Value-transformation for Monotone Prediction by Approximating Fuzzy Membership Functions

Tomáš Horváth[a,d], Alan Eckhardt[b], Krisztián Buza[c], Peter Vojtáš[b], Lars Schmidt-Thieme[a]

[a]{horvath,schmidt-thieme}@ismll.de, [b]{Alan.Eckhardt,Peter.Vojtas}@mff.cuni.cz, [c]buza@cs.bme.hu, [d]Tomas.Horvath@upjs.sk

Monotone Prediction

A labeling $l : \mathcal{X} \to \mathbb{N}$ is called monotone if for all pair of objects $\mathbf{x}, \mathbf{y} \in \mathcal{X} \subseteq \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ holds that *if* $\mathbf{x} \preceq \mathbf{y}$ *then* $l(\mathbf{x}) \leq l(\mathbf{y})$, where

$$\mathbf{x} \preceq \mathbf{y} \ \Leftrightarrow\ (\forall j \in \{1, \ldots, k\})\ x_j \leq_j y_j \qquad (1)$$

Recent Approaches

Monotone Prediction

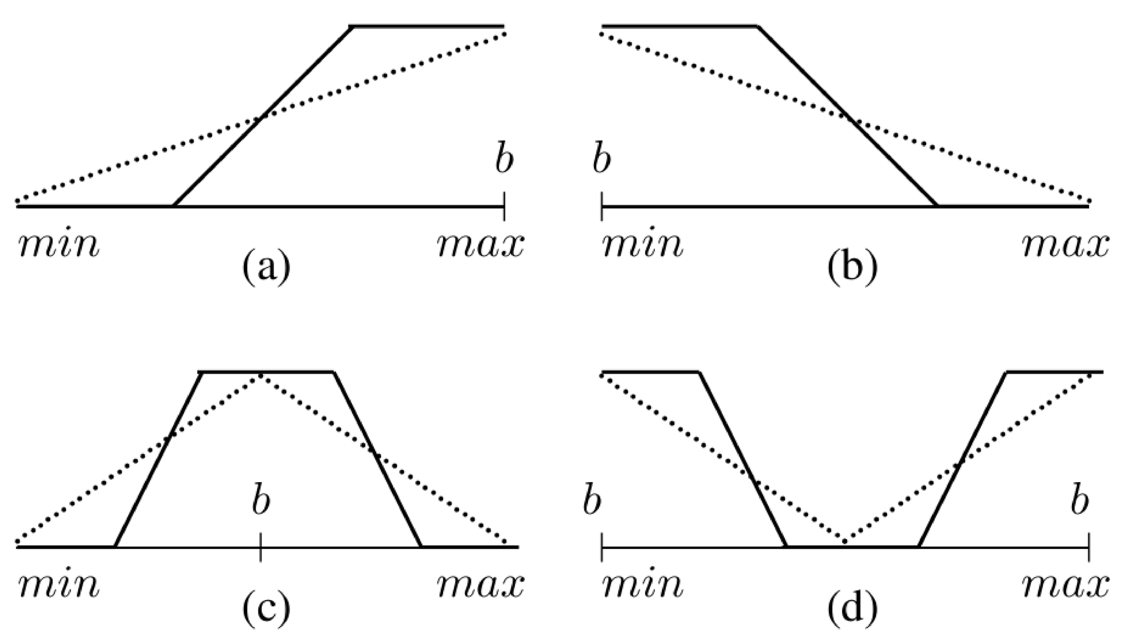
- **numerical attributes** only
- **linear ordering** $\leq_j$ on attribute domains, i.e. when the lower or higher values are the better
- Most of them **assume monotone training sets** without monotone noise

Data pre-processing

1. multiply attribute values with -1 if their negatively **correlate** with class values
2. if there is still some monotone noise, **relabel** non-monotone pairs of objects

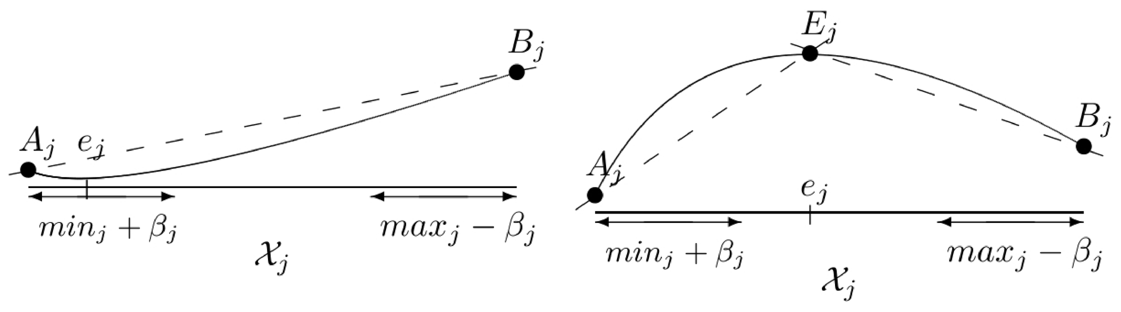
Our Approach

Replace $\leq_j$ in equation (1) by fuzzy membership functions $f_j : \mathcal{X}_j \to \mathbb{R}$ approximated from data.



