Rule Learning with Monotonicity Constraints

Wojciech Kotłowski
Roman Słowiński

Institute of Computing Science
Poznan University of Technology, Poland

ICML 2009, Montreal

Classification with Monotonicity Constraints

Classification problem with additional assumptions

- There exists a meaningful order between class labels.
- Domains of the attributes (input variables) are at least ordinal.
- Monotone relationship between values on attributes of the object and its class label: an increase in values on attributes should not decrease the label.

$\Rightarrow$ Inference with monotone functions.

Example: Windsor House Pricing

Contains $n = 546$ houses sold in Windsor, Canada (1987).
Class – selling price of the house discretized into 4 levels: *cheap, moderate, expensive, very expensive*.

Attributes:

- Size of the lot (sq. feet).
- Number of bedrooms.
- Number of bathrooms.
- Number of storeys.
- Driveway (yes/no).
- Recreation room (yes/no).
- Basement (yes/no).
- Air conditioning (yes/no).
- Number of garages.
- Desirable location (yes/no).

Why to Use Knowledge About Monotonicity?

- Monotonicity imposes constraints on the prediction function
  $\Rightarrow$ smaller hypothesis space $\Rightarrow$ less complex model
  $\Rightarrow$ increase in accuracy of predictions.
- Sometimes only the model consistent with domain knowledge will be acceptable to the domain experts.

Outline

1. Probabilistic model based on the stochastic dominance.
2. Nonparametric method of classification.
3. Learning rule ensembles with monotonicity constraints.

Some Theory of Classification

- Objects (x, y), $x \in X \subseteq \mathbb{R}^m, y \in Y = \{1, \ldots, K\}$ generated i.i.d. according to some probability distribution $P(x, y)$.
- Loss function $L(y, \hat{y})$, a penalty for predicting $\hat{y}$ when y is observed.
- The quality of a classifier $h\colon X \to Y$: expected loss (risk) according to $P(x, y)$:

$$R(h) = \mathbb{E} L(y, h(x))$$

- The minimizer of the risk, h^*, called Bayes classifier.

Problem

How can we incorporate monotonicity constraints into this formalism, i.e. express monotonicity constraints in term of $P(x, y)$?

Dominance Relation

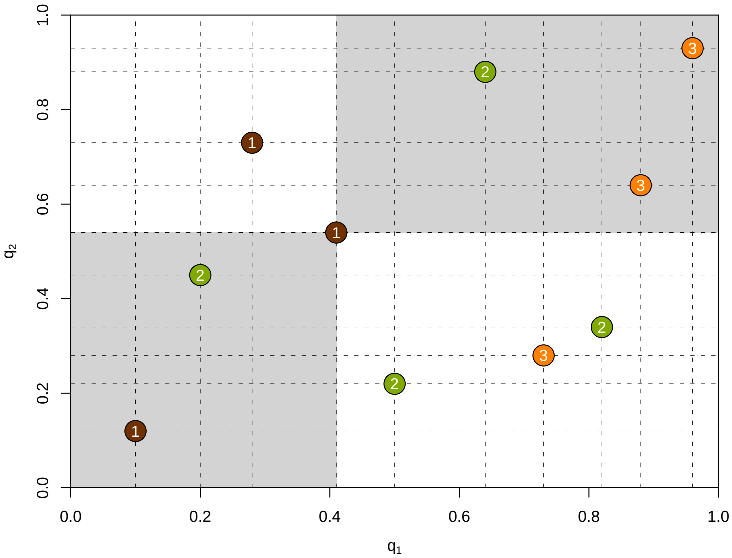
For each $x, x' \in X$, x dominates x', $x \succeq x'$, if x has higher or equal values on all attributes: $x_k \geq x'_k \; \forall k = 1, \ldots, m$.

Monotone Function

Function $f \colon X \rightarrow Y$ is monotone if for any $x, x' \in X$ it holds:

$$x \succeq x' \rightarrow f(x) \geq f(x')$$

Dominance relation – Example



Intuitively. . .

