Non-Disjoint Discretization for Aggregating One-Dependence Estimator Classifiers

HAIS 2012
on 29/03/2012

Ana M. Martínez
Geoffrey I. Webb, M. Julia Flores and José A. Gámez
Intelligent Systems and Data Mining's group (SIMD)
UCLM - Albacete - Spain

Outline

1. **Motivation**
2. **Background**
   - AODE classifier
   - Hybrid AODE (HAODE) classifier
3. **Disjoint vs Non-Disjoint Discretization (DD vs NDD)**
4. **{H}AODE with Non-Disjoint Discretization**
5. **Experiments**
6. **Conclusions**

Outline

1. **Motivation**
2. Background
   - AODE classifier
   - Hybrid AODE (HAODE) classifier
3. Disjoint vs Non-Disjoint Discretization (DD vs NDD)
4. {H}AODE with Non-Disjoint Discretization
5. Experiments
6. Conclusions

Ana M. Martínez

Motivation

- Lack of clarity about the best manner in which to handle numeric attributes when applying BN classifiers.
- **Discretization methods**: unavoidable *loss of information*.
- Appropriate discretization can **outperform** straightforward use of common, but often **unrealistic parametric distribution** (e.g. Gaussian).
- 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 [4].
  - Are disjoint intervals appropriate for AODE and/or HAODE?

Outline

1 Motivation

2 **Background**
   - AODE classifier
   - Hybrid AODE (HAODE) classifier

3 Disjoint vs Non-Disjoint Discretization (DD vs NDD)

4 {H}AODE with Non-Disjoint Discretization

5 Experiments

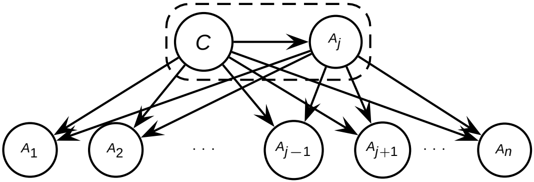
6 Conclusions

Outline

1. Motivation
2. **Background**
   - AODE classifier
   - Hybrid AODE (HAODE) classifier
3. Disjoint vs Non-Disjoint Discretization (DD vs NDD)
4. {H}AODE with Non-Disjoint Discretization
5. Experiments
6. Conclusions

AODE classifier I

- **AODE** is significantly better in terms of error reduction compared to the rest of semi-naive techniques (Zheng&Webb).



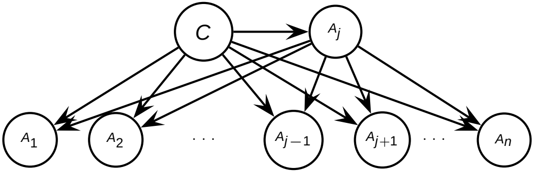
- **MAP hypothesis:**

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

Outline

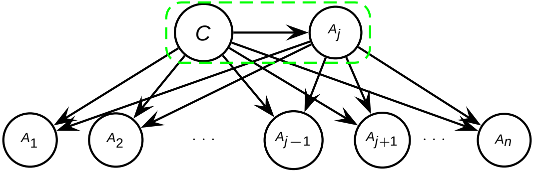
HAODE classifier

- **Discrete** superparents (A_j) in its corresponding models.



HAODE classifier

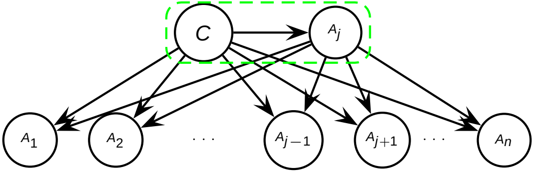
- **Discrete** superparents (A_j) in its corresponding models.



HAODE classifier

- **Discrete** superparents (A_j) in its corresponding models.

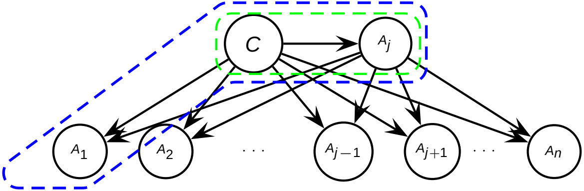
- Multinomial distribution -



HAODE classifier

- **Discrete** superparents (A_j) in its corresponding models.