Algorithm

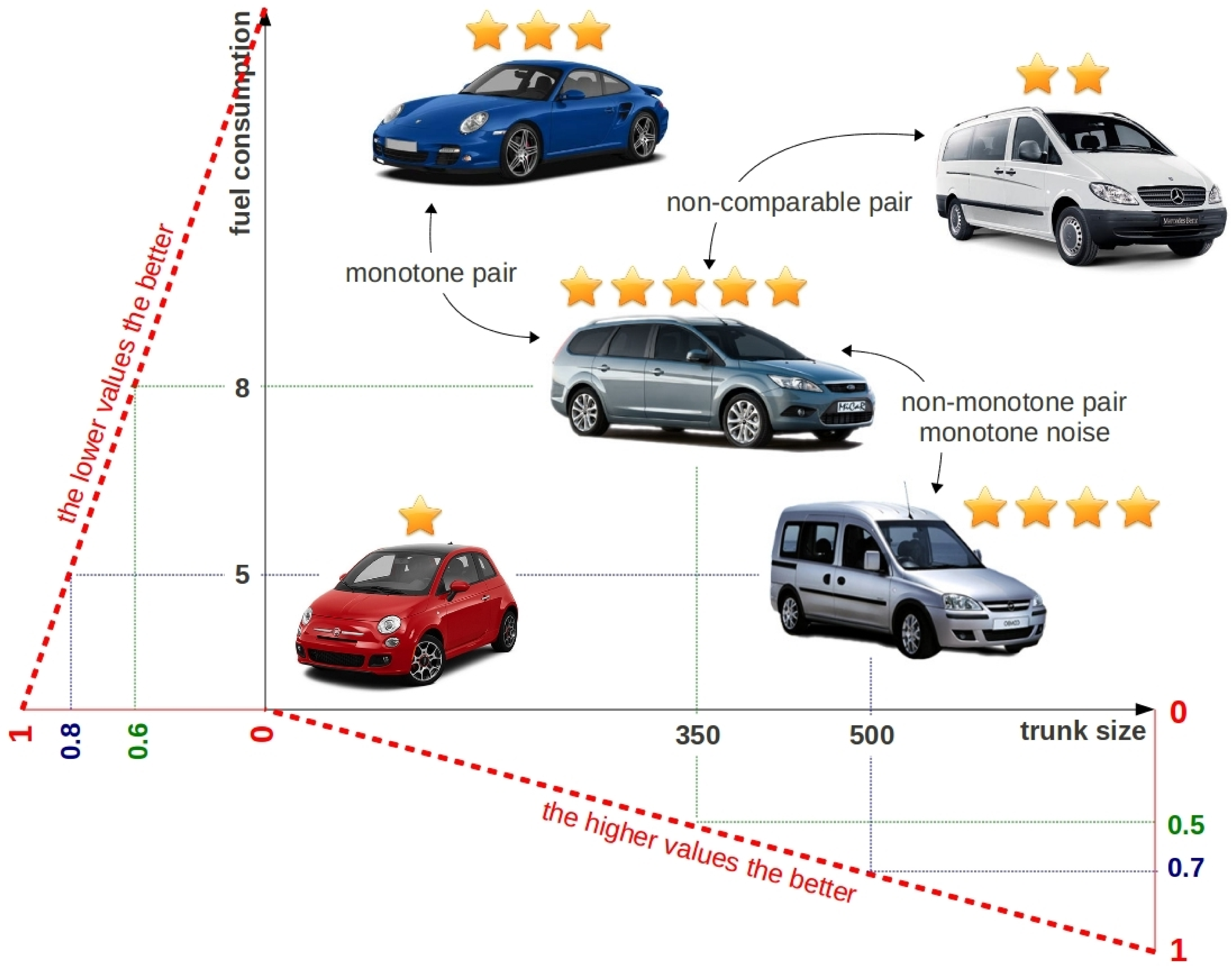
1. compute a parabola $p_j(X_j) = \alpha X_j^2 + \beta X_j + \gamma$ from the projection $\pi_j \subset \mathcal{X}_j \times \mathbb{N}$ of the values of the jth attribute to class labels
2. compute $A = p_j(min_j), B = p_j(max_j)$ and $E = p_j(e_j)$, where e_j is the extreme of p_j
3. if e_j lies on the border of the domain or outside the domain then $\hat{f}_j$ is a line defined by the points A_j and B_j. If e_j lies in the middle of the domain then $\hat{f}_j$ consists of two lines $\hat{f}_{jAE}, \hat{f}_{jEB}$ defined by the points A_j, E_j and E_j, B_j



In case of nominal attributes, we compute $\hat{f}_j$ as

$$\hat{f}_j(x) = \frac{\sum_{\{\mathbf{x}^i \in \mathcal{X} | x^i_j = x\}} l(\mathbf{x}^i)}{|\{\mathbf{x}^i \in \mathcal{X} | x^i_j = x\}|}$$

Example



Inverse mappings

The conditions in bodies of **Monotone Decision Tree** rules can be one of the following form:

$$X_j = v, \quad X_j \geq v, \quad X_j \leq v$$

where $v \in \mathbb{R}$ is a transformed value.