If $x \succeq x'$, then x probably has a higher or equal class label than x'.

Intuitively...

If $x \succeq x'$, then x probably has a higher or equal class label than x'.

Probabilistic Model

Objects are generated according to monotonically constrained probability distribution $P(x, y)$:

$$x \succeq x' \rightarrow P(y \geq k|x) \geq P(y \geq k|x') \quad \forall k = 1, \ldots, K.$$

In other words, for each k, function $P(y \geq k|x)$ is monotone.

Stochastic dominance relation.

Monotonicity of the Bayes Classifer

Let $P(x, y)$ be monotonically constrained.
Is the Bayes classifier a monotone function?

Suppose $L(y, k) = C(y - k)$. Then, Bayes classifier is monotone if and only if $C(\cdot)$ is convex.

$\Rightarrow$ Monotone for absolute or squared error loss, but not for 0-1 loss.

Learning

In general $P(x,y)$ is unknown (and so is h^*).
$\Rightarrow$ Train a classifier using a sample $D = \{(x_1,y_1),\ldots,(x_n,y_n)\}$.

Usually performed by minimization of the empirical risk:

$$\mathbb{E}_D h(x,y) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, h(x_i))$$

within restricted class of functions (e.g. linear, trees, rules, etc.).

In classification with monotonicity constraints one can consider the class of all monotone functions.

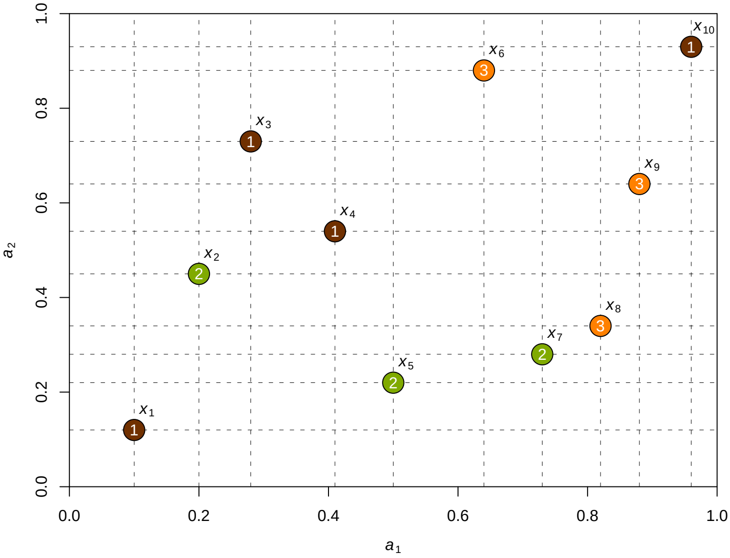
Nonparametric Classification

- Can be stated as an integer linear program:

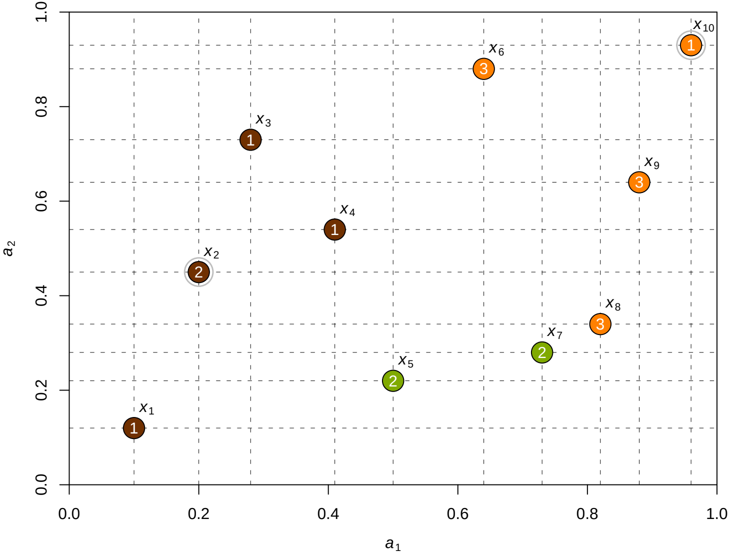
$$\begin{aligned} \min:\ & \sum_{i=1}^{n} L(y_i, d_i) \\ \text{s.t.}:\ & x_i \succeq x_j \rightarrow d_i \geq d_j \\ & d_i \in \{1, \ldots, K\} \end{aligned}$$