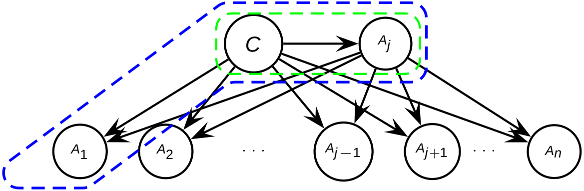
- Multinomial distribution -



HAODE classifier

- **Discrete** superparents ($\boldsymbol{A_j}$) in its corresponding models.

- Multinomial distribution -

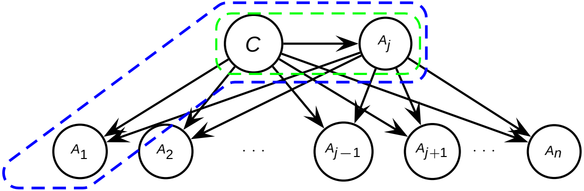


- Univariate Gaussian distribution -

HAODE classifier

- **Discrete** superparents (A_j) in its corresponding models.

- Multinomial distribution -



- Univariate Gaussian distribution -

√ Able to deal with **all kind of datasets**.

Outline

1. Motivation
2. Background
   - AODE classifier
   - Hybrid AODE (HAODE) classifier
3. **Disjoint vs Non-Disjoint Discretization (DD vs NDD)**
4. {H}AODE with Non-Disjoint Discretization
5. Experiments
6. Conclusions

Disjoint vs Non-Disjoint Discretization

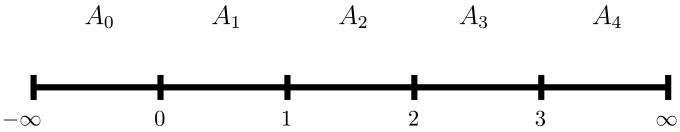
- Given $x_i, x_j \in \mathbb{R}$, any disjoint discretization (DD) method would create a **unique interval** $(a, b] \ni x_i$ and $(d, e] \ni x_j$ **for every value**.
- In DD (EF, EW, MDL): every numeric sample belongs to a single interval:
  - If x_i or x_j falls around the **center of the interval** 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)

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

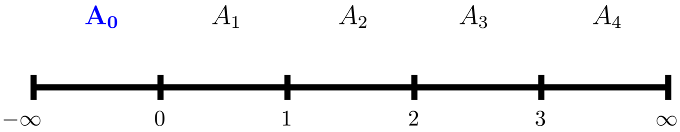
Non-Disjoint Discretization (Example)

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



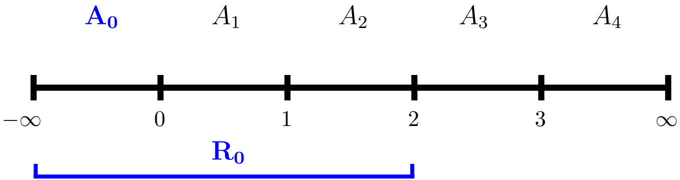
Non-Disjoint Discretization (Example)

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



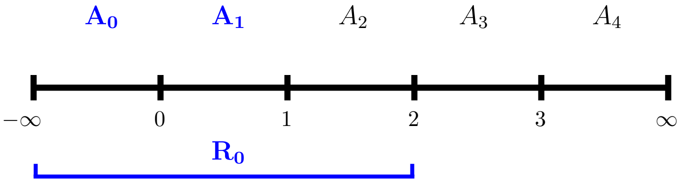
Non-Disjoint Discretization (Example)

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



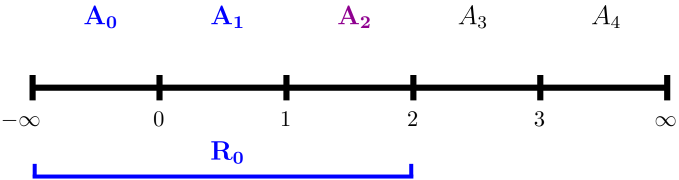
Non-Disjoint Discretization (Example)

