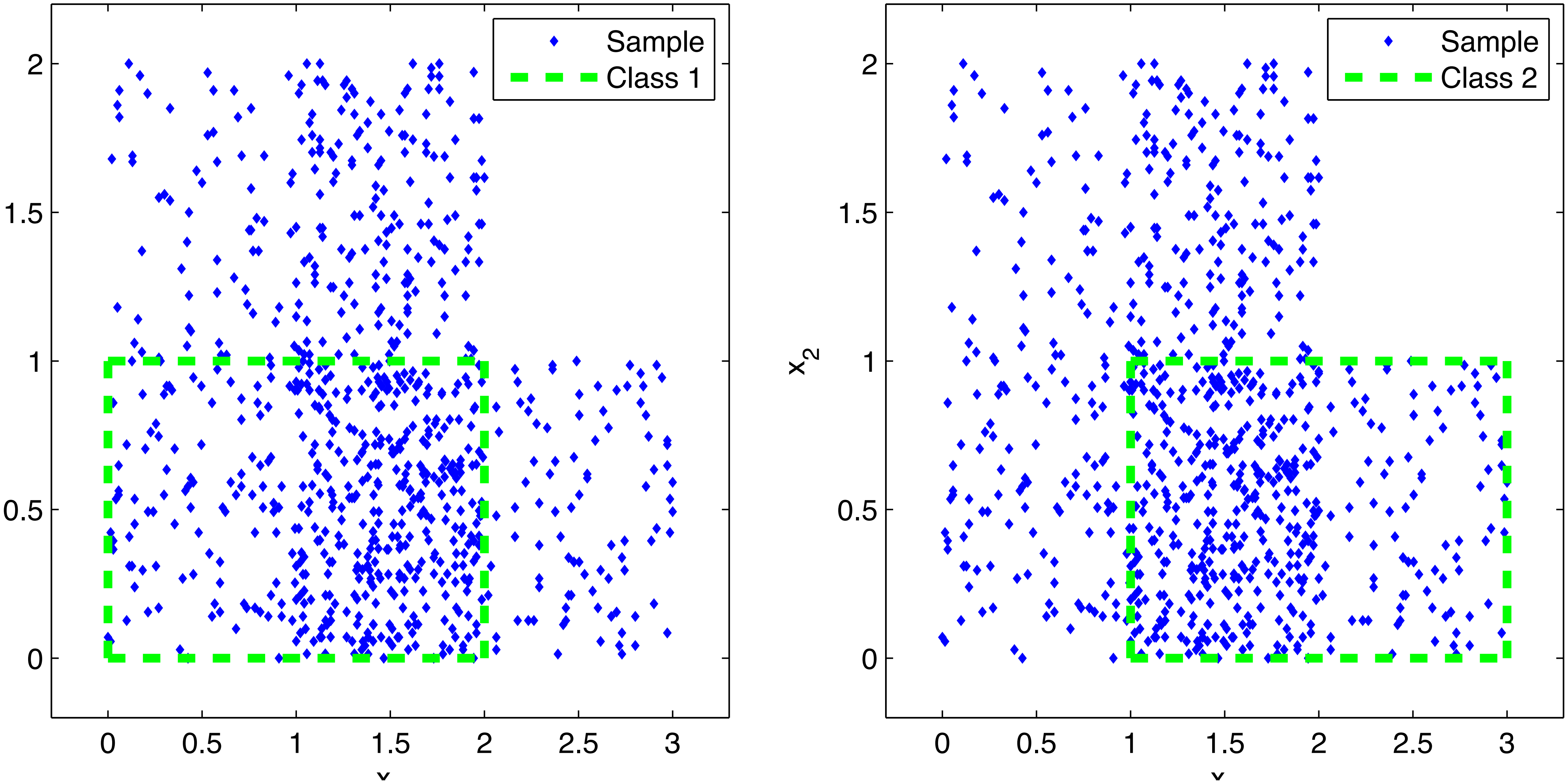
What can I deduce?

Can the points help deduction?

