Reasoning About Object-Attribute Data: Algorithms and Foundations

Radim Bělohlávek
Dept. Computer Science
Palacký Univeristy, Olomouc

2004

Goals of the Course

- **selected methods**
  - Clustering
  - Rules from data (association rules, database dependencies)
  - Formal concept analysis
  - Classification
- **selected aspects**
  - Algorithms
  - Theory
  - Problem of large data
  - Current research issues
- **further**
  - Implementation issues
  - Applications
  - Software
  - Commercial use

Literature

- Baeza-Yates R., Ribiero-Neto B.: *Modern Information Retrieval.* Addison Wesley, New York, 1999.
- Berka P.: *Dobývání znalostí z databází* (Czech). Academia, Praha, 2003.
- Bock H. H.: *Automatische Klassifikation* (German). Vandenhoeck & Ruprecht, Göttingen, 1974.
- Carpineto C., Romano G.: *Concept Data Analysis : Theory and Applications.* John Wiley & Sons, 2004.
- Everitt, Brian S.: *Cluster Analysis, 4th ed.* Edward Arnold, 2001.
- Hand D. J., Mannila H., Smyth P.: *Principles of Data Mining.* MIT Press, 2001.
- Jain A. K., Dubes R. C.: *Algorithms for Clustering Data*. Prentice Hall, NJ, 1988.
- Lukasová A., Šarmanová J.: *Metody shlukové analýzy* (Czech). SNTL, Prague, 1985.

Some useful free sources

- free **software** for DM: GUHA (`http://www.cas.cz/research/software.shtml`), KDD Package (`http://neuron.tuke.sk/ paralic/KDD`), LISp-Miner `http://lispminer.vse.cz`
- further: Rosetta (Norway), Sipina (France), Weka (New Zealand), Yale (Germany), SumatraTT (Prague), Data Minin Advisor (Univ. Porto)
- several commercial systems
- referential **data**: Machine Learning Repository `http://www1.ics.uci.edu/ mlearn/MLRepository.`, UCI KDD Archive `http://kdd.ics.uci.edu`,
- **books**: Hájek P., Havránek T.: Mechanizing Hypothesis Formation. Springer, 1978 (`http://www.cs.cas.cz/ hajek/guhabook/`)
- Michie D. et al. (Eds.): Machine Learning, Neural and Statistical Classification. Ellis Horwood, 1994 (`http://www.amsta.leeds.ac.uk/ charles/statlog/`)
- Šíma J., Neruda R.: Teoretické otázky neuronových sítí. MatFyzPress, 1996. (`http://www.cs.cas.cz/ sima/kniha.html`)

OBJECT-ATTRIBUTE DATA

Object-Attribute Data

no.	name	age	married	satisfied	language	...
1	Smith	39	Yes	1	E	...
2	Novak	58	No	0.8	E, F	...
3	Braun	23	Yes	0.1	E, S, G	...
4	Kim	36	Yes	0.5	E, F	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
134560	Lewis	31	Yes	0.8	E	...

objects x_i

- $X = \{x_1, \ldots, x_n\}$
- correspond to data items (records, ...)
- correspond to rows in table (agreement: $x_i \approx i$-th row)
- distinct objects = distinct rows

attributes y_j

- $Y = \{y_1, \ldots, y_m\}$
- alternative names: properties (features, ...)

- correspond to columns in table (agreement: $y_j \approx j$-th column)
- $\mathrm{val}(y_j)$ . . . set of values of attribute y_j (or $\mathrm{val}(y_j)$)

table entries $\mathrm{T}(x_i, y_j)$

- $\mathrm{T}(x_i, y_j) \in \mathrm{val}(y_j)$ . . . entry at position (i, j)
- alternative notation: $\mathrm{T}(i, j)$, $y_j(x_i)$, . . . (differs in literature)

Object-Attribute Data

Definition Object-attribute data table (OAD) is a structure $\mathcal{T} = \langle X, Y, T \rangle$ where

- $X \neq \emptyset$ (objects);
- $Y \neq \emptyset$ (attributes), for each $y \in Y$, $\mathrm{val}(y) \neq \emptyset$ (attribute values);
- $T : X \times Y \to \bigcup_{y \in Y} \mathrm{val}(y)$ s.t. $T(x, y) \in \mathrm{val}(y)$ (table entries).

Types of attributes

- **numeric**, i.e. $\mathrm{val}(y) \subseteq \mathbf{R}$ (age, weight, . . .)
- **categoric**, i.e. $\mathrm{val}(y) = \{c_1, \ldots, c_k\}$ (type of car, education, . . .)
- **logical**,
  - **bivalent**, i.e. $\mathrm{val}(y) = \{0, 1\}$ (married, got patent, . . .)
  - **fuzzy**, i.e. $\mathrm{val}(y) \subseteq [0, 1]$ (expensive, large, . . .)
- others (several ontologies possible)

Analysis of Object-Attribute Data

part of Data Mining, i.e. extraction of potentially useful information from data

characteristics of Data Mining

- alternative names: Knowledge Discovery in Databases, Bussiness Intelligence, (Exploratory) Data Analysis
- history: 1990s, but several methods developed earlier
- conferences: ACM KDD, IEEE DM, PAKDD, PKDD
- journals: Data Mining and Knowledge Discovery (Kluwer), IEEE Trans. Data and Knowledge Engineering (IEEE), other computer science journals
- applications: USA, Europe (both lare and small industries)
- ČR: časopis Data Mining Magazine (Adastra), společnosti zabývající se data mining, projekty u větších firem

Magagerial look on Data Mining

- problem specification
- collecting data
- selection of methods
- data preprocessing
- data mining
- interpretation of results

Technological look on Data Mining

- original data $\Rightarrow$ (selection)
- selected data $\Rightarrow$ (preprocessing)
- preprocessed data $\Rightarrow$ (transformation)
- transformed data $\Rightarrow$ (data mining)
- extracted information $\Rightarrow$ (interpretation)

- knowledge

What do we want to know?

- fundamental question
- user usually does not know
- expert needs to assist

Basic methods of analysis

- clasical statitistical (regression, testing of hypotheses, analysis of variance, . . .)
- classification (prediction about membership to classes; decision trees, neural networks, Bayesian classification, . . .)
- patterns in data (descriptive; patterns: clusters, rules from data, . . .)
- . . .

