2025 International Conference on Machine Learning

CUG-Miner

Instance Correlation Graph-based Naive Bayes

Chengyuan Li [1]; Liangxiao Jiang [1*]; Wenjun Zhang [1]; Liangjun Yu [2]; Huan Zhang [3]

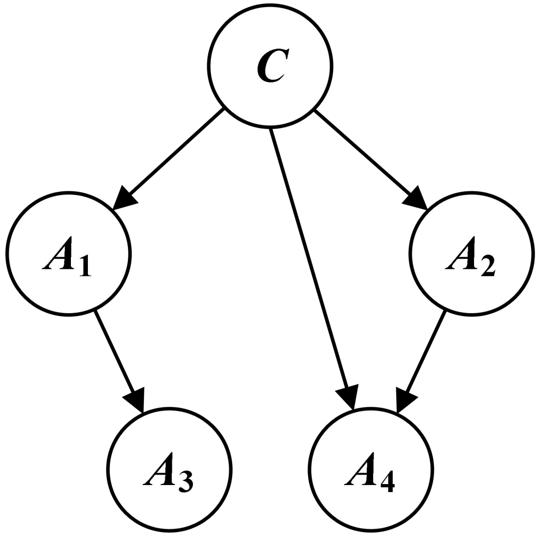
[1] School of Computer Science, China University of Geosciences, Wuhan 430074, China
[2] College of Computer, Hubei University of Education, Wuhan 430074, China
[3] School of Computer Science and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China

Background

- Supervised classification is one of the most fundamental and significant tasks in data mining.

- **Bayesian network** is commonly used in supervised classification. Its classification equation is:

$$\hat{c}(x) = \arg\max_{c \in C} \pi_c P(a_1, a_2, \ldots, a_j, \ldots, a_m \mid c).$$



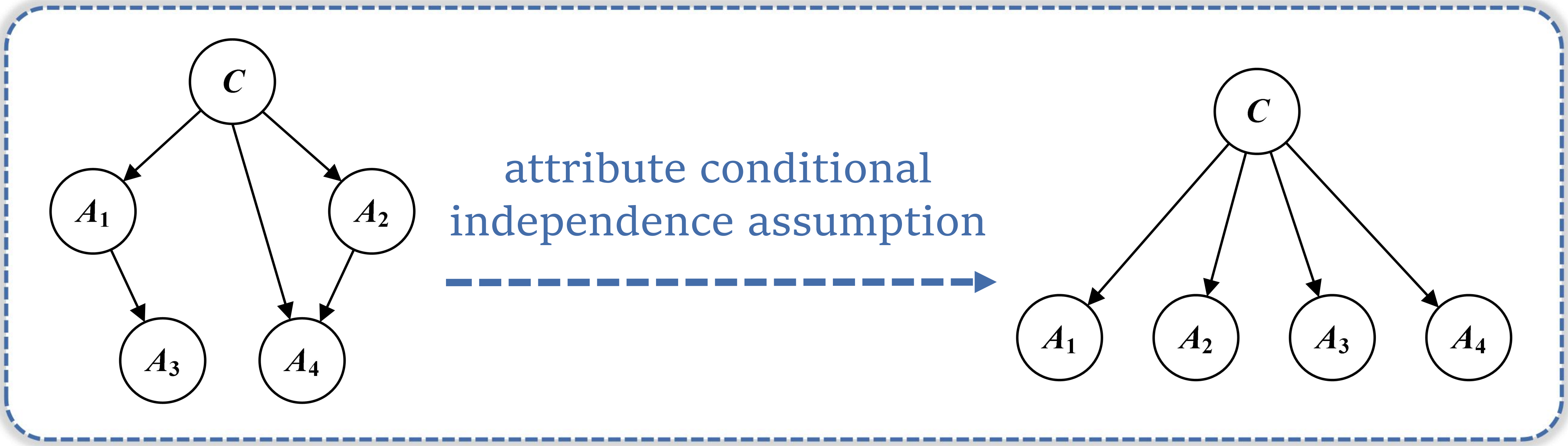
- Directly estimating the conditional probability $P(a_1, a_2, \ldots, a_j, \ldots, a_m|c)$ is an NP-hard problem.

Background

CUG-Miner

- **Naive Bayes (NB)** simplifies the estimation by leveraging an assumption that all attributes are fully independent given the class, i.e. attribute conditional independence assumption.

- The classification equation of NB is: $\hat{c}(x) = \underset{c \in C}{arg\ max}\ \pi_c \prod_{j=1}^{m} \theta_{a_j|c}$



Motivations

CUG-Miner

- In additiond to Gaussian naive Bayes (GNB), there is little work of NB focusing on numerical attributes.

