**Applying data mining to HiPR-FISH data**

People have been mining large datasets for interesting information for more than two decades

Can we apply one of these ready-made techniques to HiPR-FISH data?

I think the data is relational: i.e. there are spatial relationships between all identified cells

We would like to discover interesting relationships between taxa/host/particles

We are also interested in which relationships change between control and treatment

Inputs:

Distance

Orientation

Identity

Local quantity

Proof of concept:

Mouth bacteria – corn cob

Figures:

Heatmap of interactions on hiprfish images

**﻿﻿Data mining: An overview from a database perspective**

10.1109/69.553155

Data mining, which is also referred to as knowledge discovery in databases, means a process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in databases.

﻿In general, a data miner can be classified according to its mining of knowledge from the following different kinds of databases: relational databases, transaction databases, object-oriented databases, deductive databases, spatial databases, temporal databases, multimedia databases, heterogeneous databases, active databases, legacy databases, and the Internet information-base.

﻿Several typical kinds of knowledge can be discovered by data miners, including association rules, characteristic rules, classification rules, discriminant rules, clustering, evolution, and deviation analysis.

﻿Moreover, data miners can also be categorized according to the abstraction level of its discovered knowledge which may be classified into generalized knowledge, primitive-level knowledge, and multiple- level knowledge.

﻿Searching for similar patterns in a temporal or spatial-temporal database is essential in many data mining operations in order to discover and predict the risk, causality, and trend associated with a specific pattern. Typical queries for this type of database include identifying companies with similar growth patterns, products with similar selling patterns, stocks with similar price movement, images with similar weather patterns, geological features, environmental pollutions, or astrophysical patterns.

﻿Mining association rules in transactional or relational databases has recently attracted a lot of attention in data- base communities. The task is to derive a set of strong association rules…from the relevant data sets in a database. For example, one may find, from a large set of transaction data, such an association rule as if a customer buys (one brand of) milk, he/she usually buys (another brand of) bread in the same transaction.

﻿Data generalization and summarization presents the general characteristics or a summarized high-level view over a set of user-specified data in a database. For example, the general characteristics of the technical staffs in a company can be described as a set of characteristic rules or a set of generalized summary tables.

﻿Data classification is to classify a set of data based on their values in certain attributes. For example, it is desirable for a car dealer to classify its customers according to their preference for cars so that the sales persons will know whom to approach, and catalogs of new models can be mailed directly to those customers with identified features so as to maximize the business opportunity.

﻿﻿Basically, data clustering is to group a set of data (without a predefined class attribute), based on the conceptual clustering principle: maximizing the intraclass similarity and minimizing the interclass similarity. For example, a set of commodity objects can be first clustered into a set of classes and then a set of rules can be derived based on such a classification. Such clustering may facilitate taxonomy formation, which means the organization of observations into a hierarchy of classes that group similar events together.

**Association rules**

Could we call a “transaction” as a kernel that we iterate over the whole image? Association x -> y can then be determined based on how commonly x and y are in the same transaction.

Should we center the kernel on each cell centroid?

How do we deal with edges?

How do we account for overlapping kernels producing the same transaction multiple times?

Let ***I = {i1****,…,****im}*** be a set of taxa. ***D*** is a set of transactions, where a transaction ***T*** is a subset of ***I***. Let ***X*** be an “itemset” of taxa, where ***X*** is a subset of ***T***. We want rules of the form ***X -> Y*** where both ***X*** and ***Y*** are subsets of ***I***. Confidence is ***c*** where ***c%*** of transactions in ***D*** that contain ***X*** also contain ***Y***. Support is ***s*** where ***s%*** of transactions in ***D*** contain ***X*** or ***Y***.

Note: Quantities are not considered only presence or absence in a transaction.

High confidence and strong support define a *strong rule*.

Steps:

1. Discover the sets of “itemsets” that have support above a certain threshold
2. Generate association rules for the database

Step one determines the overall performance of the algorithm, step two is straightforward

Examples:

Apriori, DHP

Multiple Level Association rules:

Use hierarchical classification of items. Examine low level associations only when their higher level parents have a strong rule.

Interestingness:

Initially defined as - a rule is interesting if the confidence is greater than some cutoff: ***P(A and B)/P(A) > d****.* However, what if the presence of ***A*** actually reduces the likelihood of ***B***? e.g. ***B*** is present in 60% of cases where ***A*** is present, but is present in ***70%*** of all cases. A better heuristic is: ***P(A and B)/P(A) – P(B) > d*** (the probability of ***B*** given ***A*** compared to the probability of ***B***).

**Multilevel data generalization**

Is this stuff useful for taxonomic tree issues?

How do we integrate spatial info?

Mostly seems to be used for holding aggregate info.

Use data cube or attribute-oriented induction approach.

Data cube uses OLAP. Multidimensional database. Used to store aggregate functions like sum, average, etc.

Attribute-oriented induction approach looks at each attribute, assesses what the best generalization level is, and then collapses identical tuples.

With multilevel data, sometimes you want to “drill down” to more specificity or “roll up” to greater generalization.

**Data Classification**

Are there any known spatial-taxonomic patterns?

Can we use known patterns to build a classifier?

Would we would be too limited by a small training dataset?

Is it better to use similarity pattern matching?

Start with a sample database where each tuple has the same set of many attributes as the large database and where the class of each tuple is known. This is the training set used to train the classifier. Develop a model for each class using data features/attributes. Use the model to classify tuples in the large database.

Decision Trees:

Take a subset of the training set and form a decision tree for classification. Then classify the whole set. Add some of the incorrect classifications to the decision tree building set. Repeat until the decision tree correctly classifies better than some predetermined portion of the training set. The final tree has leaves for all the classes. The nodes are attributes and there are branches for all possible values of that attribute.

E.g. ID-3, C4.5: The tree is simple, it minimizes the steps needed to reach a classification, though not necessarily the simplest possible tree. More complexity (steps) means less generality. Choose steps with lots of information (that minimize the information needed in subsequent steps).

Evaluation functions: information gain, gini index, chi-square test

Other Methods:

Linear regression, combining partitioned classifiers, neural network