FORMAL CONCEPT ANALYSIS

see slides to FCA

ASSOCIATION RULES

see slides to FCA

CLUSTERING

Clustering

basic facts:

- main aim of clustering: **finding (interesting) groups/clusters in data**
- people do in everyday life; one cannot survive without clustering and classification
- data = collection of objects; objects described by their attributes
- **cluster** = collection of objects which are pairwise similar (and which are dissimilar to objects outside the cluster); vague definition but ...
- important aspects: clusters should be interesting groups (understandable); computational tractability; applicability to large data (but clustering)
- **similarity/distance** of objects: crucial role, usually computed from data table (metric, ultrametric)
- basic types: hierarchic and non-hierarchic clustering

further aspects

- relatively old (1960s-1970s), gained interest in connection with data mining

- significant area which contributed to development of clustering: clustering of biological species (Numerical taxonomy, Mathematical taxonomy)
- availability of data for clustering: (1) objects with teir descriptions available (stored in a data table); (2) objects appear one at a time (incremental methods)
- sometimes two different objectives are emphasized:
  - **cluster analysis** = to see whether data is composed of natural subclusters and what they are (the user might have no clue in advance)
  - **segmentation** = objects need to be partitioned for some practical purposes into some number of clusters (example: segmentation of customers in marketing; shirt manufacturer, clusters of customers: one shirt size for each cluster)
- "was clustering useful?" is a difficult-to-answer question; application dependent, the user judges (as with most of exploratory data analysis techniques)

typical process

- selection of method (what types of clusters do we look for?)

- selection of method parameters (measure of similarity/distance of objects, etc.)
- clustering data (run clustering algorithm on data)
- evaluation of clusters (are clusters interesting/useful?); but if clustering is used as a preprocessing step, evaluation may be omitted
- further processing (e.g. if used as a preprocessing step)

Basic types of clusterings

First: several meanings of "clustering"

- clustering as a method
- clustering as a particular algorithm
- clustering as the collection of clusters in data

Central question: what are THE right clusters?

Similarity/distance measures for clusterings

input data: $\mathcal{T} = \langle X, Y, T \rangle$
(data table, attributes numeric, categorical, logical)
$T(x, y)$ ... value of attribute y on object x

main aim: assign any two objects x_1 and x_2 a quantity (usually a real number) describing their similarity or distance

similarity vs. dissimilarity (distance): the larger the similarity, the smaller the dissimilarity (and vice versa); most often: $S = 1 - D$ (similarity S, dissimilarity D)

(dis)similarities assigned to
– pairs of objects, e.g. $D(x_1, x_2) = 0.7$
– pairs of groups of objects, e.g. $D(\{x_1, x_2\}, \{x_1, x_3, x_5\}) = 0.9$

Dissimilarity, metric and ultrametric

represent distance

Def. A **dissimilarity** on a set X is a function $d : X \times X \to [0, \infty)$ satisfying:

– $d(x, x) = 0$,
– $d(x_1, x_2) = d(x_2, x_1)$ (symmetry).

for each $x, x_1, x_2 \in X$ (sometimes additional conditions, e.g. $d(x_1, x_2) \leq 1$).

Def. A **metric** on a set X is a function $d : X \times X \to [0, \infty)$ satisfying:

– $d(x_1, x_2) = 0$ if and only if $x_1 = x_2$,
– $d(x_1, x_2) = d(x_2, x_1)$ (symmetry),
– $d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)$ (triangle inequality),

for each $x_1, x_2, x_3 \in X$.

A **pseudometric**: first condition replaced by $d(x, x) = 0$.

Def. An **ultrametric** on a set X is a function $d : X \times X \to [0, \infty)$ satisfying:

– $d(x_1, x_2) = 0$ if and only if $x_1 = x_2$,
– $d(x_1, x_2) = d(x_2, x_1)$ (symmetry),
– $d(x_1, x_3) \leq \max(d(x_1, x_2), d(x_2, x_3))$ (ultrametric inequality),

for each $x_1, x_2, x_3 \in X$.

Notes Metric well-known (calculus). Ultrametric not. Carefully! Ultrametric has some unusual properties, e.g.
– for any three objects x_1, x_2, x_3, at least two of them have the same distance;
– define $B(x, a) = \{x' \in X;\ u(x, x') \leq a\}$ (ball with center x and diameter a); then for any any two balls B_1, B_2, there are only two possibilities: (1) one of them contains the other one ($B_1 \subseteq B_2$ or $B_2 \subseteq B_1$), or (2) are disjoint $B_1 \cap B_2 = \emptyset$!

Similarity and dissimilarity of objects

Dissimilarity measures – numerical attributes

- **Euclidean metric**

$$d(x_1, x_2) = [\sum_{y \in Y} (T(x_1, y) - T(x_2, y))^2]^{\frac{1}{2}}$$

- **Manhattan (city-block) metric**

$$d(x_1, x_2) = \sum_{y \in Y} |T(x_1, y) - T(x_2, y)|$$

the name: streets and avenues on Manhattan (NY) are perpendicular to each other; If x_1 and x_2 are two crossing points (of streets on Manhattan) with coordinates $\langle y_{11}, y_{12} \rangle$ and $\langle y_{21}, y_{22} \rangle$ (that is, y_{11} and y_{12} are the x- and y-coordinates of x_1, y_{21} and y_{22} are the x- and y-coordinates of x_2) then the (walking) distance between x_1 and x_2 is just the Manhattan metric $d(x_1, x_2) = |y_{11} - y_{21}| + |y_{12} - y_{22}|$.

- L_∞ **metric**

$$d(x_1, x_2) = \max_{y \in Y} |T(x_1, y) - T(x_2, y)|$$

- L_λ **metric**: generalization of Euclidean ($\lambda = 2$), Manhattan ($\lambda = 1$), L_∞ ($\lambda \to \infty$)

$$d_\lambda(x_1, x_2) = [\sum_{y \in Y} |T(x_1, y) - T(x_2, y)|^\lambda]^{\frac{1}{\lambda}}$$

  - sometimes used: $\Delta_\lambda(x_1, x_2) = d_\lambda(x_1, x_2)/|Y|^{\frac{1}{\lambda}}$
  - then $\Delta_1(x_1, x_2) \leq \Delta_2(x_1, x_2) \leq \Delta_1(x_1, x_2) \leq \cdots$

- **weights**: we might have real coefficients $w_y \in \mathbf{R}$ assigned to attributes $y \in Y$ which express importance of attributes (higher weight means more importance, practical meaning: small difference in higly important attribute can increase distance more than a larger difference in less important attribute); distance measures modified by weights as

$$d_{w,\lambda}(x_1, x_2) = [\sum_{y \in Y} w_y \cdot |T(x_1, y) - T(x_2, y)|^\lambda]^{\frac{1}{\lambda}}$$

