PhD thesis:

# New models and algorithms for semi-naive Bayesian classification focused on the AODE paradigm

*PhD defense*
on 19/06/2012

Ana M. Martínez

Advisors: Dr. José A. Gámez and Dr. M. Julia Flores
Intelligent Systems and Data Mining's group (SIMD)
UCLM - Albacete - Spain

UNIVERSIDAD DE CASTILLA-LA MANCHA



# Outline





# Outline





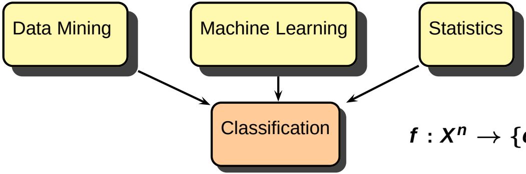
Introduction
Motivation
Objectives
I New BNCs to overcome AODE's limitations
Hidden One-Dependence Estimator
Gaussian AODE and hybrid AODE
The MTE-AODE classifier
II Discretization techniques for semi-naive BNCs
Disjoint discretization techniques
Non-disjoint discretization techniques
III Domains of competence of the semi-naive BNCs
Concluding remarks

# Motivation

Classification

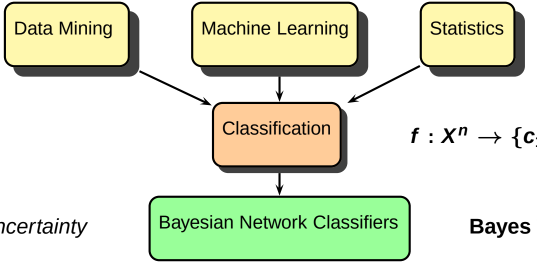
$$f : X^n \rightarrow \{c_1, \ldots, c_k\}$$

# Motivation




$$f : X^n \rightarrow \{c_1, \ldots, c_k\}$$

# Motivation

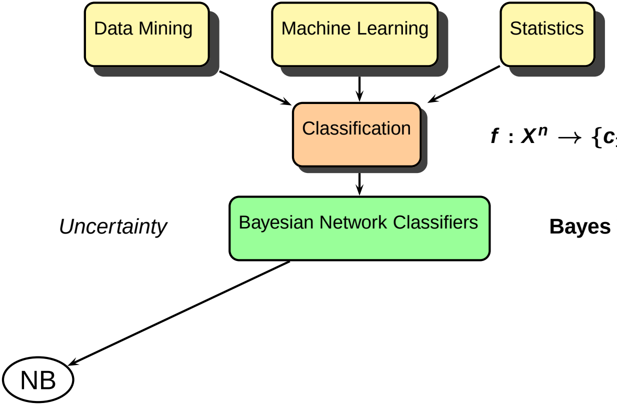



$$f : X^n \rightarrow \{c_1, \ldots, c_k\}$$

*Uncertainty*

**Bayes Theorem**

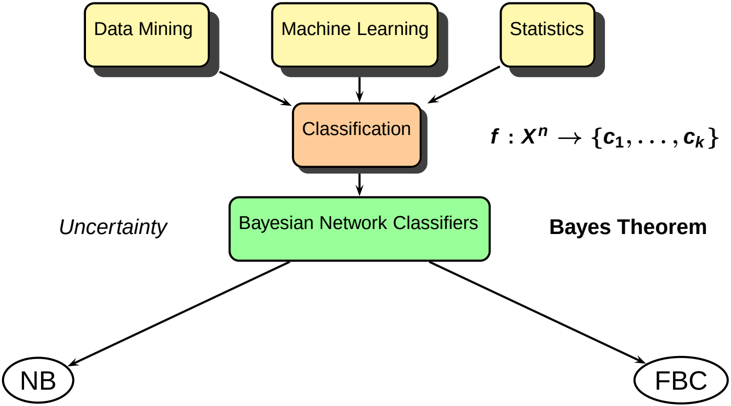
# Motivation




$$f : X^n \rightarrow \{c_1, \ldots, c_k\}$$

**Bayes Theorem**

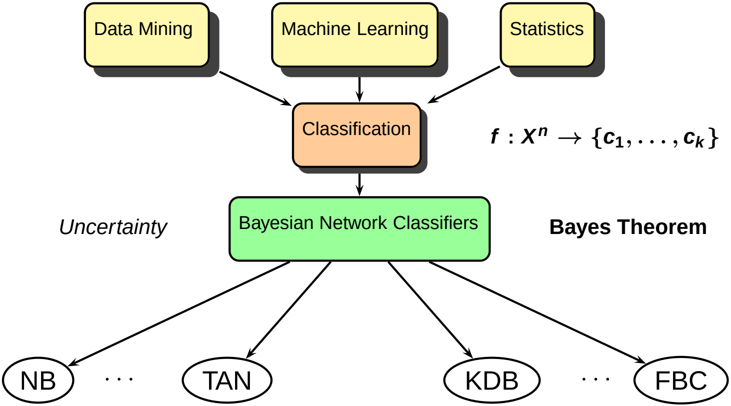
# Motivation



Data Mining
Machine Learning
Statistics
Classification
$f : X^n \to \{c_1, \ldots, c_k\}$
Uncertainty
Bayesian Network Classifiers
Bayes Theorem
NB
FBC



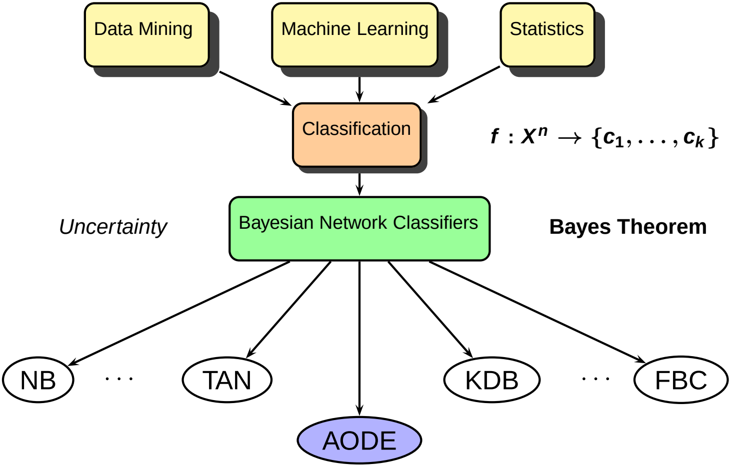

# Motivation




$$f : X^n \rightarrow \{c_1, \ldots, c_k\}$$



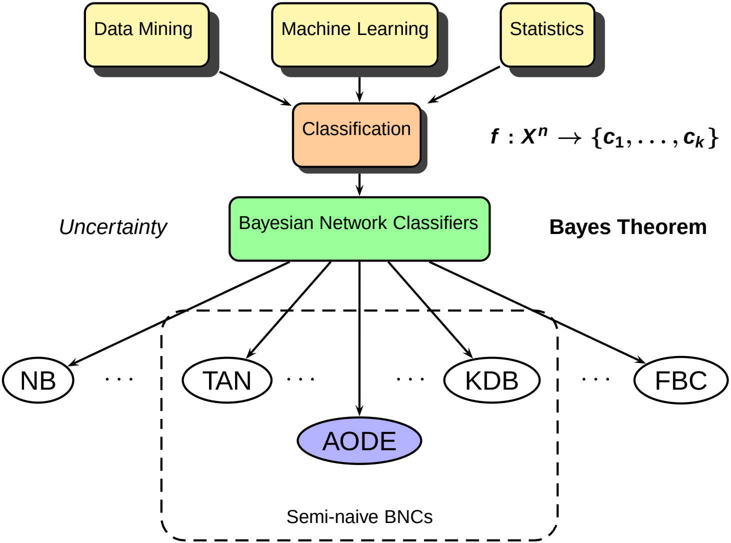
# Motivation



Data Mining
Machine Learning
Statistics
Classification
$f : X^n \rightarrow \{c_1, \ldots, c_k\}$
*Uncertainty*
Bayesian Network Classifiers
**Bayes Theorem**
NB
· · ·
TAN
KDB
· · ·
FBC
AODE

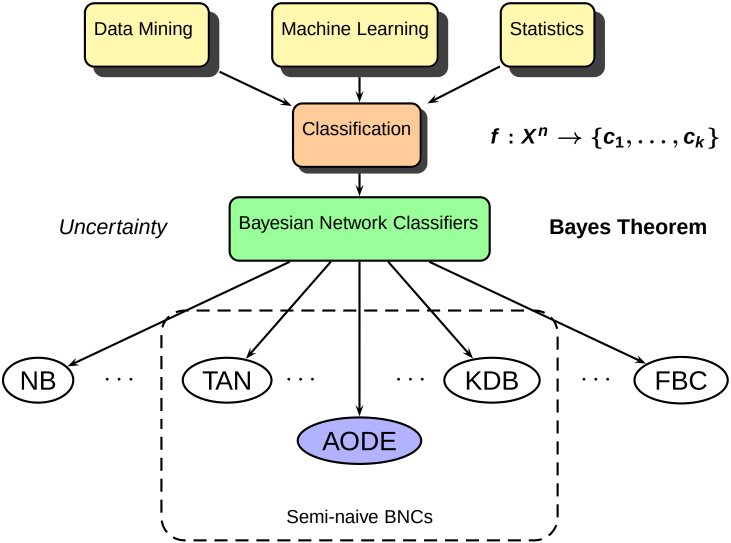



# Motivation



Data Mining
Machine Learning
Statistics
Classification
$f : X^n \rightarrow \{c_1, \ldots, c_k\}$
Uncertainty
Bayesian Network Classifiers
Bayes Theorem
NB
. . .
TAN
. . .
. . .
KDB
. . .
FBC
AODE
Semi-naive BNCs

