Pattern-Based Classification: A Unifying Perspective

LeGo
Slovenia, Bled 2009
07.09.2009

Albrecht Zimmermann, Siegfried Nijssen, Björn Bringmann
Katholieke Universiteit Leuven, Belgium
Observations

The LeGo schema

- General schema
- Augment/replaces *data mining* step in KDD
- Topic of this workshop
Observations (cont.)

- DB
- Pattern Mining
- PS
- Feature Selection
- PS
- Model Induction

- Frequent
- Closed
- Correlating
- Exhaustive
- Heuristic
- Decision Tree
- Decision List
- SVM
DB → Pattern Mining → PS → Feature Selection → PS → Model Induction

- Frequent
- Closed
- Correlating

- Exhaustive
- Heuristic

- Decision Tree
- Decision List
- SVM

No overview

Ramamohanarao et al '07
Observations (cont.)

No overview → reinventions → revisited dead ends → lost progress
DB → Pattern Mining → PS → Feature Selection → PS → Model Induction

- Frequent
- Closed
- Correlating
- Exhaustive
- Heuristic
- Decision Tree
- Decision List
- SVM

No overview → reinventions → revisited dead ends → lost progress
## What patterns and how?

| Which pattern type | Which data-structure |
|--------------------|----------------------|
| Itemsets           | FP-Trees             |
| Multi-itemsets     | ZBDDs                |
| Sequences          | TID-Lists            |
| Trees              | Bit-Vectors          |
| Graphs             |                      |
What patterns and how?

Which pattern type

- Result hold for lattices (itemsets) or even partial orders (graphs)

  Independent of Pattern Type

Which data-structure

- FP-Trees
- ZBDDs
- TID-Lists
- Bit-Vectors

Sequences ⊂ Trees ⊂ Graphs
What patterns and how?

Which pattern type

- Results hold for \textit{lattices} (itemsets) or even \textit{partial orders} (graphs)

\textbf{Independent of Pattern Type}

- Sequences $\subset$ Trees $\subset$ Graphs

Which data-structure

\textbf{Independent of Data Structure}
Why mine explicit patterns?

Rules:

\[ A_1 = v_2 \quad A_4 = v_1 \]
\[ A_3 = v_2 \quad A_2 = A_1 \]

Decision Trees:

\[ A_1 = v_2 \quad A_4 = v_1 \quad A_3 = v_2 \]

Traditional classification

Why should we care in the first place?

EXCURSUS

Why should we care in the first place?

apart from attending the workshop
Why mine explicit patterns?

Traditional classification

Attributes: \{A_1, \ldots, A_d\}

Values: \(V(A) = \{v_1, \ldots, v_r\}\)

Rules:

- \(A_1 = v_2 \land A_4 = v_1 \Rightarrow +\)
- \(A_3 = v_2 \land A_2 = v_1 \Rightarrow -\)

Decision Trees:

- \(A_1 = v_2\)
  - \(A_4 = v_1\)
  - \(A_3 = v_2\)
Why mine explicit patterns?

Pattern based classification

Transactions are Structured

t ⊆ \{i_1, \ldots, i_3\}

- Patterns provide instance description
- Models can be built independent of data type
- Yield interpretable classifiers
- Alternatives are opaque (Kernels, NN, ...)

Structured Transactions
Thus leverage pattern mining techniques

Advantages:

- 15 years of research → fast and scaleable
- Described in structured language → persistent, not opaque

Challenge(s):

- (Re-)Entangle instance description and classification
Roadmap

Class-sensitive patterns & the mining thereof

- Model-independence
  - Post-processing
  - Iterative Mining

- Model-dependence
  - Post-processing
  - Iterative Mining
Roadmap

Class-sensitive patterns & the mining thereof

Model-independence

Post-processing

Iterative Mining

Model-dependence

Post-processing

Iterative Mining

D I S C L A I M E R

We will probably miss some approaches that should have been included in the presentation.

which just proves our point
Should we use frequent patterns?

- Well-researched
- Frequent $\rightarrow$ expected to hold on unseen
- Efficient mining

- Which threshold?
- Frequent $\rightarrow$ no/anti-correlation w/classes
- (Too) many patterns

DB → Pattern Mining → PS → Feature Selection → PS → Model Induction → M
Class-sensitive patterns

Taking relationship to class-labels into account

- Interesting Rules ’98 (IR)
- Jumping Emerging Patterns ’01 (JEP)
- Nuggets ’94
- Subgroup Descriptions ’96 (SGD)
- Contrast Sets ’99 (CS)
- Correlating Patterns ’00 (CP)
- Discriminative Patterns ’07 (DP)
- Emerging Patterns ’99 (EP)
- Version Space Patterns ‘01
- Class-Association Rules ’98 (CAR)

Taking no sides/not subscribing to particular universe
Evaluating class-sensitivity

- Confidence, Lift, WRAcc (Novelty), $X^2$, Correlation Coefficient, Information Gain, Fisher Score

- Some of them mathematically equivalent, some semantically

- Lavrac et al. ‘09
How to mine them?