- Due to unimodularity of the constraints matrix integer conditions can be relaxed $\Rightarrow$ linear program.
- Interpretation: relabel the objects to remove inconsistencies with respect to monotonicity constraints.
- New labels are always monotone. $\Rightarrow$ data monotonization.
- Convergence to Bayes classifier.

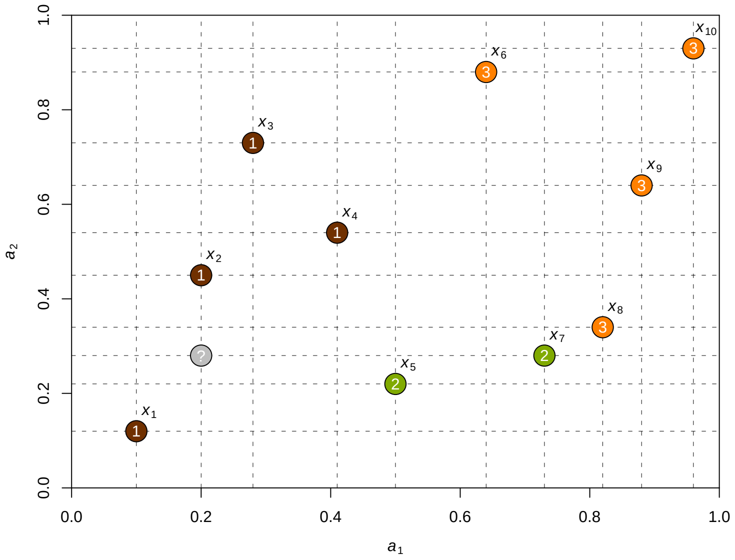
Original data set with violations of monotonicity constraints.



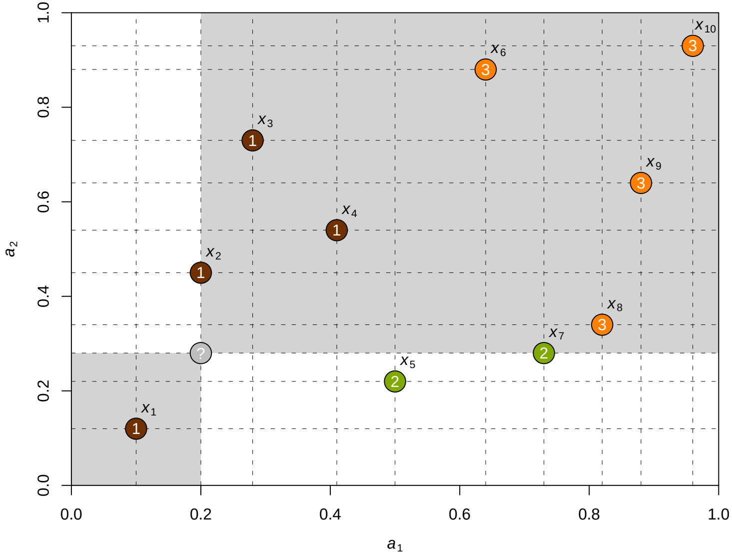
Monotonized data set (with use of absolute error loss)



Prediction.