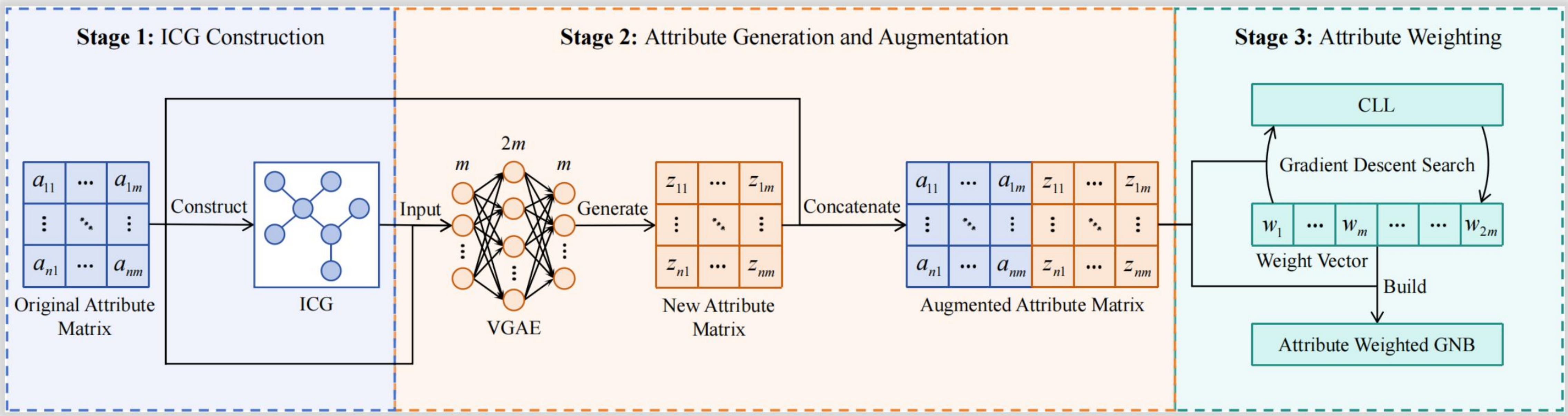
- There is no improved algoithm of NB takes into account the correlations among instances.

Method

CUG-Miner

◆ To address these two existing issues, we propose an instance correlation graph-based naive Bayes (**ICGNB**).

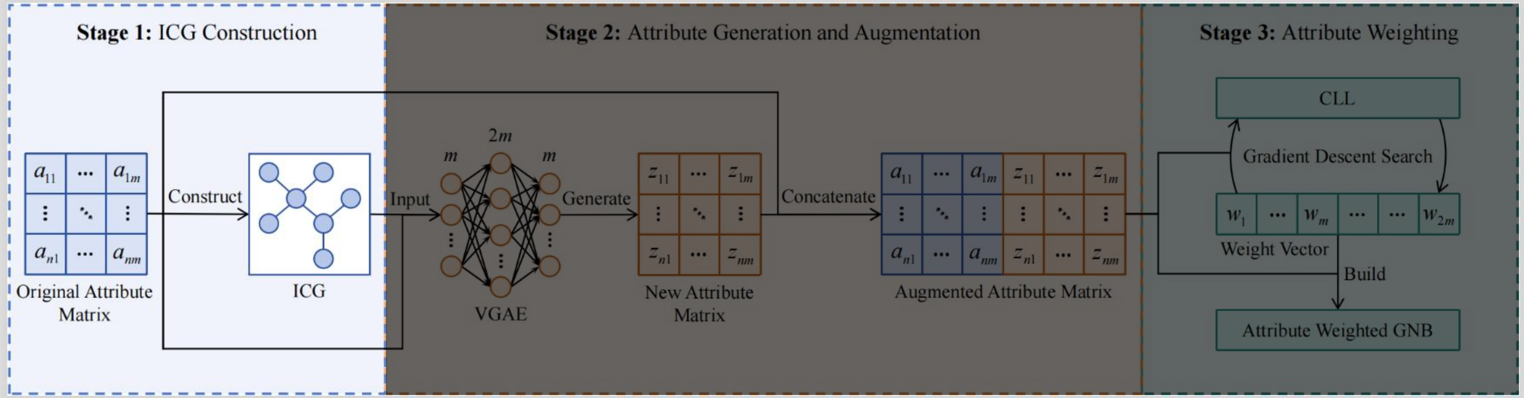
◆ As seen from the framework below, ICGNB consists of three stages:



Framework of ICGNB.

Method

◆ In Stage 1, we mine the correlations among instances from the original attribute matrix and construct an instance correlation graph (ICG) to capture the correlations.