- statistically based distance measures (eliminate the influence of correlated attributes)
  - **Mahalanobis distance**
  - **correlation coefficient**

- further dissimilarity measures:
  - **non-metric coefficient** (Lance, Williams) for $T(x, y) \geq 0$

$$d(x_1, x_2) = \frac{\sum_{y \in Y} |T(x_1, y) - T(x_2, y)|}{\sum_{y \in Y} T(x_1, y) + T(x_2, y)}$$

  - **Canberra metric** for $T(x, y) \geq 0$

$$d(x_1, x_2) = \sum_{y \in Y} \frac{|T(x_1, y) - T(x_2, y)|}{T(x_1, y) + T(x_2, y)}$$

Similarity measures – logical (binary) attributes

for each $x \in X, y \in Y$: $T(x, y) \in \{0, 1\}$

contingency table: for a data table $\mathcal{T} = \langle \mathcal{X}, \mathcal{Y}, \mathcal{T} \rangle$ with binary attributes, a contingency table for objects x_1, x_2 is a table

	$T(x_2, y) = 1$	$T(x_2, y) = 0$	$\sum$
$T(x_1, y) = 1$	a_{11}	a_{10}	$a_{1\_}$
$T(x_1, y) = 0$	a_{01}	a_{00}	$a_{0\_}$
$\sum$	$a_{\_1}$	$a_{\_0}$	$\lvert Y \rvert$

$a_{kl} = |\{y;\, T(x_1, y) = k \text{ and } T(x_2, y) = l\}|$, i.e.

a_{00} . . . #attributes for which x_1 has value 0 and x_2 has value 0

. . .

a_{11} . . . #attributes for which x_1 has value 1 and x_2 has value 1

$a_{\_1}$ . . . #attribute for which x_2 has value 1, etc.

shortly

	1	0
1	a_{11}	a_{10}
0	a_{01}	a_{00}

or the like

more often: **similarity** rather than dissimilarity measures are considered for binary data

a family of similarity measures of the form

$$s(x_1, x_2) = S(a_{00}, a_{01}, a_{10}, a_{11})$$

where S comes from intuition/expert opinion

common requirements: $S(a_{00}, a_{01}, a_{10}, a_{11})$ is
– nondecreasing in a_{00} and a_{11}
– nonincreasing in a_{01} and a_{10}
– symmetric in a_{01} and a_{10}: $S(a_{00}, b, c, a_{11}) = S(a_{00}, c, b, a_{11})$

Examples of similarity measures

- **simple matching coefficient**

$$s(x_1, x_2) = \frac{a_{11} + a_{00}}{a_{00} + a_{01} + a_{10} + a_{11}}$$

  – $1 - s(x_1, x_2)$ is the (normalized) Hamming distance of x_1 and x_2

- **Jaccard coefficient**

$$s(x_1, x_2) = \frac{a_{11}}{a_{01} + a_{10} + a_{11}}$$

– for situations where non-presence of any attribute should not influence similarity

- **Dice coefficient**

$$s(x_1, x_2) = \frac{2a_{11}}{a_{01} + a_{10} + 2a_{11}}$$

  – extends the argument fo Jaccard coef.: presence of attribute in both objects is twice as important as its presence in only one object
  – example of weighted coefficient

- more generally one could take weights $w^u_{00}, \ldots, w^u_{11}$ and $w^l_{00}, \ldots, w^l_{11}$ and have

$$s_w(x_1, x_2) = \frac{w^u_{00}a_{00} + w^u_{01}a_{01} + w^u_{10}a_{10} + w^u_{11}a_{11}}{w^l_{00}a_{00} + w^l_{01}a_{01} + w^l_{10}a_{10} + w^l_{11}a_{11}},$$

- **weights for attributes** w_y $(y \in Y)$ and consider e.g.

$$d(x_1, x_2) = \frac{\sum_{y \in Y} w_y \cdot |T(x_1, y) - T(x_2, y)|}{\sum_{y \in Y} w_y}$$

  which is a normalized weighted Hamming distance; the corresponding similarity is $s_w = 1 - d_w$

Example: data table with binary attributes, contingency table, . . .

	fund	type	1	2	3	4	5	6	7	8	9
1	CPI Penezniho trhu	money	1	0	0	0	1	0	0	1	0
2	CSOB Akciovy	stock	1	0	0	0	0	1	0	0	1
3	CSOB Bond mix	bond	0	1	0	1	0	0	0	1	0
4	IKS Dluhopisovy	bond	0	1	0	1	0	0	1	0	0
5	IKS Globalni	mixed	0	1	0	0	1	0	0	1	0
6	IKS Penezni trh	money	1	0	0	0	1	0	0	1	0
7	ISCS Sporoinvest	money	1	0	0	0	1	0	0	1	0
8	ISCS Sporotrend	stock	0	0	1	0	0	1	0	0	1
9	ISCS Trendbond	bond	0	0	1	1	0	0	1	0	0
10	ISCS Vynosovy	mixed	0	0	1	0	1	0	0	1	0

attributes: 1 - rating for 1 week $<= 0,5$, 2 - rating for 1 week $> 0,5$ and $<= 1$, 3 - rating for 1 week > 1, 4 - rating for 26 weeks $<= 0,5$, 5 - rating for 26 weeks $> 0,5$ and $<= 4$, 6 - rating for 26 weeks > 4, 7 - rating for 52 weeks $<= 0,5$, 8 - rating for 56 weeks $> 0,5$ and $<= 10$, 9 - rating for 56 weeks > 10

contingency table for $x_1 = 3$, $x_2 = 4$:

	1	0
1	2	1
0	1	5

Execrice: compute various similarity measures for x_1, x_2; take one similarity measure and compute the similarity matrix ($X \times X$ matrix filled with $s(x_i, x_j)$)

Similarity measures − categoric attributes

several possibilities, e.g. (the most simple)

$s(x_1, x_2) = S(a, b)$ where

a . . . #attributes for which x_1 and x_2 have the same value
b . . . #attributes for which x_1 and x_2 have different values

Similarity and dissimilarity of groups of objects

we assume $A, B \subseteq X$ (sets of objects)

we are interested in $s(A, B)$ (similarity) and $d(A, B)$ (dissimilarity)

basic intuitive requirements:
$s(A, B) = s(B, A) \geq 0$
$d(A, B) = d(B, A) \geq 0$

Measures using (dis)similarity on objects

assume s is a similarity on objects (see above), and define similarity on sets of objects (defined also s):