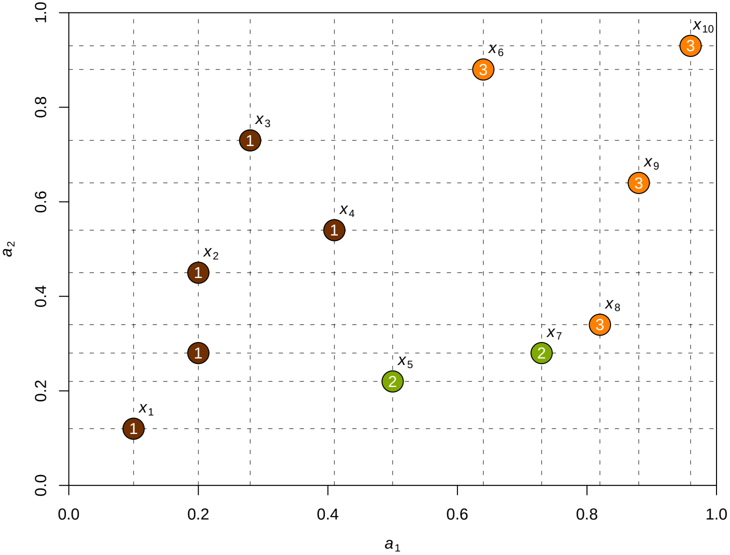


Prediction.



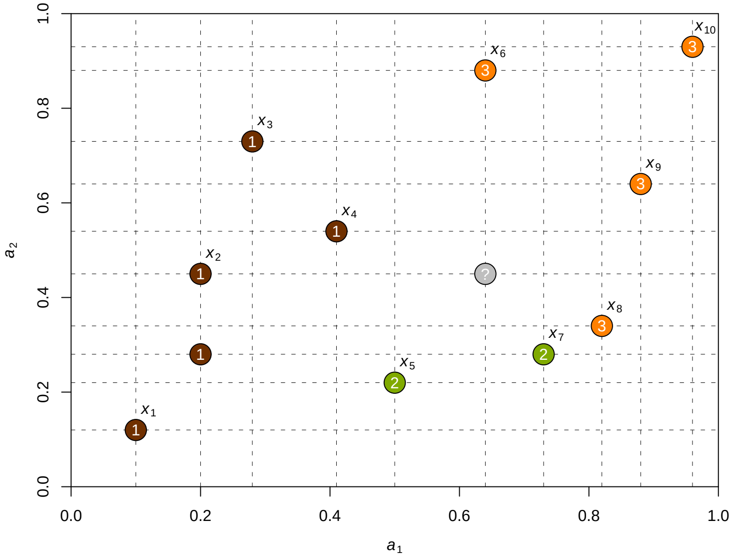
1.0
0.8
0.6
0.4
0.2
0.0
a_2
x_{10}
x_6
x_3
x_9
x_4
x_2
x_8
x_7
?
x_5
x_1
1
2
3
0.0
0.2
0.4
0.6
0.8
1.0
a_1

Prediction.

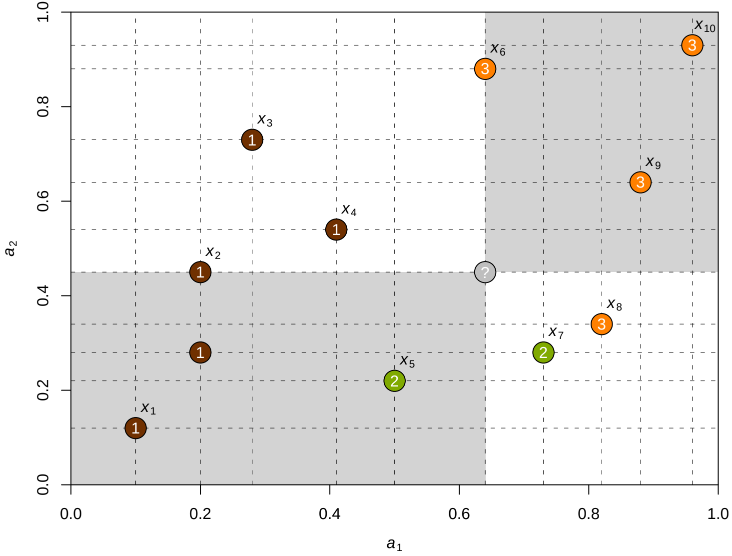


1.0
0.8
0.6
0.4
0.2
0.0
a_2
0.0
0.2
0.4
0.6
0.8
1.0
a_1
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
1
1
1
1
1
2
2
3
3
3
3

