Exceptional Model Mining with Tree-Constrained Gradient Ascent

Thomas Krak Ad Feelders

Department of Information and Computing Sciences
Faculty of Science
Utrecht University

Presentation Overview

- Problem Introduction
  - Exceptional Model Mining
- Algorithm
  - Motivation
  - Tree-Constrained Gradient Ascent
  - Algorithm Sketch
- Experimental Results
  - Synthetic Data
  - Real Data
- Conclusion

Exceptional Model Mining (EMM)

EMM generalizes Subgroup Discovery (SD).

Given:

- Data set $\mathcal{D}$, containing n records.
- Record $r_i \equiv \langle a_1^i, \ldots, a_k^i, x_1^i, \ldots, x_p^i \rangle$, for $i = 1, \ldots, n$.
  - $\mathbf{a}^i \equiv \langle a_1^i, \ldots, a_k^i \rangle$ are *attributes*, domain $\mathcal{A}$.
  - $\mathbf{x}^i \equiv \langle x_1^i, \ldots, x_p^i \rangle$ are *targets*, domain $\mathcal{X}$.
- Model class $\mathcal{M}$ on $\mathcal{X}$.
  - E.g., linear regression.
- Quality function $\varphi_{\mathcal{D}} : \mathcal{P}(\mathcal{D}) \to \mathbb{R}$.

Exceptional Model Mining (cont.)

A *pattern* is a function $P : \mathcal{A} \rightarrow \{0, 1\}$ that induces a *subgroup* $G_P \subseteq \mathcal{D}$,

$$G_P \equiv \{ r_i \mid P(\mathbf{a}^i) = 1 \} .$$

Example:

$$P(\mathbf{a}^i) = \begin{cases} 1 & \text{if } (\texttt{age>23}) \wedge (\texttt{sex=F}) \text{, and} \\ 0 & \text{otherwise.} \end{cases}$$

Exceptional Model Mining (cont.)

Given two models from $\mathcal{M}$:

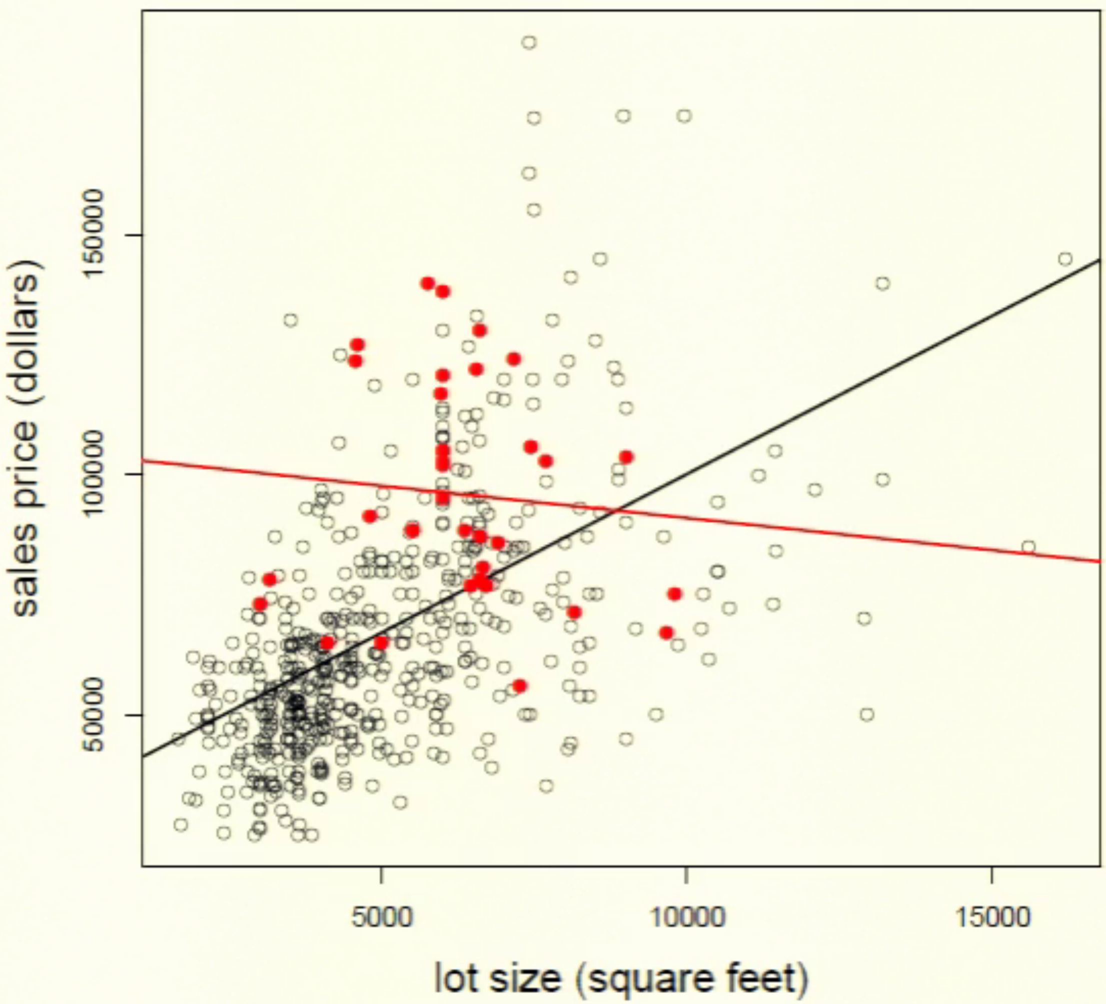
- Model $M_{\mathcal{D}}$ fitted to entire data set $\mathcal{D}$,
- Model M_{G_P} fitted to subgroup induced by pattern P.