- Mining frequent patterns & post-processing
  - Liu et al. ’98 (CAR)
  - Kavask et al. ’06 (SGD)
  - Atzmüller et al. ’06 (SGD)
  - Cheng et al. ’07 (DP)

- Bounding specific measure
  - Wrobel ’97 (SGD)
  - Bay et al. ’99 (CS)
  - Wang et al. ’05 (CAR)
  - Arunasalam et al. ’06 (CAR)
  - Nowozin et al. ’07 (CAR)
  - Cheng et al. ’08 (DP)
    (1 bound)

| CAR       | - Class Association Rules |
|-----------|---------------------------|
| CS        | - Contrast Sets           |
| DP        | - Discriminative Patterns |
| SGD       | - SubGroup Descriptions   |
How to? (cont.)

- **General Branch-and-bound**
  - Webb '95 (CAR)
  - Klösgen '96 (SGD)
  - Morishita et al. '00 (2-bounds)
  - Grosskreutz et al. '08 (SGD)
  - Nijssen et al. '09 (4-bounds)*

- **Iterative deepening**
  - Bringmann et al. '06 (CP)
  - Cerf et al. '08 (CAR)
  - Yan et al. '08 (DP)

- **Sequential sampling**
  - Scheffer et al. '02 (SGD)

Earlier than most specifics, subsumes them!

*) itemset-specific, constraint programming
What traversal strategy

Seriously ?
Result sets

- Are still too big
- May include irrelevant patterns
- May include much redundancy
The (extended) LeGo

DB → Pattern Mining → PS → Feature Selection → PS → Model Induction → M

Pattern set constraint
Model constraint
The (extended) LeGo

Model constraint

Optimisation Criteria

Mining Constraint

DB Pattern Mining PS Feature Selection PS Model Induction M

Model constraint
The (extended) LeGo

Model constraint

Model constraint

Model-Independent Iterative Mining

Model-Independent Post-Processing

DB
Pattern Mining
PS
Feature Selection
PS
Model Induction
M

Iterative Mining

Model-Independent

Post-Processing

Model constraint
The (extended) LeGo

Model-Independent Iterative Mining

Model-Independent Post-Processing

DB → Pattern Mining → PS → Feature Selection → PS → Model Induction → M

Optimisation Criteria

Mining Constraint
The (extended) LeGo

- Model-Independent Iterative Mining
- Pattern Mining
- PS
- Feature Selection
- PS
- Model Induction

- Model-Dependent Iterative Mining
- Model-Independent Post-Processing
- Model-Dependent Post-Processing

DB
Model-independence

- Only patterns affect other patterns’ selection
- Modular: usable in any classifier (often SVM)
Model independent Post-processing

- Mine large set of patterns
- Select subset
  - Exhaustively: too expensive
  - Heuristically: usually ordered
- Use measure to quantify combined worth
Pattern Set Scores

- Pattern sets can be scored based on

  - **TID lists** of patterns only
    - significance: incorporate support/class-sensitivity
    - redundancy: similarity between TID lists

  - **Pattern structure** & TID lists
    - using a **pattern distance** measure
    - by computing how well the patterns **compress** data

* computable for all data types
* requires specialization
Model independent Post-Processing

Exhaustive

- Exhautive enumeration
- Explicit size constraint
- Boundable pruning
- Implicit redundancy control

De Raedt et al. '07
- Exhaustive enumeration
- Arbitrary constraints
- Monotone, boundable pruning
- Explicit redundancy control

Exhaustive

Extremely large search space -> scalability issues

Counter-intuitive result: all sets

DISCLAIMER

The following algorithms should be considered illustrating examples, NOT recommendations!

other approaches vary
Model independent Post-Processing

Exhaustive

- Knobbe et al. ’06
  - Exhaustive enumeration
  - **Explicit size constraint**
  - Boundable pruning
  - Implicit redundancy control (entropy)

- De Raedt et al. ’07
  - Exhaustive enumeration
  - Arbitrary constraints
  - Monotone, boundable pruning
  - Explicit redundancy control

Extremely large search space -> scalability issues

Counter-intuitive result: all sets
Model independent Post-Processing

Heuristic Search Strategies

- **Fixed Order:** Scan patterns in (possibly random) fixed order, add each pattern that improves running score (O(n))