Framework of ICGNB.

Algorithm 1 ICG-construction($\boldsymbol{X}$)

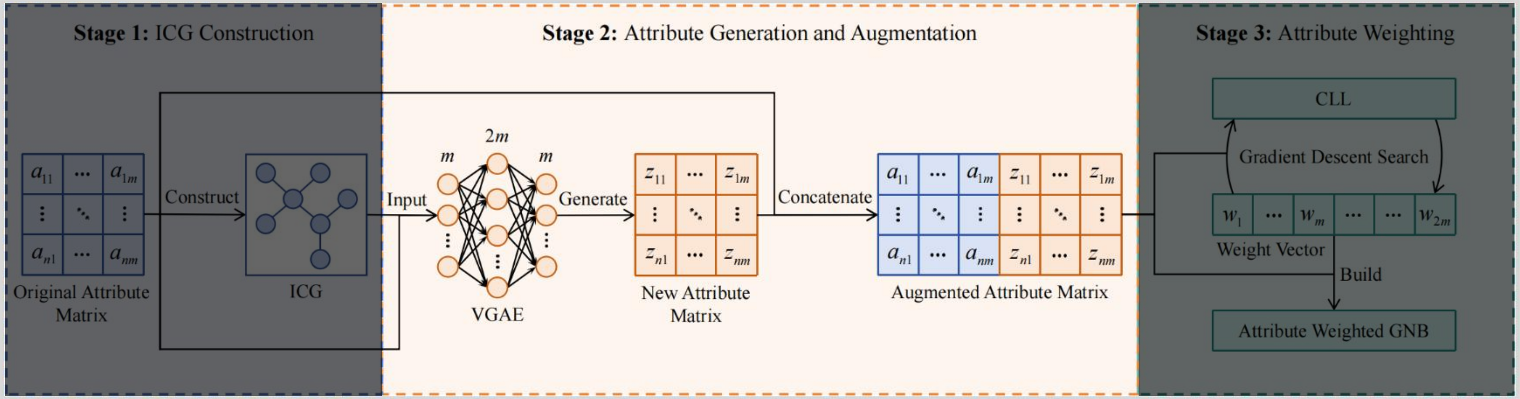
Input: $\boldsymbol{X}$ - the original attribute matrix.
Output: E - the set of edges in ICG.
Construct a full connection graph of instances and store its edges in E_F;
for i = 1 to n **do**
  for t = 1 to n **do**
    Calculate $d(\boldsymbol{x}_i, \boldsymbol{x}_t)$ between $\boldsymbol{x}_i$ and $\boldsymbol{x}_t$ by Eq. (3);
  end for
end for
Sort edges in E_F by Euclidean distances in ascending order;
Initialize an empty set E;
for i = 1 to $n(n-1)/2$ **do**
  if two vertices connected by the i-th edge in E_F are not reachable through the edges in E **then**
    Add the the i-th edge in E_F to E;
    if E contains n-1 edges **then**
      Break;
    end if
  end if
end for
for i = 1 to n **do**
  Add a self-connecting edge for the i-th vertex to E;
end for
return E.

$$d(\boldsymbol{x}_i, \boldsymbol{x}_t) = \sqrt{\sum_{j=1}^{m} (a_{ij} - a_{tj})^2}$$

Method

CUG-Miner

◆ In Stage 2, we input ICG and the original attribute matrix into variational graph auto-encoder (VGAE) to generate a new attribute matrix and augment the original attribute matrix by it.



Framework of ICGNB.

◆ Encoder:

$$q(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{G}) = \prod_{i=1}^{n} \mathcal{M}(\boldsymbol{z}_i|\phi_i)$$

◆ Convolution:

$$\boldsymbol{H}^{r+1} = \delta(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{G}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{r}\boldsymbol{W}^{r})$$

$$\boldsymbol{\mu} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{G}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{W}_{\boldsymbol{\mu}}$$

$$\boldsymbol{\sigma} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{G}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{W}_{\boldsymbol{\sigma}}$$