# Motivation


Data Mining
Machine Learning
Statistics
Classification
$f : X^n \rightarrow \{c_1, \ldots, c_k\}$
*Uncertainty*
Bayesian Network Classifiers
**Bayes Theorem**
NB
TAN
KDB
FBC
AODE
Semi-naive BNCs

# Naive Bayes classifier

- The attributes are conditionally independent given the class value $I(A_i, A_j | C)$.

$$c_{MAP} = argmax_{c \in \Omega_C}\, p(c) \prod_{i=1}^{n} p(a_i | c)$$

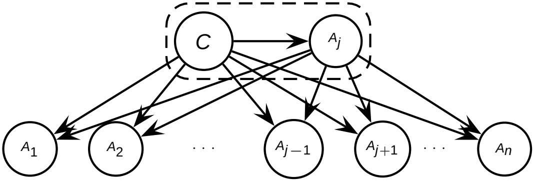
- **Time complexity**:
  - **Training**: $\mathcal{O}(mn)$
  - **Classification**: $\mathcal{O}(cn)$
- **Space complexity**:
  - **Training**: $\mathcal{O}(cnv)$
  - **Classification**: $\mathcal{O}(cnv)$

- Limitations:
  - ✗ : It does not work properly in certain datasets.
  - ✗ : Dependencies between attributes reduce, unavoidably, the prediction capability of NB.
  - ✗ : Not only interesting to be right in the classification in certain applications.



# AODE classifier I
## *(Averaged One-Dependence Estimators)*

- **AODE** is significantly better in terms of error reduction compared to the rest of semi-naive techniques.




- **MAP hypothesis:**

$$argmax_{c\in\Omega_C}\left(\sum_{j=1,N(a_j)>m}^{n} p(c,a_j)\prod_{i=1,i\neq j}^{n} p(a_i|c,a_j)\right)$$



# AODE classifier II

- **Time complexity**:
  - **Training**: $\mathcal{O}(mn^2)$
  - **Classification**: $\mathcal{O}(cn^2)$

- **Space complexity**:
  - **Training**: $\mathcal{O}(c(nv)^2)$
  - **Classification**: $\mathcal{O}(c(nv)^2)$



# AODE classifier II

- **Time complexity**:
  - **Training**: $\mathcal{O}(mn^2)$
  - **Classification**: $\mathcal{O}(cn^2)$
- **Space complexity**:
  - **Training**: $\mathcal{O}(c(nv)^2)$
  - **Classification**: $\mathcal{O}(c(nv)^2)$
- Limitations:



# AODE classifier II

- **Time complexity**:
  - **Training**: $\mathcal{O}(mn^2)$
  - **Classification**: $\mathcal{O}(cn^2)$
- **Space complexity**:
  - **Training**: $\mathcal{O}(c(nv)^2)$
  - **Classification**: $\mathcal{O}(c(nv)^2)$

- Limitations:
  - × : **Quadratic order time in classification**.
  - × : **High demand of RAM memory**.



# AODE classifier II

- **Time complexity**:
  - **Training**: $\mathcal{O}(mn^2)$
  - **Classification**: $\mathcal{O}(cn^2)$
- **Space complexity**:
  - **Training**: $\mathcal{O}(c(nv)^2)$
  - **Classification**: $\mathcal{O}(c(nv)^2)$

- Limitations:
  - × : **Quadratic order time in classification**.
  - × : **High demand of RAM memory**.
  - × : Only discrete variables.



# AODE classifier II

- **Time complexity**:
  - **Training**: $\mathcal{O}(mn^2)$
  - **Classification**: $\mathcal{O}(cn^2)$
- **Space complexity**:
  - **Training**: $\mathcal{O}(c(nv)^2)$
  - **Classification**: $\mathcal{O}(c(nv)^2)$

- Limitations:
  - × : **Quadratic order time in classification**.
  - × : **High demand of RAM memory**.
  - × : Only discrete variables.

- Attempts to improve AODE's **accuracy**
  - Two algorithms based on EM and constrained optimization for learning MAP weights for the SPODEs **[Cerquides & de Mántaras, 2005]**
  - To select or to weight: A comparative study of model selection and model weighing for SPODE ensembles **[Yang et al., 2006]**.
  - **WAODE**: Model weighting with $MI(C, A_j)$ **[Jiang & Zhang, 2006]**.



# Outline

1. **Introduction**
   - Motivation
   - Objectives
2. **I New BNCs to overcome AODE's limitations**
   - Hidden One-Dependence Estimator
   - Gaussian AODE and hybrid AODE
   - The MTE-AODE classifier
3. **II Discretization techniques for semi-naive BNCs**
   - Disjoint discretization techniques
   - Non-disjoint discretization techniques
4. **III Domains of competence of the semi-naive BNCs**
5. **Concluding remarks**



# Objectives

1. Reduce AODE's computational needs in terms of memory storage and classification time.
2. Propose different alternatives for handling numeric attributes, commonly present in real databases:
   1. Gaussian distributions.
   2. Mixtures of truncated exponentials.
   3. Different discretization techniques (disjoint vs non-disjoint).
3. Study the domain of competence for several semi-naive classifiers according to different complexity measures **[Ho & Basu, 2002]**.



# Outline



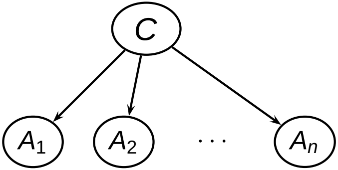


# Motivation

- AODE is **quadratic** in **training** and **classification time**.
  - Can be a handicap in some real applications where the **response time** is **critical**.
- AODE needs to store the *n* models.
  - Can be problematic when the size of the database (mainly the number of attributes or attribute labels) is very large.
  - Real examples: microarrays or DNA chips or large marketing databases (Orange)
- <u>Our solution</u>: new classifier which **estimates a new variable** which gathers the dependencies represented by every superparent in AODE in *a single model*.

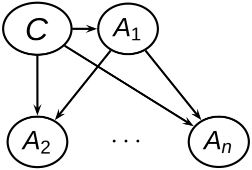


# HODE Classifier

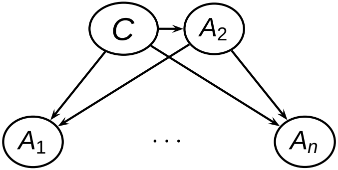




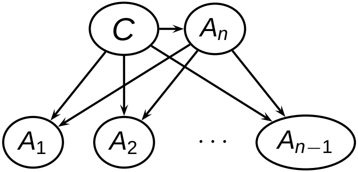

# HODE Classifier



# HODE Classifier

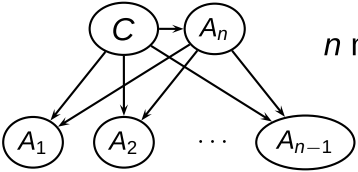


# HODE Classifier



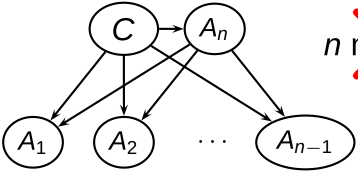



# HODE Classifier



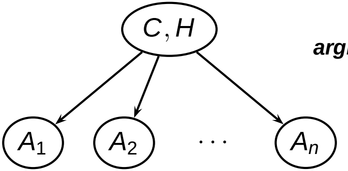

$n$ models

# HODE Classifier



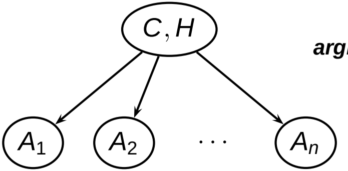

~~$n$ models~~

# HODE Classifier




$$\mathrm{argmax}_{c\in\Omega_C}\left(\sum_{j=1}^{\#H} p(c,h_j)\prod_{i=1}^{n} p(a_i|c,h_j)\right)$$

# HODE Classifier




$$argmax_{c \in \Omega_C} \left( \sum_{j=1}^{\#H} p(c, h_j) \prod_{i=1}^{n} p(a_i | c, h_j) \right)$$

- Necessary to estimate the probability of every attribute value conditioned by the class and $H$.
    - **Expectation-Maximization algorithm**.

# Number of states for the $H$ variable

- **Greedy technique**:
  - First step: $\#H = 1$ (NB).
  - Execution of EM algorithm and build models: $\#H = \#H + 1$.
  - Evaluate model: if worse than previous model then stop process.