$$\mathcal{C} \models \phi$$

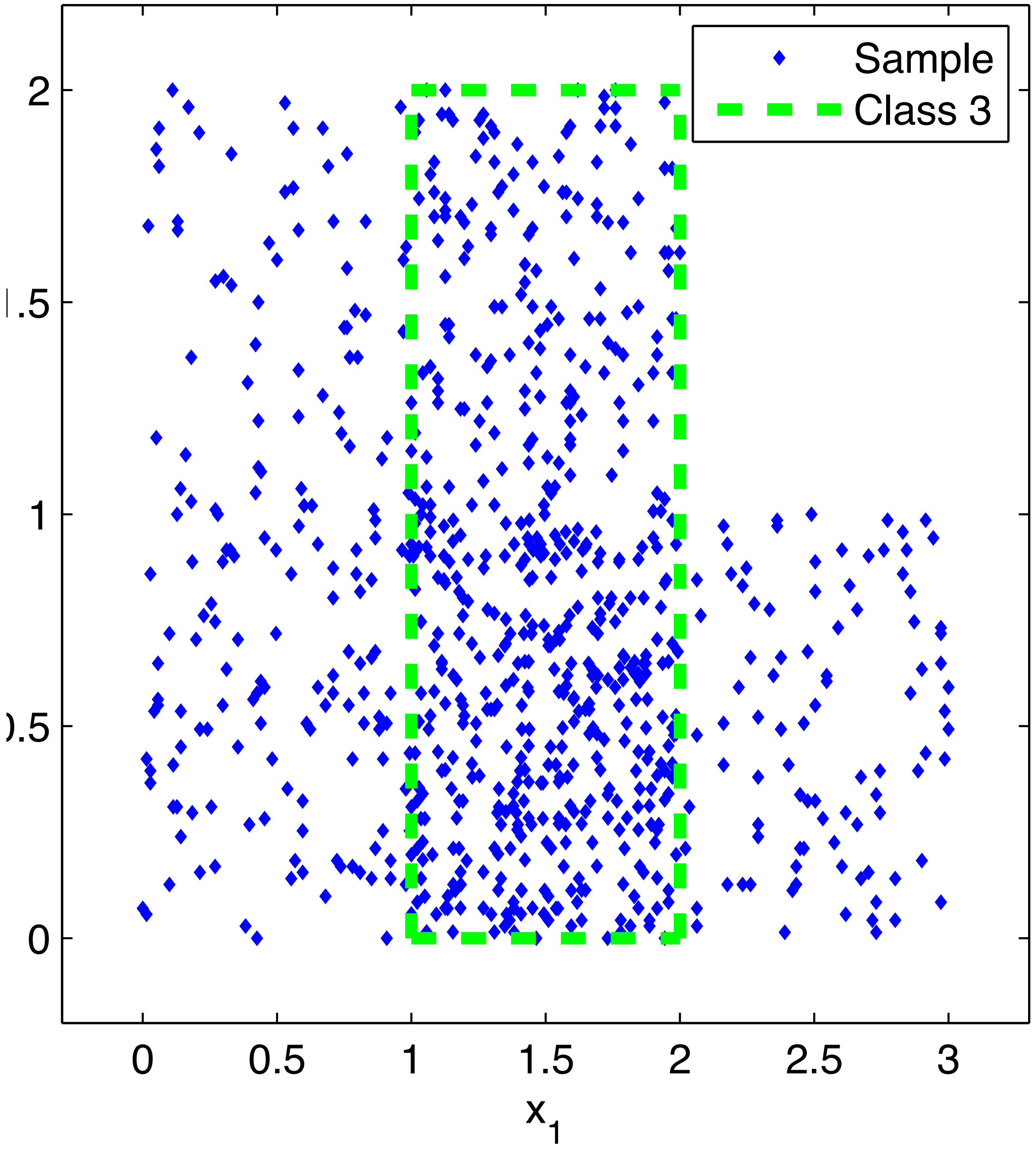
WU - 6/1/2012

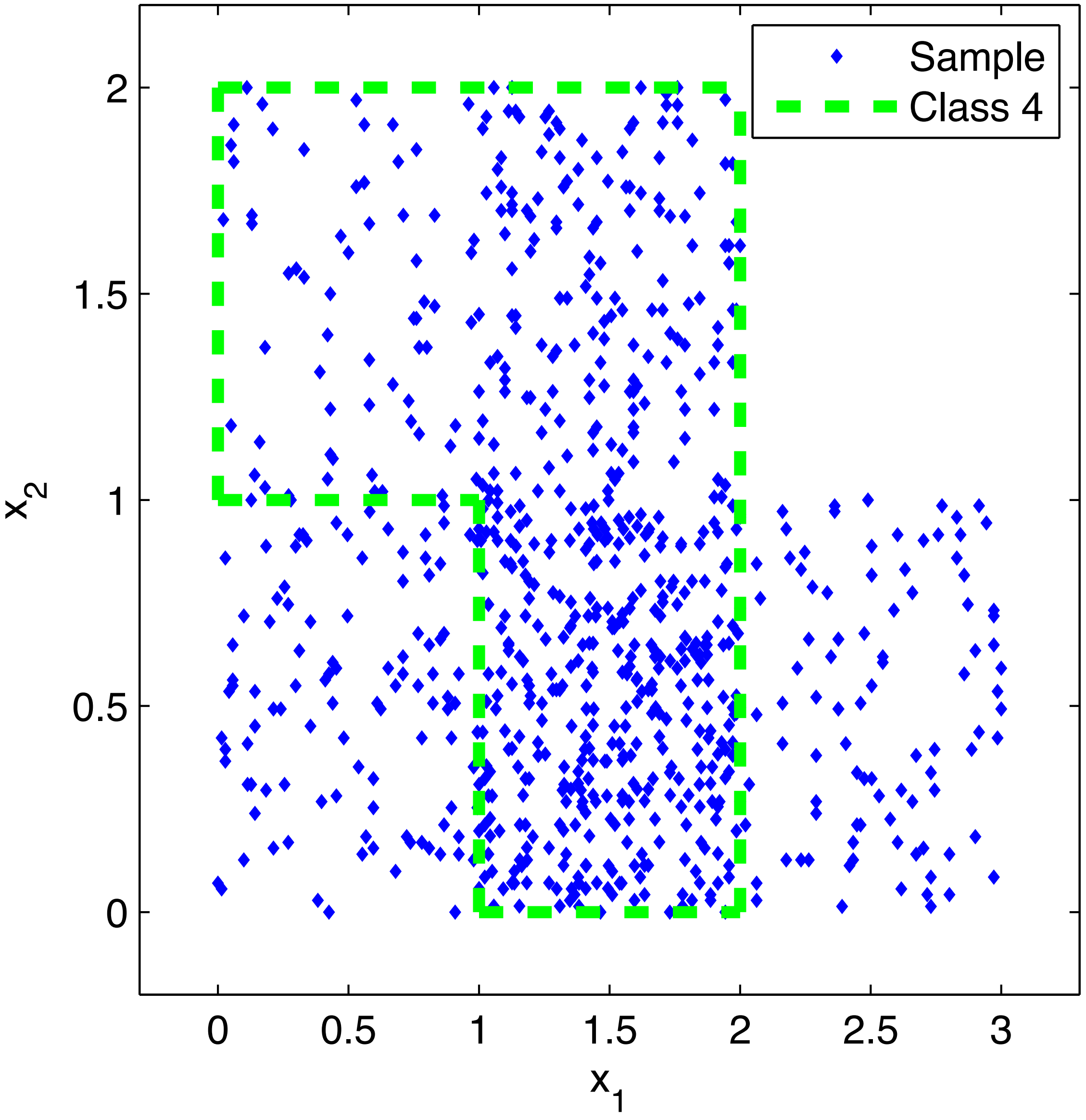
