UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Applying distances between terms to both flat and hierarchical data

J.A. Bedoya-Puerta,
C. Ferri,
J. Hernández-Orallo,
M.J. Ramírez Quintana.

DSIC, Universitat Politècnica de València

Contents

Contents

1. Introduction

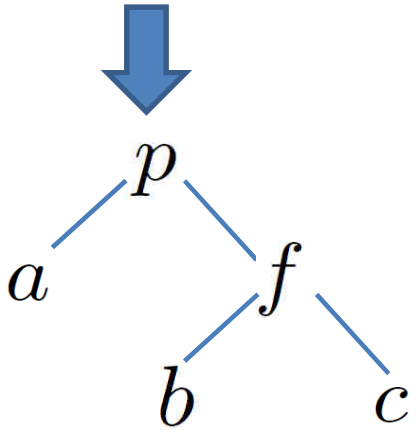
- Distances are measures of dissimilarity with some special properties, such as symmetry and triangle inequality, which make them more advantageous for many algorithms.
- There are distances for virtually any kind of object, including complex or highly structured ones, such as tuples, sets, lists, trees, graphs, etc.
- Distances between atoms may not only be useful in ILP, but also in other areas where structured (hierarchical) information is involved, such as learning from ontologies or XML documents.
- The advantage of term distances is that they are able to consider context and, some of them, repetitions.

1. Introduction

- Tree structures and functional terms have strong similarities.

- Some popular languages for information representation are based on trees or hierarchies, such as XML.

$$t = p(a, f(b, c))$$




```
<?xml version="1.0"?>
<p>
  <p1> a </p1>
  <f>
    <f1> b </f1>
    <f2> c </f2>
  </f>
</p>
```

1. Introduction

- Objectives:
  - Define a transformation procedure to convert semi-structured data into a term-based representation in XML.
  - Define several mechanisms to extract hierarchies from flat data, using the attribute names or their correlation.
  - Experimentally analyse the performance of term distances over the transformed data.

Contents

2. Distances over terms

- In order to use with a general inductive method, such as *k-NN,* we worked with the following distances over terms:
  - Nienhuys-Cheng's distance
  - Estruch et al.'s distance

2. Distances over terms

Nienhuys-Cheng distance:

Given two ground expressions: $s = s_0(s_1, \ldots, s_n)$
$t = t_0(t_1, \ldots, t_n)$

distance between them is recursively defined as:

$$d_N(s,t) = \begin{cases} 0, & \text{if } s = t \\ 1, & \text{if } \neg Compatible(s,t) \\ \frac{1}{2n} \sum_{i=1}^{n} d(s_i, t_i), & otherwise \end{cases}$$

It takes into account the depth of the symbol occurrences.

2. Distances over terms

Nienhuys-Cheng distance example:

$$d_N(s,t) = \begin{cases} 0, & \text{if } s = t \\ 1, & \text{if } \neg Compatible(s,t) \\ \frac{1}{2n}\sum_{i=1}^{n} d(s_i,t_i), & otherwise \end{cases}$$

Given two ground expressions: $s = p(a,a)$ $t = p(f(b),f(b))$

$$d_N(s,t) = \frac{1}{4} \cdot \big(d(a,f(b) + d(a,f(b))\big)$$

$$d_N(s,t) = \frac{1}{4} \cdot (1+1) = \frac{1}{2}$$

2. Distances over terms

Estruch et al. distance:

Given two ground expressions: $s = s_0(s_1, \ldots, s_n)$
$t = t_0(t_1, \ldots, t_n)$

distance between them is defined as:

$$d_E(s,t) = \sum_{o \in O^\star(s,t)} \frac{w(o)}{C(o)} \left(Size'(s|_o) + Size'(t|_o) \right)$$

It takes into account:

- Context of the differences *C(o).*
- Syntactical complexity through size of the expression *Size'(s|$_{o'}$)*
- The weight *w(o)* associated with the repetitions, using equivalence relations.

2. Distances over terms

Estruch et al. distance example:

$$d_E(s,t) = \sum_{o \in O^\star(s,t)} \frac{w(o)}{C(o)} \left(Size'(s|_o) + Size'(t|_o) \right)$$

Given two ground expressions: $s = p(a,a) \quad t = p(f(b), f(b))$

$$C(1) = C(2) = 2 \cdot (2+1) = 6$$

$$Size'(a) = 1/4 \text{ and } Size'(f(b)) = 5/16$$

$$w(1) = 1 \text{ and } w(2) = 7/8$$

$$d_E(s,t) = \frac{1}{6}\left(\frac{1}{4} + \frac{5}{16}\right) + \frac{7}{48}\left(\frac{1}{4} + \frac{5}{16}\right)$$

Contents

3. Transforming semi-structured data into a term-based representation using XML

- A common language for representing several datatypes is XML.

