Diterpene Structure Elucidation from NMR

Task:
- given: the C13-NMR (nuclear magnetic resonance) spectra of a Diterpene molecule
- find: the skeleton class it belongs to (e.g. "Labdan")

Approach:
- Multirelational algorithm RIBL [Emde & Wettschereck 96, Horvath, Wrobel, Bohnebeck 01]

Result:
- promises to greatly reduce the very high manual effort required so far
Further application fields

- Bioinformatics, chemistry, biology
- Environmental sciences
- Astronomy
- Engineering

Text Mining:
- E-Mail and Webpage classification
- Information Extraction
- Structuring of Document Collections

Discovery in Geo-Data, Audio-Data, Image-Data, ...

Ecological Prediction of Plant Growth

[Kirsten/Wrobel/Dahmen/98]

Task:
- given a database of plant site recordings
  - site conditions (climate, soil)
  - plants recognized at site
- find characterizations of plant's growth conditions
- an interesting problem: several hundred thousands of such recordings available in Germany alone!

Result: characterizations produced by DM methods no worse than manual characterizations - with much less effort

Instance-based learning (IBL)

Idea:
- store all training instances $x$ along with their prediction value $y$:
  $$x_1, y_1$$
  $$x_2, y_2$$
  ...
- When asked to predict for an unseen instance,
  - find stored instance “closest” to it: the “nearest neighbor”
  - and simply predict its stored class value
- Do not produce an explicit generalization (lazy learning)

K-nearest neighbor learning
- Find $k$ most similar instances and take a (weighted) majority vote

Comparative Accuracies
Remarks on Distance Functions

- In real-valued spaces, simply Euclidean distances
- Easily generalizes to categorical attributes
- Distances have also been defined for more complex representations
  - E.g. first-order/multirelational data: RIBL2 [Horvath, Wrobel, Bohnebeck 01]
  - This was used for the Diterpene application!
- Distance-functions can be generalized to kernel functions and give rise to kernel methods
  - E.g. support vector machines [Vapnik 94, 97]

First-order similarity

Assume we want to use kNN to classify access rights.
Problem: What is the similarity of:

```
May_operate(smith,sys1)  May_operate(miller,sys2)
Employer(smith,co1)      Employer(miller,co2)
Size(co1,large)          Size(co1,medium)
Age(smith,47)            Age(miller,42)
Skill(smith,sk1)         Skill(miller,sk2)
Skill(smith,sk2)         Skill(miller,sk3)
```

Idea [Bisson, KBG system]: compute similarity recursively based on similarity of related objects!

Recursive Similarity Computation

[Bisson/92, Emde/Wettschereck/96]

- Compute distance for nominal and numeric attributes as before
- For “name” attributes (identifiers), collect two sets of facts describing the two given identifiers, respectively
- Find a “suitable” mapping between the different elements of these two sets
- Use distance measure recursively to compute similarity of each mapped pair (recurse until no more identifiers or depth bound reached)
- Take weighted average (unmapped elements have distance 1)

```
Employer(smith,co1)    Employer(miller,co2)
Age(smith,47)          Age(miller,42)
Skill(smith,sk1)       Skill(miller,sk2)
Skill(smith,sk2)       Skill(miller,sk3)
```

RIBL1, RIBL2

[Emde/Wettschereck/96, Bohnebeck/Horvath/Wrobel/98,01]

- Uses recursive similarity metric as defined above
- Constructs cases based on mode declarations and depth limit
- Adds features weight learning (using ReliefF [Kononenko])
- RIBL.2: extended to handle lists and terms using edit distances
Other FO Distance Proposals

Š semantical: DISTILL (Sebag ILP’97)

1. construct d hypotheses
2. \( \theta_i : E \rightarrow \mathbb{N} \) for \( i = 1, \ldots, d \)
   let \( \theta (E) = (\theta_1(E); \ldots; \theta_d(E)) \)
3. \( d(E; F) \): Euclidean distance between \( \theta (E) \) and \( \theta (F) \)

Š syntactical: (Hutchinson, ECML’97, Nienhuys-Cheng, ILP’97)

1. distance between terms
2. distance between point sets

syntactical: (Ramon et.al., 1998,01,02)

Hierarchical Clustering

Goal:

- find sequence of nested partitions \( C^1, C^2, \ldots, C^n \) of \( I \)
- where \( C^i = \{ C_{i,k} \} \) and \( C^n = \{ I \} \)
- and for each \( i \leq m - 1 \)
  \[ \forall C_i \in C^i : \exists C_{i+1} \subseteq C_{i+1} \text{ such that } C_i \subseteq C_{i+1} \]

N. B. Every clustering method can be used recursively to produce hierarchical clusterings
Example

Hierarchical clustering visualized as a dendrogram

\[ \{ (x_1,_,), (x_2,_,), (x_3,_,), (x_4,_,), (x_5,_,) \} \]
\[ \{ (x_1, x_2, ), (x_3, x_4, ), (x_5, ) \} \]
\[ \{ (x_1, x_2, x_3, ), (x_4, ), (x_5, ) \} \]
\[ \{ (x_1, x_2, x_3, x_4, x_5) \} \]

Notes on algorithms

Weighted/unweighted:
- unweighted: each instance counted equally
- weighted: each cluster counted equally, i.e. instances weighted by cluster size

Distances:
- average: average of all pairs (in between min and max!)
- centroid: compute two centroids
  \[ \text{centroid} (C) = \frac{1}{|C|} \sum_{x \in C} x \]
  possible only if \( X = \mathbb{R}^m \)
- Ward: square error minimization

Proximity Dendrograms

- use vertical axis to show proximity at which clusters were merged
- identical to normal dendrogram if distances relabeled to 1, 2, ...

Partitional Clustering

- Hierarchical clustering feasible for small data sets only
- users may want single partition

Task:
- assume \( X \) is \( \mathbb{R}^d \) (d-dimensional metric space)
- distance is Euclidean
- quality criterion square-error
- assume size of clustering \( |C| = k \) given
- clusters may not overlap: partitioning
Searching for good clusterings

- Brute-force?
  - 10 objects in 4 clusters
    - 34,105 possibilities
  - 19 objects in 4 clusters
    - 11,259,660,00 possibilities!

- Must use “heuristic” search

  > K-means algorithm

  - hill-climbing
  - multiple hill-climbing

General k-means type algorithm

- Select k cluster centers
- REPEAT
  - assign each instance to closest center
  - compute centers of the thus-formed clusters
- UNTIL quality does not improve any more

Optional:
- adjust cluster number and restart at REPEAT

RETURN last clustering

Beispiel (1. Iteration)

Beispiel (2. Iteration)
### Properties

- Almost certain convergence - to a local minimum!
  - use non-local “jumps”
  - use multiple hill-climbing searches
- Result depends on initial “seeds”
  - choose systematically: e.g.
    - Start with data centroid, add most distant
- Distorted by outliers
  - preprocess or recognize in method

### Systems

[FORC (Kirsten, Wrobel, 01):]
- “First-order relational clustering”
- k-means type clustering with prototype or medoids

[RDBC (Kirsten, Wrobel 98):]
- “Relational Distance-Based Clustering”
- Hierarchical agglomerative clustering with cut-off selection