- $s(A, B) = \min\{s(x_1, x_2);\ x_1 \in A, x_2 \in B\}$
- $s(A, B) = \frac{1}{|A| \cdot |B|} \sum_{x_1 \in A} \sum_{x_2 \in B} s(x_1, x_2)$
- $s(A, B) = \max\{s(x_1, x_2);\ x_1 \in A, x_2 \in B\}$

and similarly for dissimilarity measures

Measures using numeric attributes

for a cluster A, one can take the center $x_A = \frac{1}{|A|} \sum_{x \in A} x$

and then

$$s(A, B) = S(x_A, x_B)$$

with S being a suitable similarity function

Measures using categoric attributes

for $A \subseteq X$, $y \in Y$, $c \in \mathrm{val}(y)$, put

$$p^A_{y,c} = \frac{|\{x \in A;\, T(x, y) = c\}|}{|A|}$$

. . . frequency of objects from A havin value of y equal c

then (e.g. Sokal+Sneath):

$$s(A, B) = \frac{1}{|Y|} \sum_{y \in Y} \sum_{c \in \mathrm{val}(y)} p^A_{y,c} \cdot p^B_{y,c}$$

observe that

$$\sum_{c \in \mathrm{val}(y)} p^A_{y,c} \cdot p^B_{y,c}$$

can be seen as the probability that selecting randomly $x_1 \in A$ and $x_2 \in B$, x_1 and x_2 will agree on attribute y

$\Rightarrow$ $s(A, B)$ can be seen as the probability that selecting randomly $x_1 \in A$ and $x_2 \in B$, and selecting an attribute $y \in Y$, x_1 and x_2 will agree on y

Basic types of clustering – an overview

several taxonomies of clusterings possible, based e.g. on

- types of clusters: **crisp** (clusters are ordinary sets) vs. **fuzzy** (clusters are fuzzy sets)
- relationship between clusters: **non-overlapping** (different clusters have no objects in common) vs. **overlapping** (different clusters may have onjects in common)
- **hierarchic** (clusters may be subgouprs of other clusters; namely, those which are more general) vs. **non-hierachic** (the other case)

and we may have hierarchic clusters which are fuzzy sets, hierarchic clusters which are crisp sets, etc.

In the following we present **selected clustering types**.

Preliminaries for clustering (recalling well-known facts)

$R \subseteq X \times X$ . . . a **binary relation** on X

R is called a
–**tolerance** if it is reflexive and symmetric
equivalence if it is reflexive, symmetric, and transitive

–**class of** R induced by $x \in X$: a set $[x]_R = \{x' \in X; \langle x, x' \rangle \in R\}$

a system Π of subsets of X, i.e. $\Pi = \{C_i; i \in I, C_i \subseteq X\}$ is called a
–**covering** of X if (1) each $C_i \in \Pi$ is nonempty; (2) each $x \in X$ belongs to some $C_i \in \Pi$
–**partition** of X if (1) each $C_i \in \Pi$ is nonempty; (2) each $x \in X$ belongs to some $C_i \in \Pi$; any two distinct $C_i, C_j \in \Pi$ are disjoint, i.e. $C_i \cap C_j = \emptyset$

there is a **one-to-one relationship between equivalence relations and partitions** on X:
–from equivalence R to partition Π_R: $\Pi_R = \{[x]_R; x \in X\}$ (partition consists of classes of R)
–from partition Π to equivalence R_Π: $\langle x, x' \rangle \in R_\Pi$ iff there is $C_i \in \Pi$ such that $x, x' \in C_i$

Hierarchic clustering

- useful if we want nested/hierachically ordered clusters
- result is a tree (hierarchy; or an indexed tree, so-called **dendrogram**) with nodes labeled by clusters of objects (subsetes of X)
- **root** is labeled by X (largest cluster), **leaves** labeled by $\{x\}$ (smallest clusters, singletons for each $x \in X$)
- for clusters $A, B \subseteq X$: $A \subseteq B$ iff the node labeled by A is a descendant of a node labeled by B
- basic **algorithms** for obtaining dendrograms:
  - **agglomerative**: starts with singleton clusters $\{x\}$, in each step selects two most similar clusters and joins them, repeats until the largest cluster X is obtained
  - **divisive** (less used, computatinoally more demanding): start with the largest cluster X which is split into smaller clusters, division is repeated until singleton clusters are obtained; two approaches to division of clusters: **monothetic** (one attribute is used to determine division) and **polythetic** (all attributes are used)
- well-elaborated theoretical foundations (ultrametrics,)

Algorithms for hierarchic clustering: agglomerative

INPUT: data table $\mathcal{T} = \langle X, Y, T \rangle$, a similarity measure $S : X \times X \to [0, \infty)$

based on S, select an extension of S to a similarity on groups of objects, i.e. a function assigning to any $A, B \subseteq X$ a number $S(A, B) \in [0, \infty)$ (see later)

AGGLOMERATIVE ALGORITHM

1. (initialization) $\Pi_0 = \{\{x\}; x \in X\}$ (partition with singleton classes); $h_0 := 0$; $t := 1$;

2. (closest clusters) take distinct $C_1, C_2 \in \Pi_{t-1}$ for which

$$S(C_1, C_2) = \max_{C, C' \in \Pi_{t-1}, C \neq C'} S(C, C');$$

3. (merging clusters) $\Pi_t := \Pi_{t-1} - \{C_1, C_2\} \cup \{C_1 \cup C_2\}$; $h_t := 1 - S(C_1, C_2)$.

4. (termination test) if Π_t contains more than one cluster, put $t := t + 1$ and go to step 2; otherwise stop.

OUTPUT: usually considered as $\mathcal{C} = \{C; C \in \Pi_t$ for some $t\}$ (collection of **resulting clusters**) or alternatively $\mathcal{C} = \{\langle C, h_t \rangle ; t = \min\{t'; C \in \Pi_{t'}\}\}$.

Now:
ordering $\mathcal{C}$ by set inclusion gives a partial order
the Hasse diagram of the partial order is a tree (we label nodes by $C \in \mathcal{C}$ or by $\langle C, h_t \rangle \in \mathcal{C}$)

Remarks to agglomerative algorithm

(1) We get so-called **single-linkage** version for

$$S(A, B) = \max\{S(x_1, x_2);\ x_1 \in A, x_2 \in B\},$$

and so-called **complete linkage** version for

$$S(A, B) = \min\{S(x_1, x_2);\ x_1 \in A, x_2 \in B\}.$$

Single-linkage and complete linkage are two boundary cases. In addition to these, there are various "average-linkage" versions (average similarity of two clusters instead of maximal or minimal).