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



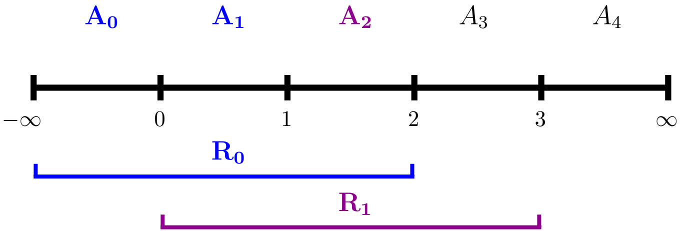
Non-Disjoint Discretization (Example)

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



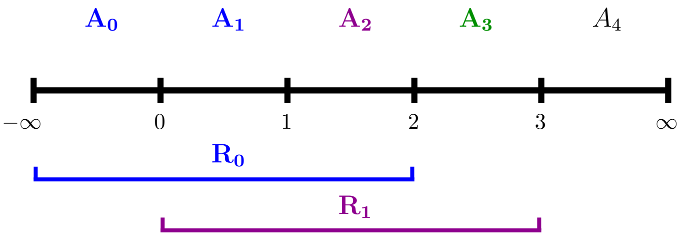
Non-Disjoint Discretization (Example)

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



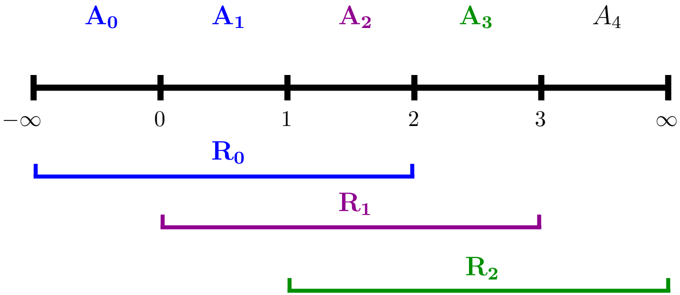
Non-Disjoint Discretization (Example)

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



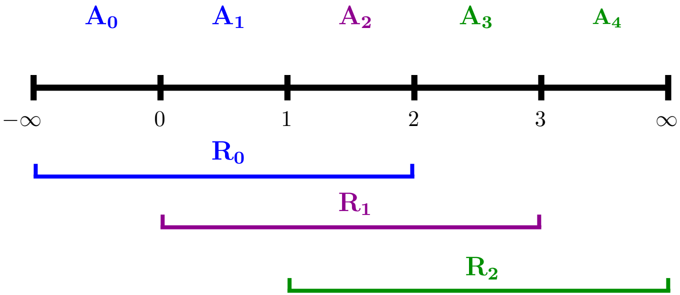
Non-Disjoint Discretization (Example)

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



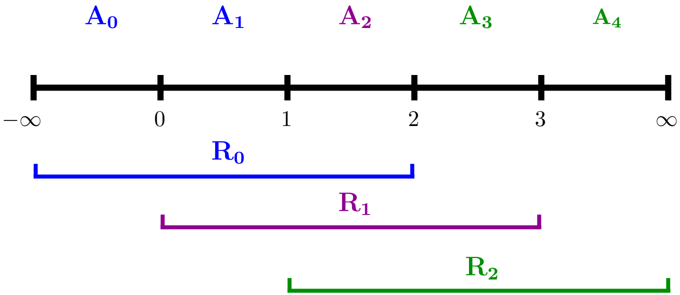
Non-Disjoint Discretization (Example)

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



Non-Disjoint Discretization (Example)

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



Outline

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

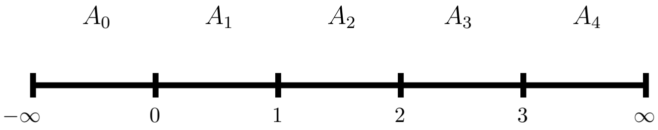
MODIFICATIONS:

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

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.