- **How is the fitness of the model evaluated?**
  - **Log-likelihood (LL)**:

$$LL = \sum_{i=1}^{I} \log \left( \sum_{t=1}^{\#H} p(c^i, a_1^i, \cdots, a_n^i, h_t) \right) = \sum_{i=1}^{I} \log \left( \sum_{t=1}^{\#H} p(c^i, h_t) \prod_{r=1}^{n} p(a_r^i | c^i, h_t) \right)$$

  - **Penalization**: **Akaike Information Criterion or AIC**:

$$AIC = LL - C(M) .$$

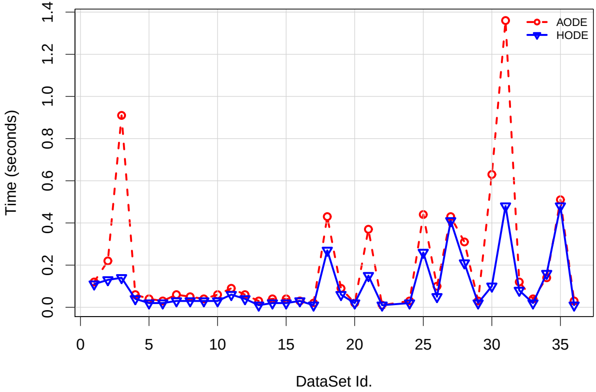
- **Results:** $(w/t/l)$: $16/6/14$.



# What could make us vote for one or the other?

## Time complexity

- At **training time** HODE is **quadratic in the worst case**:
  - $1mn + 2mn + \cdots + nmn$
  - AODE is usually faster in model construction (because of EM).
- At **classification time:**



# What could make us vote for one or the other?

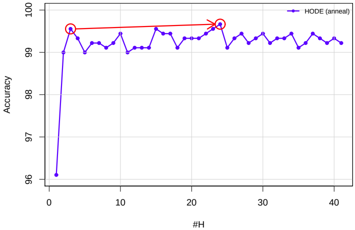
## Space complexity

- **Lower than AODE's**, it needs to store less CPTs.
  - $\mathcal{O}(cn\#Hv)$ for HODE vs $\mathcal{O}(c(nv)^2)$ for AODE.
- AODE demands more RAM memory, problems in large databases with a high number of attributes or even attributes with too many states.
- Experiments in 7 datasets (microarrays or DNA chips).

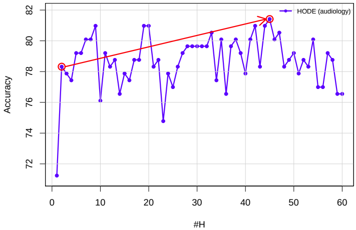
| Dataset | $k$ | $n$ | $l$ | NB | AODE | HODE |
|---|---|---|---|---|---|---|
| colon | 2 | 2000 | 62 | 93.5484 | 91.9355 | **96.7742** |
| DLBCL-Stanford | 2 | 4026 | 47 | 100 | 100 | 100 |
| GCM | 14 | 16063 | 190 | 60.5263 | OutOfMem | **70** |
| leukemia | 2 | 7129 | 72 | **100** | OutOfMem | 98.6111 |
| lungCancerHarvard2 | 2 | 12533 | 181 | 98.895 | OutOfMem | **99.4475** |
| lymphoma | 9 | 4026 | 96 | **96.875** | OutOfMem | 75 |
| prostate_tumorVS | 2 | 12600 | 136 | 80.1471 | OutOfMem | **95.5882** |

- OutOfMem: problems of overflow with a maximum of 8 gigabytes.
- HODE terminated without problems, even with a lower need for memory.

# HODE's parallelization



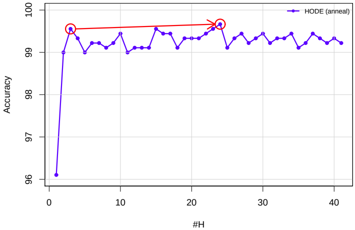

anneal



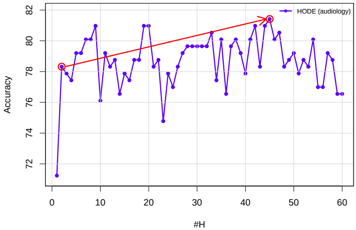

audiology



# HODE's parallelization



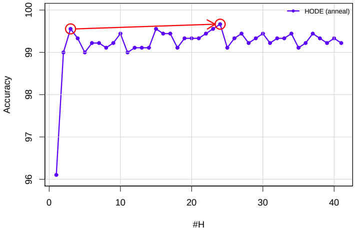

anneal



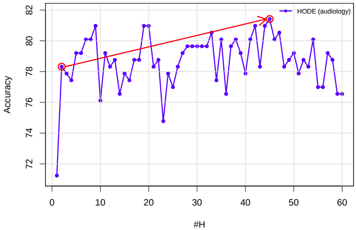

audiology



# HODE's parallelization



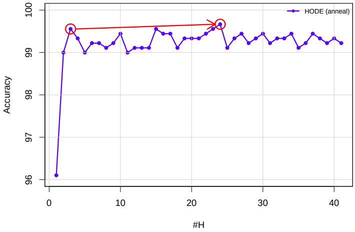

anneal



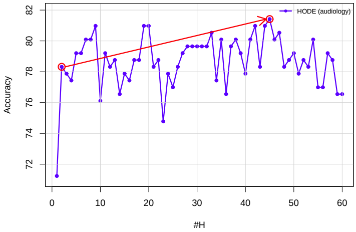

audiology



# HODE's parallelization




anneal




audiology

- mpiJava library.

# Partial conclusions and future work

- **HODE**: alternative to the AODE classifier (similar accuracy results).
  - **Reduction** in *space complexity* and *classification time*.

√ HODE tested in a **parallel environment**: global optimum for $\#H$.

√ Direct adaptation for the imputation of **missing values** in the dataset, use of **EM**.
  - Better records (12-0-6) than imputation with the global mean/mode.

- Future improvements:
  - Average the constructed models.

# Outline

1 Introduction
   Motivation
   Objectives

2 I New BNCs to overcome AODE's limitations
   Hidden One-Dependence Estimator
   Gaussian AODE and hybrid AODE
   The MTE-AODE classifier

3 II Discretization techniques for semi-naive BNCs
   Disjoint discretization techniques
   Non-disjoint discretization techniques

4 III Domains of competence of the semi-naive BNCs

5 Concluding remarks



# Motivation

- A BN assumes **all** the variables are **discrete**.
- It is common the **coexistence** of **discrete** and **continuous** variables in the same problem.
- Direct solution $\longrightarrow$ **discretization**.
  - ✗ : Unavoidable loss of precision.
  - ✗ : Which discretization method should we choose?.
- Some alternative solutions:
  1. **Kernel-based distributions**. Space and temporal undesired restrictions for AODE (concerning the number of instances).
  2. **Conditional Gaussian Networks**.
  3. **Mixtures of truncated exponentials**.



# Conditional Gaussian Networks

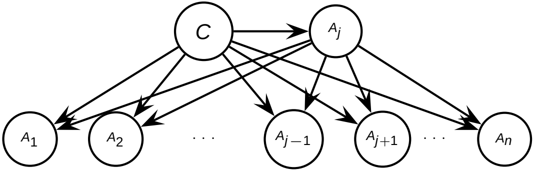
- Every **continuous variable** $X$ is modeled with a **conditional Gaussian distribution**.

$$f(X|\mathbf{Y}=y,\mathbf{Z}=z;\Theta)=\mathcal{N}\left(x:\mu_X(y)+\sum_{j=1}^{s}b_{XZ_j}(y)(z_j-\mu_{Z_j}(y)),\sigma^2_{X|\mathbf{Z}}(y)\right)$$

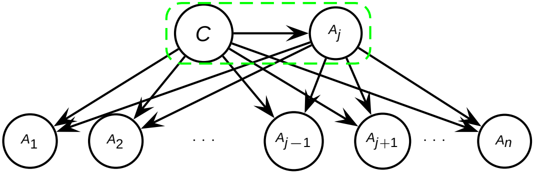
- $b_{XZ_j}(y)$, **regression term** that measures the strength of the connection between $X$ and every continuous parent.
- $\sigma^2_{X|\mathbf{Z}}(y)$ is the **conditional variance** of $X$ over its continuous parents.



# GAODE classifier



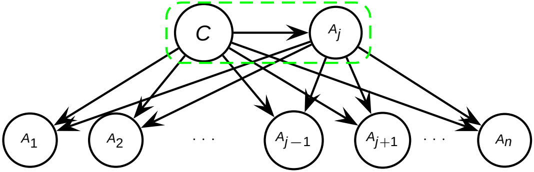
# GAODE classifier






# GAODE classifier

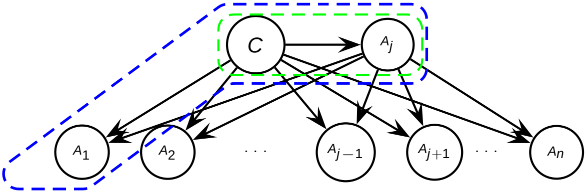
## - Univariate Gaussian Distribution -





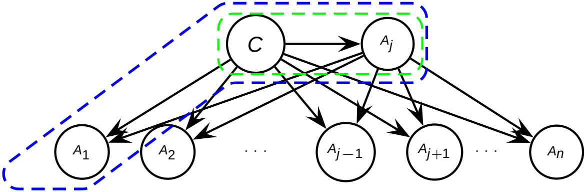

# GAODE classifier

## - Univariate Gaussian Distribution -



# GAODE classifier

## - Univariate Gaussian Distribution -




## - Conditional Gaussian Distribution -

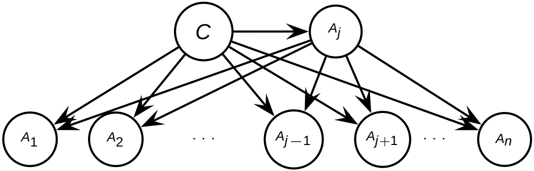
# GAODE classifier

- **Time complexity**:
  - The same as AODE (incremental computation of parameters).
- **Space complexity**:
  - **Training & Classification**: $\mathcal{O}(kn^2)$ (independent from $v$).