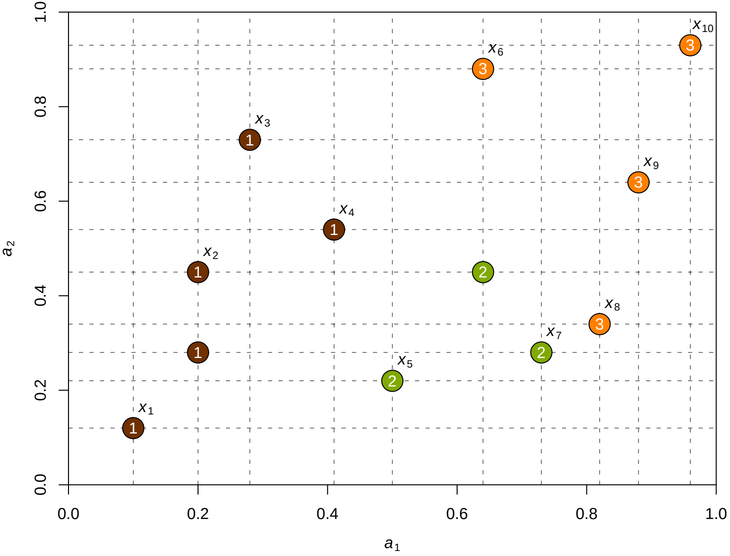
Ambiguous prediction.



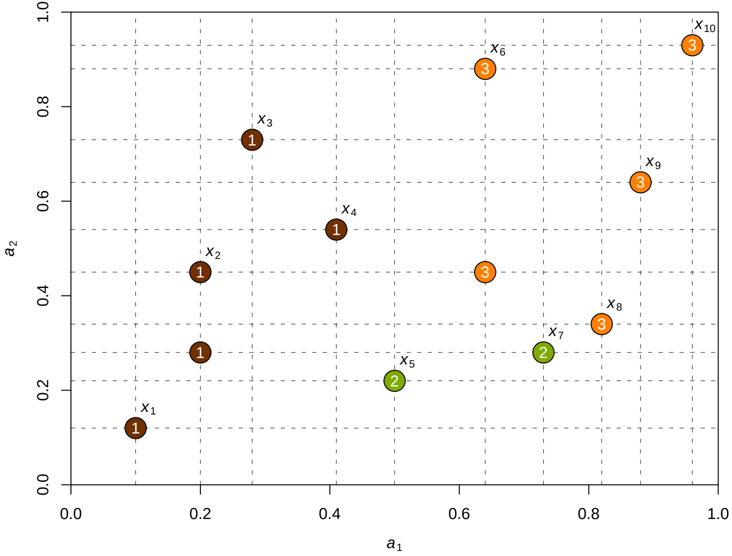
Ambiguous prediction.



Ambiguous prediction.



Ambiguous prediction.



Beyond Nonparametric Classification

Problems

- Nonparametric classification requires memorization of a large part of the training set.
- Can give an imprecise prediction.

Solution

- Apply nonparametric classification to D in order to obtain a monotonized data set D'.
- "Compress" the monotonized dataset D' using a set of rules (rule ensemble).