Quality measure $\varphi_{\mathcal{D}}$ defines a *distance function* between $M_{\mathcal{D}}$ and M_{G_P}.

Goal is to find P s.t. $\varphi_{\mathcal{D}}(G_P)$ has high value.

I.e., we want to find subgroups with models that differ from the norm.

Exceptional Model Mining (cont.)



Pattern: $(\texttt{drive=1}) \wedge (\texttt{rec\_room=1}) \wedge (\texttt{nbath} \geq 2)$.

Exceptional Model Mining (cont.)

So, goal is to find P s.t. $\varphi_{\mathcal{D}}(G_P)$ has high value.

Problem (in general):

- Checking all patterns is intractable.

Hence, heuristics are often used.

Heuristically search space of all patterns.

- Beam search is commonly used.

Question:

- Can we do better?

Motivation

Actually two different search spaces:

- All patterns (*pattern language*)
- All subgroups (*extension space*)

These spaces do not (necessarily) "contain the same information".

- See [van Leeuwen, 2010].

Idea:

- Use information from both spaces instead of just searching in one.

Extension Space

Consider extension space.

Subgroup represented with *inclusion indicators* $\mathbf{w} = \langle w_1, \ldots, w_n \rangle, w_i \in \{0, 1\}$.

Quality of subgroup could be optimized using e.g. a hillclimber.

Our approach:
Generalize to *soft* subgroup, with *inclusion weights*:

$$w_i \in [0, 1] .$$

Extension Space (cont.)

Parameterize $\varphi_{\mathcal{D}}(\cdot)$ as objective function $O : [0,1]^n \to \mathbb{R}$.

- Use weighted-data scheme to estimate M_G.

Use numerical optimization to maximize $O(\mathbf{w})$.

- We use gradient ascent to find (local) optimum $\mathbf{w}^*$.

Extension Space (cont.)

Parameterize $\varphi_{\mathcal{D}}(\cdot)$ as objective function $O : [0,1]^n \rightarrow \mathbb{R}$.

