On the specificity of distance-based generalisation operators

V. Estruch, C. Ferri, J. Hernández-Orallo & M.J. Ramírez-Quintana

Universitat Politècnica de València
Departament de Sistemes Informàtics i Computació
(Spain)

The 16th International Conference on Inductive Logic Programming
ILP’06
Index of contents

- Introduction
- Preliminaries
- Aim of this work: minimal distance-based generalisation operators ($mg$)
- Trivial attempt: $mg$ via the inclusion operator
- $mg$ via a distance-based cost function: $k(E,p)$
- An illustrative example
- Conclusions and future work
Introduction

- In some learning problems, data is not only described by nominal or numerical features but also by other sorts of data (sets, lists, etc).
  - E.g.: Text and web mining, molecular classification, etc.
- Methods dealing with structured information are needed.
- Distance-based methods are widely used in structured learning:
  - They have been successfully tested in several real-world domains: E.g.: image recognition.
  - They can be easily upgraded: distance functions can be found for several data types.
Introduction

- Distance-based methods transform the similar traits between two objects into a numerical value

\[ d(\text{Once upon a time, in a faraway kingdom...}, \text{Once the time interval is selected...}) = n \]

- As a consequence,
  - The information of the matches is lost.
  - Sometimes, a pattern/model informing about the similarities among several objects could be useful → Some issues arise...
Let \((X,d)\) and \(L\) be a metric space and a pattern language respectively. The mapping
\[
\Delta : E \subset 2^X \rightarrow p \in L
\]
is a \textbf{distance-based generalisation operator} if for every \(E, p\) "explains" the distances among the elements in \(E\).

\(X=\text{space of finite lists} \quad \& \quad d=\text{edit distance}\) permitting insertions and deletions only. Given \(w_1=abb\) and \(w_2=bba:\)
\[
d(w_1, w_2) = \begin{array}{c}
\text{abb} \\
\text{bba}
\end{array} = 2
\]

\(\Delta_1(\{w_1, w_2\}) = *a^* \ (\text{no distance-based}) \quad \Delta_2(\{w_1, w_2\}) = *bb^* \ (\text{distance-based})\)
Aim of this work

How general an admissible pattern is?

- Minimal generalisations are important if we want a pattern/model to fit a set of examples as much as possible.

- We focus on defining minimal distance-based generalisation operators.
Trivial attempt: mg via the inclusion operator

- Every pattern $p$ represents a set denoted by $\text{Set}(p)$.
- Let $E$ be a set of elements in $(X,d)$, $\Delta_1(E)=p_1$ and $\Delta_2(E)=p_2$, then we could have:

$$\text{Set}(p_1) \subseteq \text{Set}(p_2)$$

$p_1$ is less general than $p_2$ iff $\text{Set}(p_1) \subseteq \text{Set}(p_2)$

Some drawbacks appear
Trivial attempt: mg via the inclusion operator

- The inclusion operator ignores the underlying distance

1. The patterns $p_1$ and $p_2$ cannot be compared.
2. Intuitively, $p_1$ seems less general than $p_2$
Trivial attempt: mg via the inclusion operator

- The minimal generalisation might not exist
  - E.g.: The space $\mathbb{R}^2$ with the Euclidean distance and $L = \{\text{rectangles in } \mathbb{R}^2 \text{ along with their finite unions}\}$
mg via a distance-based cost function: \( k(E,p) \)

- The level of “complexity” of a pattern is reasonable only if a sufficient number of examples justifies it (MDL/MML).
- A distance-based cost function \( k(E,p) \) is introduced for this purpose:

\[
K(\cdot, \cdot): E \subset 2^X \times p \in L \rightarrow \mathbb{R}
\]

**Goal:** Given \( \Delta(E)=p \), then \( k(E,p) \) measures the complexity of the pattern \( p \) and how good \( \text{Set}(p) \) fits \( E \).

**Property:** Given \( \Delta_1(E)=p_1 \) and \( \Delta_2(E)=p_2 \) such that \( \text{Set}(p_1) \subseteq \text{Set}(p_2) \) and both patterns are equally complex, then \( k(E,p_1) \leq k(E,p_2) \).

**Definition:** \( \Delta \) is a mg operator if \( k(E,\Delta(E)) \leq k(E,\Delta'(E)) \) for all \( E \) and \( \Delta' \).
The function $k(E, p)$ can be expressed as:

$$K(E, p) = c(p) + c(E \mid p)$$

### Complexity of the pattern

<table>
<thead>
<tr>
<th>Sort of data</th>
<th>$L$</th>
<th>$c(p)$</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical</td>
<td>Closed intervals</td>
<td>Length of the interval</td>
<td>$c([a, b]) = b-a$</td>
</tr>
<tr>
<td>First order atoms</td>
<td>Herbrand base with variables</td>
<td>Number of symbols</td>
<td>$c(q(a,X,X)) = 4$</td>
</tr>
<tr>
<td>Any</td>
<td>Any</td>
<td>Constant function</td>
<td>$c(p) = \text{cte.}$</td>
</tr>
</tbody>
</table>
The function $k(E,p)$ can be expressed as:

$$K(E,p) = c(p) + c(E|p)$$

| Sort of data | $L$ | $C(E|p)$ |
|--------------|-----|----------|
| Any          | Any | $c(E,p) = \sum_{e \in E} \min_{e' \in \partial \text{Set}(p)} d(e,e')^*$ |
| Any          | Any | $c(E,p) = \sum_{e \in E} \min_{e' \in \partial \text{Set}(p)} d(e,e')$ $+ \max_{e'' \in \partial \text{Set}(p)} d(e,e'')^{**}$ |