Checking (logic) constraints



WU - 6/1/2012

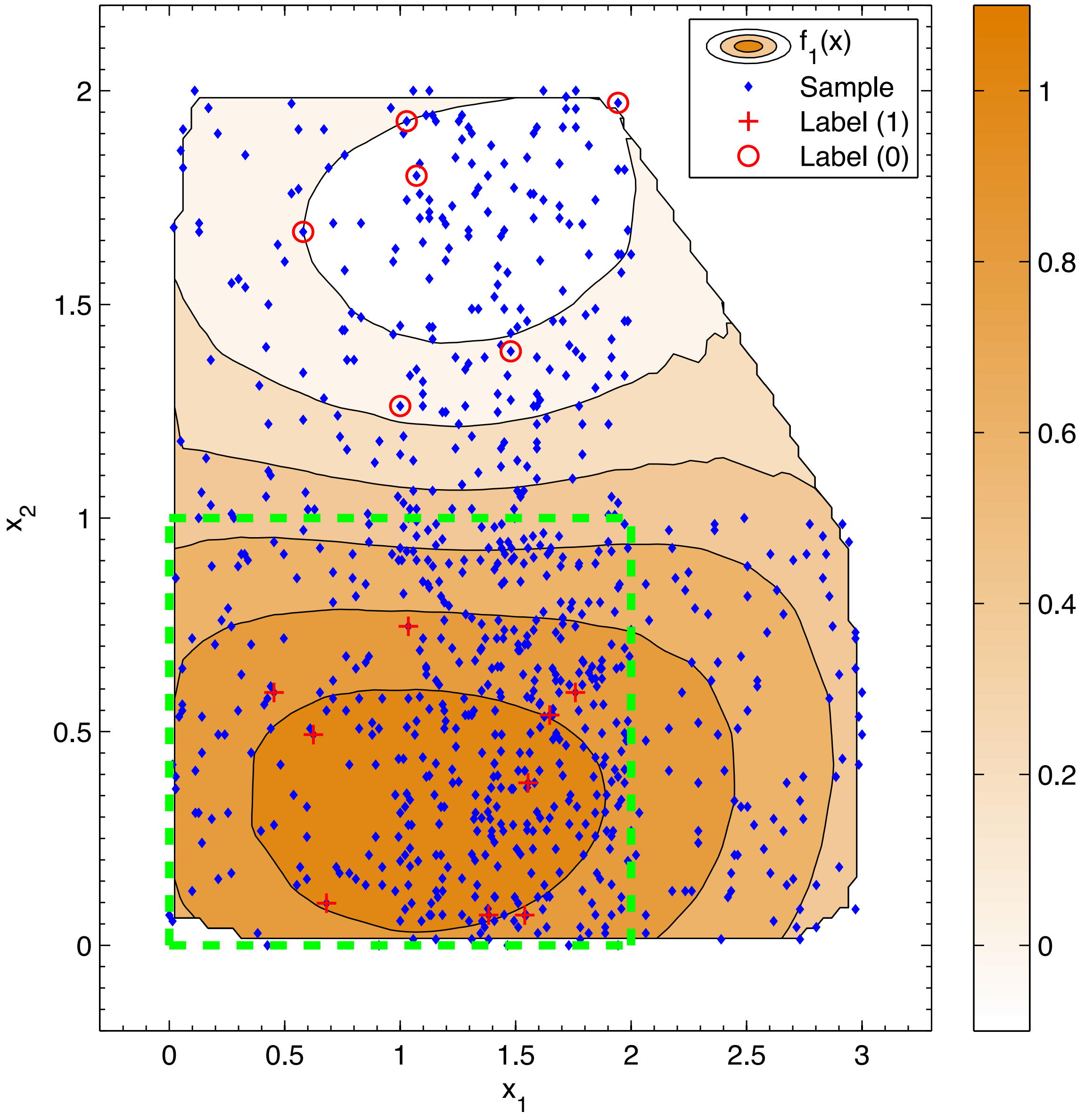
Checking (logic) constraints



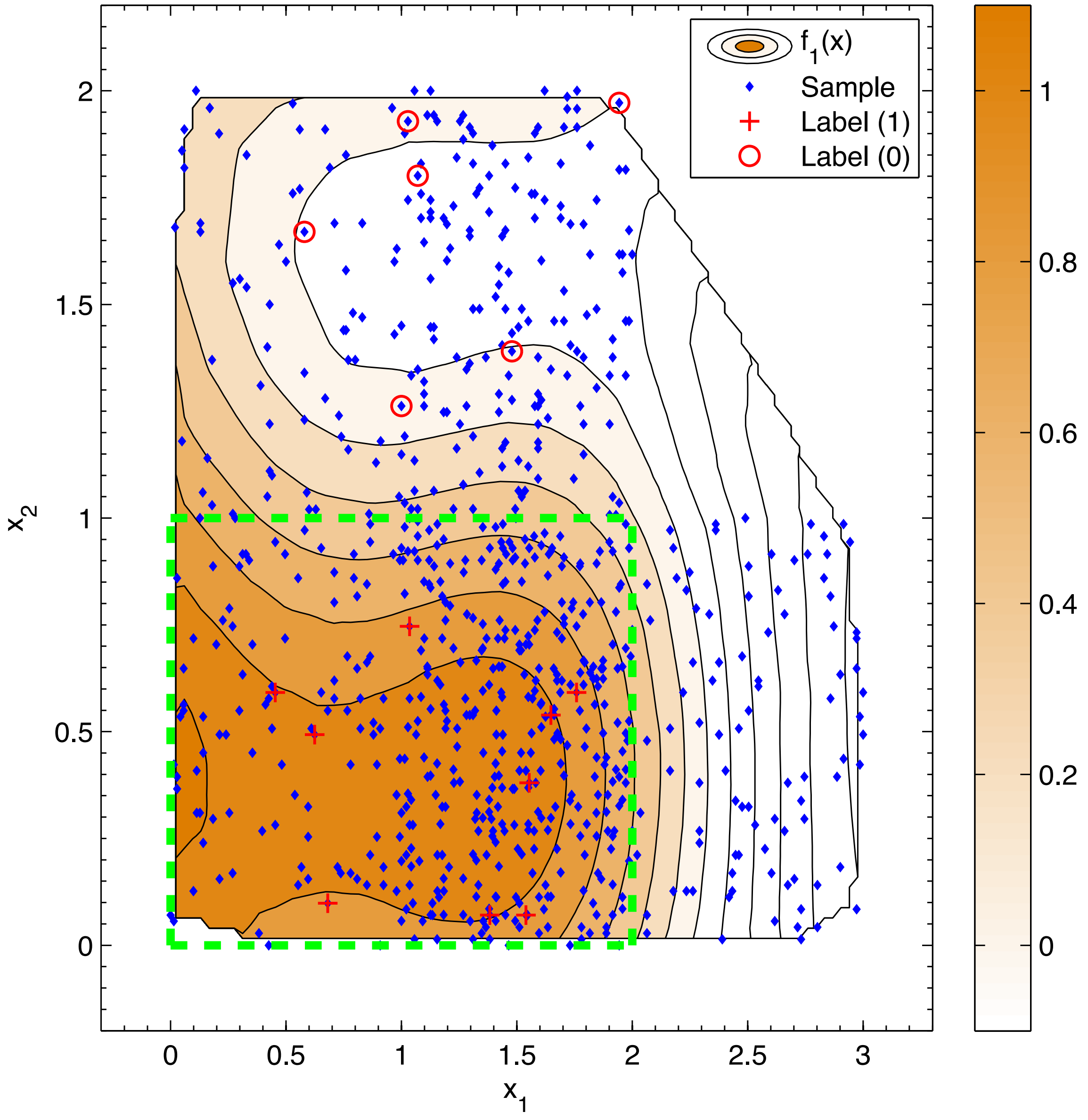