- We distinguished two situations:

  - **Flat data**: are given as a table of attribute-value pairs.

  - **Hierarchical data at source**: consist of data with a hierarchical structure which is represented by a tree or a functional term.

3. Transforming semi-structured data into a term-based representation using XML

Schema definition:

- We need to make some decisions in order to use XML to represent functional terms, and also to adapt the previous types of data to a common representation.

- It is necessary to ensure that distance calculations are not affected by the absence of features or their order.

- This difficulty requires the creation of a general schema that allows each instance, with its own features, to be properly adjusted, without losing any element or content, and in a defined order.

3. Transforming semi-structured data into a term-based representation using XML

Schema definition:

```
...
<COLOUR>
  <INTERNAL>WHITISH</INTERNAL>
  <EXTERNAL>GREY</EXTERNAL>
</COLOUR>
...
```

Hierarchy 1

```
...
<COLOUR>
  <INTERNAL>WHITISH</INTERNAL>
</COLOUR>
<PIGMENTATION>YES</PIGMENTATION>
...
```

Hierarchy 2

```
<COLOUR>
    <INTERNAL/>
    <EXTERNAL/>
</COLOUR>
<PIGMENTATION/>
```

3. Transforming semi-structured data into a term-based representation using XML

Deriving hierarchical XML schemas from flat data

This work considered three possible sources of structures from flat representations:

1. Value equality
2. Name-induced hierarchies
3. Attribute-similarity hierarchy

3. Transforming semi-structured data into a term-based representation using XML

1. Value equality

A detailed inspection shows that some attributes are related by the values they take.

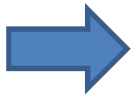
For instance:

Two variables X_1 and X_2 can take the values ***East***, ***West***, ***North*** and ***South***; there is clearly a connection between them that can be exploited, especially through the use of equalities.

3. Transforming semi-structured data into a term-based representation using XML

2. Name-induced hierarchies

1. cap-shape
2. cap-surface
3. cap-color
4. bruises
5. odor
6. gill-attachment
7. gill-spacing
8. gill-size
9. gill-color
10. stalk-shape
11. stalk-root
12. stalk-surface-above-ring
13. stalk-surface-below-ring
14. stalk-color-above-ring
15. stalk-color-below-ring
16. veil-type
17. veil-color
18. ring-number
19. ring-type
20. spore-print-color
21. population
22. habitat



```
<Mushroom>
  <cap>
    <shape>CONVEX</shape>
    <surface>SMOOTH</surface>
    <color>WHITE</color>
  </cap>
  <bruises>BRUISES</bruises>
  <odor>ALMOND</odor>
  <gill>
    <attachment>FREE</attachment>
    <spacing>CROWDED</spacing>
    <size>NARROW</size>
    <color>WHITE</color>
  </gill>
  <stalk>
    <shape>TAPERING</shape>
    <root>BULBOUS</root>
    <surface>
      <above_ring>SMOOTH</above_ring>
      <below_ring>SMOOTH</below_ring>
    </surface>
    <color>
      <above_ring>WHITE</above_ring>
      <below_ring>WHITE</below_ring>
    </color>
  </stalk>
  <veil>
    <type>PARTIAL</type>
    <color>WHITE</color>
  </veil>
  ...
```