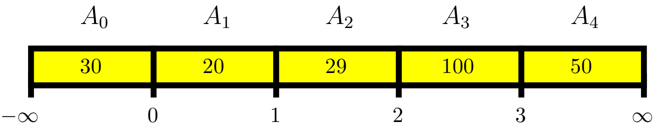


NDD in AODE and HAODE

MODIFICATIONS:

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

$$\mathbf{minFreq} = \mathbf{100}$$

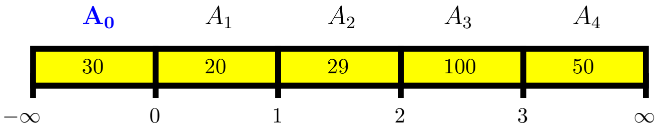


NDD in AODE and HAODE

MODIFICATIONS:

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

$$\mathbf{minFreq} = \mathbf{100}$$

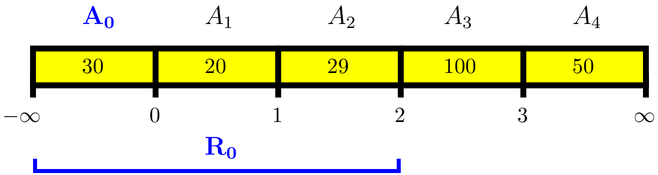


NDD in AODE and HAODE

MODIFICATIONS:

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

$$\mathbf{minFreq} = \mathbf{100}$$

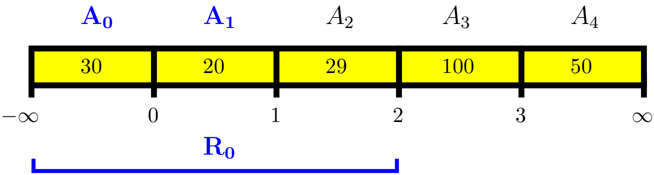


NDD in AODE and HAODE

MODIFICATIONS:

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

$$\mathbf{minFreq} = \mathbf{100}$$

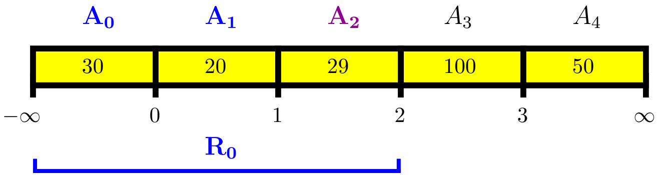


NDD in AODE and HAODE

MODIFICATIONS:

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

$\mathbf{minFreq} = \mathbf{100}$

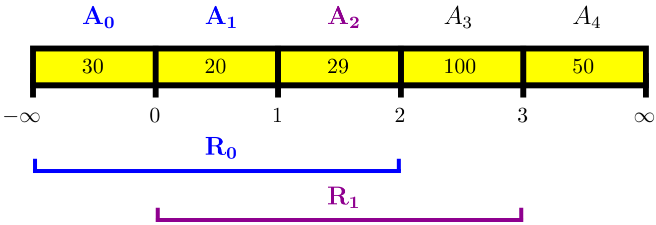


NDD in AODE and HAODE

MODIFICATIONS:

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

$$\mathbf{minFreq} = \mathbf{100}$$

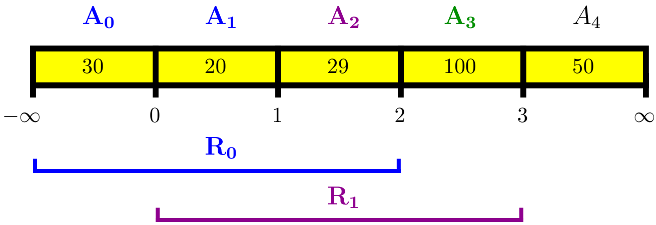


NDD in AODE and HAODE

MODIFICATIONS:

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

$$\mathbf{minFreq} = \mathbf{100}$$

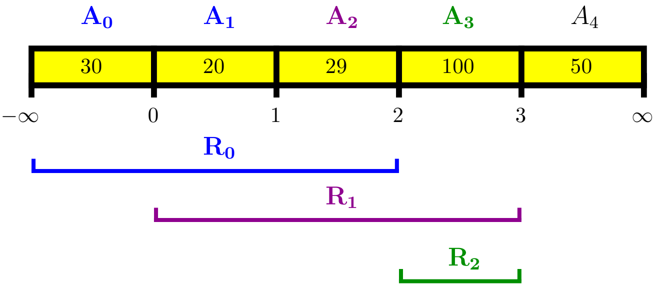


NDD in AODE and HAODE

MODIFICATIONS:

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

$$\mathbf{minFreq} = \mathbf{100}$$

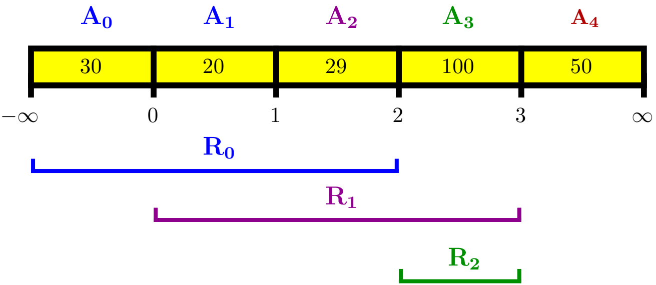


NDD in AODE and HAODE

MODIFICATIONS:

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

$$\mathbf{minFreq} = \mathbf{100}$$

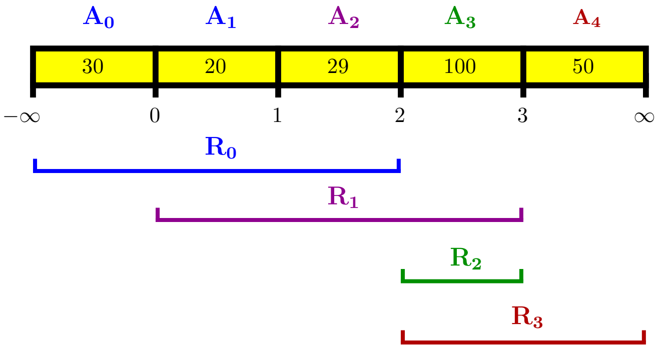


NDD in AODE and HAODE

MODIFICATIONS:

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

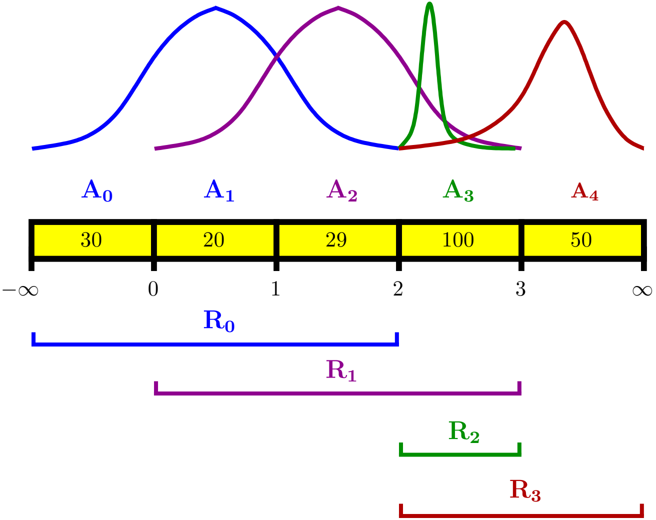
$$\mathbf{minFreq} = \mathbf{100}$$



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.



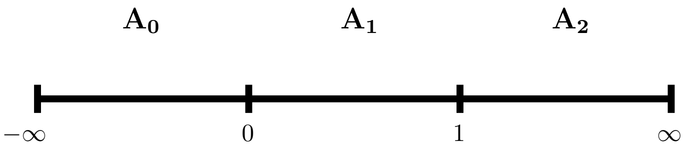
NDD in {H}AODE MODIFICATIONS

2. The interval size is not equal to the interval number ($\approx \sqrt{m}$).

MODIFICATIONS

2. The interval size is not equal to the interval number ($\approx \sqrt{m}$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.