$$a_1(x) \rightsquigarrow f_1(x)$$

points only

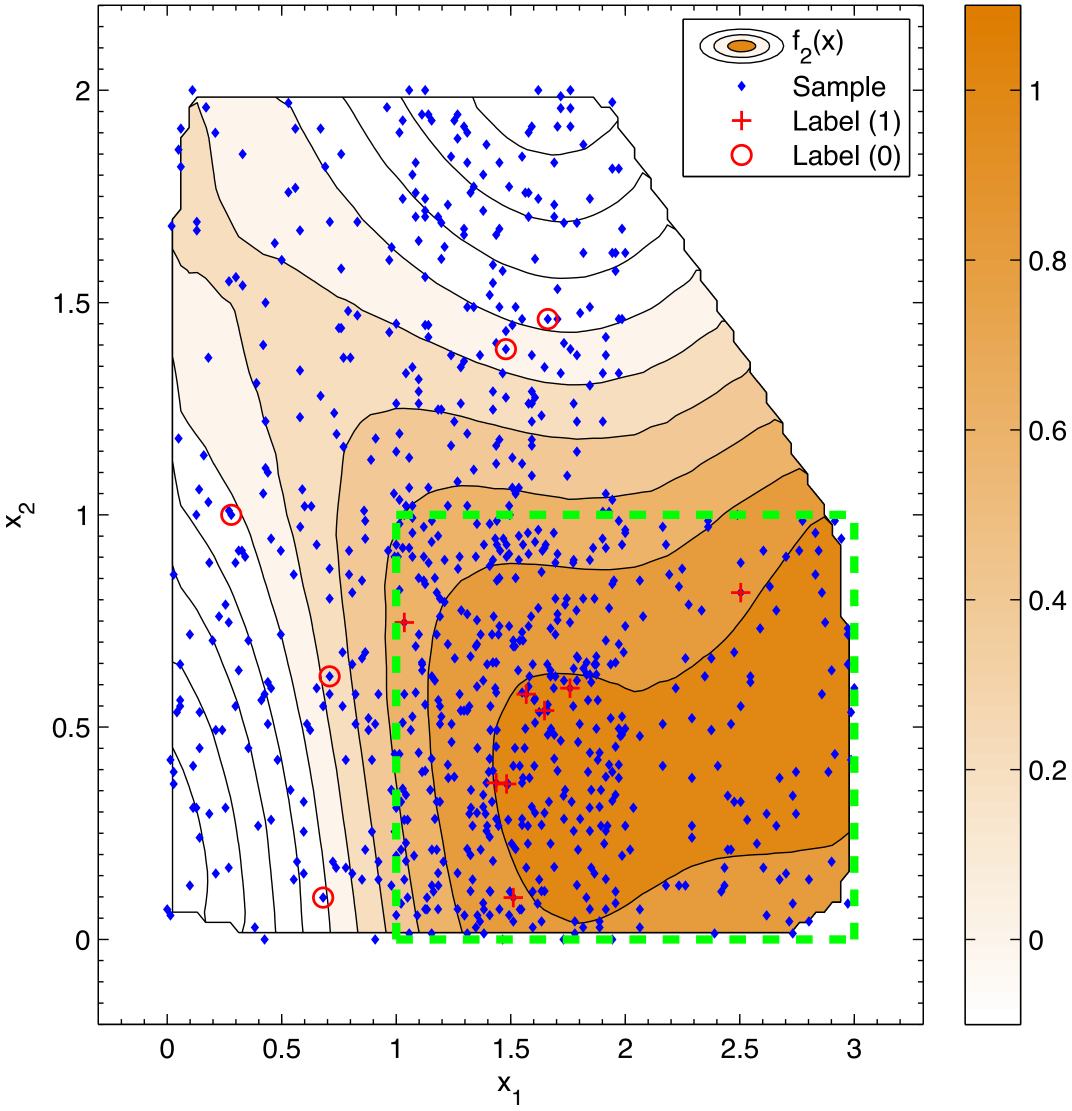


points and “logic rules”