(2) Can be equivalently formulated using dissimilarity D instead of S (replace S by D and interchange $\min$ and $\max$).

Agglomerative algorithm: example

data table:

	fund	type	1	2	3	4	5	6	7	8	9
1	CPI Penezniho trhu	money	1	0	0	0	1	0	0	1	0
2	CSOB Akciovy	stock	1	0	0	0	0	1	0	0	1
3	CSOB Bond mix	bond	0	1	0	1	0	0	0	1	0
4	IKS Dluhopisovy	bond	0	1	0	1	0	0	1	0	0
5	IKS Globalni	mixed	0	1	0	0	1	0	0	1	0
6	IKS Penezni trh	money	1	0	0	0	1	0	0	1	0
7	ISCS Sporoinvest	money	1	0	0	0	1	0	0	1	0
8	ISCS Sporotrend	stock	0	0	1	0	0	1	0	0	1
9	ISCS Trendbond	bond	0	0	1	1	0	0	1	0	0
10	ISCS Vynosovy	mixed	0	0	1	0	1	0	0	1	0

attributes: 1 - rating for 1 week $<= 0,5$, 2 - rating for 1 week $> 0,5$ and $<= 1$, 3 - rating for 1 week > 1, 4 - rating for 26 weeks $<= 0,5$, 5 - rating for 26 weeks $> 0,5$ and $<= 4$, 6 - rating for 26 weeks > 4, 7 - rating for 52 weeks $<= 0,5$, 8 - rating for 56 weeks $> 0,5$ and $<= 10$, 9 - rating for 56 weeks > 10

we use dissimilarity: (non-normalized) weighted Hamming distance

$$D_w(x_i, x_j) = \sum_{y \in Y} w_y \cdot |T(x_i, y) - I(x_j, y)|, \tag{1}$$

with $w_1, \ldots, w_9$ (weights for attributes 1–9) equal to 0.4, 0.3, 0.3, 0.3, 0.5, 0.2, 0.2, 0.6, 0.2, respectively

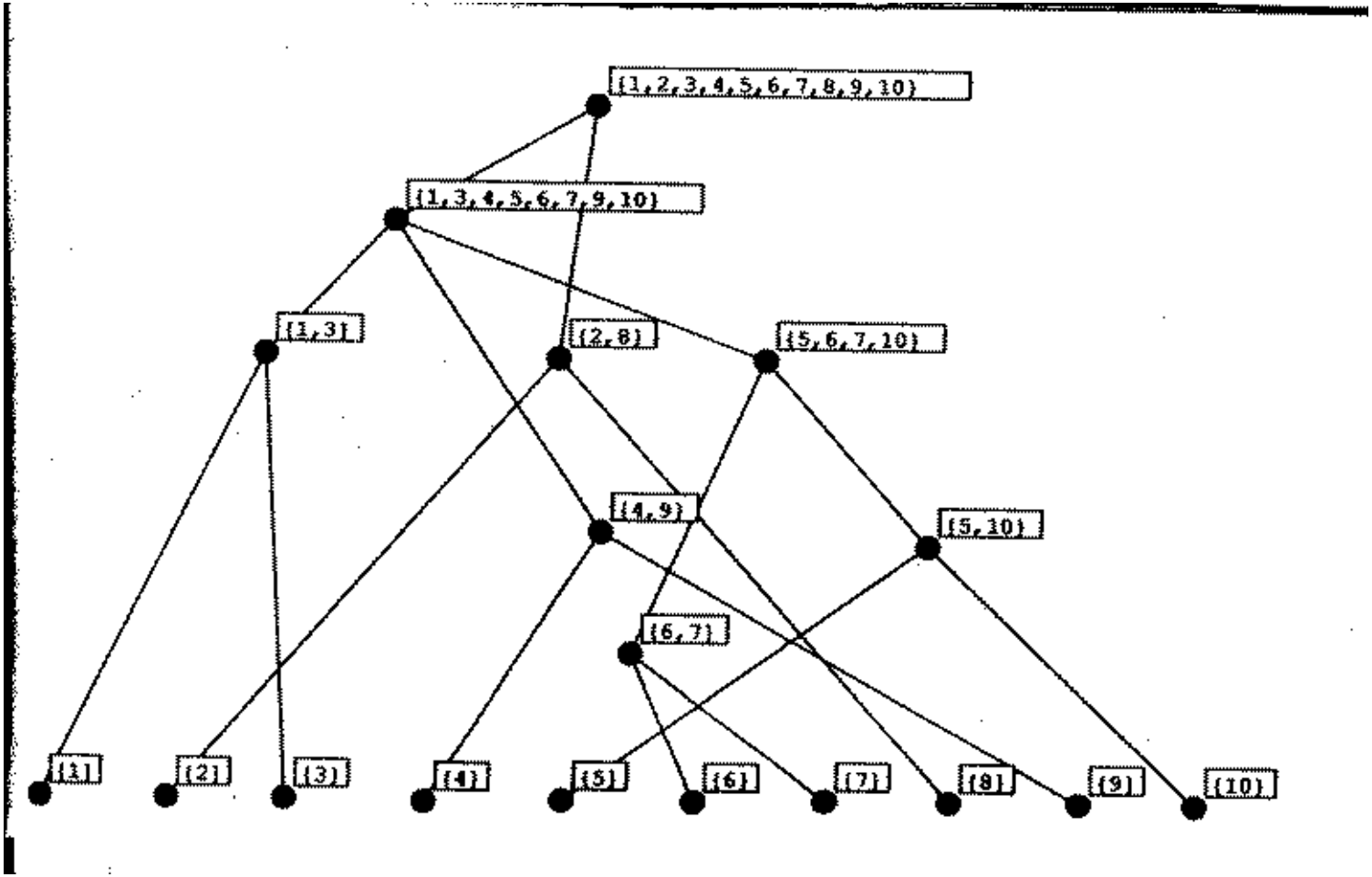
dissimilarity coefficients:

	1	2	3	4	5	6	7	8	9	10
1	0	1.4	0.7	1.5	1.5	0.8	0.8	2.1	1.5	1.5
2	1.4	0	2.1	1.7	2.1	1.4	1.4	0.7	1.7	2.1
3	0.7	2.1	0	0.8	0.8	1.5	1.5	2	1.4	1.4
4	1.5	1.7	0.8	0	1.6	2.3	2.3	1.6	0.6	2.2
5	1.5	2.1	0.8	1.6	0	0.7	0.7	2	2.2	0.6
6	0.8	1.4	1.5	2.3	0.7	0	0	2.1	2.3	0.7
7	0.8	1.4	1.5	2.3	0.7	0	0	2.1	2.3	0.7
8	2.1	0.7	2	1.6	2	2.1	2.1	0	1	1.4
9	1.5	1.7	1.4	0.6	2.2	2.3	2.3	1	0	1.6
10	1.5	2.1	1.4	2.2	0.6	0.7	0.7	1.4	1.6	0

by agglomerative hierarchical clustering (single-linkage), we get a collection of nested partitions $\Pi_1, \ldots, \Pi_5$ and the corresponding equivalence relations $\equiv_1$, $\ldots$, $\equiv_5$:

$$\begin{aligned}
\Pi_1 &= \{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{10\}\},\\
\Pi_2 &= \{\{1\},\{2\},\{3\},\{4\},\{5\},\{8\},\{9\},\{10\},\{6,7\}\},\\
\Pi_3 &= \{\{1\},\{2\},\{3\},\{8\},\{4,9\},\{5,10\},\{6,7\}\},\\
\Pi_4 &= \{\{4,9\},\{1,3\},\{2,8\},\{5,6,7,10\}\},\\
\Pi_5 &= \{\{2,8\},\{1,3,4,5,6,7,9,10\}\},\\
\Pi_6 &= \{\{1,2,3,4,5,6,7,8,9,10\}\}.
\end{aligned}$$

corresponding dendrogram:



Resolving ties

tie = situation when there are more then one candidate pairs of clusters to merge (step 2. of agglomerative algorithm)

So suppose in step 2 that we have

$$s = \max_{C,C' \in \Pi_{t-1}, C \neq C'} S(C, C') = S(C_{l_1}, C_{r_1}) = \cdots = S(C_{l_k}, C_{r_k})$$

Common ways to resolve ties:

1. Each C_{l_i} is merged with at most (and if possible, with exactly) one cluster C for which $S(C_{l_i}, C) = s$. If there are more possibilities to choose merges, select the merges at random.

2. Each C_{l_i} is merged with each C_u with $S(C_{l_i}, C_u) = s$, then each of these C_u's is merged with again with all C_v's with $S(C_u, C_v) = s$, etc. In other words, the newy formed clusters are the components of a graph where clusters C and D are linked by an edge iff $S(C, D) \geq s$. (note: contrary to the above approach, this is one yields a unique solution to ties; the resulting dendrogram need not be binary)

Example: Clusters are $C_1, \ldots, C_6$, $S(C_1, C_2) = S(C_3, C_4) = S(C_3, C_5) = s$. Then method 1 yields clusters $C_1 \cup C_2$, $C_3 \cup C_4$, C_5, C_6 or $C_1 \cup C_2$, $C_3 \cup C_5$, C_4, C_6, method 2 yields clusters $C_1 \cup C_2$, $C_3 \cup C_4 \cup C_5$, C_6.

Single-linkage is more robust w.r.t. the way ties are resolved. We will not go into this issue in more detail. Assuming ties makes theoretical consideration more difficult. In the following we assume that no ties occur.

Foundations of hierarchical clustering: dendrograms and ultrametrics

Def. A **hierarchy** on a set X (objects) is a system $\mathcal{H} \subseteq 2^X$ of subsets of X such that for any two distinct $A, B \in \mathcal{H}$ we have $A \cap B = \emptyset$ od $A \subseteq B$ or $B \subseteq A$. $A \in \mathcal{H}$ are called classes of $\mathcal{H}$.

Rem. (1) For a hierarchy $\mathcal{H}$, the pair $\langle \mathcal{H}, \subseteq \rangle$ (i.e. hierarchy equipped with subsethood) is a **tree order**, i.e. an order for which its Hasse diagram is a tree.
(2) A hierarchy $\mathcal{H}$ is called **binary** if each $A \in \mathcal{H}$ has either no or just two successors (in the corresponding tree); **complete** if $X \in \mathcal{H}$ and $\{x\} \in \mathcal{H}$ for each $x \in X$ (we will mostly aaume that we deal with complete hierarchies).
(3) The output $\mathcal{C} = \{C;\, C \in \Pi_t$ for some $t\}$ of the agglomerative algorithm is a complete hierarchy.

Def. A **dendrogram** is a pair $\langle \mathcal{H}, h \rangle$ where $\mathcal{H}$ is a hierarchy and $h : \mathcal{H} \to [0, \infty)$ is a function (index function) satisfying: (a) for each $A, B \in \mathcal{H}$, $A \subset B$ implies $h(A) < h(B)$; (b) $h(A) = 0$ iff for each $x_1, x_2 \in A$, x_1 and x_2 have the same attribute values.

Rem. (1) visualization of $\langle \mathcal{H}, h \rangle$: draw a tree corresponding to $\mathcal{H}$, along with a vertical axis for values of h such that each $A \in \mathcal{H}$ is drawn at the level $h(A)$ on the vertical axis.

(2) dendrogram from the agglomerative algorithm: The hierarchy is $\mathcal{H} := \mathcal{C} = \{C;\, C \in \Pi_t \text{ for some } t\}$; h is given by

$$h(A) = h_t \text{ where } t = \min\{t';\, C \in \Pi_{t'}\}.$$

(3) Each hierarchy $\mathcal{H}$ cen be made into a dendrogram $\langle \mathcal{H}, h \rangle$. Indeed, take any dissimilarity d on X. Then both $h(A) := \max_{x_1,x_2 \in A} d(x_1, x_2)$ and $h(A) := \sum_{x_1,x_2 \in A} d(x_1, x_2)$ are index functions for $\mathcal{H}$.

(4) $A \in \mathcal{H}$ is a **class of level** a ($a \geq 0$) if $h(A) \leq a$ and there is no $B \supset A$ with $h(A) \leq a$.

(5) If $\langle \mathcal{H}, h \rangle$ is a dendrogram with $\mathcal{H}$ being a complete hierarchy, then for each $a \geq 0$, the system

$$\Pi(a) = \{A \in \mathcal{H};\, A \text{ is a class of level } a\}$$

is a partition, so-called **partition of level** a. Geometric interpretation: In the tree corresponding to $\langle \mathcal{H}, h \rangle$, draw a horizontal line at the level a; elements of $\Pi(a)$ are just the classes right below the line.

Theorem (**induced ultrametric**) For a dendrogram $D = \langle \mathcal{H}, h \rangle$ with a complete hierarchy $\mathcal{H}$, put

$$u_D(x_1, x_2) = \min\{h(A);\ A \in \mathcal{H}, x_1, x_2 \in A\}$$

for any $x_1, x_2 \in X$. Then u_D is an ultrametric.