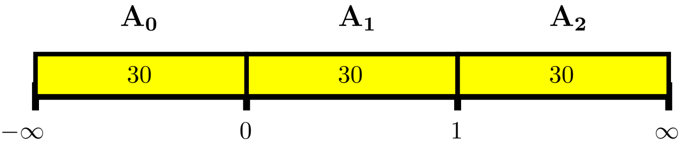
2. The interval size is not equal to the interval number ($\approx \sqrt{m}$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.



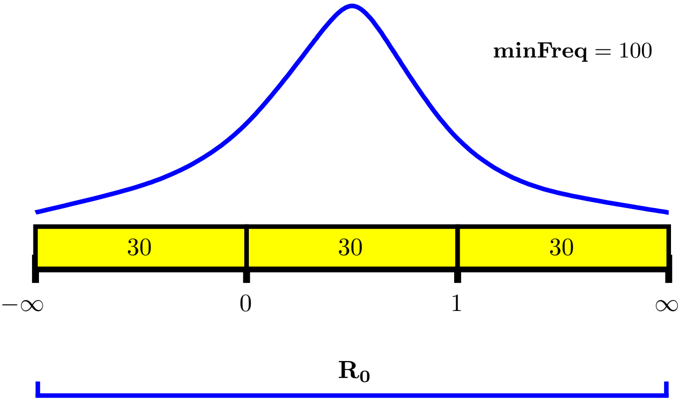
MODIFICATIONS

2. The interval size is not equal to the interval number ($\approx \sqrt{m}$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.

$$\mathbf{minFreq} = 100$$

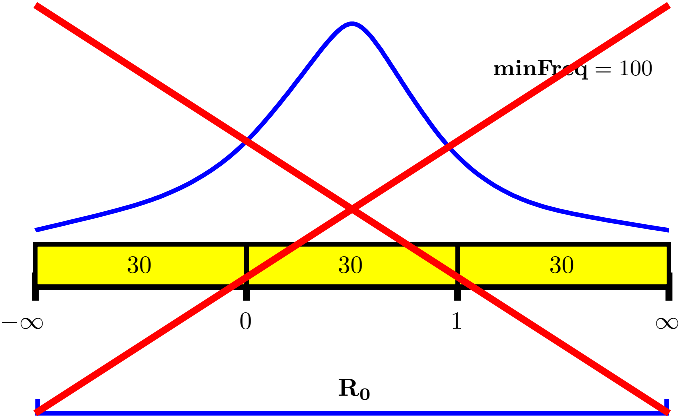


2. The interval size is not equal to the interval number ($\approx \sqrt{m}$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.



MODIFICATIONS

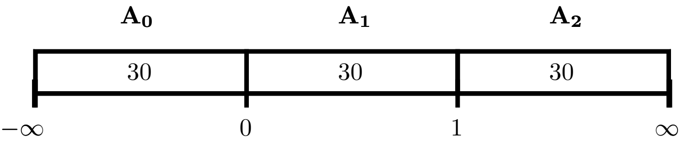
2. The interval size is not equal to the interval number ($\approx \sqrt{m}$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.



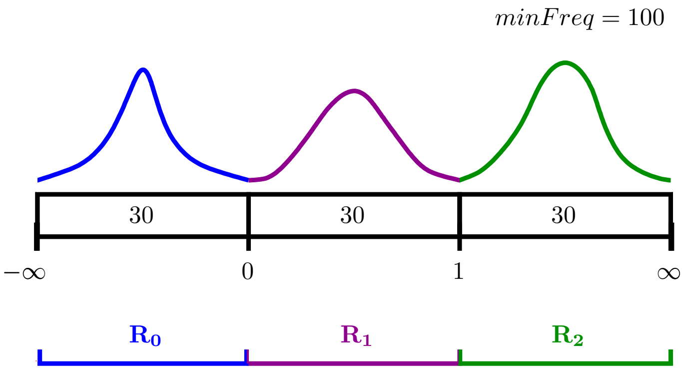
MODIFICATIONS

2. The interval size is not equal to the interval number ($\approx \sqrt{m}$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.

$$minFreq = 100$$



2. The interval size is not equal to the interval number ($\approx \sqrt{m}$).
3. When the number of **cut-points is lower than 3** $\rightarrow$ **Equal Frequency** discretization will be kept.