Rule ensemble

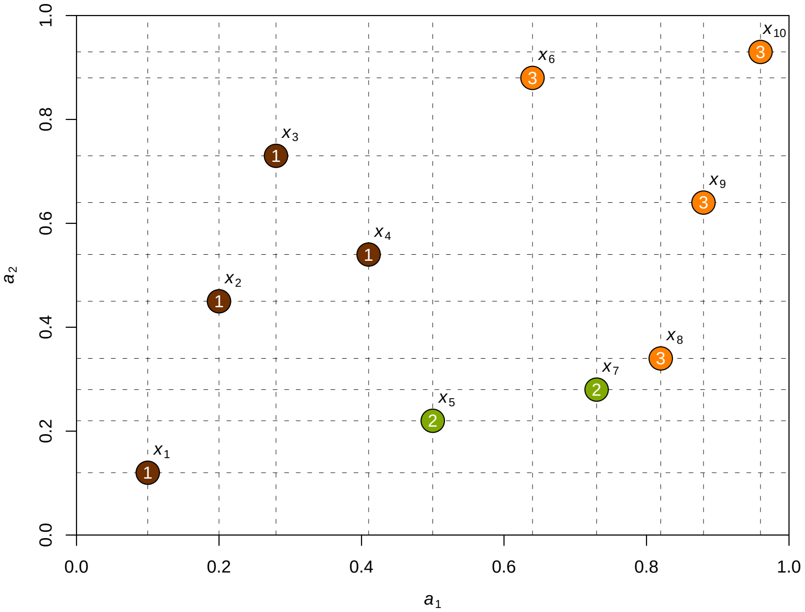
A (decision) rule is a logical expression of the form "if *conditions* then *decision*".

- condition part is a conjunction of constraints of the form $x_i \leq s_i$ or $x_i \geq s_i$,
- decision is a vote for a given cumulation of classes ("class at least k" or "at most k").

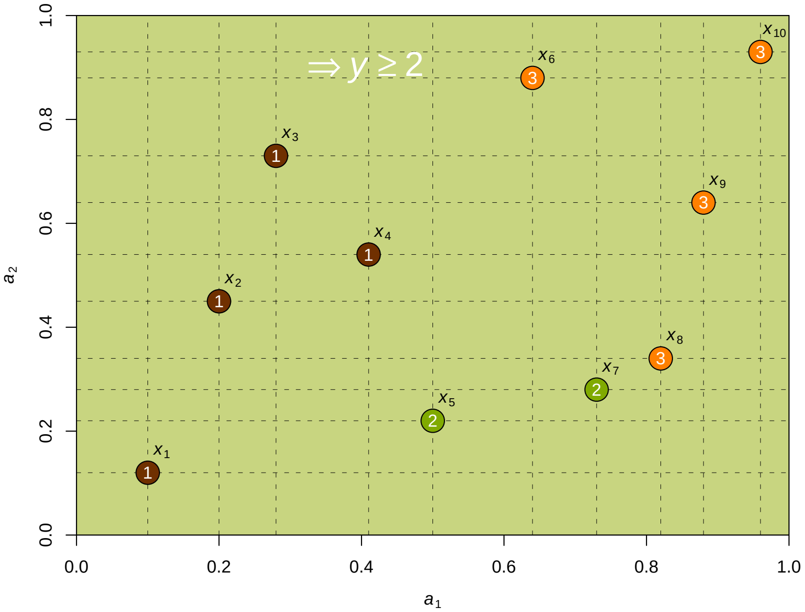
Example

if $lot\_size \geq 80\ 000$ and $nstoreys \geq 2$ then $price\_level \geq 3$.