3. Transforming semi-structured data into a term-based representation using XML

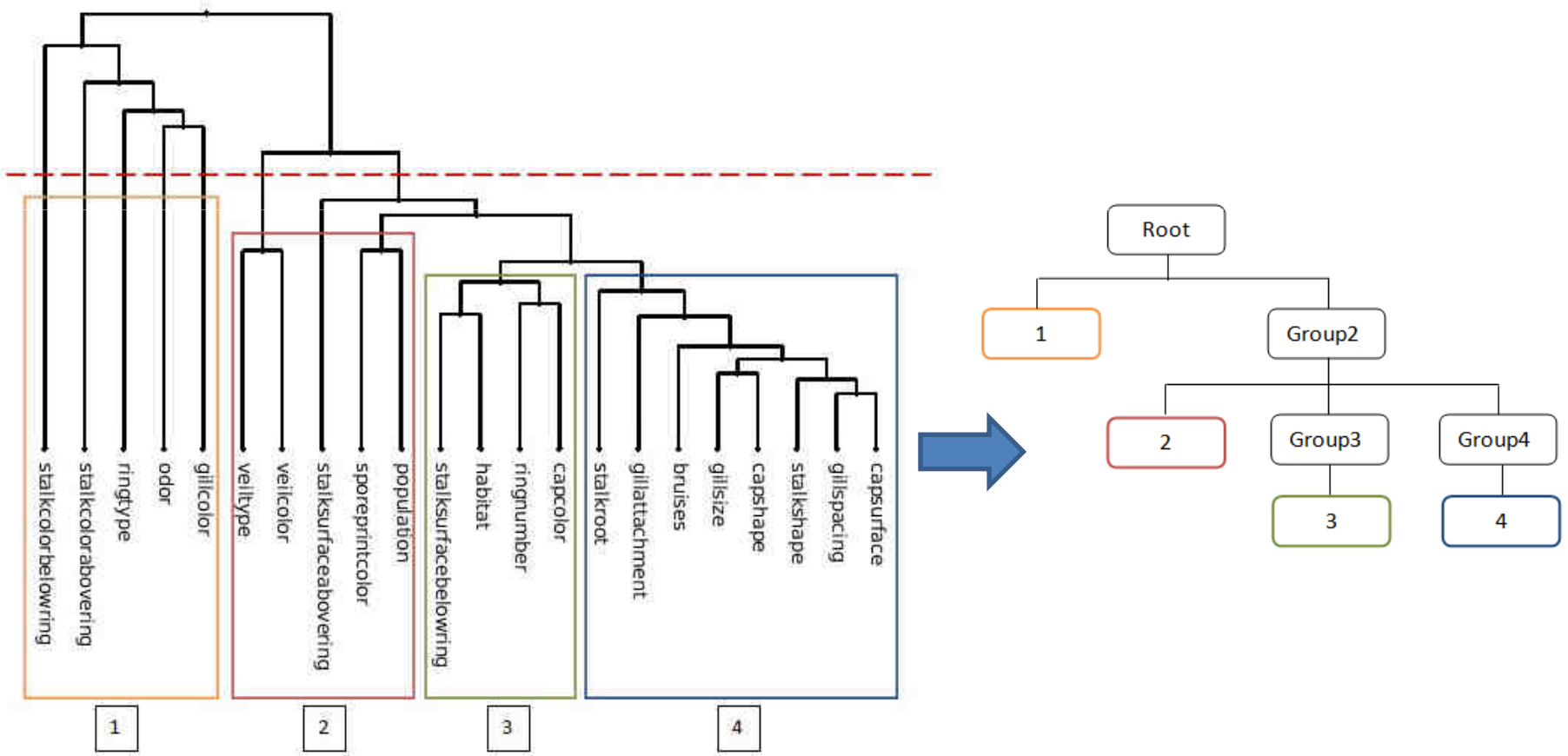
2. Name-induced hierarchies

```
Mushroom (
        cap (CONVEX, SMOOTH, WHITE),
        BRUISES, ALMOND,
        gill (FREE, CROWDED, NARROW, WHITE),
        stalk (TAPERING, BULBOUS,
                surface(SMOOTH, SMOOTH),
                color (WHITE, WHITE)),
        veil (PARTIAL, WHITE),
        ring(ONE, PENDANT),
        spore (print (BROWN)),
        SEVERAL, WOODS
)
```

```
<Mushroom>
  <cap>
    <shape>CONVEX</shape>
    <surface>SMOOTH</surface>
    <color>WHITE</color>
  </cap>
  <bruises>BRUISES</bruises>
  <odor>ALMOND</odor>
  <gill>
    <attachment>FREE</attachment>
    <spacing>CROWDED</spacing>
    <size>NARROW</size>
    <color>WHITE</color>
  </gill>
  <stalk>
    <shape>TAPERING</shape>
    <root>BULBOUS</root>
    <surface>
      <above_ring>SMOOTH</above_ring>
      <below_ring>SMOOTH</below_ring>
    </surface>
    <color>
      <above_ring>WHITE</above_ring>
      <below_ring>WHITE</below_ring>
    </color>
  </stalk>
  <veil>
    <type>PARTIAL</type>
    <color>WHITE</color>
  </veil>
  ...
```