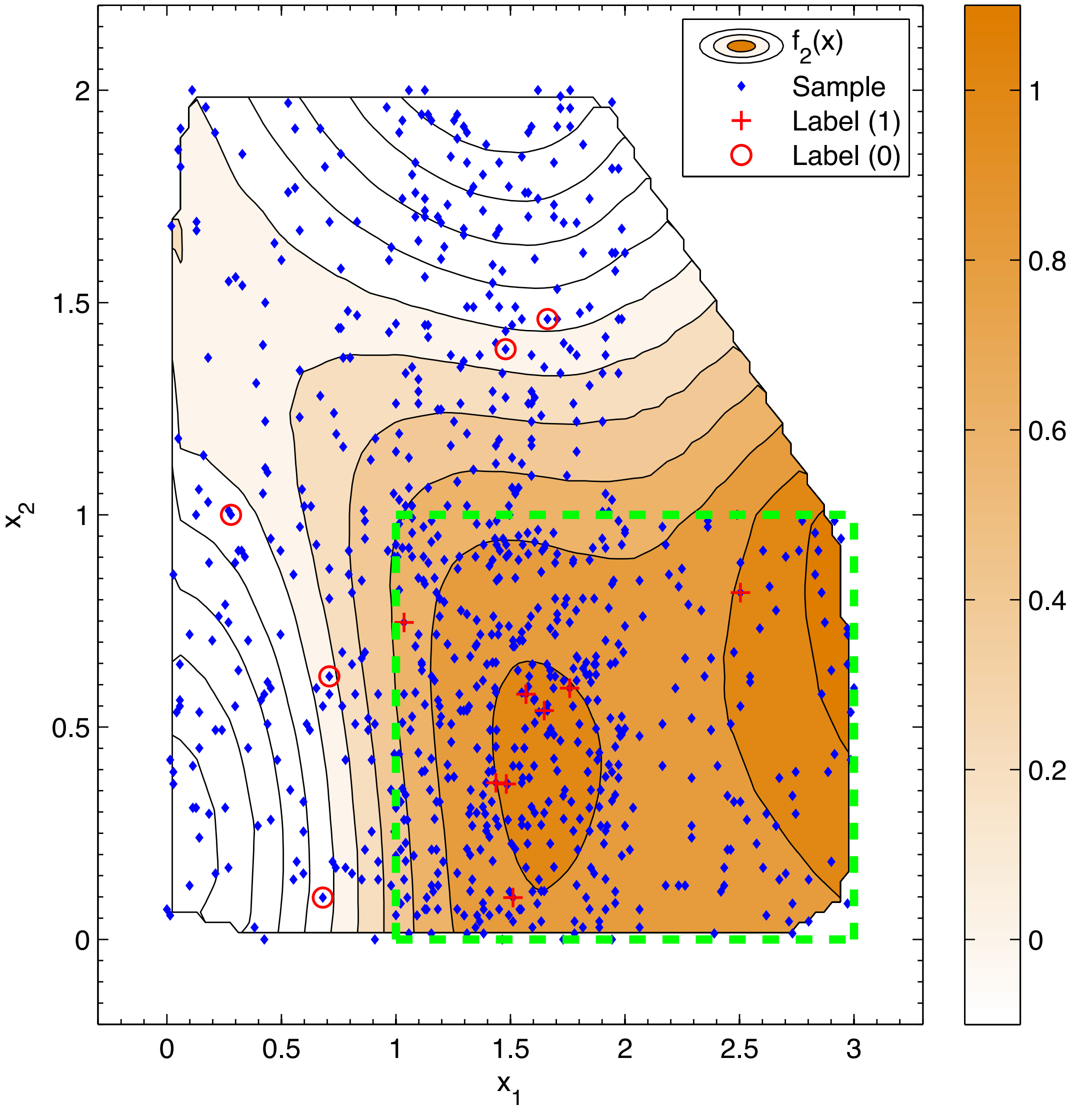


$$a_2(x) \rightsquigarrow f_2(x)$$

points only

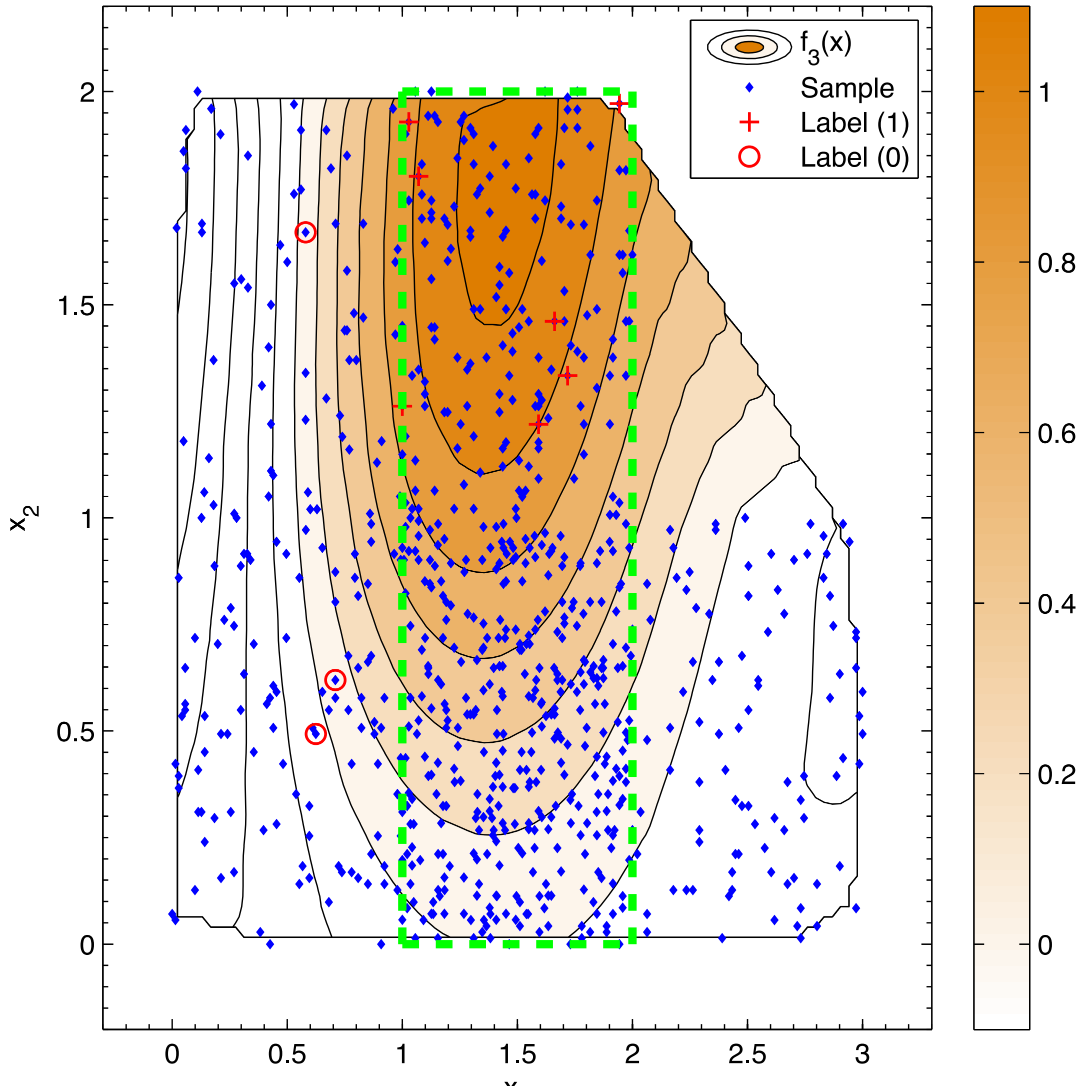


points and “logic rules”