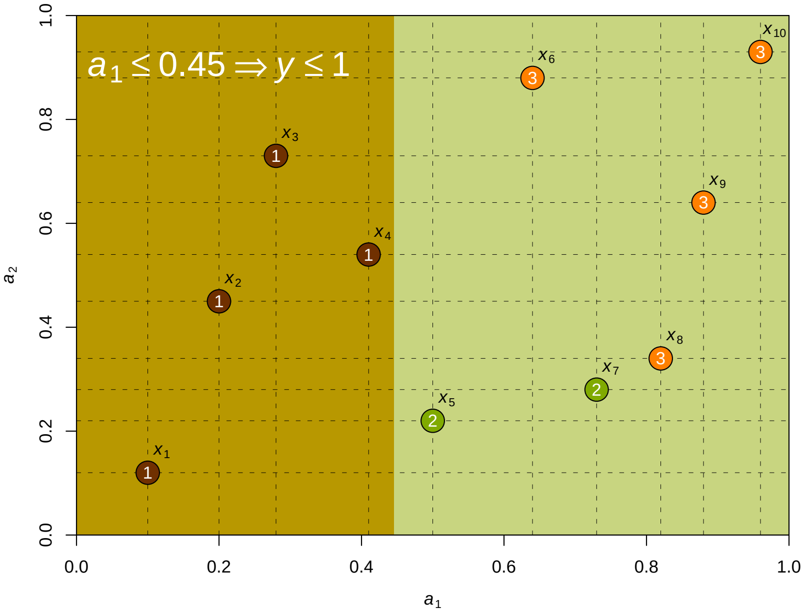
A single rule is too weak ⇒ a set of rules needed (rule ensemble).



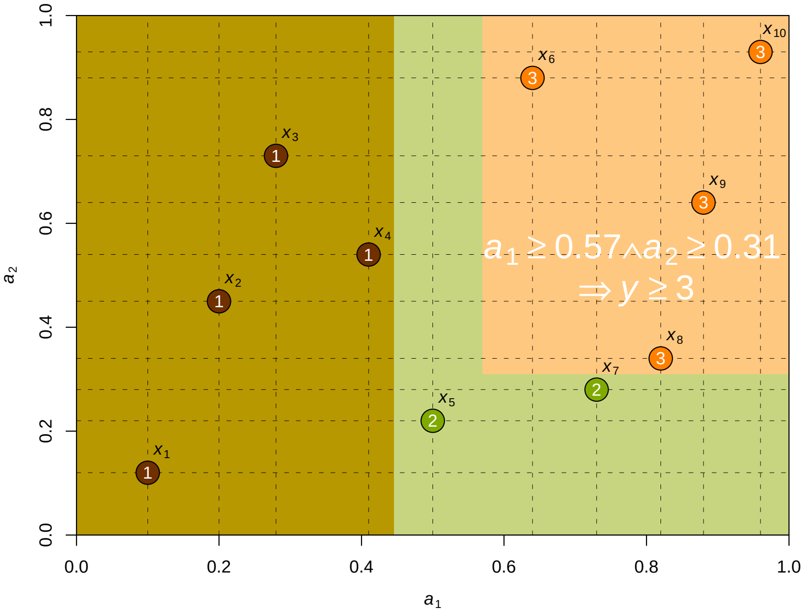
1.0
0.8
0.6
0.4
0.2
0.0
a_2
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
1
2
3
0.0
0.2
0.4
0.6
0.8
1.0
a_1



⇒ y ≥ 2
x1
x2
x3
x4
x5
x6
x7
x8
x9
x10
1
1
1
1
2
3
2
3
3
3
a1
a2
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0



$a_1 \leq 0.45 \Rightarrow y \leq 1$
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
1
2
3
0.0
0.2
0.4
0.6
0.8
1.0
a_1
a_2



1.0
0.8
0.6
0.4
0.2
0.0
a_2
0.0
0.2
0.4
0.6
0.8
1.0
a_1
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
$a_1 \geq 0.57 \wedge a_2 \geq 0.31$
$\Rightarrow y \geq 3$

Combining Rules

Rule ensemble is a set of $K-1$ convex combinations of rules.

$$f_k(x) = \sum_{t=1}^{T_k} \alpha_{kt} r_{kt}(x) \qquad k = 2, \ldots, K$$

where $\sum_t \alpha_{kt} = 1$ and $\alpha_{kt} \geq 0$.

For each $k = 2, \ldots, K$, $f_k(x)$ aims at separating class "at least k" from class "at most $k-1$".