To interpret these rules we use the inverse mappings $\hat{f}_j^{-1}$ depending on the type of $\hat{f}_j$.
For example, if $\hat{f}_j$ is of type (c), then the above conditions will be interpreted as

$$X_j = \hat{f}^{-1}_{jAE}(v) \wedge X_j = \hat{f}^{-1}_{jEB}(v)$$

$$X_j \geq \hat{f}^{-1}_{jAE}(v) \wedge X_j \leq \hat{f}^{-1}_{jEB}(v)$$

$$X_j \leq \hat{f}^{-1}_{jAE}(v) \wedge X_j \geq \hat{f}^{-1}_{jEB}(v)$$

Experiment (2)

Measuring the **influence of pre-processing** to Monotone Classification

Baselines: *orig*, *corr*

Monotone Classification **algorithm used**:
– W. Duivesteijn and A. Feelders: *Nearest Neighbour Classification with Monotonicity Constraints.* Proceedings of ECML/PKDD '08

10-fold cross-validation to measure the average accuracy of classification on 10 folds

Results:
– Algorithm computed only for 27 datasets
– in 17 cases *fuzz* outperformed *orig*
– in 14 cases *fuzz* outperformed *corr*
– in 2 cases, the improvement was significant

Experiment (1)

Baselines – use only the numerical attributes

- *orig*: no pre-processing
- *corr*: correlation-based pre-processing

Monotonicity degree δ of the transformed (pre-processed) datasets $\mathcal{X}'$ measured

$$\delta = \frac{\#\text{ of monotone pairs in } \mathcal{X}'}{\#\text{ of comparable pairs in } \mathcal{X}'}$$

Tested on 40 **datasets** from the UCI machine learning repository

Results

Name	#Obj	#Num/#Nom	δ_{orig}	δ_{corr}	δ_{fuzzy}
auto-mpg	398	7 / 2	0.204	0.977	**0.979**
breast-c.	699	10 / 1	1	1	1
communit.	1994	123 / 5	NaN	NaN	**1**
concrete	1030	9 / 0	0.953	**0.986**	0.984
forestf.	517	11 / 2	0.733	0.719	**0.775**
machine	209	7 / 3	0.918	0.954	**0.958**
servo	167	3 / 2	0.333	0.704	**0.85**
slump	103	8 / 3	NaN	1	1
wdbc	569	31 / 1	1	1	1
wine	178	13 / 1	0.333	1	1
wineq.-r	1599	12 / 0	0.82	0.965	**0.966**
wineq.-w	4898	12 / 0	0.786	**0.919**	0.899
wpbc	198	33 / 2	0.962	1	1
abalone	4177	8 / 1	0.829	0.829	**0.854**
adult	32561	7 / 8	0.966	0.965	**0.988**
agaricus	8124	1 / 22	–	–	**1**
austral.	690	15 / 0	0.977	0.991	0.991
bands	540	21 / 19	1	0.957	1
car	1728	7 / 0	0.902	1	1
cmc	1473	8 / 2	0.8	0.818	**0.845**
crx	690	7 / 9	0.923	0.963	**0.991**
diagnosis	120	2 / 6	0.677	0.813	**1**
haberman	306	4 / 0	**0.893**	0.878	0.874
heart	270	14 / 0	0.988	**0.996**	0.995
hepatitis	155	7 / 13	0.98	0.993	**0.996**
horse-c.	300	17 / 11	1	1	1
house-v.	435	1 / 16	–	–	**0.986**
ionosph.	351	35 / 0	1	1	1
magic04	19020	11 / 0	0.446	0.979	**0.985**
mammogr.	961	4 / 2	0.944	0.944	**0.969**
nursery	12960	2 / 7	0.694	0.694	**1**
parkins.	195	23 / 1	NaN	0.995	**0.997**
pima	768	9 / 0	0.978	**0.978**	0.977
poker	25010	11 / 0	0.693	**0.735**	0.734
post-oper.	90	9 / 0	0.772	0.825	**0.857**
shuttle-l.	15	7 / 0	0.833	0.833	**1**
spambase	4601	58 / 0	0.977	**0.998**	0.987
tae	151	2 / 4	0.644	0.677	**0.882**
tic-tac-t.	958	1 / 9	–	–	**0.961**
transfu.	748	5 / 0	0.807	**0.94**	0.936

Acknowledgements

Tomáš Horváth was supported by the grant VEGA 1/0131/09 and the centre of excellence in computer science and knowledge systems (CaKS - ITMS 26220120007). Krisztián Buza is supported by the grant TÁMOP-4.2.2.B-10/1–2010-0009. We would like to thank to Alexandros Nanopoulos for his comments and suggestions. Finally, our special thanks go to Wouter Duivesteijn and Ad Feelders for providing us their implementation of their algorithm.