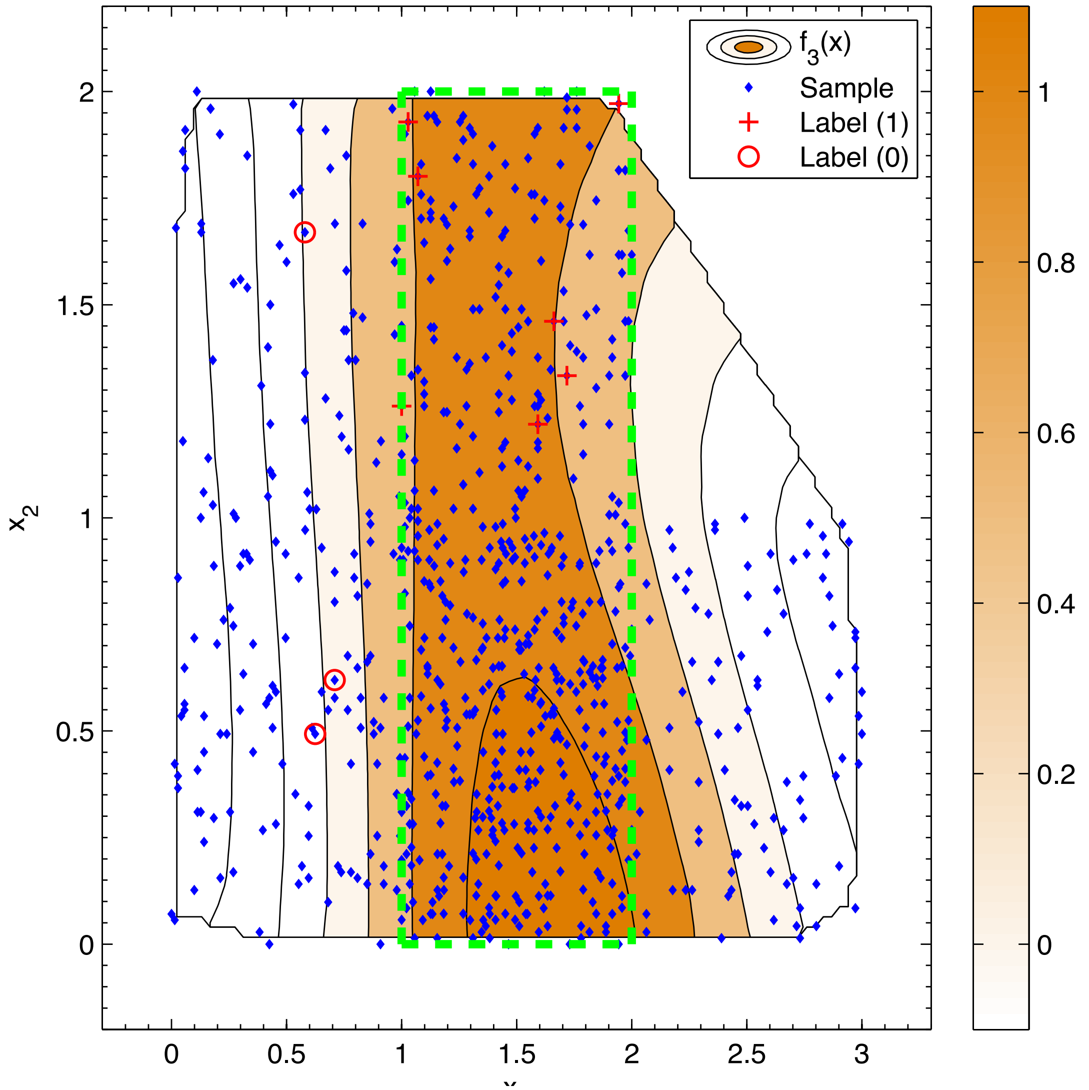


$$a_3(x) \rightsquigarrow f_3(x)$$

points only

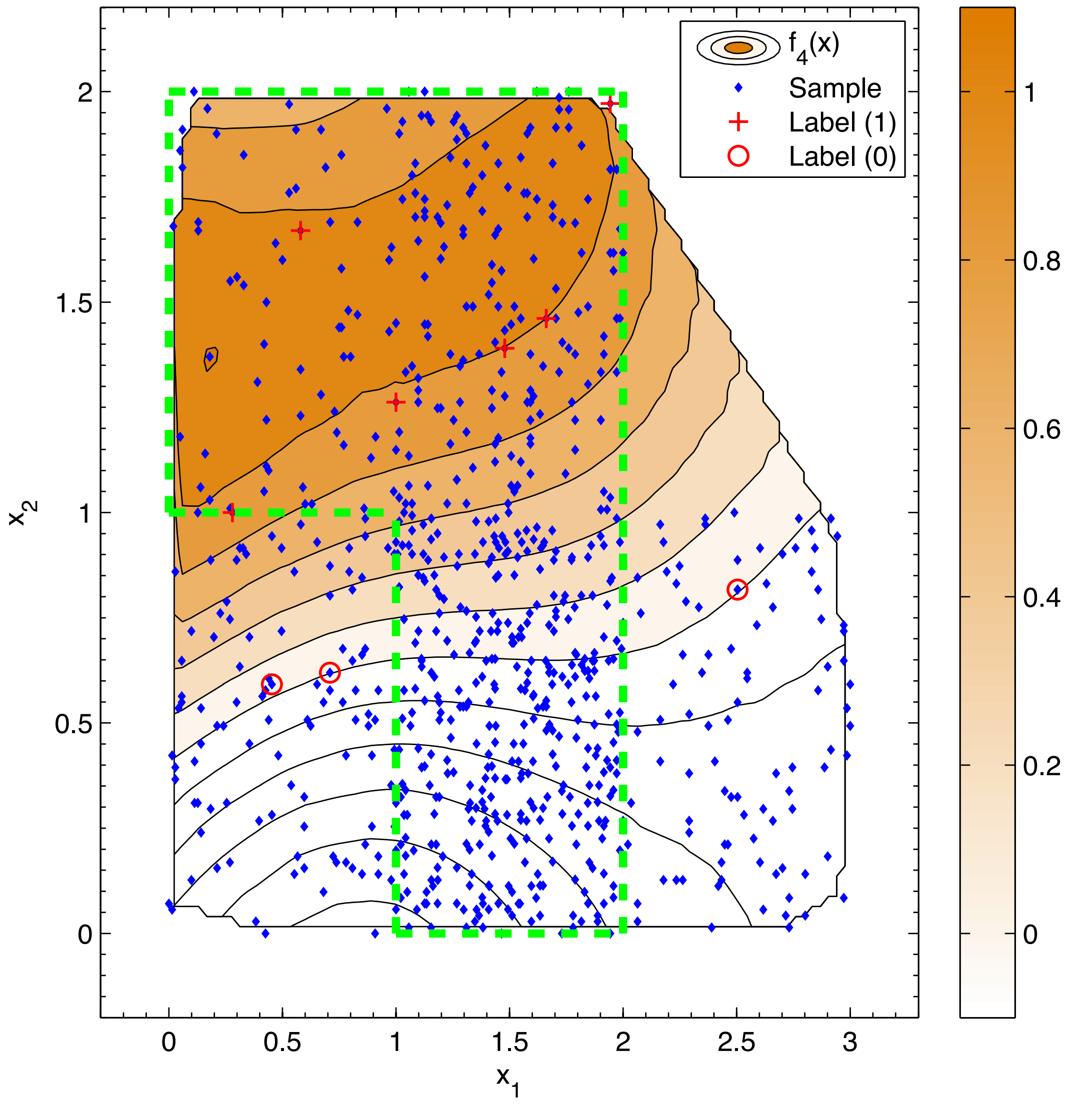


points and “logic rules”