Rem. $u_D(x_1, x_2)$ is the level of the least class containing both x_1 and x_2 (x_1 and x_2 meet in A).

Proof (of Theorem) $u_D(x, x) = 0$ is true since $\{x\} \in \mathcal{H}$ and $h(\{x\}) = 0$. $u_D(x_1, x_2) = u_D(x_2, x_1)$ is obvious.
$u_D(x_1, x_3) \leq \max(u_D(x_1, x_2), u_D(x_2, x_3))$: Let $u_D(x_1, x_2) = h(A)$ and $u_D(x_2, x_3) = h(B)$. Since $x_2 \in A \cap B$, we have $A \cap B \neq \emptyset$ and so, since $\mathcal{H}$ is a hierarchy, $A \subseteq B$ or $B \subseteq A$. Suppose $A \subseteq B$ (for $B \subseteq A$ we can proceed analogously). Then $x_1, x_2, x_3 \in B$ and so the least class C containing both x_1 and x_3 is contained in B, whence $u_D(x_1, x_3) = h(C) \leq h(B) = \max(h(A), h(B)) = \max(u_D(x_1, x_2), u_D(x_2, x_3))$. □

From ultrametric to dendrogram: Let u be an ultrametric on X. Recall: For $x \in X$, $a \geq 0$, $B_u(x, a) = \{x'; u(x, x') \leq a\}$ (a ball with center x and diameter a). Any two balls are either disjoint or one contains the other.

For $A \subseteq X$, the u-diameter of A is defined by

$$u(A) = \max\{u(x_1, x_2); x_1, x_2 \in A\}.$$

Furthermore: From ultrametric inequality we easily see that: (1) The diameter of a ball is equal to its radius, i.e. $u(B(x, a)) = a$; (2) any point of a ball is its center, i.e. for each $x' \in B(x, a)$ we have $B(x, a) = B(x', a)$.

For an ultrametric u, put

$$\mathcal{H}_u = \{B(x, a); x \in X, a \geq 0\}$$

(system of all u-balls) and for $A \in \mathcal{H}_u$,

$$h_u(A) = u(A).$$

Theorem (induced dendrogram) For an ultrametric u, $D_u = \langle \mathcal{H}_u, h_u \rangle$ is a dendrogram with a complete hierarchy $\mathcal{H}_u$.

Proof First, $\mathcal{H}_u$ is a complete hierarchy. $\mathcal{H}_u$ is a hierarchy because of the properties of ultrametric balls. $\mathcal{H}_u$ is complete since $X = B(x, \infty)$ (for any $x \in X$), and $\{x\} = B(x, 0)$ for each $x \in X$. Second, h_u is an index for $\mathcal{H}_u$: directly by definition. □

Lemma For a dendrogram $D = \langle \mathcal{H}, h \rangle$ and $A \in \mathcal{H}$ we have

$$A = B_{u_D}(x, h(A)) \quad \text{for any } x \in A, \tag{2}$$

$$h(A) = u_D(A). \tag{3}$$

Proof (2): Take any $x' \in X$ ane let C be the least $C \in \mathcal{H}$ where x and x' meet. If $x' \in A$ then C is included in A and so $u_D(x, x') = h(C) \leq h(A)$, i.e. $x' \in B(x, h(A))$. Conversely, if $x' \in B(x, h(A))$ then $u_D(x, x') = h(C) \leq h(A)$. Now, since $x \in A \cap C$ and $h(C) \leq h(A)$, we must have $C \subseteq A$ (since D is a dendrogram; namely, $A \cap C \neq \emptyset$ and $A \subset C$ cannot be the case because of $h(C) \leq h(A)$). But then $x' \in C$ yields $x' \in A$. We proved that A is an u_D-ball.

(3): Follows from (2) and the fact that the diameter and radius of a ball are the same. □

Theorem (dendrograms vs. ultrametrics) Let $D = \langle \mathcal{H}, h \rangle$ be a dendrogram with a complete hierarchy, u be an ultrametric (both on a set X). Then
(1) u_D is an ultrametric;
(2) D_u is a dendrogram with a complete hierarchy;
(3) $D = D_{u_D}$ and $u = u_{D_u}$.

Rem.: (3) says that the mappings $D \mapsto u_D$ and $u \mapsto D_u$ are mutually inverse bijective mappings between the set of all dendrograms with complete hierarchies on X and the set of all ultrametrics on X. Loosely speaking, dendrograms with complete hierarchies and ultrametrics describe the same phenomenon.

Proof (of Theorem) For (1) and (2), see the above theorems. (3): We prove $D = D_{u_D}$.
First, "$\mathcal{H} \subseteq \mathcal{H}_{u_D}$": Let $A \in \mathcal{H}$. We have to show $A \in \mathcal{H}_{u_D}$, i.e. we have to show that A is an u_D-ball. This fact follows from Lemma.
Second, "$\mathcal{H} \supseteq \mathcal{H}_{u_D}$": Let $A = B(x, a) \in \mathcal{H}_{u_\mathcal{H}}$ be an u_D-ball, let $r = r(A) = u_D(A)$ be the radius/diameter of A. Then $A = B(x, r)$ and $r = u_D(x, x')$ for some $x, x' \in A$ (by definition of diameter). By definition of u_D, we have $r = h(C)$ (C is the least one from $\mathcal{H}$ where x and x' meet). Since C is an element of $\mathcal{H}$ and $h(C) = r$, we can show as above that $C = B(x, r)$, i.e. $A = C$ which means that $A \in \mathcal{H}$.

We have shown $\mathcal{H} \supseteq \mathcal{H}_{u_D}$. The fact $h = h_{u_D}$ follows by $h_{u_D}(A) = u_D(A) = h(A)$, see (3). $\square$

Optimal hierarchies

PROBLEM

We are given a set X objects with a (dis)similarity function (matrix) $d : X \times X \to [0, \infty)$. Alternatively, d is computed from object-attribute data table $\langle X, Y, I \rangle$. The aim is to construct a dendrogram D. Why a dendrogram (and not just d)? Because it is a "user-friendly" graphical way to look at the data. Intuitively, we want the dendrogram to represent the (dis)similarity structure given by d as close as possible. We know that a dendrogram corresponds to a unique ultrametric u. u can be thougt of as representing the dissimilarity structure contained in the dendrogram D. Therefore, starting from a dissimilarity d, we construct an ultrametric u. In a sense, constructing "a good" D from d is to look for an ultrametric u which approximates d well enough.

In the following, we show selected results along this line.