- **Greedy:** Repeatedly reorder patterns to pick pattern that improves score most (O(n^2))
Model independent Post-Processing

Heuristic Search Strategies

- **Fixed Order**: Scan patterns in (possibly random) fixed order, add each pattern that improves running score ($O(n)$)

- **Greedy**: Repeatedly reorder patterns to pick pattern that improves score most ($O(n^2)$)
Model independent Post-Processing

Example I
(Siebes et al ’06)

- Score pattern set by MDL encoding of db:
  \[ L_C(db) = L(C, S_C)(db) + L(CT_C) \]

- Order patterns by size and support

- Fixed order scan
  - Pick first improving score
  - Some pruning

- Also:
  - Bringmann et al ’07
  - Al Hasan et al ’07
Significance $S$ traded off against redundancy $L$:

$$G_{gen}(P^k) = \sum_{i=1}^{k} S(p_i) - L(P^k)$$

**Use TIDs only**

**Greedy:**
- Add pattern improving $G$ most
- Until $|S| = k$

**Also:**
- Garriga et al ’07
- Cheng et al ’07
- Miettinen et al ’08
- Bringmann et al ’09
- Thoma et al ’09
Model independent Iterative Mining

- Mine (set of) pattern(s)
- Adjust scoring function according to pattern
- Re-Mine
Sequential Mining
(Cheng et al ‘08)

- Information Gain
- Sequential covering:
  - Mine most discriminating pattern
  - Add to set
  - Remove covered instances
  - Until $|S| = k$

Also:
- Rückert et al ‘07
- Thoma et al ‘09
Model dependence

- Final model influences patterns’ selection
- Can be used in any model, optimized for one
- Less modular, stages need to coordinate
Model dependent techniques

Model types

- **Votes** of patterns
  - Weighted votes
  - Compression-based

- **Ordered list** of patterns
  - Some of which can be compressed into trees

- **Tree** of patterns
- Mine large set of patterns
- Post-process depending on model constraints
- (Check on model effectiveness)
Model dependent Post-Processing

Fixed order scan

- Sorting order
  - Confidence/support
  - Growth rate/support
  - Size/support
  - $X^2$/support
  - Unimportant - every pattern above threshold chosen

- Patterns chosen
  - Independent of particular classes
  - Per class
Model dependent Post-Processing

**Example I**

(Zaki et al '03)

- Model: weighted vote
- Fix measure for predictive strength
- Filter patterns on strength threshold

Also:

- Wang et al '05
- Arunasalam et al ‘06

Threshold Selection: 3
Model dependent Post-Processing

**Example II**
(Liu et al ’98)

- Model: ordered list
- Order: confidence/support
- Hill-climbing:
  - Pick first pattern correctly predicting at least one training instance
  - Remove covered training data

Also:
- Dong et al ’99
- Li et al ’01
- Zimmermann et al ’05
- Van Leeuwen et al ‘06

**Fixed Order: 5**
Model dependent Post-Processing

Example II

(Liu et al ’98)

- Model: ordered list
- Order: confidence/support
- Hill-climbing:
  - Pick first pattern correctly predicting at least one training instance
  - Remove covered training data

Also:
- Dong et al ’99
- Li et al ’01
- Zimmermann et al ’05
- Van Leeuwen et al ’06

Also:
- Siebes et al ’06!
Model dependent Post-Processing

Example II
(Liu et al ’98)

Model: ordered list
Order: confidence/support
Hill-climbing:
- Pick first pattern correctly predicting at least one training instance
- Remove covered training data

Also:
- Dong et al ’99
- Li et al ’01
- Zimmermann et al ’05
- Van Leeuwen et al ’06

Fixed Order: 8
Model dependent Post-Processing

Example III
(Nijssen et al ’07)

- Model: patterns as tree
- Mine/filter patterns based on model constraints
- Each itemset a DT branch
- Scan lattice bottom up, enforcing model constraints

Also:
- Gay et al ‘07
Model dependent

Iterative Mining

- Clearest connection to ML
- Features made-to-fit
- Overfitting danger
Sequential Covering
(Galiano et al ‘04)

- Model: ordered list
- Algorithm:
  - Mine patterns
  - Select set of mutually exclusive patterns
  - Remove covered data

Also:
- Yin et al ‘03
Decision Tree Construction
(Bringmann et al. ‘05)

- Model: tree of patterns
- Algorithm:
  - Mine most discriminating pattern (information gain)
  - Split data into covered and uncovered
- Also:
  - Geamsakul et al. ’03
  - Fan et al. ‘08
Lazy Learning
(Li et al ’00)

- Model: weighted vote
- For each testing instance:
  - Project db on syntactic elements
  - Mine highly predictive patterns
- Also:
  - Veloso et al ‘06