- Use weighted-data scheme to estimate M_G.

Use numerical optimization to maximize $O(\mathbf{w})$.

- We use gradient ascent to find (local) optimum $\mathbf{w}^*$.

This representation gives useful information:

$$\texttt{Sign}\left\{\frac{\partial O(\mathbf{w})}{\partial w_i}\right\}$$

- If positive, increasing w_i improves subgroup.
- If negative, decreasing w_i improves subgroup.

Information about influence of individual records on quality.

Extension Space (cont.)

However:

- Interested in P^*, not (really) in $\mathbf{w}^*$.

Solution:

- Fit classifier to $\mathbf{w}^*$ to find P^*.
  - See [van Leeuwen, 2010].

Problems:

- P^* could be very complex.
- No guarantees that P^* even exists.

Tree-Constrained Gradient Ascent (TCGA)

Tree-Constrained Gradient Ascent

- Numerically optimize $O(\mathbf{w})$ to find $\mathbf{w}^*$.
- Constrain search to ensure P^* exists and is simple.
- Ensure that constraint hinders search as little as possible.

TCGA Algorithm Sketch

Basic idea:

- Construct classification tree on $\mathcal{A}$ with

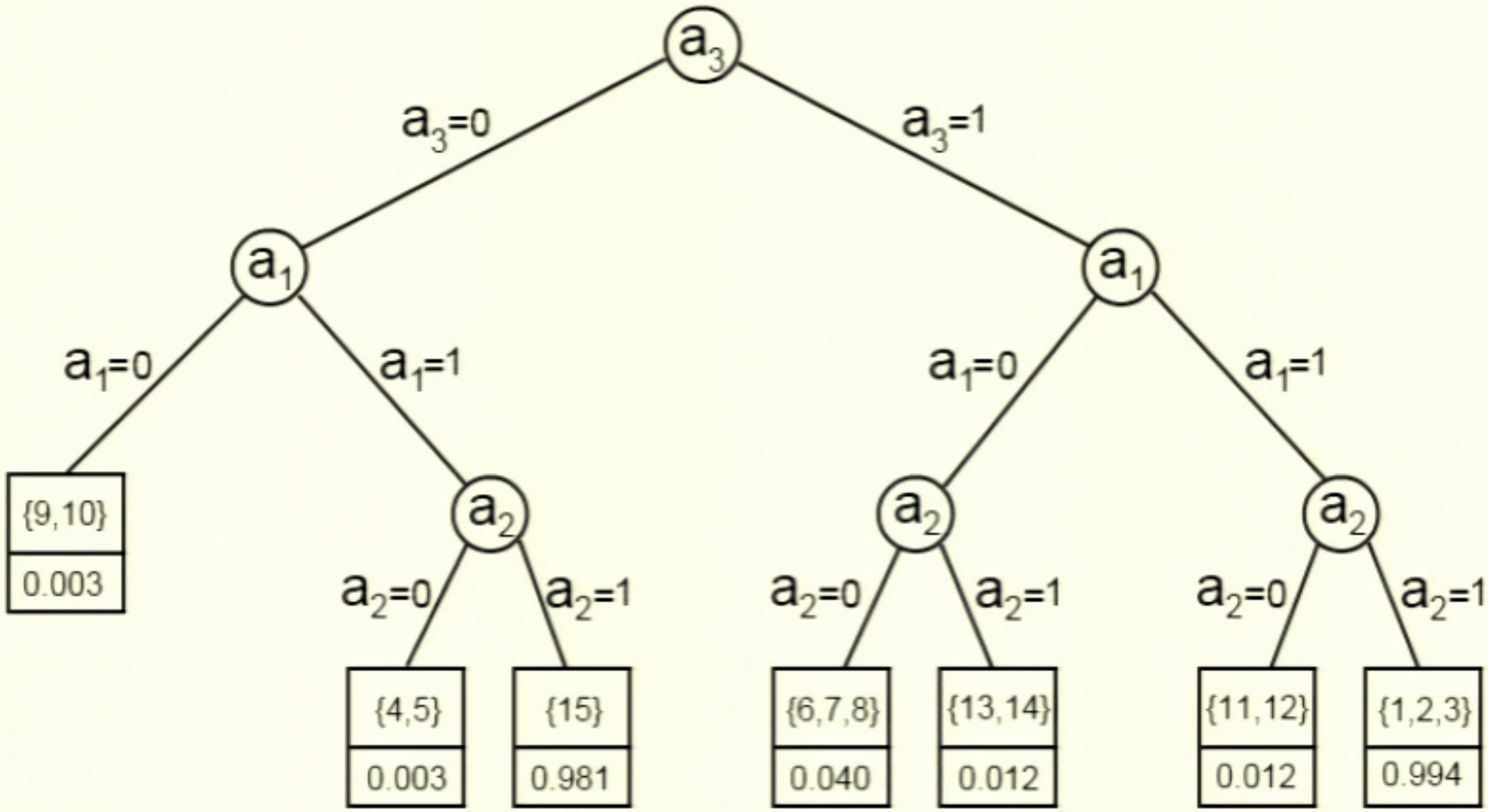
$$\texttt{class\_label}(\mathbf{a}^i) = \texttt{Sign}\left\{\frac{\partial O(\mathbf{w})}{\partial w_i}\right\}$$