NDD in {H}AODE MODIFICATIONS

MODIFICATIONS

4. Weighting importance: samples to be placed in its corresponding interval (when classification) are given more importance when training.

4. Weighting importance: samples to be placed in its corresponding interval (when classification) are given more importance when training.

	1 interval	2 intervals	3 intervals
w_1	1	0.75	0.75
w_2		0.25	0.125
w_3			0.125

$I = \{A_0 = 3.5, A_1 = 2, C = c_1\}$ → NDD → $I_{NDD} = \{A_0 = (L_0, L_1, L_2), A_1 = (L'_0, L'_1), C = c_1\}$

$L_0 : w_0 = 0.125$ $L'_0 : w'_0 = 0.25$
$L_1 : w_1 = 0.75$ $L'_1 : w'_1 = 0.75$
$L_2 : w_2 = 0.125$

When updating $\mathbf{p(A_0|A_1, C)}$:

- $\{L_0, L'_0, c_1\}$ $w = w_0 * w'_0 = 0.125 * 0.25$
- $\{L_0, L'_1, c_1\}$ $w = w_0 * w'_1 = 0.125 * 0.75$
- $\{L_1, L'_0, c_1\}$ $w = w_1 * w'_0 = 0.75 * 0.25$
- $\{L_1, L'_1, c_1\}$ $w = w_1 * w'_1 = 0.75 * 0.75$
- $\{L_2, L'_0, c_1\}$ $w = w_2 * w'_0 = 0.125 * 0.25$
- $\{L_2, L'_1, c_1\}$ $w = w_2 * w'_1 = 0.125 * 0.75$

$\sum_w = 1$

When classifying I:

$I'_{NDD} = \{A_0 = L_1, A_1 = L'_1, C = ?\}$

Outline

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:** supposing there is a fixed dataset size, the larger the number of intervals, the smaller the number of instances per interval is.

Experiments

- Group of 28 hibrid datasets (same as in [4]).
- 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
Error	21-1-6	19-3-6	16-3-9	18-1-9

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**
Error	0.1737	**0.1643**	0.1836	**0.1758**

Outline

1. Motivation
2. Background
   - AODE classifier
   - Hybrid AODE (HAODE) classifier
3. Disjoint vs Non-Disjoint Discretization (DD vs NDD)
4. {H}AODE with Non-Disjoint Discretization
5. Experiments
6. **Conclusions**

Conclusions

- **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.
- NDD will be strengthened when applied to ***AnDE*** (Aggregating *n*-dependence estimators).
- Test wNDD in high-dimensional datasets (variance component should be reduced).
- Drawback of NDD: additional parameters (apart from the number of bins):
  - Number of atomic bins per operational interval.
  - Minimum frequency per interval.

Thank you

Questions, ideas or comments are welcome :)

References I

- Zheng, F., Webb, G.I.: A Comparative Study of Semi-naive Bayes Methods in Classification Learning. In: 4th Australasian Data Mining Conference (AusDM05), Simoff, S.J, Williams, G.J, Galloway, J., Kolyshkina, I. editors, pp. 141–156. University of Technology, Sydney (2005)
- Flores, M. J., Gámez, J. A., Martínez, A. M., Puerta J. M.: GAODE and HAODE: Two Proposals based on AODE to Deal with Continuous Variables. In: 26th International Conference on Machine Learning: 40. Montreal, Canada (2009)
- Webb, G. I., Boughton, J. R., Wang, Z.: Not So Naive Bayes: Aggregating One-Dependence Estimators. J. Mach. Learn. 58 (1), 5-24 (2005)
- Yang Y., Webb G. Non-Disjoint Discretization for Naive-Bayes Classifiers. Proceedings of the 19th International Conference on Machine Learning (ICML-2002), 666-673. (2002).