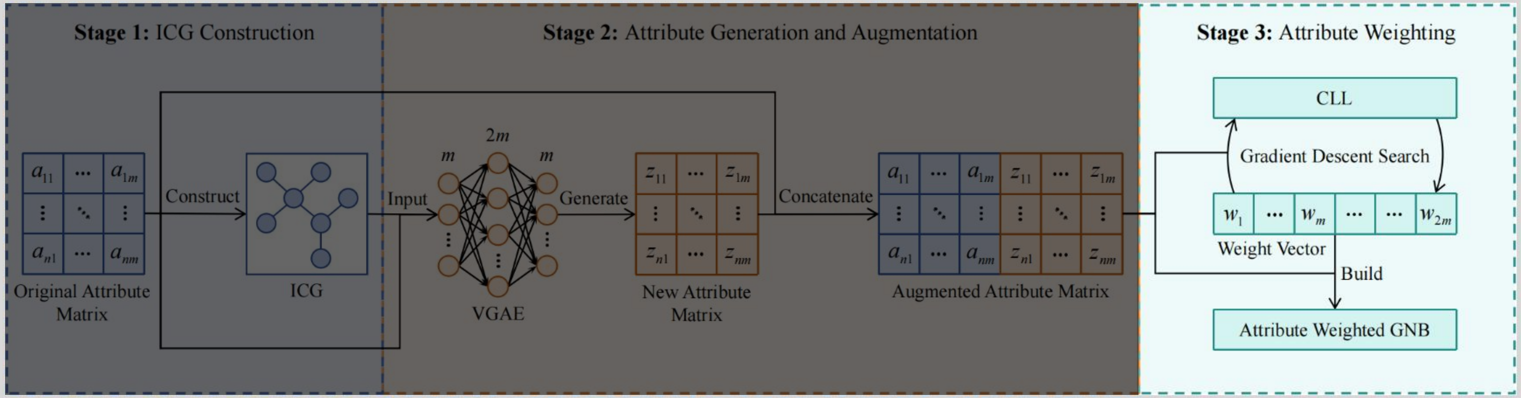
◆ Decoder:

$$p(\boldsymbol{G}|\boldsymbol{Z}) = \prod_{i=1}^{n}\prod_{j=1}^{n} \zeta(\boldsymbol{z}_i^T \boldsymbol{z}_j)$$

Method

CUG-Miner

◆ In Stage 3, we maximize the CLL by the gradient descent search to optimize the weight vector and finally build attribute weighted GNB on augmented attributes.



Framework of ICGNB.

◆ Classfication:

$$\hat{c}(\boldsymbol{x}) = \underset{c \in C}{arg\ max}\, \pi_c \prod_{j=1}^{2m} \theta_{a_j|c}^{w_j}$$

◆ Gradient:

$$\frac{\partial}{\partial w_j}\mathrm{CLL}(\boldsymbol{w})$$

$$= \frac{\partial}{\partial w_j} \sum_{i=1}^{n_t} \left(\log\left(\gamma_{c_i \boldsymbol{x}_i}(\boldsymbol{w})\right) - \log\left(\sum_{c=1}^{k} \gamma_{c\boldsymbol{x}_i}(\boldsymbol{w}) \right) \right)$$

$$= \sum_{i=1}^{n_t} \left(\frac{\gamma_{c_i \boldsymbol{x}_i}(\boldsymbol{w}) \log(\theta_{a_j|c_i})}{\gamma_{c_i \boldsymbol{x}_i}(\boldsymbol{w})} - \frac{\sum_{c=1}^{k} \gamma_{c\boldsymbol{x}_i}(\boldsymbol{w}) \log(\theta_{a_j|c})}{\sum_{c=1}^{k} \gamma_{c\boldsymbol{x}_i}(\boldsymbol{w})} \right)$$

$$= \sum_{i=1}^{n_t} \left(\log(\theta_{a_j|c_i}) - \sum_{c=1}^{k} \hat{P}(c|\boldsymbol{x}_i, \boldsymbol{w}) \log(\theta_{a_j|c}) \right).$$

Method

CUG-Miner

◆ The process of ICGNB:

ICGNB-training

ICGNB-classification

Algorithm 2 ICGNB-training($\mathcal{D}$)

Input: $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{c}\}$ - the dataset.
Output: $\boldsymbol{Z}$ - the new attribute matrix, $\boldsymbol{w}$ - the weight vector.
Construct ICG by **Algorithm 1**;
Transform ICG into an adjacency matrix $\boldsymbol{G}$;
Initialize the parameters of VGAE as $\mathcal{W}_1$;
for $p = 1$ to P **do**
  Generate the embedding matrix by Eq. (4) with $\mathcal{W}_p$;
  Reconstruct the original adjacency matrix by Eq. (9);
  Calculate the loss by Eq. (10);
  Update the parameters by Eq. (11);
end for
Generate the new attribute matrix $\boldsymbol{Z}$ by Eq. (4) with $\mathcal{W}_{P+1}$;
for $i = 1$ to n_t **do**
  for $j = 1$ to m **do**
    Define the z_{ij} as the $(m+j)$-th attribute value of $\boldsymbol{x}_i$;
  end for
end for
for $c = 1$ to k **do**
  Estimate the prior probability π_c by Eq. (13);
  for $j = 1$ to $2m$ **do**
    Estimate the conditional probability $\theta_{a_j|c}$ by Eq. (14);
  end for
end for
Initialize each weight in the weight vector $\boldsymbol{w}$;
Optimize the initialized weight vector $\boldsymbol{w}$ by Eqs. (15) - (19);
return $\boldsymbol{Z}$, $\boldsymbol{w}$.