√ Probabilities estimated can be more reliable comparing to the multinomial version as they are modeled from more samples.

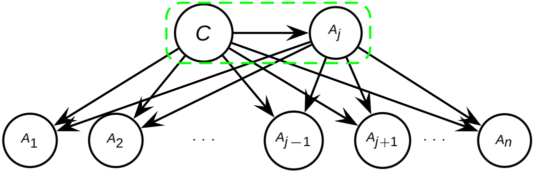
× : Not possible to define the corresponding probability function for a discrete variable conditioned to a numeric attribute. **Restricted to Numerical datasets**.

# HAODE classifier



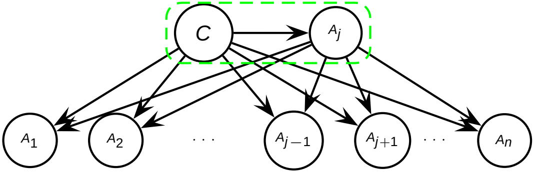



# HAODE classifier



# HAODE classifier

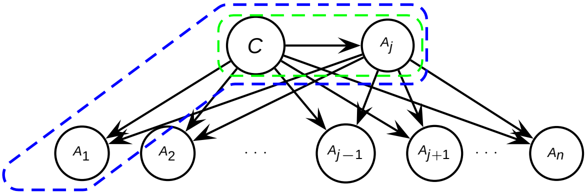
## - Multinomial distribution -






# HADOE classifier

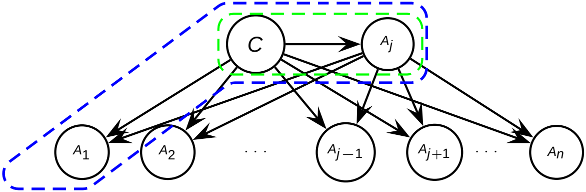
## - Multinomial distribution -






# HAODE classifier

## - Multinomial distribution -




## - Univariate Gaussian distribution - (Multinomial distribution)

# HAODE classifier

- **Time complexity**:
  - The same as AODE (incremental computation of parameters).
- **Space complexity**:
  - **Training & Classification**: The same as AODE $\mathcal{O}(k(nv)^2)$ in the worst case.

√ Able to deal with **hybrid datasets** too.

# Results

- **Datasets with only continuous attributes**
  - √ Both of our classifiers are significantly better than NB.
  - √ **GAODE** obtains competitive results comparing to AODE.
  - √ **HAODE** offers an even higher advantage, HAODE **significantly improves AODE in numeric datasets**.
- **Hybrid datasets?**
  - *Wilcoxon test*: No statistical difference.
  - No significant pattern found when the percentage of numerical variables with respect to discrete ones was analysed.
  - Apparent tendency of HAODE to **penalize datasets with missing values**. Statistical difference (Wilcoxon) when only datasets with missing values are considered.



# Partial conclusions and future work

- **GAODE**: applies CGNs.
  - √ **Competitive results** comparing to AODE (5 – 16 – 5).
  - √ **Reduction** in the **space complexity**.
  - √ Maintains AODE's time complexity.
  - × : Restricted to continuous datasets.
- **HAODE**: discretizes the superparents.
  - √ **Significantly better** than AODE in continuous datasets (6 – 19 – 1).
  - √ Able to deal with **all kind of datasets**.
  - × : Clear preference for datasets with continuous attributes and absence of missing data.
- Reconsider Gaussian assumption.



# Outline

1. Introduction
   - Motivation
   - Objectives
2. I New BNCs to overcome AODE's limitations
   - Hidden One-Dependence Estimator
   - Gaussian AODE and hybrid AODE
   - The MTE-AODE classifier
3. II Discretization techniques for semi-naive BNCs
   - Disjoint discretization techniques
   - Non-disjoint discretization techniques
4. III Domains of competence of the semi-naive BNCs
5. Concluding remarks



# The MTE model (Moral et al., 01)

## Definition (MTE potential)

- $\mathbf{X}$: mixed $n$-dimensional random vector. $\mathbf{Y} = (Y_1, \ldots, Y_d)$, $\mathbf{Z} = (Z_1, \ldots, Z_c)$ its discrete and continuous parts. A function $f : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$ is a Mixture of Truncated Exponentials potential (MTE potential) if for each fixed value $\mathbf{y} \in \Omega_{\mathbf{Y}}$ of the discrete variables $\mathbf{Y}$, the potential over the continuous variables $\mathbf{Z}$ is defined as:

$$f(\mathbf{z}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{ \sum_{j=1}^{c} b_i^{(j)} z_j \right\}$$

  for all $\mathbf{z} \in \Omega_{\mathbf{Z}}$, where $a_i$, $b_i^{(j)}$ are real numbers.
- Also, $f$ is an MTE potential if there is a partition $D_1, \ldots, D_k$ of $\Omega_{\mathbf{Z}}$ into hypercubes and in each $D_i$, $f$ is defined as above.



# Learning MTEs from data:

## Univariate case

Learning a univariate MTE density involves three basic steps:

# Learning MTEs from data:

## Univariate case

Learning a univariate MTE density involves three basic steps:

- Determination of the splits into which $\Omega_X$ will be partitioned (Convexity/Concavity).

# Learning MTEs from data:

## Univariate case

Learning a univariate MTE density involves three basic steps:

- Determination of the splits into which $\Omega_X$ will be partitioned (Convexity/Concavity).
- Determination of the number of exponential terms in the mixture for each split (usually 2).

# Learning MTEs from data:

## Univariate case

Learning a univariate MTE density involves three basic steps:

- Determination of the splits into which $\Omega_X$ will be partitioned (Convexity/Concavity).
- Determination of the number of exponential terms in the mixture for each split (usually 2).
- Estimation of the parameters.

# Learning MTEs from data:

## Univariate case

Learning a univariate MTE density involves three basic steps:

- Determination of the splits into which $\Omega_X$ will be partitioned (Convexity/Concavity).
- Determination of the number of exponential terms in the mixture for each split (usually 2).
- Estimation of the parameters.

## Alternatives explored in literature

- Least squared estimation (LS).
- Maximum likelihood estimation (ML).



# Learning MTEs from data:

## Univariate case

Learning a univariate MTE density involves three basic steps:

- Determination of the splits into which $\Omega_X$ will be partitioned (Convexity/Concavity).
- Determination of the number of exponential terms in the mixture for each split (usually 2).
- Estimation of the parameters.

## Alternatives explored in literature

- Least squared estimation (LS).
- Maximum likelihood estimation (ML).

## Conditional case

- Mixed tree: efficient data structure for dealing with conditional MTE densities.



# MTE-AODE classifier

**Definition**

An **MTE-AODE classifier** classifies an individual as in AODE, where all the probability functions involved are of class MTE.

- **Advantages**:
  - (over GAODE) the underlying distribution is not assumed to be of Gaussian type, as MTEs are able to accurately represent the most common distributions.
  - (over HAODE) there is no need to discretize the super-parent nodes, as the MTE mode allows discrete variables with continuous parents.
- **Disadvantage**: the learning phase is slower.
- We can trivially extend this definition to the MTE-NB classifier as a particular case of MTE-AODE.

# Accuracy comparisons

- Study the application of MTEs over a **large group of datasets** without individual parametrization.
  - 16 hybrid datasets.

| | Naive Bayes | | AODE | |
|---|---|---|---|---|
| | MTE vs G | MTE vs D | MTE vs H | MTE vs D |
| EF10 | 10-6 | 5-11 | 8-8 | •5-11 |
| EF5 | 11-5 | 7-9 | 9-7 | 7-9 |
| F&I | 10-6 | 9-7 | 8-8 | 8-8 |

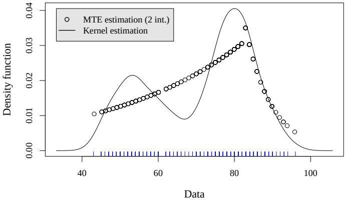


# Accuracy comparisons

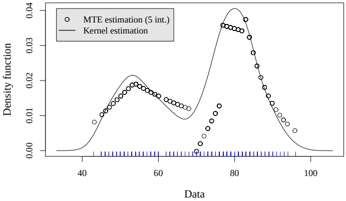
- Study the application of MTEs over a **large group of datasets** without individual parametrization.
  - 16 hybrid datasets.

| | Naive Bayes | | AODE | |
|---|---|---|---|---|
| | MTE vs G | MTE vs D | MTE vs H | MTE vs D |
| EF10 | 10-6 | 5-11 | 8-8 | •5-11 |
| EF5 | 11-5 | 7-9 | 9-7 | 7-9 |
| **F&I** | **10-6** | **9-7** | **8-8** | **8-8** |

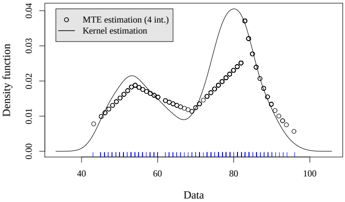
# Estimation of MTEs when selecting different number of cutpoints




(a) 2 intervals




(b) 5 intervals




(c) 4 intervals



## Partial conclusions and future work

- Generalization of AODE to all kind of datasets through MTEs.
  - ✓ Extensible to NB and other semi-naive classifiers.