3. Transforming semi-structured data into a term-based representation using XML

3. Attribute-similarity hierarchy



3. Transforming semi-structured data into a term-based representation using XML

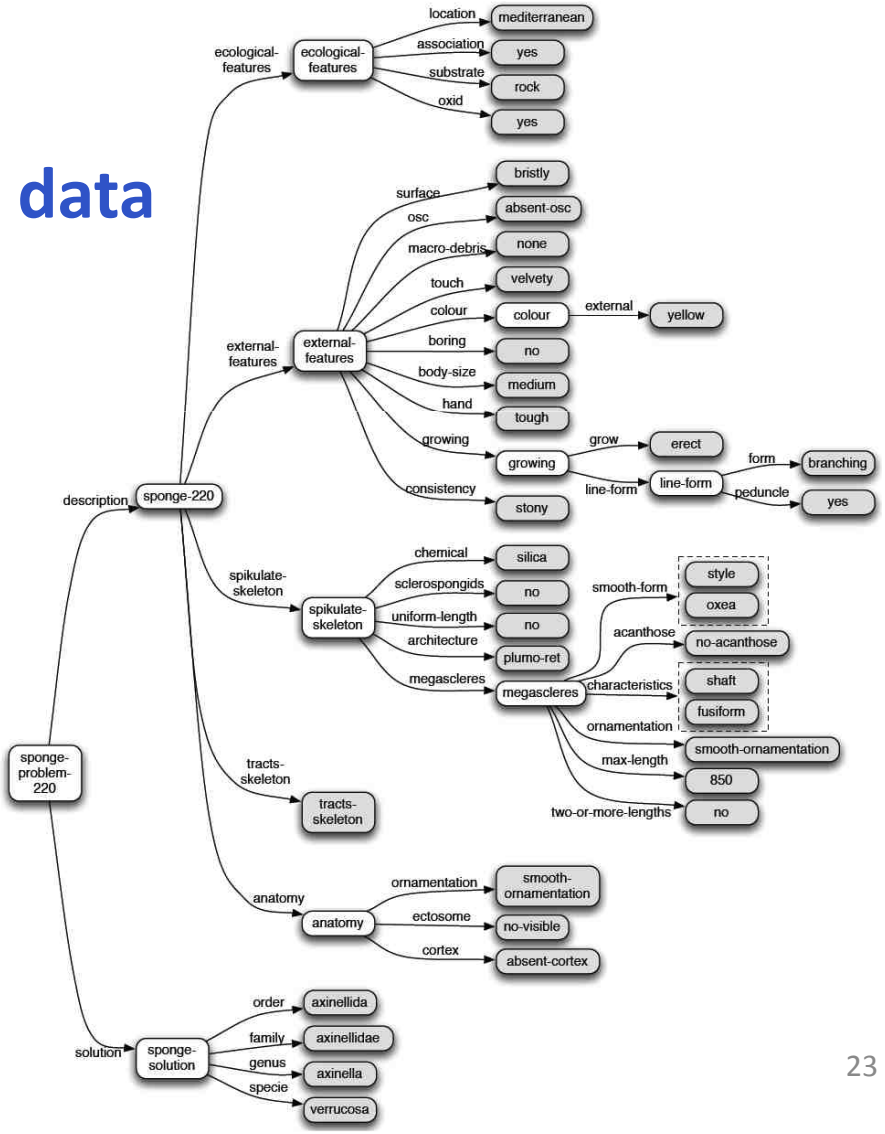
Deriving hierarchical XML schemas from hierarchical data

- It is necessary to determine whether order, repetitions and labels are relevant, in order to determine the features and place them correctly in the XML document.

3. Transforming semi-structured data into a term-based representation using XML

Deriving hierarchical XML schemas from hierarchical data

```
<BODY-SIZE>MEDIUM</BODY-SIZE>
<CONSISTENCY>STONY</CONSISTENCY>
<HAND>TOUGH</HAND>
<TOUCH>VELVETY</TOUCH>
<SURFACE>BRISTLY</SURFACE>
<GROWING>
  <DEF>GROWING</DEF>
  <GROW>ERECT</GROW>
  <LINE-FORM>
    <DEF>LINE-FORM</DEF>
    <FORM>BRANCHING</FORM>
    <PEDUNCLE>YES</PEDUNCLE>
  </LINE-FORM>
</GROWING>
<BORING>NO</BORING>
<COLOUR>
  <DEF>COLOUR</DEF>
  <EXTERNAL>YELLOW</EXTERNAL>
</COLOUR>
<CRIBLE>NO</CRIBLE>
<HOLLOW>NO</HOLLOW>
<BRUSH>NO</BRUSH>
<BRIOZOA>NO</BRIOZOA>
<MACRO-DEBRIS>NONE</MACRO-DEBRIS>
```



Contents

4. Experiments

- k-nearest neighbor (*k*-NN) algorithm.
- Weighted *k*-NN variant, using an attraction function which gives more or less weight to each of the k-most nearest examples, defined as

$$\frac{1}{d^i}$$ (*i* varying from *0* to *3*).