Algorithm 3 ICGNB-classification($\boldsymbol{Z}$, $\boldsymbol{w}$, $\boldsymbol{x}$)

Input: $\boldsymbol{Z}$ - the new attribute matrix, $\boldsymbol{w}$ - the weight vector, $\boldsymbol{x}$ - a test instance.
Output: $\hat{c}(\boldsymbol{x})$ - the predicted class label of $\boldsymbol{x}$.
Extract the embedding vector $\boldsymbol{z}$ corresponding to $\boldsymbol{x}$ from $\boldsymbol{Z}$;
for $j = 1$ to m **do**
  Define z_{ij} as the $(m+j)$-th attribute value of $\boldsymbol{x}$;
end for
for $c = 1$ to k **do**
  Estimate the prior probability π_c by Eq. (13);
  for $j = 1$ to $2m$ **do**
    Estimate the conditional probability $\theta_{a_j|c}$ by Eq. (14);
  end for
end for
Predict the class label $\hat{c}(\boldsymbol{x})$ of $\boldsymbol{x}$ by Eq. (12);
return $\hat{c}(\boldsymbol{x})$.

Experiments on real-world datasets

CUG-Miner

◆ Classification accuracy

ICGNB is the best among all the competitors.

Dataset	ICGNB	WANBIA	CFWNB	AG-NBC	AE-NBC	GNB
appendicitis	89.09±7.66	88.18±7.66	**90.00±6.03**	83.64±6.80	83.18±10.77	86.36±6.10
balance	88.00±3.12	90.24±2.84	88.32±3.87	**91.68±1.53**	79.84±5.22	90.24±2.84
banana	69.53±0.80	62.00±0.91	59.74±1.48	**84.80±3.83**	62.41±3.19	62.00±0.91
cleveland	57.67±4.84	**58.00±4.27**	54.83±4.86	53.67±3.86	55.50±7.82	51.67±10.22
ecoli	**79.12±5.46**	79.12±6.06	60.29±11.60	70.00±6.35	78.53±2.81	60.74±6.61
glass	**63.02±8.85**	59.07±7.44	51.40±11.74	60.70±6.61	60.47±8.06	47.21±9.70
iris	**96.33±3.14**	96.00±3.27	95.33±3.71	90.33±6.90	91.00±7.31	95.33±3.71
led7digit-01	**72.40±4.27**	70.40±5.90	64.20±9.11	71.30±4.73	71.70±3.26	63.30±12.12
magic	**78.38±0.78**	77.05±0.67	74.56±0.56	75.04±1.33	77.69±1.35	72.56±0.64
movement_libras	56.11±4.31	62.92±4.97	62.22±4.76	69.17±6.77	**70.97±6.38**	61.94±5.63
phoneme	76.46±1.10	75.91±1.40	76.85±1.58	**77.22±1.69**	76.91±1.62	75.97±1.65
pima	75.32±2.29	**75.52±2.69**	75.06±2.99	73.12±2.23	73.70±2.94	74.61±3.45
ring	**97.98±0.36**	97.90±0.20	97.96±0.30	93.34±1.18	94.73±0.51	97.92±0.28
segment	**90.41±1.59**	88.81±1.26	80.52±1.26	88.01±1.76	83.35±2.60	79.42±1.48
sonar	**78.57±5.73**	78.33±5.05	67.86±5.46	76.67±7.81	67.15±8.05	66.67±5.11
spambase	**90.67±1.08**	89.99±1.08	83.71±1.26	86.52±1.83	86.51±0.98	82.08±1.25
texture	**96.84±0.52**	84.47±1.00	78.35±1.38	94.91±0.78	94.22±0.74	77.45±1.39
titanic	77.41±1.19	**77.64±1.21**	76.98±0.89	75.51±1.70	76.98±0.89	76.98±0.89
twonorm	97.66±0.23	**97.72±0.27**	97.71±0.29	96.37±0.61	95.31±0.91	97.70±0.28
wdbc	**96.32±1.51**	96.40±1.54	93.95±1.90	94.65±1.69	85.53±2.97	92.98±2.29
wine	96.94±1.94	97.50±1.50	96.94±1.94	**97.50±1.94**	92.22±5.53	97.50±1.50
winequality-red	**59.53±2.56**	58.44±1.78	58.47±1.49	58.84±3.19	57.16±2.93	54.72±2.56
winequality-white	**52.64±1.27**	52.21±1.51	49.23±1.14	51.02±1.94	51.51±1.43	44.38±1.61
yeast	**56.53±3.26**	54.28±3.20	18.22±3.83	50.03±3.07	55.49±1.87	14.41±3.38
(W / T / L)		17/0/7	18/1/5	19/0/5	22/0/2	20/0/4
Average	78.87	77.84	73.03	77.67	75.92	71.84

Classification accuracy (%) comparisons for ICGNB versus its competitors.