1. The selection of the number of intervals is not trivial:
   - It is advisable to find the optimum for every data set, but also,
   - To find the optimum for every domain to partition in each leaf of the mixed tree independently, requiring a more sophisticated supervised “discretization” technique oriented to the estimation of MTEs.
2. Time complexity:
   - The estimation of MTEs requires more time than EF discretization.
   - This time can be controlled: limiting the number of exponential terms and the maximum number of intervals into which divide the domain.

- Good alternative to CGs and discretization for some datasets.



# Outline

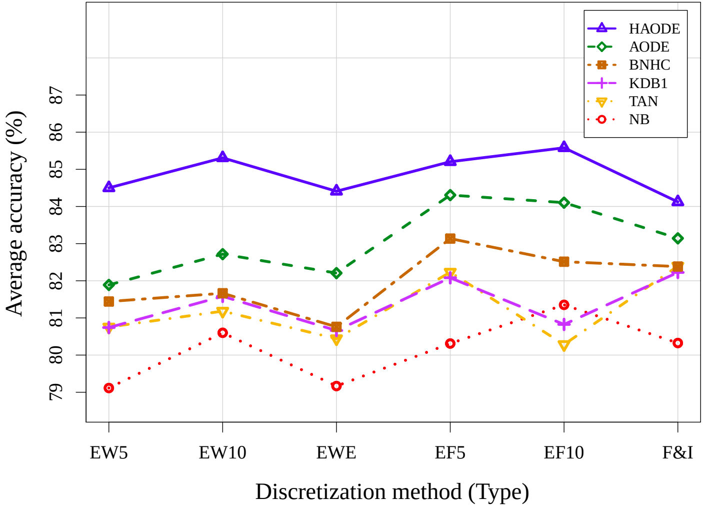
1. **Introduction**
   - Motivation
   - Objectives
2. **I New BNCs to overcome AODE's limitations**
   - Hidden One-Dependence Estimator
   - Gaussian AODE and hybrid AODE
   - The MTE-AODE classifier
3. **II Discretization techniques for semi-naive BNCs**
   - Disjoint discretization techniques
   - Non-disjoint discretization techniques
4. **III Domains of competence of the semi-naive BNCs**
5. **Concluding remarks**



# Motivation

- Discretization is probably one of the **pre-processing techniques most broadly used** in machine learning.
- The real distribution of the data is replaced with a mixture of uniform distributions.
- **Discretization methods**: unavoidable *loss of information*.
- Appropriate discretization can **outperform** straightforward use of common, but often **unrealistic parametric distribution** (e.g. Gaussian).
- Many distinct techniques for discretization can be found in literature.
- *Should we worry about the discretization method applied when designing the set of experiments?*.
  - HAODE vs AODE's results with F&I can be generalized.

# Study in terms of accuracy

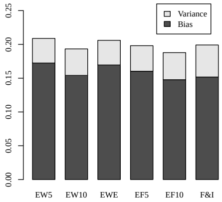




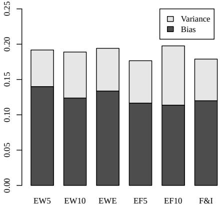

# Before we go: discretization bias and variance

- **In practice:**
  - **Discretization Bias:** Intuitively, discretization resulting in *large interval numbers* tends to have *low* bias (any given interval is less likely to include a decision boundary of the original numeric attribute).
  - **Discretization Variance:** discretization resulting in *intervals with a large number of instances* tends to have *low* variance (as the probability estimations are more stable and reliable).
- **Problem:** assuming there is a fixed dataset size, the larger the number of intervals, the smaller the number of instances per interval is.

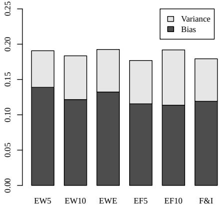
# Study in terms of bias/variance



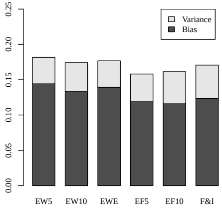

NB



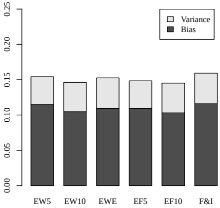

TAN



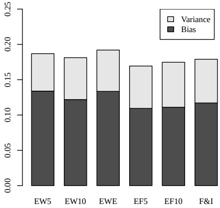

KDB1




AODE



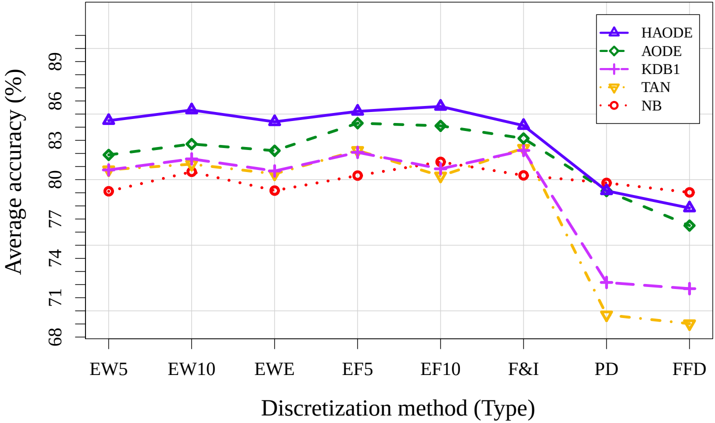

HAODE




BNHC

# NB-tailored discretization techniques

## Study in terms of accuracy






# Partial conclusions and future work

- **Outcomes:**
  1. Can the use of a particular discretization method **alter the ranking** between classifiers?
     - **Not** for HAODE, AODE and BNHC.
     - As NB, TAN and KDB obtain very similar results, their position in the ranking can vary in a particular case
  2. Comparison between AODE and HAODE with several discretization methods:
     - In all cases HAODE's average accuracy is higher than AODE's (significantly better except for EF10).
- If discretization technique tailored to NB not extensible to other classifiers of the same family.
- **Future work:** considering a wider range of discretization techniques (e.g. multivariate discretization).



# Outline

1 Introduction
Motivation
Objectives

2 I New BNCs to overcome AODE's limitations
Hidden One-Dependence Estimator
Gaussian AODE and hybrid AODE
The MTE-AODE classifier

3 II Discretization techniques for semi-naive BNCs
Disjoint discretization techniques
Non-disjoint discretization techniques

4 III Domains of competence of the semi-naive BNCs

5 Concluding remarks



# Motivation

- For **AODE** and **HAODE**, previous studies have shown its robustness towards the discretization method applied.
  - **Only disjoint discretization** techniques taken into account so far.
- **Non-Disjoint discretization**:
  - Good performance in Naive Bayes **[Yang & Webb, 2002]**.
  - Are disjoint intervals appropriate for AODE and/or HAODE?

# Disjoint vs Non-Disjoint Discretization

- Given $x_i, x_j \in \mathbb{R}$, any disjoint discretization (DD) method would create a **unique interval** $(a, b] \ni x_i$ and $(d, e] \ni x_j$ **for every value**.
- In DD (EF, EW, MDL): every numeric sample belongs to a single interval:
  - If $x_i$ or $x_j$ falls around the **center of the interval** assigned, we could expect more **distinguishing information** than when it falls near one of the boundaries of the interval.
- In NDD: with an **odd number of intervals** (in training time) every numeric value will always be located toward the middle of the final interval when classifying each test instance (e.g. three intervals).