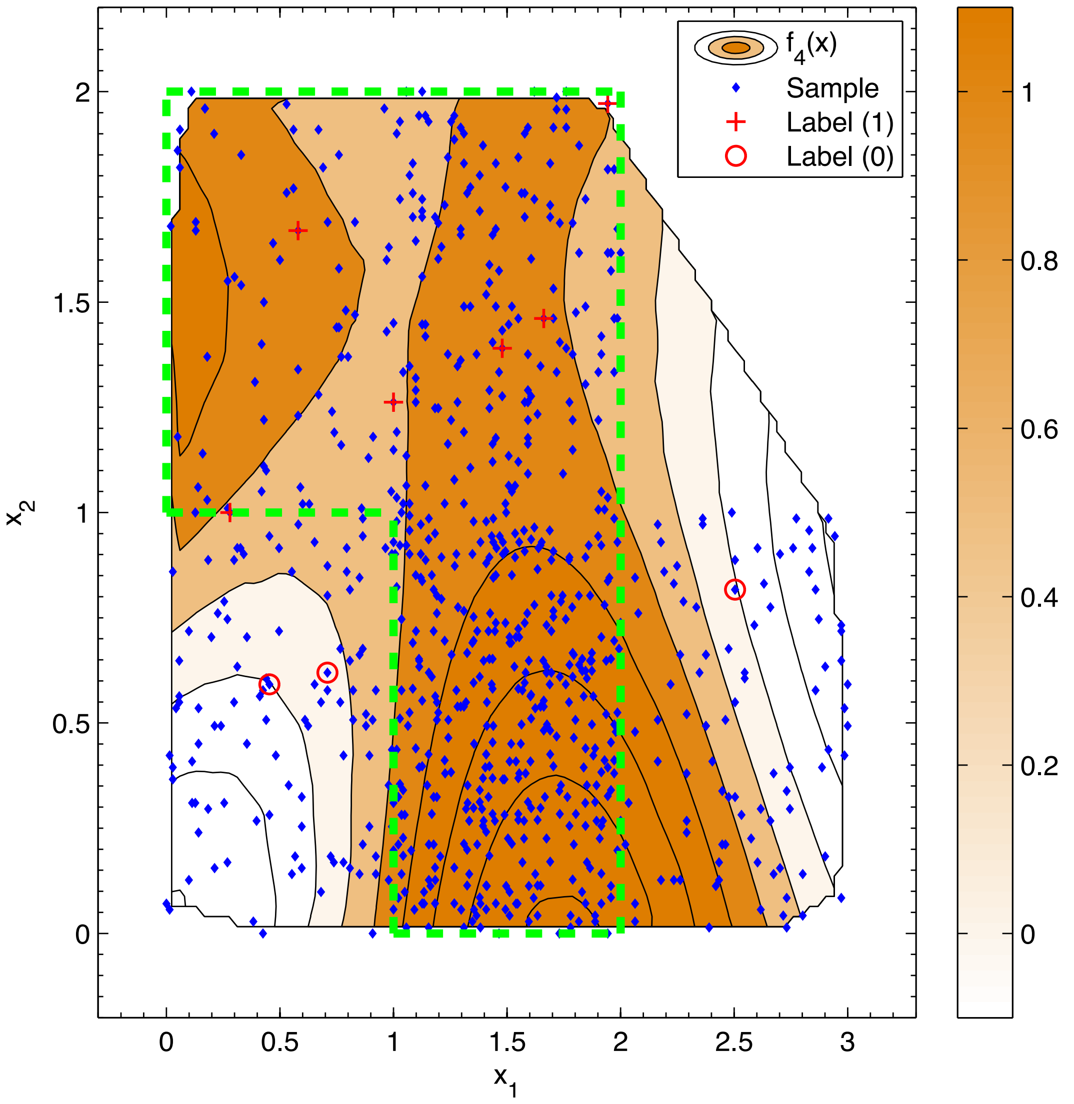


$a_4(x) \rightsquigarrow f_4(x)$

points only

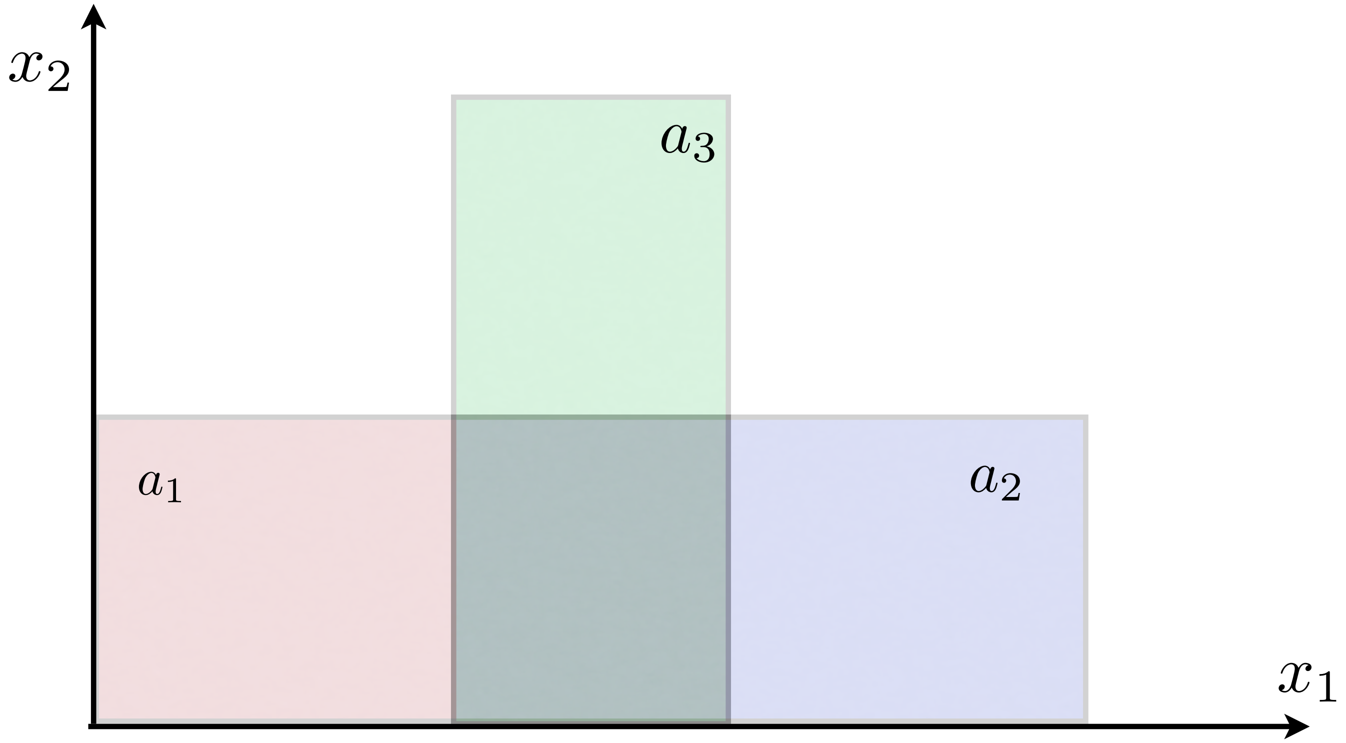


points and “logic rules”



Checking in the environment!

$$a_1(x) \ \wedge \ a_2(x) \ \Rightarrow \ a_3(x)$$
$$a_3(x) \ \Rightarrow \ a_4(x)$$
$$a_1(x) \ \vee \ a_2(x) \ \vee a_3(x)$$



Formally false ? but true in this environment!

$$a_1(x) \ \wedge \ a_3(x) \ \Rightarrow \ a_2(x) \quad a_1 = 1, \quad a_2 = 0 \quad a_3 = 1$$
$$a_3(x) \ \wedge \ a_2(x) \ \Rightarrow \ a_1(x) \quad a_1 = 0, \quad a_2 = 1 \quad a_3 = 1$$

Checking constraints

FOL clause	**Category**	**Average Truth Value**		
$a_1(x) \wedge a_2(x) \Rightarrow a_3(x)$	KB	98.26% (1.778)		
$a_3(x) \Rightarrow a_4(x)$	KB	98.11% (2.11)		
$a_1(x) \vee a_2(x) \vee a_3(x)$	KB	96.2% (3.34)		
$a_1(x) \wedge a_2(x) \Rightarrow a_4(x)$	LD	96.48% (3.76)	√	True
$a_1(x) \wedge a_3(x) \Rightarrow a_2(x)$	ENV	91.32% (5.67)		
$a_3(x) \wedge a_2(x) \Rightarrow a_1(x)$	ENV	91.7% (4.57)		
$a_2(x) \wedge a_3(x) \Rightarrow a_4(x)$	LD	96.58% (4.13)	√	