Experiments on real-world datasets

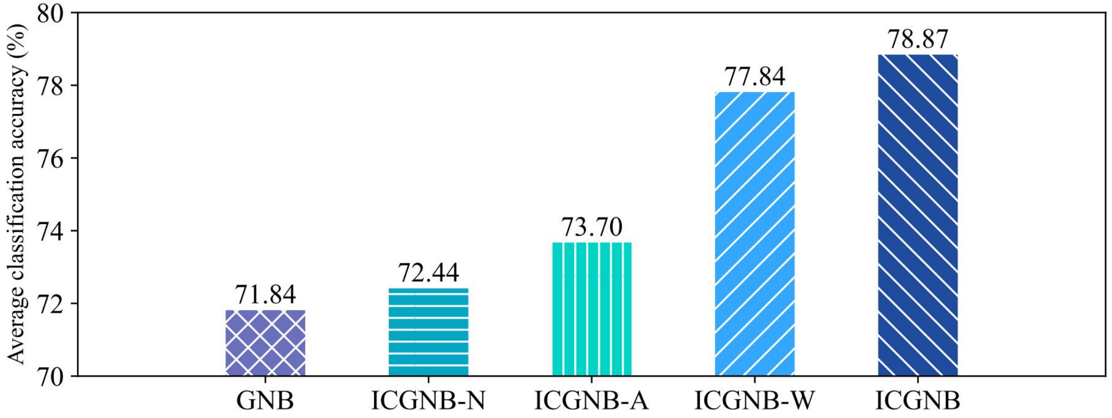
CUG-Miner

◆ Wilcoxon tests ICGNB significantly outperforms all the competitors.

Algorithm	ICGNB	WANBIA	CFWNB	AG-NBC	AE-NBC	GNB
ICGNB	-	○	○	○	○	○
WANBIA	●	-	○		○	○
CFWNB	●	●	-	●		○
AG-NBC	●		○	-		○
AE-NBC	●	●			-	○
GNB	●	●	●	●		-

Wilcoxon tests for ICGNB versus its competitors.

◆ Ablation study Each part in ICGNB is necessary.



Variant	Generation	Augmentation	Weighting
ICGNB-N	√	×	×
ICGNB-A	√	√	×
ICGNB-W	×	×	√

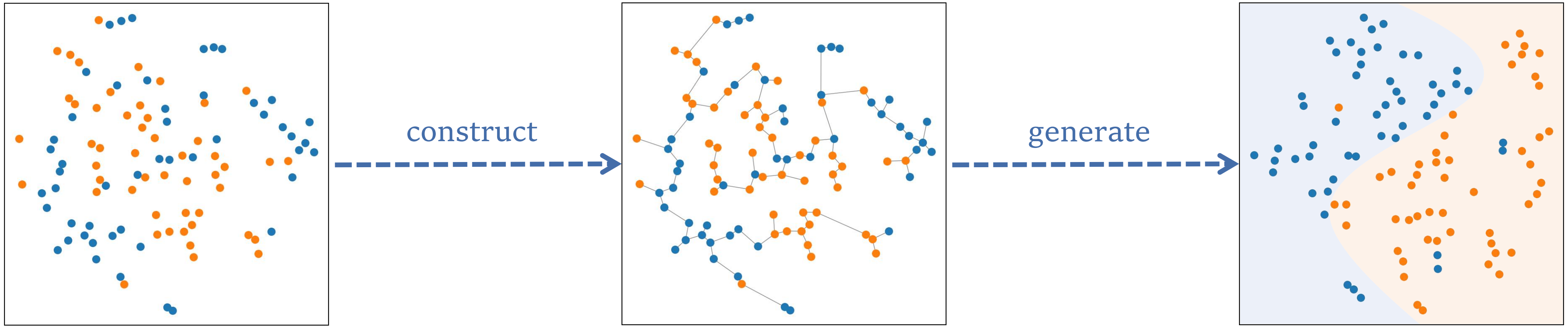
Experiments on the synthetic dataset

CUG-Miner

◆ Effectiveness

In Figure (b), ICG effectively connects instances of the same class.