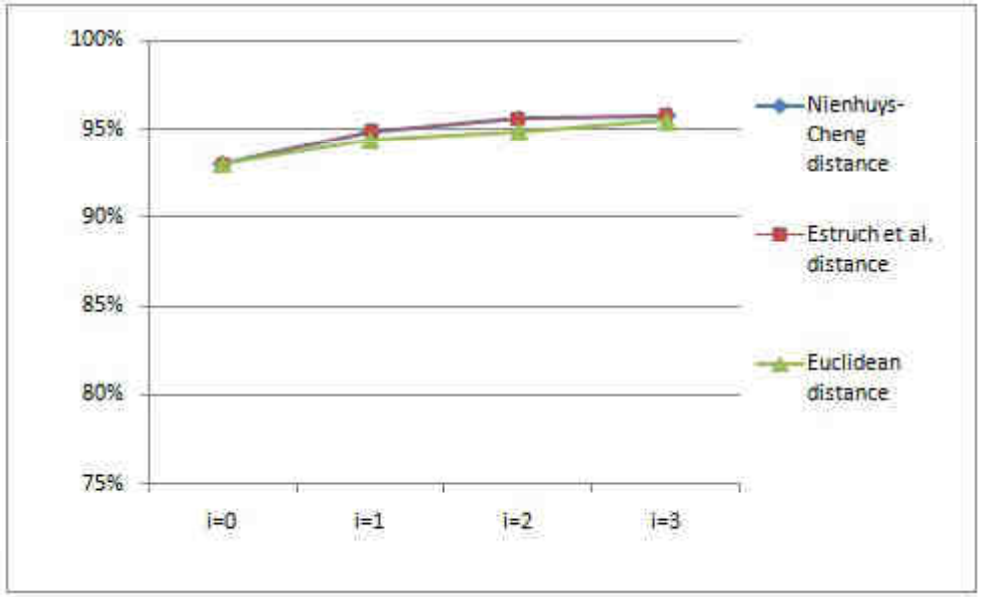
- Three distances: the Nienhuys-Cheng distance, the Estruch et al. distance and the Euclidean distance.
- Datasets:
  - Flat: Mushroom (1000 examples) and Soybean:
    - 10-fold cross validation experiment.
    - A paired t-test between the methods (confidence 95%).
  - Hierarchy: Demospongiae (sponge) dataset.
    - 60% for training examples and a 40% for test examples.

4. Experiments

Mushroom dataset

- Distances without hierarchy data.

	i=0 %	i=1 %	i=2 %	i=3 %
d_N	93.0	94.9	95.6	95.8
d_E	93.0	94.9	95.6	95.8
d_U	93.0	94.4	94.9	95.5



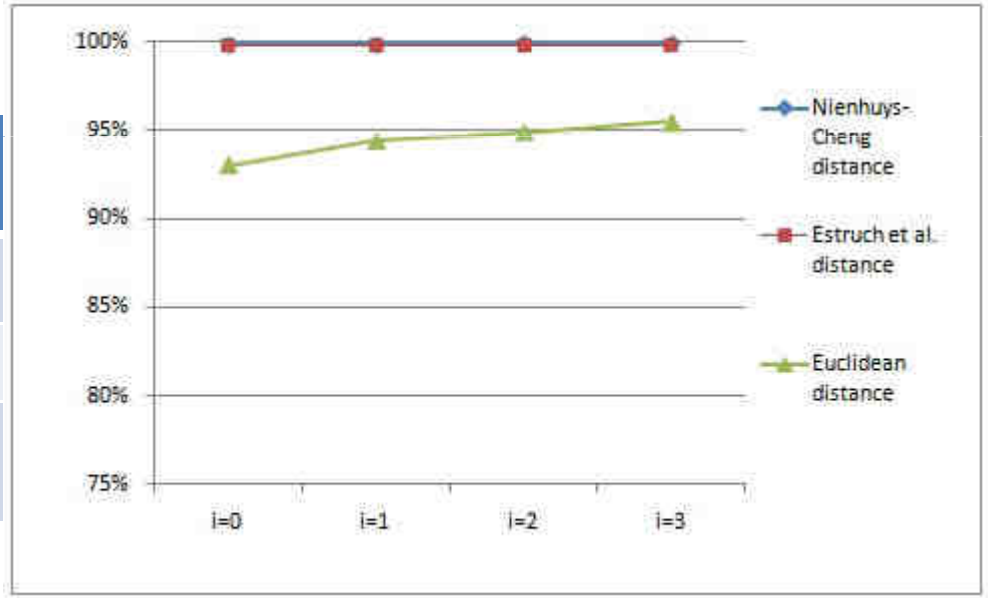
- The differences between distances were not significant.