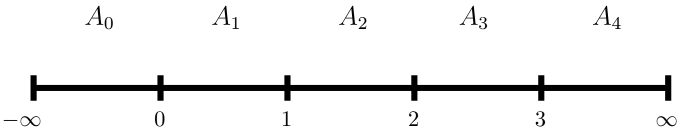


# Non-Disjoint Discretization (Example)

$\lfloor \sqrt{m} \rfloor$ intervals with $\lfloor \sqrt{m} \rfloor$ samples

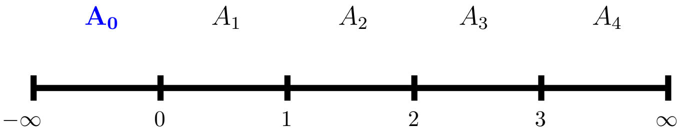
# Non-Disjoint Discretization (Example)

$\lfloor\sqrt{m}\rfloor$ intervals with $\lfloor\sqrt{m}\rfloor$ samples



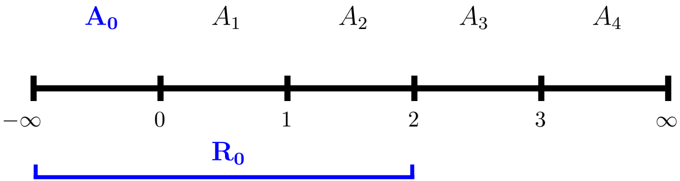
# Non-Disjoint Discretization (Example)

$\lfloor\sqrt{m}\rfloor$ intervals with $\lfloor\sqrt{m}\rfloor$ samples



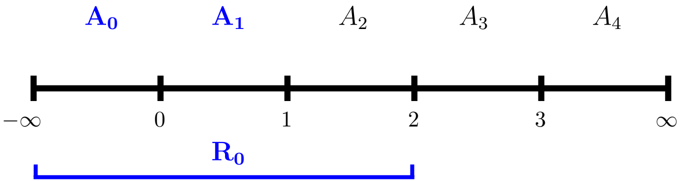
# Non-Disjoint Discretization (Example)

$\lfloor\sqrt{m}\rfloor$ intervals with $\lfloor\sqrt{m}\rfloor$ samples



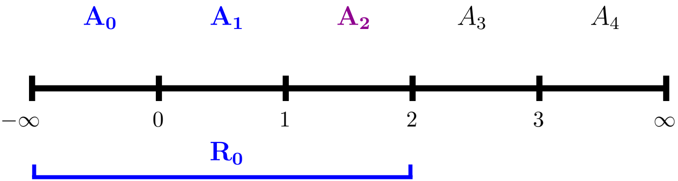
# Non-Disjoint Discretization (Example)

$\lfloor \sqrt{m} \rfloor$ intervals with $\lfloor \sqrt{m} \rfloor$ samples



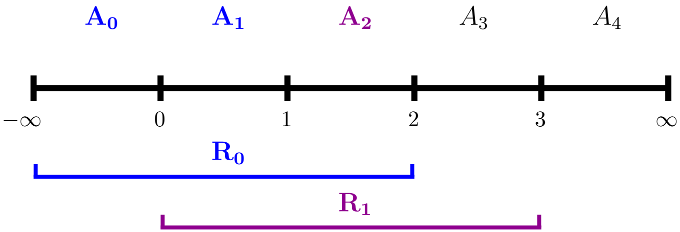
# Non-Disjoint Discretization (Example)

$\lfloor \sqrt{m} \rfloor$ intervals with $\lfloor \sqrt{m} \rfloor$ samples



# Non-Disjoint Discretization (Example)

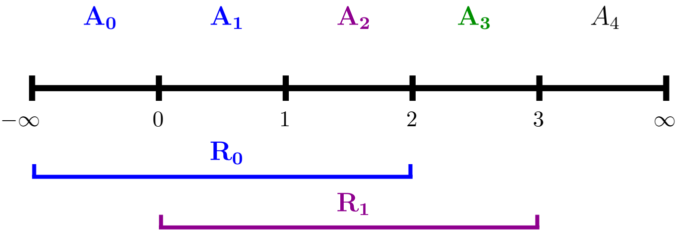
$\lfloor \sqrt{m} \rfloor$ intervals with $\lfloor \sqrt{m} \rfloor$ samples





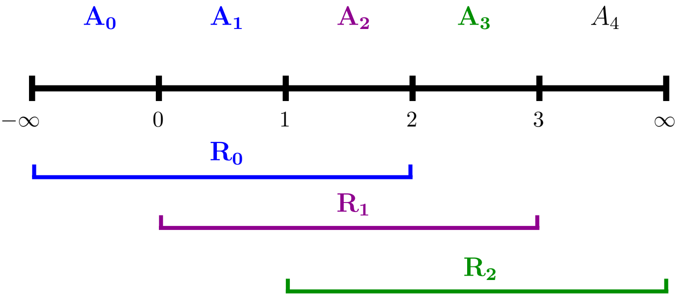

# Non-Disjoint Discretization (Example)

$\lfloor \sqrt{m} \rfloor$ intervals with $\lfloor \sqrt{m} \rfloor$ samples



# Non-Disjoint Discretization (Example)

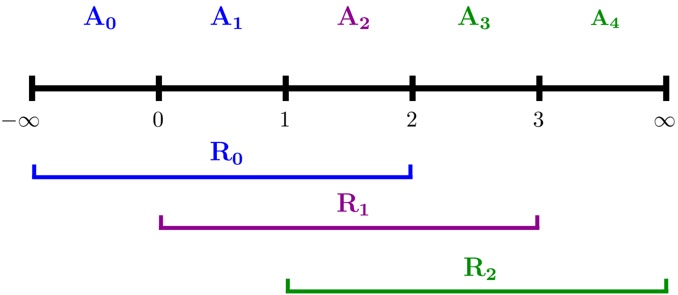
$\lfloor\sqrt{m}\rfloor$ intervals with $\lfloor\sqrt{m}\rfloor$ samples






# Non-Disjoint Discretization (Example)

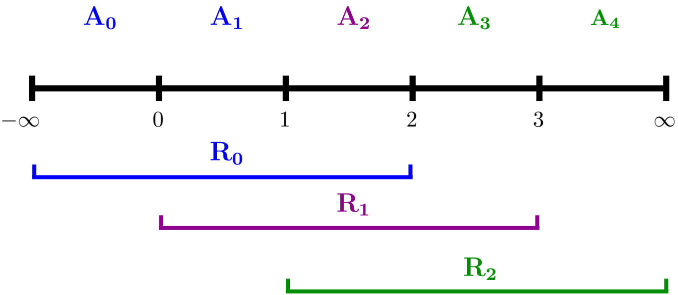
$\lfloor \sqrt{m} \rfloor$ intervals with $\lfloor \sqrt{m} \rfloor$ samples






# Non-Disjoint Discretization (Example)

$\lfloor \sqrt{m} \rfloor$ intervals with $\lfloor \sqrt{m} \rfloor$ samples






# NDD in AODE and HAODE

**MODIFICATIONS**:

# NDD in AODE and HAODE

**MODIFICATIONS**:

1. A threshold to mark the **minimum frequency** from which an atomic interval will not be merged with its neighbours.



## NDD in AODE and HAODE

**MODIFICATIONS**:

1. A threshold to mark the **minimum frequency** from which an atomic interval will not be merged with its neighbours.
2. The interval size is not equal to the interval number ($\approx \lfloor \sqrt{m} \rfloor$).



# NDD in AODE and HAODE

**MODIFICATIONS**:

1. A threshold to mark the **minimum frequency** from which an atomic interval will not be merged with its neighbours.
2. The interval size is not equal to the interval number ($\approx \lfloor \sqrt{m} \rfloor$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.



# NDD in AODE and HAODE

**MODIFICATIONS**:

1. A threshold to mark the **minimum frequency** from which an atomic interval will not be merged with its neighbours.
2. The interval size is not equal to the interval number ($\approx \lfloor\sqrt{m}\rfloor$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.
4. Weighting importance (wNDD): samples to be placed in its corresponding interval (when classification) are given more importance when training.



# Experiments

- Group of 28 hybrid datasets (same as in **[Yang & Webb, 2002]**).
- Parameters: #atomic_bins= 15 (with equal frequency)
  min_freq_not_merging= 100 (per operational int.)

**Win-Draw-Lose**

| | non-weighted | | weighted | |
|---|---|---|---|---|
| | AODE NDD vs EF5 | HAODE NDD vs EF5 | AODE NDDw vs EF5 | HAODE NDDw vs EF5 |
| Accuracy | 23-0-5 | 21-1-6 | 22-1-5 | 18-2-8 |
| Bias | 14-3-11 | 21-1-6 | 15-3-10 | 22-0-6 |
| Variance | 18-2-8 | 14-4-10 | 13-2-13 | 10-0-14 |

**Average Values**

| | AODE | | HAODE | |
|---|---|---|---|---|
| | EF5 | NDD | EF5 | NDD |
| Accuracy | 82.4169 | **83.5873** | 82.2658 | **82.4935** |
| Bias | 0.1298 | **0.1250** | 0.1348 | **0.1275** |
| Variance | 0.0395 | **0.0355** | 0.0440 | **0.0435** |



# Partial conclusions and future work

- **Main conclusion:** whereas some of the most common disjoint discretization techniques have failed to demonstrate consistent improvement relative to alternatives, NDD demonstrates better win-draw-loss records and significant overall improvement.
- Test wNDD in high-dimensional datasets (variance component should be reduced).
- NDD will be strengthened when applied to ***AnDE*** (Aggregating *n*-dependence estimators).
- Drawback of NDD: additional parameters (apart from the number of bins):
  - Number of atomic bins per operational interval.
  - Minimum frequency per interval.