The class distribution in Figure (c) demonstrates distinguishability compared to that with original attributes in Figure (a), in which instances of different classes are scattered.



(a) original attributes (b) ICG (c) new attributes

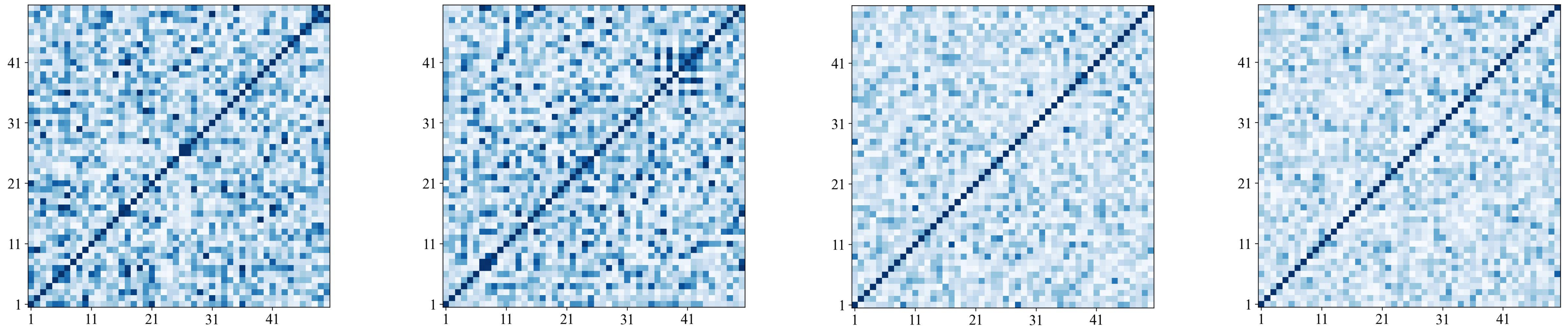
Experiments on the synthetic dataset

CUG-Miner

◆ Independence

In heat maps of attribute correlations, Figure (c) and Figure (d) are much lighter in color than Figure (a) and Figure (b).

New attributes have lower correlations with each other given the class than original attributes.



(a) original atributes - class 1 (b) original atributes - class 2 (c) new atributes - class 1 (d) new atributes - class 2