For two functions $d_i : X \times X \to [0, \infty)$ $(i = 1, 2)$ we put $d_1 \leq d_2$ iff for every $x_1, x_2 \in X$ we have $d_1(x_1, x_2) \leq d_2(x_1, x_2)$ (coordinatewise partial order).

Maximal dominated ultrametric u^-

The problem is to find an ultrametric u^- which
– is dominated by d (i.e. $u^- \leq d$), and

– dominates any other ultrametric dominated by d (i.e. if $u \leq d$ for some ultrametric u then $u \leq u^-$).

To make the dependence on d explicit, u^- will also be denoted by $u^-(d)$. Put $U^-(d) = \{u;\ u \text{ is an ultrametric and } u \leq d\}$.

Lemma (existence of u^-) Given a dissimilarity d on X, u^- is given by $u^-(x_1, x_2) = \sup_{u \in U^-(d)} u(x_1, x_2)$.

Proof We have to show that u^- as defined above is an ultrametric and that $u^- \leq d$. The fact $u^- \leq d$ follows from $u \leq d$ for each $u \in U^-(d)$. In order to show that u^- is an ultrametric, we need to verify the ultrametric inequality (the other conditions are obvious): We have $u^-(x_1, x_3) = \sup_{u \in U^-(d)} u(x_1, x_3) \leq \sup_{u \in U^-(d)} \max(u(x_1, x_2), u(x_2, x_3)) = \max(\sup_{u \in U^-(d)} u(x_1, x_3), \sup_{u \in U^-(d)} u(x_3, x_2)) = \max(u^-(x_1, x_3), u^-(x_3, x_2))$. □

Theorem (**single linkage gives** $u^-(d)$) Let d be a dissimilarity, $D = \langle \mathcal{H}, h \rangle$ the dendrogram obtained by single linkage agglomerative algorithm, u_D be the ultrametric corresponding to D. Then $u_D = u^-(d)$.

Proof Nebude pozadovan.

Minimal dominating ultrametric u^+

The problem is to find an ultrametric u^+ which
– dominates d (i.e. $u^+ \geq d$), and
– is minimal among all ultrametrics dominating d (i.e. if $u^+ \geq u \geq d$ for some ultrametric u then $u = u^+$).

Put $U^+(d) = \{u;\ u \text{ is an ultrametric and } u \geq d\}$.

Note: u^+ nedd not be unique. Using an analogical formula (to that for u^-), i.e. $u^+(x_1, x_2) = \inf_{u \in U^+(d)} u(x_1, x_2)$, we do not obtain an ultrametric.

There is in general not "the least" dominating ultametric. Example: Consider $X = \{x, y, z\}$ and dissimilarity d (it is not an ultrametric) given by

$$d(x, y) = 0.1, d(y, z) = 0.2, d(x, z) = 0.3.$$

Consider u_1, u_2 given by

$$u_1(x, y) = 0.1, u_1(y, z) = 0.3, u_1(x, z) = 0.3,$$
$$u_2(x, y) = 0.3, u_2(y, z) = 0.2, u_2(x, z) = 0.3.$$

One can see that both u_1 and u_2 are minimal ultrametrics dominating d but they are incomparable.

Theorem (**complete linkage gives u^+**) Let d be a dissimilarity, $D = \langle \mathcal{H}, h \rangle$ the dendrogram obtained by complete linkage agglomerative algorithm, u_D be the ultrametric corresponding to D. Then $u_D = u^+$, i.e. u_D is a minimal ultrametric dominating d (on of the possibly several minimal dominating ultrametrics).

Proof Nebude pozadovan.

Corollary If the dissimilarity matrix which inputs the agglomerative algorithm is an ultrametric then both single-linkage and complete-linkage algorithms yield the same dendrogram.

Disjoint clustering

- results into a partition of objects
- many particular approaches
- usually: user has to know (expected) number of clusters
- we focus only on: **competition learning**
- we do not discuss statistically-based clustering procedures, e.g. EM clustering (expectation maximization); theyr are well-elaborated

Competition learning: neural clustering

- set of points that need to be clustered: $T = \{x^p \in \mathbf{R}^n; p \in P\}$; $x^p = \langle x^p_1, \ldots, x^p_n \rangle$
- "neural network" scheme: n input neurons $x_1, \ldots, x_n$, m output neurons $y_1, \ldots, y_m$
- each output neuron represents one cluster
- parameters (weights) $w_{ij} \in R$ ($i = 1, \ldots, n$, $j = 1, \ldots, m$): cluster corresponding to j-th neuron is represented by a point with coordinates $\langle w_{1j}, \ldots, w_{nj} \rangle$
- PROBLEM: find good parameters w_{ij}

LEARNING ALGORITHM
INPUT: T, m
OUTPUT: $w_{ij} \in R$ ($i = 1, \ldots, n$, $j = 1, \ldots, m$)

1. set $t = 0$ (time/step)
2. $w_{ij}^0 \in R$ ($i = 1, \ldots, n$, $j = 1, \ldots, m$) at random (or by some heuristic based on knowledge of T)
3. set ν (learning rate, usually $0 < \nu < 1$)
4. for each $p \in P$: select the output neuron closest to x^p (winner): that with index j^* for which $d(x^p, w^t_{\_j})$ is minimal
5. update weights: $w_{ij^*}^{t+1} := w_{ij^*}^t + \nu(x_i^p - w_{ij^*}^t)$ for $i = 1, \ldots, n$
6. if clusters did not change in the last update cycle for $p \in P$, STOP; otherwise $t := t + 1$ and go to 4.

usually: $d(x^p, w_{\_j}) = \sum_{i=1}^n (x_i^p - w_{ij})^2$

Remarks

- meaning of weights update: New weight $w_{-j^*}^{t+1}$ (as a point in $\mathbf{R}^n$) is obtained by moving from the old weight $w_{-j^*}^{t}$ along the vector $x^p - w_{-j^*}^{t}$. Parameter ν says how far we move.
- meaning of "if clusters did not change in the last update cycle for $p \in P$": in every step t, every point x^p is assigned its corresponding winner $j^*(t,p)$ (step 4). "Clusters did not change" means that for each point $x^p \in T$, the winners $j^*(t,p)$ and $j^*(t-1,p)$ are the same.
- What is the resulting clustering? It is a partition Π of T into m classes given by the output neurons by the "winner takes all" principle:

$$\Pi = \{\{x^p \in T;\ \text{neuron } j \text{ is the winner for } x^p\};\ j = 1 \ldots, m\}.$$

CLASSIFICATION

Classification

pristi semestr