  Intuition:
  - Separate what you want to include from what you want to exclude.
- Assign same inclusion weight to all records in same leaf of tree.
- Optimize these weights numerically.

Because derivatives (class labels!) can change sign:

- Alternate tree construction and weight optimization.

TCGA Algorithm Output



Finally:

- Round inclusion weights to $\{0, 1\}$.
- Read P^* from the tree.
  - Here: $(a_3 = 0 \land a_1 = 1 \land a_2 = 1) \lor (a_3 = 1 \land a_1 = 1 \land a_2 = 1)$.

TCGA (cont.)

Some details:

- Find multiple subgroups by random restarts.
- Perform post-processing on output.

Experiments

TCGA with linear regression model class.

Experiments on:

- Synthetic data.
- Real data.

Comparison to:

- Beam search (BS).
- Beam search with post-processing (BSPP).

Synthetic Data

Known high-quality subgroups.

Performance measured in F_1 score and $\varphi_{\mathcal{D}}(\cdot)$-based measure.

Results (at $\alpha = 0.01$ level):

- TCGA significantly outperformed both BS and BSPP.
- BSPP significantly outperformed BS.

Synthetic Data

Known high-quality subgroups.

Performance measured in F_1 score and $\varphi_{\mathcal{D}}(\cdot)$-based measure.

Results (at $\alpha = 0.01$ level):

- TCGA significantly outperformed both BS and BSPP.
- BSPP significantly outperformed BS.

Further experiments showed significant correlation between:

- TCGA's relative performance and global model R^2.
- TCGA's relative performance and subgroup quality.

(TCGA performed worse than BS when R^2 or quality were low).

Real Data

10 dataset/model pairs from different sources.

Performance measured in $\varphi_{\mathcal{D}}(\cdot)$-based measure.

Results (at $\alpha = 0.05$ level):

- No significant difference between TCGA and BS/BSPP.
- BS significantly outperformed BSPP.

Here also:

- Significant correlation between TCGA's relative performance and global model R^2.

BS performed better when global R^2 was low, TCGA performed better when it was high.

Summary & Conclusion

Tree-Constrained Gradient Ascent (TCGA):

- New heuristic for EMM.
- Performs numerical optimization in extension space.
- Constrains search to ensure corresponding pattern exists.
- Tries to hinder search as little as possible.

TCGA outperforms BS when:

- Quality of subgroups is not too low.
- Global model R^2 is not too low.

And, these are really the cases that matter.