Model dependent Iterative Mining
Model dependent Iterative Mining

Boosting/Regression
(Nowozin et al '07)

- Model: weighted vote
- Algorithm
  - Mine predictive pattern
  - Re-weight mis-classified training instances as in Linear Programming Boosting
- Weights derived from mining
- Also:
  - Saigo et al '08

![Diagram of DB, Pattern Mining, Feature Selection, Model Induction]
Conclusions

Let’s Count

Model-Independent Post-Processing
- Fixed Order: 3
- Greedy: 6

Model-Dependent Post-Processing
- Threshold Selection: 3
- Fixed Order: 5
- Decision Tree Construction: 2

Model-Independent Iterative Mining
- Sequential Mining: 3

Model-Dependent Iterative Mining
- Sequential Mining: 2
- Lazy Learners: 2
- DT Construction: 3
- Boosting-Like: 2
Conclusions

Let’s Count

- **Model-Independent**
  - Post-Processing
  - Fixed Order: 3
  - Greedy: 6
- **Model-Dependent**
  - Post-Processing
  - Fixed Order: 8
  - Decision Tree Construction: 2

- **Iterative Mining**

- **Sequential Mining**: 3
- **Lazy Learners**: 2
- **DT Construction**: 3
- **Boosting-Like**: 2
- **Iterative Mining**

- **Fixed Order**: 3
- **Greedy**: 6
- **Sequential Mining**: 3
Conclusions

Let’s Count

Post-Processing

- Fixed Order: 11
- Greedy: 6
- Decision Tree Construction: 2

Model-Independent Iterative Mining

- Sequential Mining: 3

Model-Dependent Iterative Mining

- Sequential Mining: 2
- Lazy Learners: 2
- DT Construction: 3
- Boosting-Like: 2
Conclusions

Let's Count

**Post-Processing**
- Fixed Order: 11
- Greedy: 6
- Decision Tree Construction: 2

**Iterative Mining**
- Sequential Mining: 5
- Lazy Learners: 2
- DT Construction: 3
- Boosting-Like: 2
Conclusions

Let’s Count

Post-Processing
- Fixed Order: 11
- Greedy: 6
- Decision Tree Construction: 2

Iterative Mining
- Sequential Mining: 5
- Lazy Learners: 2
- DT Construction: 3
- Boosting-Like: 2
Conclusions

Let’s Count

WE BROUGHT YOU

31

LeGo techniques
Conclusions

- Large number of existing LeGo approaches
- Two main dimensions
  - Model (in)dependence
  - Post-Processing & Iterative Mining
  - Boundaries blur
- Mostly very flexible

- Few studies in relative effectiveness
  - Deshpande et al ’05
  - Wale et al ’08
  - Janssen et al ’09
### The exact picture

#### Model independent PP

|                  | TID Score | Pattern Structure Score | Search         | Score used     |
|------------------|-----------|-------------------------|----------------|----------------|
|                  | Sig       | Red                     | Distance       | Compress       | Fixed | Greedy | Approx | Score used |
| Siebes et al '06 | X         |                         |                | X              | X     |        |        | MDL        |
| Xin et al '06    |           | X                       | X              |                |        |        |        | mutual distance |
| Bringmann et al '07 |         | X                       |                |                | X     |        |        | partition based |
| Garriga et al '07 |           |                          |                |                |        |        | X      | marginal gain |
| Al Hasan et al '07 |         |                          |                |                |        |        | X      | clique based |
| Cheng et al '06  |           |                          |                |                |        |        |        | Jaccard coeff. |
| Miettinen et al '08 |        |                          |                |                | X     |        |        | discrete basis |
| Bringmann et al '09 |        | X                       |                |                |        |        | X      | partition based |
| Thoma et al '09  |           |                          |                |                |        |        | X      | pairs of misclass |

Some greedy algorithms approximate a well-defined global optimum
### The exact picture
#### Model dependent PP

| Model Type | Order | Selection |
|------------|-------|-----------|
| Voting     | Compress | List | Conf. | Growth | $X^2$ | Threshold | Per class | Indep |
| Liu et al '98 | X | X | | | | | X |
| Dong et al '99 | | X | | | | | X |
| Li et al '01 | | X | | | | | X |
| Zaki et al '03 | | | X | | | | X |
| Wang et al '05 | | X | | | | | X |
| Zimmermann et al '05 | | X | | | | | X |
| Van Leeuwen et al '06 | | X | | | | | X |
| Arunasalam et al '06 | | X | | | | | X |

### Diagram

- DB → Pattern Mining → PS → Feature Selection → PS → Model Induction → M