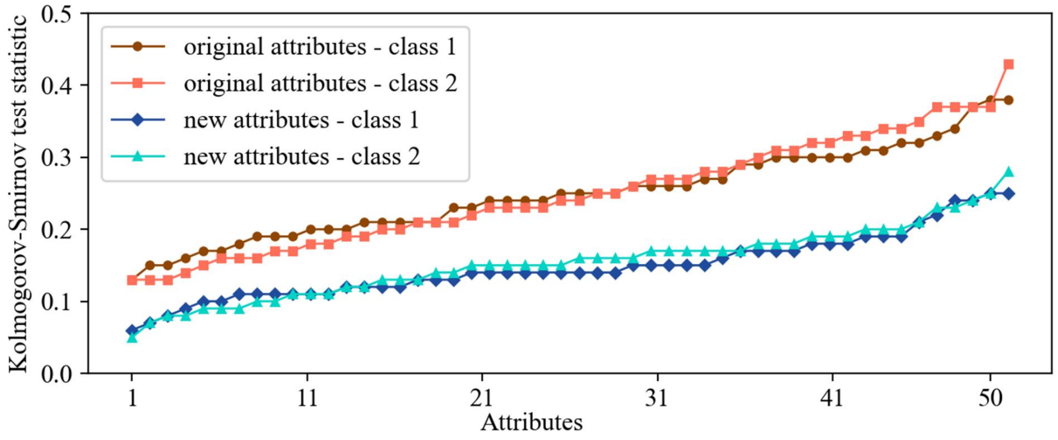
Experiments on the synthetic dataset

CUG-Miner

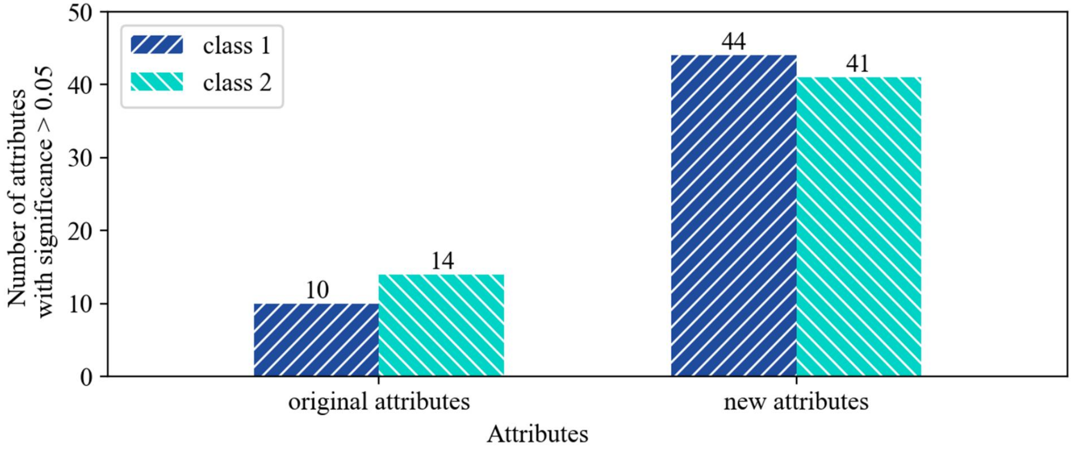
◆ Gaussianity

In Kolmogorov-Smirnov test, new attributes consistently exhibit lower statistics compared to original attributes in both class 1 and class 2.

There are 10, 14 (original) and 44, 41 (new) attributes in class 1 and class 2 demonstrating significant Gaussianity, respectively.



(a) statistics



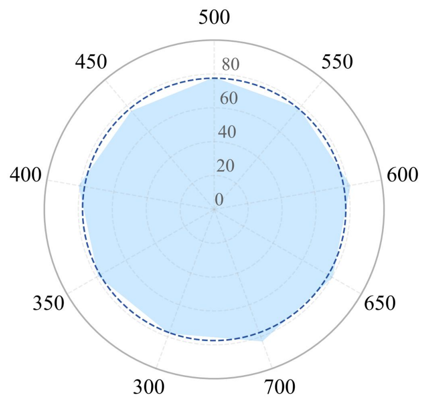
(b) significances

Experiments on the synthetic dataset

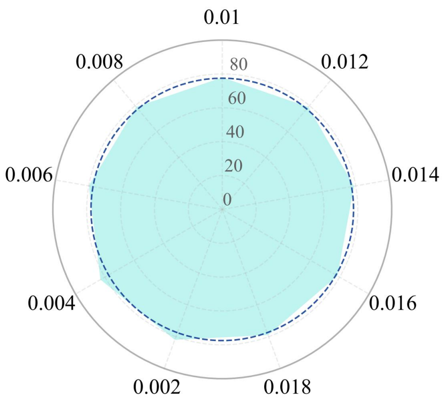
◆ Sensitivity

Using different parameters and graph convolution functions, the average classification accuracy is consistently near that of the default state.

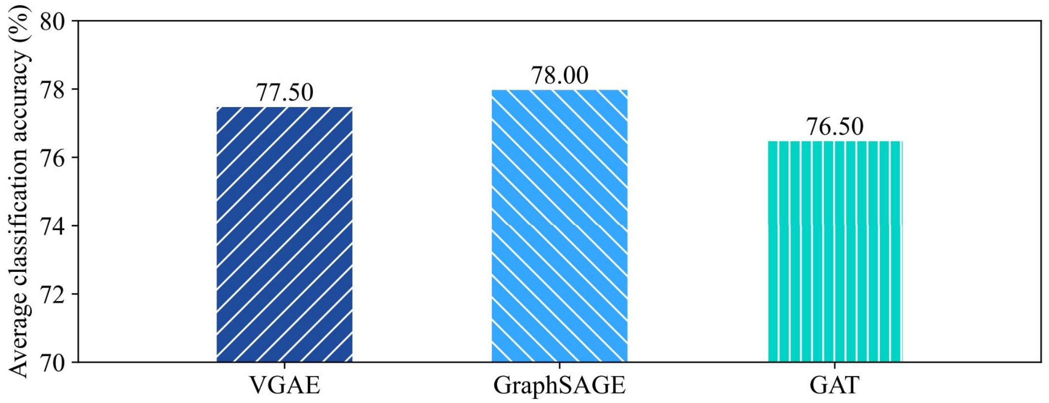
ICGNB is not sensitive to the number of iterations, the learning rate and the graph convolution function.



(a) number of iterations



(b) learning rates



(c) graph convolution functions

Conclusions and future work

CUG-Miner

◆ Conclusions

1. We develop an instance correlation graph (ICG)-based representation learning method to leverage the correlations among instances.

2. We propose a novel algorithm called instance correlation graph-based naive Bayes (ICGNB) based on the representation learning method.

3. We validate the performance of our proposed ICGNB on 24 real-world datasets and a synthetic dataset.

◆ Future work

1. Exploring how to construct ICG with supervised information.

2. Exploring how to design a strategy with lower computational cost.

Thank You！