(*) Note that the concept of border can be expressed in any metric space

(**) Only for bounded sets
An illustrative example

- \((\mathbb{R}^2, d=\text{Euclidean dist.}) \& L=\{\text{rectangles in } \mathbb{R}^2 \text{ with their finite unions}\}\)
- \(k(E,p) = c(p) + c(E|p)\)
  - \(c(p) = \text{(number of squares) } \times \alpha(E,d) \rightarrow (\alpha(\cdot,\cdot) = \text{scale factor})\)
  - \(c(E|p) = \sum d(e_i, \text{nearest element in } \partial p)\)

\[\Delta_1(A,B)=p_1 \quad \Delta_3(A,B)=p_3 \quad B(3,4)\]

- \(p_1\) is the mg for \(E=\{A,B\}\)
- \(c(p_1)=\alpha(E,d)\)
- \(c(E|p_1)=0,\) since \(A\) and \(B\) are in \(\partial\text{Set}(p_1)\)
- \(c(p_3)=4\cdot\alpha(E,d)\)
- \(c(E|p_3)=0,\) since \(A\) and \(B\) are in \(\partial\text{Set}(p_3)\)
Conclusions and future work

- *Mg* operators are important if we want a generalisation to fit a set of examples as much as possible.
  - E.g. Conceptual clustering
- This work introduces the notion of minimal distance-based generalisation operator via a distance-based cost function $k(\cdot, \cdot)$ for data embedded in a metric space.
  - *mg* operators can be defined for several sorts of data (sets, lists, graphs, etc.)
- It can be easily shown that the *Plotkin’s lgg* is a particular case of the *mg* operator.
  - Different *mg* operators can be obtained by defining other cost functions $k(\cdot, \cdot)$
Appendix (I): Plotkin’s lgg is a mg operator

- (X= first-order atoms, d= J. Ramon et al. distance) & L={Herbrand base with variables}
- \( k(E,p) = c(p) + c(E|p) \)
  - \( c(p) = \) constant function \( x \) \( a(E,d) \)
  - \( c(E|p) = \sum d(e_i, \text{nearest element in } \partial p) \)
- Given a set of atoms \( E=\{e_1,\ldots, e_n\} \)
  - \( \Delta(E)=lgg(E) \) is a distance-based operator
  - Let \( p \) be an atom such that for all \( i=1,\ldots,n \), there exists a substitution \( \theta_i \) satisfying \( p\theta_i=e_i \)
- (sketch of proof) By definition, \( p \) is more general than \( lgg(E) \). Then for all \( E \), \( Set(lgg(E)) \) is included in \( Set(p) \) and \( k(E,lgg(E)) \leq k(E,p) \).
Appendix (II): working with lists

(X=space of finite lists, d= edit distance) & \( L = \{ X_1aa, X_1X_2, bX_1X_2, \ldots \} \)

\[ k(E, p) = c(p) + c(E|p) \]

- \( c(p) \) = number of symbols of \( p \) x D(E) (diameter)
- \( c(E|p) = \sum d(e_i, \text{nearest element not belonging to } Set(p)) \)

Given the set \( E = \{ abb, bba \} \rightarrow D(E)=2 \)

- \( p_1 = X_1bbX_2 \) and \( p_2 = X_1bX_2 \) are distance-based patterns
- \( k(E, p_2) \leq K(E, p_1). \) That is,

<table>
<thead>
<tr>
<th>( p_1 = X_1bbX_2 )</th>
<th>( p_2 = X_1bX_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c(p_1) = 4 \cdot D(E) = 8 )</td>
<td>( c(p_2) = 3 \cdot D(E) = 6 )</td>
</tr>
<tr>
<td>( c(E</td>
<td>p_1) = d(abb, babb) + d(bba, bbab) = 2 )</td>
</tr>
</tbody>
</table>
Appendix (III): Distance for terms & atoms (*J. Ramon et al.*)

Bidimensional distance (lex. order)

\[ \text{lgg}(a_1, a_2) = p(X) \]
\[ S(p(X)) = (1, 1) \]

\[ S(a_1) = (F, V) \]
\[ d(a_1, a_2) = \Delta S |^{a_1}_{\text{lgg}(a_1, a_2)} + \Delta S |^{a_2}_{\text{lgg}(a_1, a_2)} \]
\[ d(a_1, a_2) = [(2, 0) - (1, 1)] + [(2, 0) - (1, 1)] \]
\[ = (2, -2) \]

\[ a_1 = p(a) \]
\[ S(a1) = (2, 0) \]

\[ a_2 = p(b) \]
\[ S(a2) = (2, 0) \]