$a_3(x) \Rightarrow a_1(x) \vee a_2(x) \vee a_4(x)$	LD	99.7% (0.54)	√	
$a_1(x) \wedge a_4(x)$	ENV	45.26% (5.2)		False
$a_2(x) \vee a_3(x)$	ENV	78.26% (6.13)		
$a_1(x) \vee a_2(x) \Rightarrow a_3(x)$	ENV	68.28% (5.86)		
$a_1(x) \wedge a_2(x) \Rightarrow \neg a_4(x)$	ENV	3.51% (3.76)		
$a_1(x) \wedge \neg a_2(x) \Rightarrow a_3(x)$	ENV	27.74% (18.96)		
$a_2(x) \wedge \neg a_3(x) \Rightarrow a_1(x)$	ENV	5.71% (5.76)		

Based on fixed-point iteration

Conclusions

What's next?

examples are constraints!

there is no need to distinguish perceptual and logic constraints

Do you want know more?

visit https://sites.google.com/site/semanticbasedregularization/

A s/w simulator will be released soon for public use. Drop me an e-mail (marco@dii.unisi.it) if you want to try it!

Developmental Agents

- What if constraints are not available?

- don't use all the info at once!
  "easy-first" policy to select constraints?

reformulation based on information-theoretic principles for feature generation, constraint selection and generation