# Outline

1. **Introduction**
   - Motivation
   - Objectives

2. **I New BNCs to overcome AODE's limitations**
   - Hidden One-Dependence Estimator
   - Gaussian AODE and hybrid AODE
   - The MTE-AODE classifier

3. **II Discretization techniques for semi-naive BNCs**
   - Disjoint discretization techniques
   - Non-disjoint discretization techniques

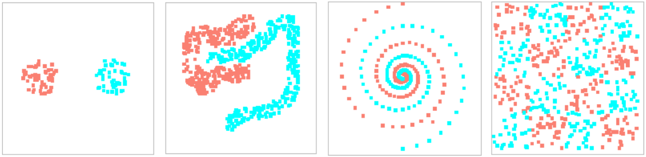
4. **III Domains of competence of the semi-naive BNCs**

5. **Concluding remarks**



# Motivation

- No clear winners for all problems.



- Can we measure geometrical complexity of classification problems?

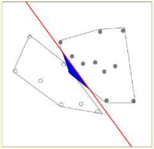


# Some Useful Measures of Geometric Complexity (Ho&Basu 2002)

## Degree of Linear Separability

Find separating hyper-plane by linear programming

Error counts and distances to plane measure separability



## Fisher's Discriminant Ratio

Classical measure of class separability

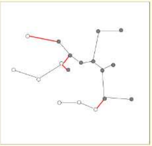
Maximize over all features to find the most discriminating

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

## Length of Class Boundary

Compute minimum spanning tree

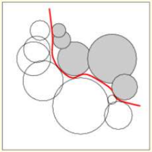
Count class-crossing edges



## Shapes of Class Manifolds

Cover same-class pts with maximal balls

Ball counts describe shape of class manifold



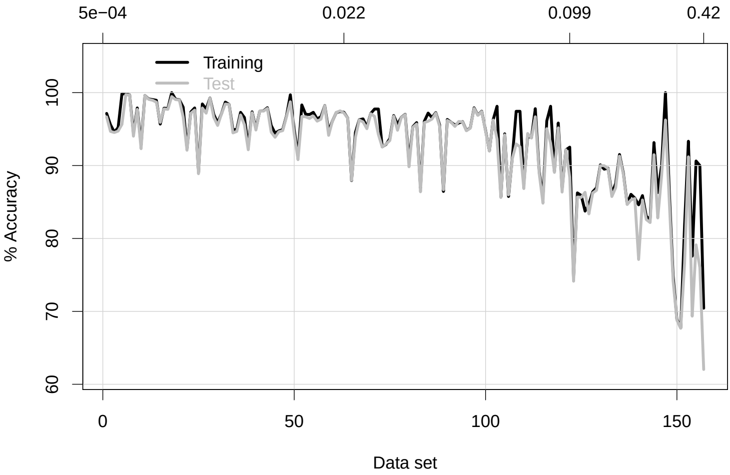
# Some Useful Measures of Geometric Complexity

| Type | Id. | Name | Simpler if... |
|---|---|---|---|
| Overlaps in the feature values from different classes | F1 | Maximum Fisher's discriminant ratio | + |
| | F1v | Directional-vector maximum Fisher's discriminant ratio | + |
| | F2 | Overlap of the per-class bounding boxes | − |
| | F3 | Maximum (individual) feature efficiency | + |
| | F4 | Collective feature efficiency | + |
| Measures of class separability | L1 | Minimized sum of the error distance of a linear classifier | − |
| | L2 | Training error of a linear classifier | − |
| | N1 | Fraction of points on the class boundary | − |
| | N2 | Ratio of average intra/inter class nearest neighbor distance | − |
| | N3 | Leave-one-out error rate of the one-nearest neighbor classifier | − |
| Measures of geometry, topology, and density of manifolds | L3 | Nonlinearity of a linear classifier | − |
| | N4 | Nonlinearity of the one-nearest neighbor classifier | − |
| | T1 | Fraction of maximum covering spheres | − |
| | T2 | Average number of points per dimension | + |



# NB: Discrete datasets organised according to L2

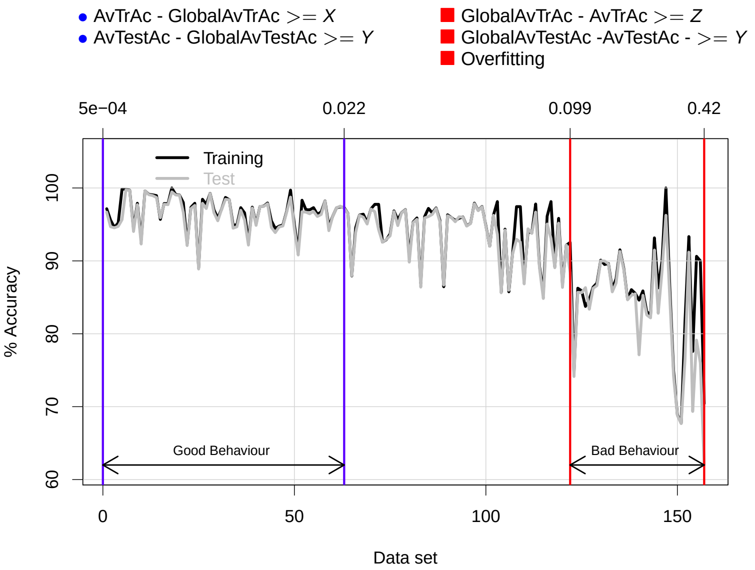
## Good pattern





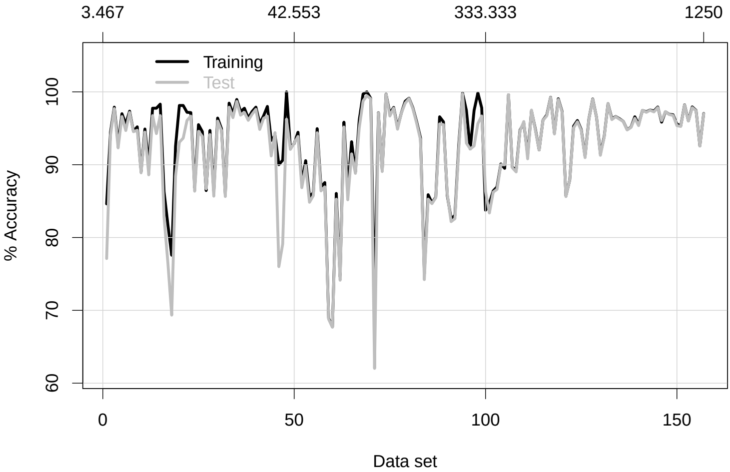

# NB: Discrete datasets organised according to L2

## Good pattern



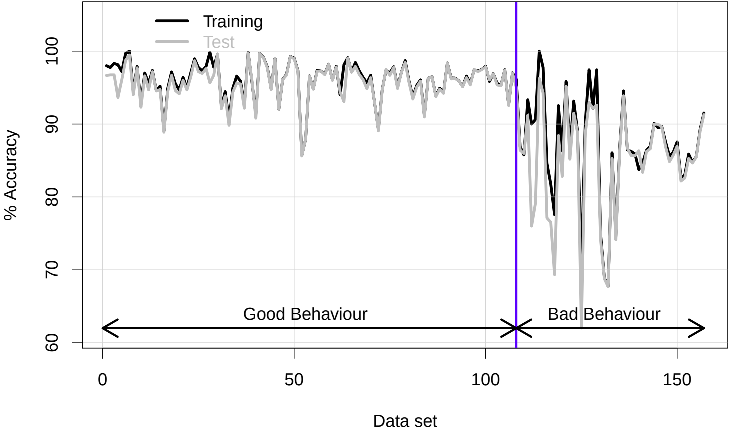
# NB: Discrete datasets organised according to T2

