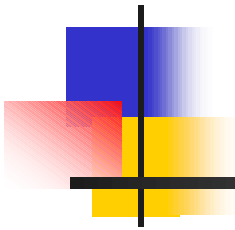

CLASTERING WITH AUTOCLASS

Milan Nikolic

Communication Networks Laboratory

http://www.ensc.sfu.ca/research/cnl

School of Engineering Science

Simon Fraser University

What is AutoClass?

- AutoClass is an unsupervised Bayesian classification system that seeks a maximum posterior probability classification

AutoClass information

- AutoClass on the Internet
- http://ic.arc.nasa.gov/ic/projects/bayes-group/autoclass/autoclass-c-program.html

- AutoClass contacts
- Dr. Peter Cheeseman (cheesem@ptolemy.arc.nasa.gov), Principal Investigator - NASA Ames, Computational Sciences Division
- John Stutz (stutz@ptolemy.arc.nasa.gov), Research Programmer - NASA Ames, Computational Sciences Division
- Will Taylor (taylor@ptolemy.arc.nasa.gov), Support Programmer - NASA Ames, Computational Sciences Division

Key features

- Automatically determines the number of classes
- Uses mixed discrete and real valued data
- Handles missing values
- Processing time is roughly linear in the amount of data
- Cases have probabilistic class membership
- Allows correlation between attributes within a class
- Generates reports describing the classes found
- Predicts “test” case class membership from a “training” classification

How it works?

- Input consists of a database of attribute vectors (cases) and a class model
- AutoClass uses:
  - Gaussian distributions over the real valued attributes
  - Bernoulli distributions over the discrete attributes
- AutoClass finds the set of classes maximally probable with respect to the data and the model
- Output is a set of class descriptions and partial membership of the cases in the classes

How to get started?

- The usual way of using AutoClass is to put all the data in a data file, describe that data with model and header files, describe the search with search parameters file, and run AutoClass in a “search” mode.

- The command to invoke AutoClass search is:

```
% autoclass –search <.db2 file path> <.hd2 file path>
  <.model file path> <.s-params file path>
```

Database file example

```
!#; AutoClass C data file -- database.db2
!#; prior to the first non-comment line being read
!#; the following chars in column 1 make the line a comment:
!#; '!', '#', ';', ' ', and '\n' (empty line)

!#; after the first non-comment line is read, the only column 1 comment characters are
!#; ' ', '\n' (empty line), and comment_char (data file format def in .hd2 file)

;;; Index Project Database
;;; from UC Berkeley

; 84 Data, 24 attributes
1 1 0 0 1 3 0 0 1 0 0 0 1 0 yes 1 0 0 1 0 yes no yes yes
2 3 2 0 0 2 0 0 1 1 1 1 1 0 no 1 1 1 1 1 yes yes yes yes
3 3 1 0 0 3 0 0 1 1 0 1 0 0 no 0 1 0 1 0 yes yes yes yes
4 2 4 1 0 2 0 0 1 1 0 0 1 0 no 0 1 0 0 1 yes yes yes yes
5 1 2 0 0 2 0 0 1 1 0 0 1 0 no 0 0 0 1 1 yes yes yes no
6 1 4 1 0 1 0 0 1 1 0 1 1 0 yes 0 1 0 0 1 no no yes yes
7 2 4 0 1 1 0 0 1 0 0 1 1 0 no 0 1 0 1 0 yes yes yes yes
. . . . . .
. . . . . .
```

Header file example

```
!#; AutoClass C header file -- database.hd2
!#; the following chars in column 1 make the line a comment:
!#; '!', '#', ';', ' ', and '\n' (empty line)

;;; Index Project Database
;;; from UC Berkeley

; 84 Data, 24 attributes
;#! num_db2_format_defs <num of def lines -- min 1, max 4>
num_db2_format_defs 2
;; required
number_of_attributes 24
;; optional - default values are specified
separator_char  ' '

;; <zero-based att#>  <att_type>  <att_sub_type>  <att_description>  <att_param_pairs>
0  dummy nil "case number"
1 discrete nominal "income" range 3
2 discrete nominal "profession" range 5
3 discrete nominal "engineer" range 2
. . . . . .
. . . . . .
```

Model file example

```
!#; AutoClass C model file -- database.model
!#; the following chars in column 1 make the line a comment:
!#; '!', '#', ';', ' ', and '\n' (empty line)

;;; Index Project database
;;; from UC Berkeley

;; 1 or more model definitions
;; model_index <zero-based index> <number of model definition lines>
model_index 0 1
;; <model type> <zero-based attribute number> <zero-based attribute number> ...
single_multinomial 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

Parameters file example

```
# PARAMETERS TO AUTOCLASS-SEARCH -- AutoClass C
# ------------------------------------------------------------------------------------
# as the first character makes the line a comment, or
! as the first character makes the line a comment, or
; as the first character makes the line a comment, or
;;; '\n' as the first character (empty line) makes the line a comment.
# to override the following default parameters,
# enter below the line => #!;#!;#!;#!;#!;#!;#!;#!;#!;#!;#!;
# <parameter_name> = <parameter_value>, or
# <parameter_name> <parameter_value>, or      # separator is a space
# <parameter_name>\tab<parameter_value>.
# ------------------------------------------------------------------------------------
#  DEFAULT PARAMETERS
# ------------------------------------------------------------------------------------
# rel_error = 0.01
!       passed to clsf-DS-%= when deciding if a new clsf is duplicate of old
# start_j_list = 2, 3, 5, 7, 10, 15, 25
!       initially try these numbers of classes, so not to narrow the search
!       too quickly.  the state of this list is saved in the <..>.search file
!       and used on restarts, unless an override specification of start_j_list
!       is made in this file for the restart run.
. . . . . .
```

Generating reports

- The command to invoke AutoClass reports is:

```
% autoclass –reports <.results[-bin] file path>
<.search file path> <.r-params file path>
```

- The standard reports are:
  - attribute influence values
  - cross-reference by case number
  - cross-reference by class number

Prediction using classification

- The command to invoke AutoClass prediction is:

```
% autoclass –predicts <-predict.db2 file path>
   <.results[-bin] file path> <.search file path>
   <.r-params file path>
```

AutoClass results (1)

- Analyzed database was collected from the INDEX project demographic questionnaire and consists of 84 cases with 23 attributes
- AutoClass found classification with 3 classes:
- CLASS 1 (47 cases):
  - students and technicians
  - in their 20's or 30's
  - with low or mid income

AutoClass results (2)

- CLASS 2 (28 cases):
  - professors, technicians and administrators
  - in their 30's or 50's
  - with high or mid income
  - Internet usage at home paid by university
- CLASS 3 (9 cases):
  - employed people
  - with high or mid income
  - Internet usage at home paid by employer
  - Internet usage at work paid by employer

References

- Peter Cheeseman and John Stutz, “Bayesian Classification (AutoClass): Theory and Results”, 1995.
- Robin Hanson, John Stutz and Peter Cheeseman, “Bayasian Classification Theory”, 1990.