- $\{r_{kt}, t = 1, \ldots, T_k\}$ is a set of rules voting for a class "at least k" (then $r_{kt}(x) = 1$) or "at most $k-1$" ($r_{kt}(x) = -1$).
- The final response of the classifier is:

$$h(x) = 1 + \sum_{k=2}^{K} \operatorname{sgn}(f_k(x))$$

Absolute error of $h(x)$ does not exceed sum of 0-1 errors of $f_k(x)$.

Generating Rules

For each $k = 2, \ldots, K$:

- Let $y_{ki} = 1$ if $y_i \geq k$ and $y_{ki} = -1$ if $y_i < k$. Rules are generated by maximization of the minimum margin:

$$\max \min_i y_{ik} f_k(x_i)$$

- A linear program. Can be solved efficiently via column generation algorithm (c.f. LPBoost).
- A solution with positive margin exists if and only if the dataset is monotone.

Generalization Bound

Theorem

Assume $P(x, y)$ is monotonically constrained. Let $h(x)$ be the final classifier and let $f_k(x)$, $k = 2, \ldots, K$, be the k-th rule ensemble trained on the monotonized data set D'_k achieving minimum margin γ_k. Then, with probability at least $1 - \delta$ for every $\gamma_2, \ldots, \gamma_K$:

$$\mathbb{E}L(y, h(x)) - \mathbb{E}L(y, h^*(x)) \leq M\left(2(K-1)\sqrt{\frac{\log \frac{2(K-1)}{\delta}}{n}} + \sqrt{\frac{m}{n}}\sum_{k=2}^{K}\frac{1}{\gamma_k}\right)$$

for some universal constant M.

Experimental Results

Ten datasets for which the monotone relationships are observed.

Five classifiers:

- Two state-of-the-art methods in classification with monotonicity constraints: Ordinal Learning Model (OLM), Isotonic Separation (IsoSep).
- Two classifier which does not take monotonicity into account, run in the ordinal setting : Support Vector Machines (SVM), decision tree (J48).
- Our methods: Linear Programming Rules (LPRules)

The error measure (and loss function): mean absolute error.

10-fold cross validation, repeated 10 times to improve replicability.

Results of Experiment

Dataset	OLM	IS	LPRules	J48	SVM
DenBosch	0.282 ±0.039	**0.183** ±0.037	**0.168** ±0.034	**0.172** ±0.032	**0.202** ±0.036
ESL	0.371 ±0.024	**0.328** ±0.023	**0.323** ±0.024	0.369 ±0.022	0.355 ±0.023
SWD	**0.452** ±0.017	**0.442** ±0.018	**0.435** ±0.017	**0.442** ±0.016	**0.435** ±0.016
LEV	0.427 ±0.018	**0.398** ±0.017	**0.396** ±0.016	0.415 ±0.018	0.444 ±0.016
ERA	1.256 ±0.031	1.271 ±0.034	1.263 ±0.033	**1.217** ±0.032	1.271 ±0.029
Housing	0.527 ±0.032	**0.286** ±0.02	**0.274** ±0.021	0.332 ±0.023	0.314 ±0.025
CPU	0.29 ±0.035	0.099 ±0.02	**0.073** ±0.018	0.1 ±0.019	0.371 ±0.03
Balance	0.224 ±0.02	0.19 ±0.017	**0.063** ±0.009	0.271 ±0.021	0.137 ±0.017
Windsor	0.576 ±0.028	0.52 ±0.028	**0.516** ±0.026	0.565 ±0.025	**0.491** ±0.026
Car	0.084 ±0.01	0.045 ±0.006	**0.03** ±0.004	0.09 ±0.008	0.078 ±0.007

Summary

- Statistical theory of classification with monotonicity constraints.
- Nonparametric classification: no additional assumptions on the model, learning in the class of all monotone functions.
- Compressing the training data with rule ensemble.