## Bad pattern



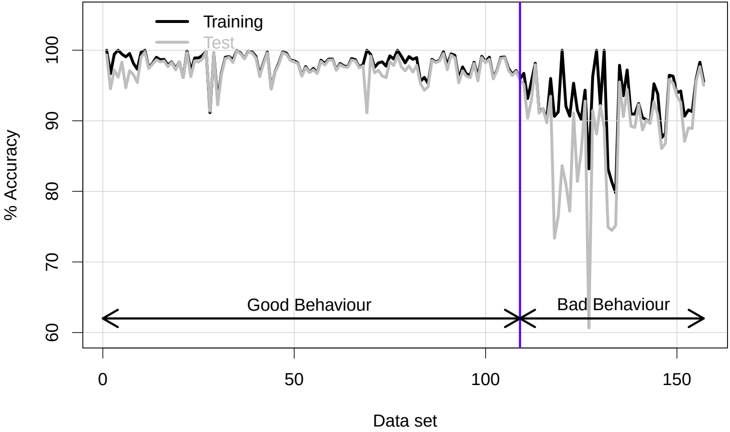



# NB characterization in terms of good and bad behaviour by the use of 6 CMs: F1v, F3, F4, L2, N1 and N3

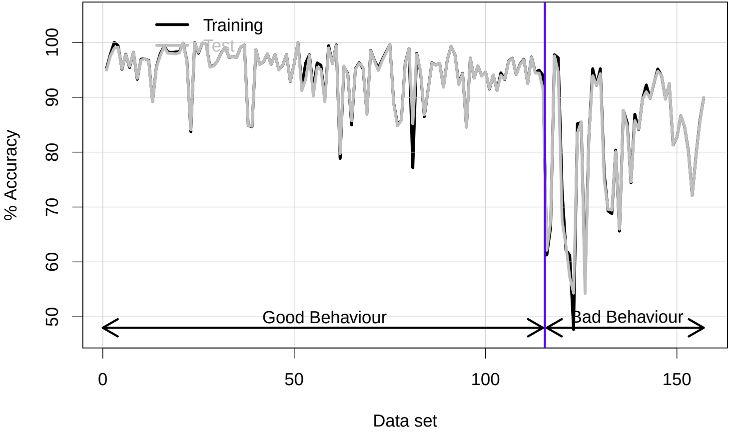




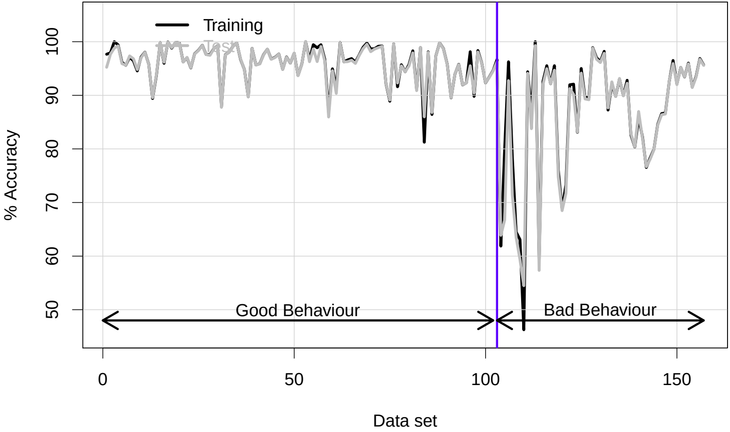

# AODE characterization in terms of good and bad behaviour by the use of 6 CMs: F1v, F3, F4, L2, N1 and N3



# Gaussian NB characterization in terms of good and bad behaviour by the use of 6 CMs: F1, L1, N1, N2, N3 and N4



# GAODE characterization in terms of good and bad behaviour by the use of 6 CMs: F1, L1, N1, N2, N3 and N4

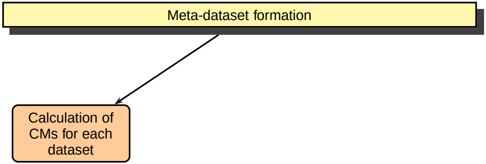


# Schema of the meta-classification process

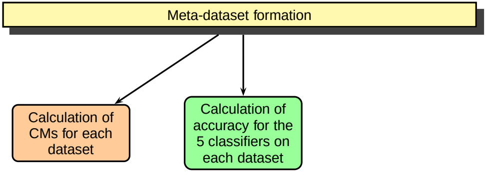
Meta-dataset formation



# Schema of the meta-classification process

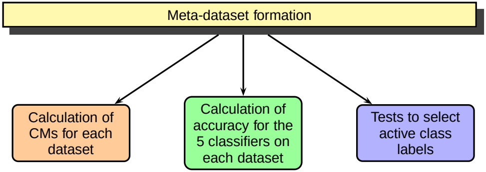


# Schema of the meta-classification process

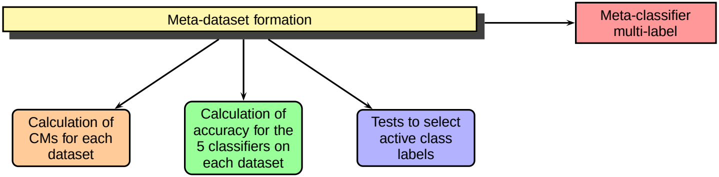




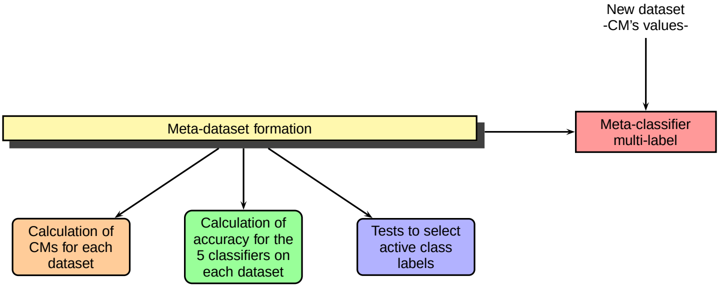

# Schema of the meta-classification process



# Schema of the meta-classification process

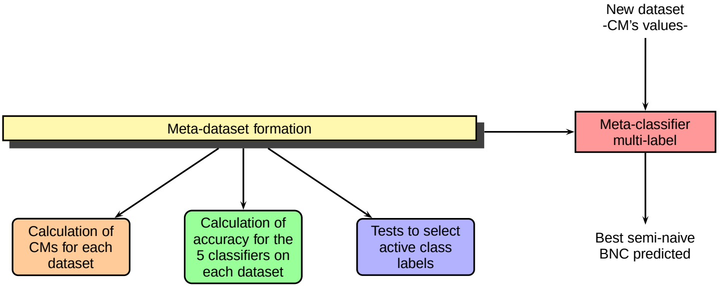


# Schema of the meta-classification process

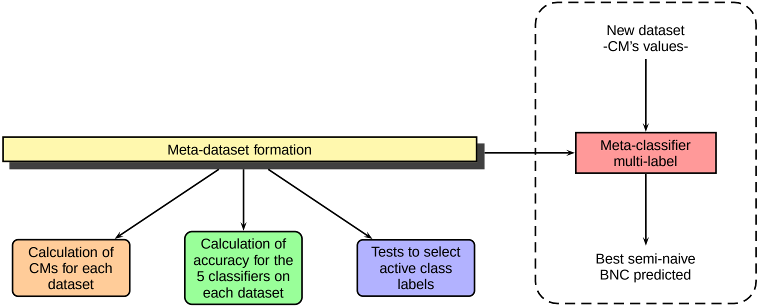





# Schema of the meta-classification process

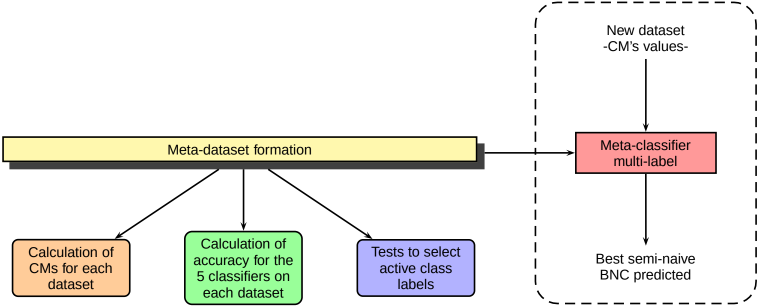


# Schema of the meta-classification process






# Schema of the meta-classification process




- Example-based precision = **87.38 %**

# Partial conclusions and future work I

- **Conclusions**
  - Easier to characterize datasets in discrete domains.
  - √ It is possible to characterize both NB and AODE for both domains and obtain disjoint rules to predict if the classifier will perform well or poorly, depending on the values of some of the complexity measures.
  - √ Automatic process to advise on the best semi-naive BNC to use for classification.
- **Future work**
  - Datasets from the Landscape contest as test bed (wider range of the complexity measurement space).
  - Theoretical way to find the suitability of a measure to characterize a particular classifier.
  - Study bivariate/multivariate relationships between the CMs.



# Outline

1 Introduction
   Motivation
   Objectives

2 I New BNCs to overcome AODE's limitations
   Hidden One-Dependence Estimator
   Gaussian AODE and hybrid AODE
   The MTE-AODE classifier

3 II Discretization techniques for semi-naive BNCs
   Disjoint discretization techniques
   Non-disjoint discretization techniques

4 III Domains of competence of the semi-naive BNCs

5 Concluding remarks



# Summary and contributions to literature

Contribution to the state of the art of semi-naive BNCs, focused on the AODE paradigm.

- New proposals to overcome AODE's limitations.
  1. HODE (**ECSQARU'09**).
  2. GAODE and HAODE (**ICML'09**,JFRB'10).
  3. MTE-AODE (**ISDA'11**).
- Impact of several discretization paradigms on the family of semi-naive BNCs:
  1. Comparison of disjoint/traditional discretization methods (**APPLIED INTELLIGENCE'11**, **IEA/AIE'10**, CAEPIA'09).
  2. Non-disjoint discretization for AODE and HAODE (**HAIS'12**).
- Domains of competence of this family of classifiers.
  1. Rules of good and bad behaviour.
  2. Meta-classification process to select the best semi-naive BNCs.
     (*submitted to* **INFORMATION SCIENCES'12**, **PGM'12**).



# Future work

1. HODE: reuse estimations on one step in the EM algorithm used in HODE on posterior steps.
2. MTE-AODE: new supervised method to dynamically search for the optimum number in every case into every dataset.
3. NDD for {H}AODE: test bed of very high dimensional datasets.
4. Domains of competence: higher dimensionality relations between complexity measures, and datasets created for the Landscape contest.
5. AnDE: in high-dimensional datasets.

- Multinets or recursive nets.
- Multi-label classification.
- Multi-instance learning.

PhD thesis:

# New models and algorithms for semi-naive Bayesian classification focused on the AODE paradigm

## Thank you for your attention

Questions, ideas or comments are welcome