4. Experiments

Mushroom dataset

- Distances with hierarchy induced from the attributes names.

	i=0 %	i=1 %	i=2 %	i=3 %
d_N	99.8	99.8	99.9	99.9
d_E	99.8	99.8	99.8	99.8
d_U	93.0	94.4	94.9	95.5



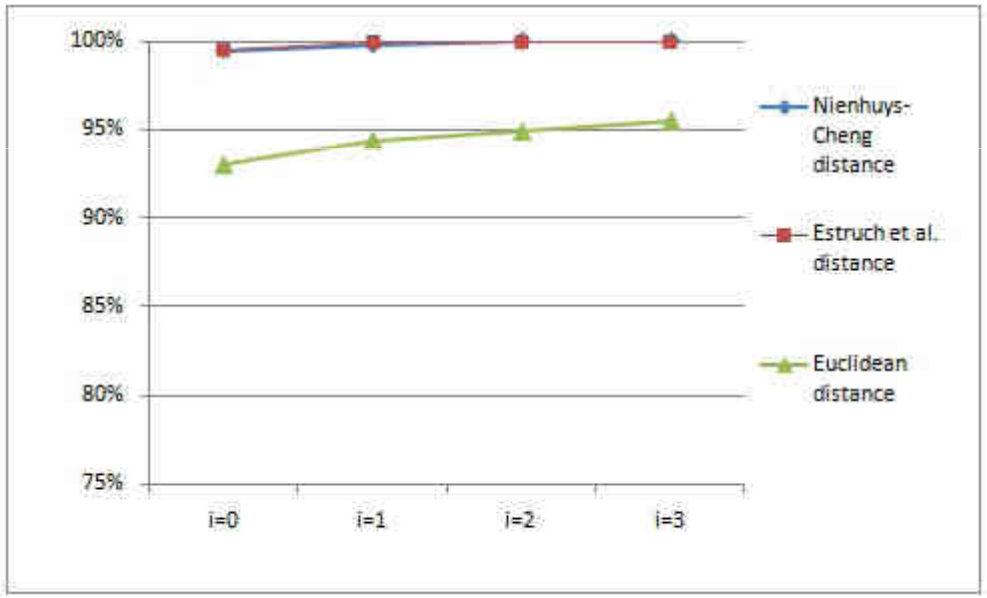
- The performance of Estruch et al's and Nienhuys-Cheng's distances was optimum.

4. Experiments

Mushroom dataset

- Distances with hierarchy using similarities between attributes.

	i=0 %	i=1 %	i=2 %	i=3 %
d_N	99.5	99.8	100	100
d_E	99.5	100	100	100
d_U	93.0	94.4	94.9	95.5



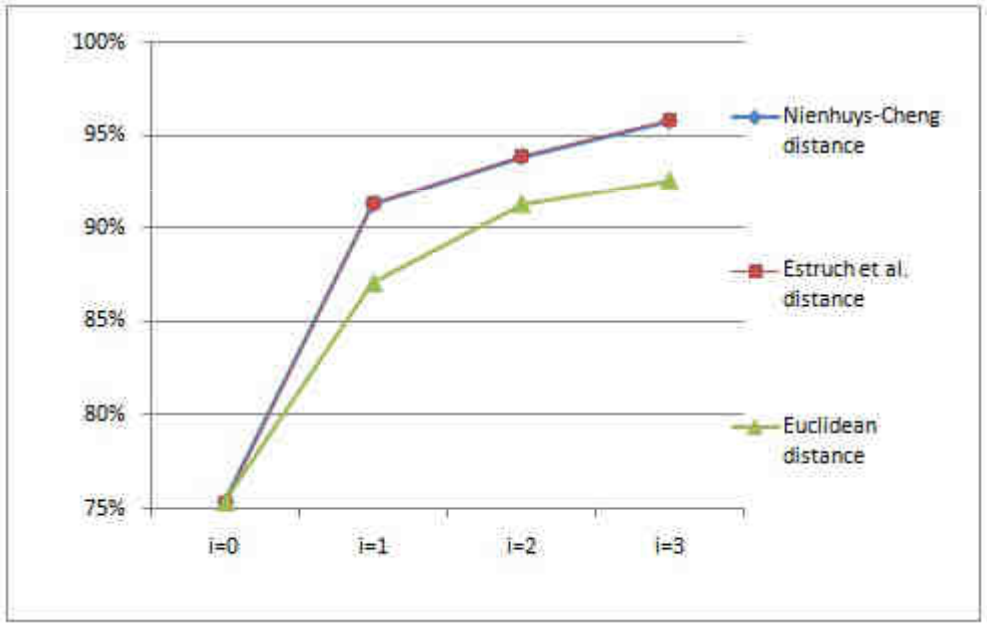
- The distances between terms exploited this structure and got much better results than the Euclidean distance.

4. Experiments

SoyBean dataset

- Distances without hierarchy data.

	i=0 %	i=1 %	i=2 %	i=3 %
d_N	75.3	91.3	93.9	95.8
d_E	75.3	91.3	93.9	95.8
d_U	75.3	87.1	91.3	92.6



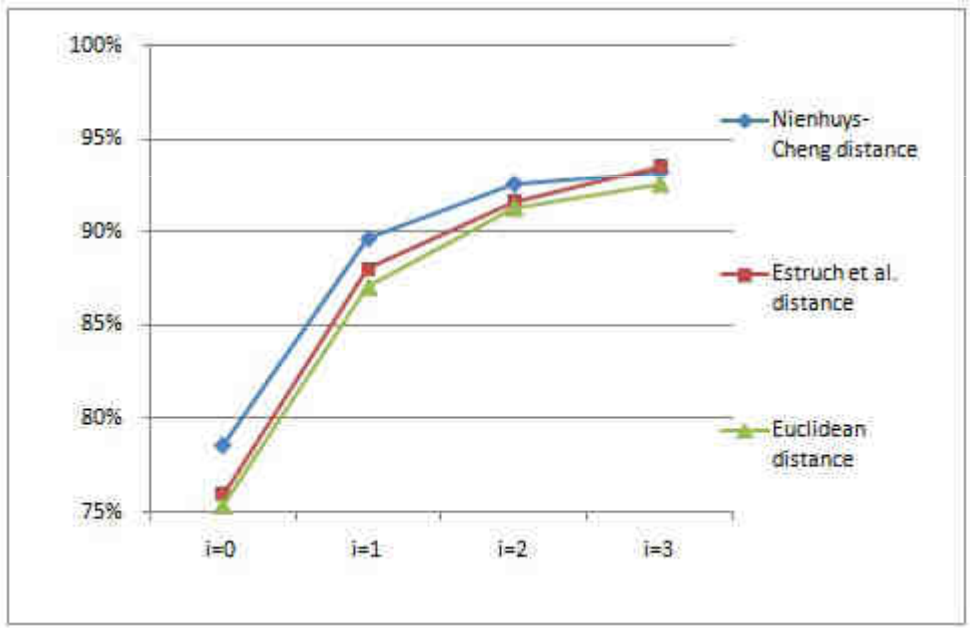
- The Euclidean distance obtained less favorable results than the two term distances.

4. Experiments

SoyBean dataset

- Distances with hierarchy induced from the attributes names.

	i=0 %	i=1 %	i=2 %	i=3 %
d_N	78.6	89.6	92.6	93.2
d_E	76.0	88.0	91.6	93.5
d_U	75.3	87.1	91.3	92.6



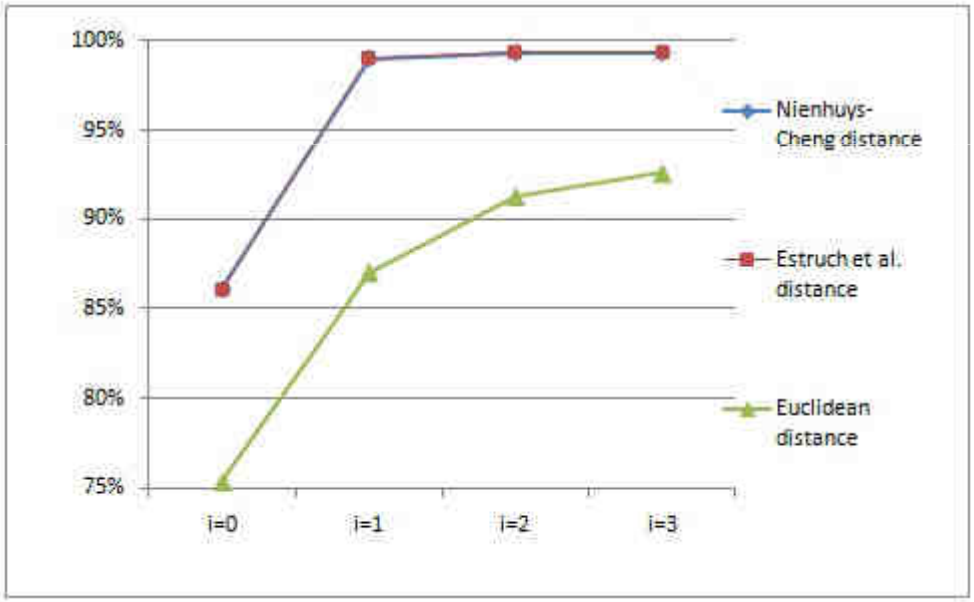
- The behavior of the distances was similar and the differences were not statistically significant.

4. Experiments

SoyBean dataset

- Distances with hierarchy using similarities between attributes.

	i=0 %	i=1 %	i=2 %	i=3 %
d_N	86.1 d_U	99.6 d_U	99.4 d_U	99.4 d_U
d_E	86.1 d_U	99.0 d_U	99.4 d_U	99.4 d_U
d_U	75.3 $d_N d_E$	87.1 $d_N d_E$	91.3 $d_N d_E$	92.6 $d_N d_E$



- The term distances obtained much better results than the Euclidean distance.

4. Experiments

Demospongiae dataset

- It is a hierarchical dataset represented as a tree using terms in Lisp language.
- Each tree has a depth between 5 and 8 levels and its number of leaves varies between 17 and 51.
- In order to do the transformations from Lisp, each line in Lisp was converted into one or several well-formed XML elements.

For instance:

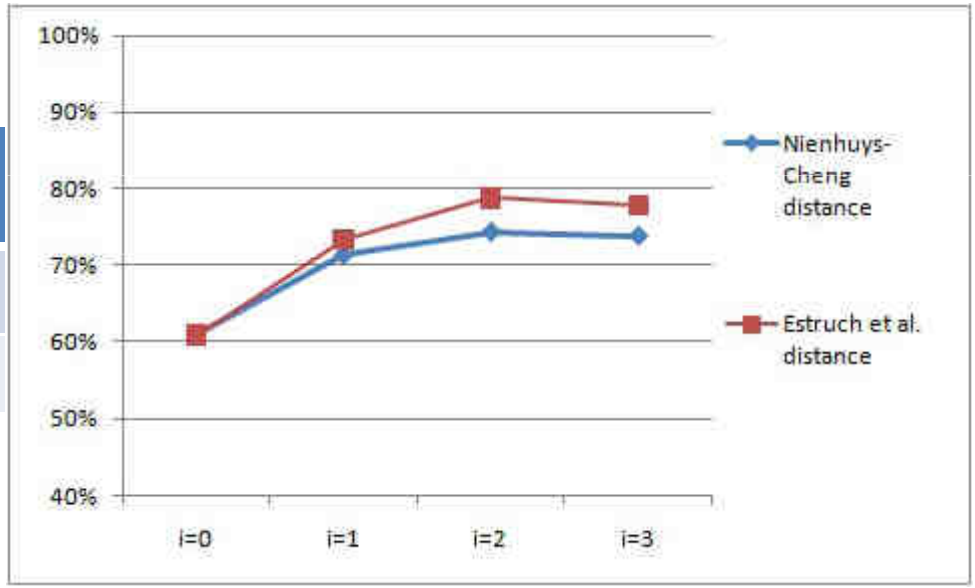
- Each simple feature such as: (BODY-SIZE SMALL)
  was converted into an element <BODY-SIZE>SMALL</BODY-SIZE>

4. Experiments

Demospongiae dataset

- Differences between the term-based distances using semi-structured data.

	i=0 %	i=1 %	i=2 %	i=3 %
d_N	60.9	71.3	74.3	73.8
d_E	60.9	73.3	78.7	77.7



- The Estruch et al's distance showed a better accuracy than Nienhuys-Cheng's distance.

Contents

5. Conclusions

- Regarding the three transformations to adapt different degrees of structures and hierarchical data to be used with term distances:
  - The method for constructing the hierarchical from the attribute names does not always seem to provide good results.
  - Using the similarity between the attributes to construct a dendrogram from which a hierarchy is constructed improves the results of the flat dataset.
  - Different transformations take place when the original dataset already has a hierarchy, here, the relevance of using repetitions or not can be seen.
- We have seen a promising application of term distances to different types of datasets, which suggests that the use of term distances can be broader than it is now.

Thank you
for your attention!