国防科技大学

National University of Defense Technology

统计决策理论与贝叶斯分析

姓　名	刘悦
学　号	21023115
学　院	计算机学院
专　业	计算机科学与技术
导　师	刘新旺

单位：国防科技大学计算机学院学院五队

My work

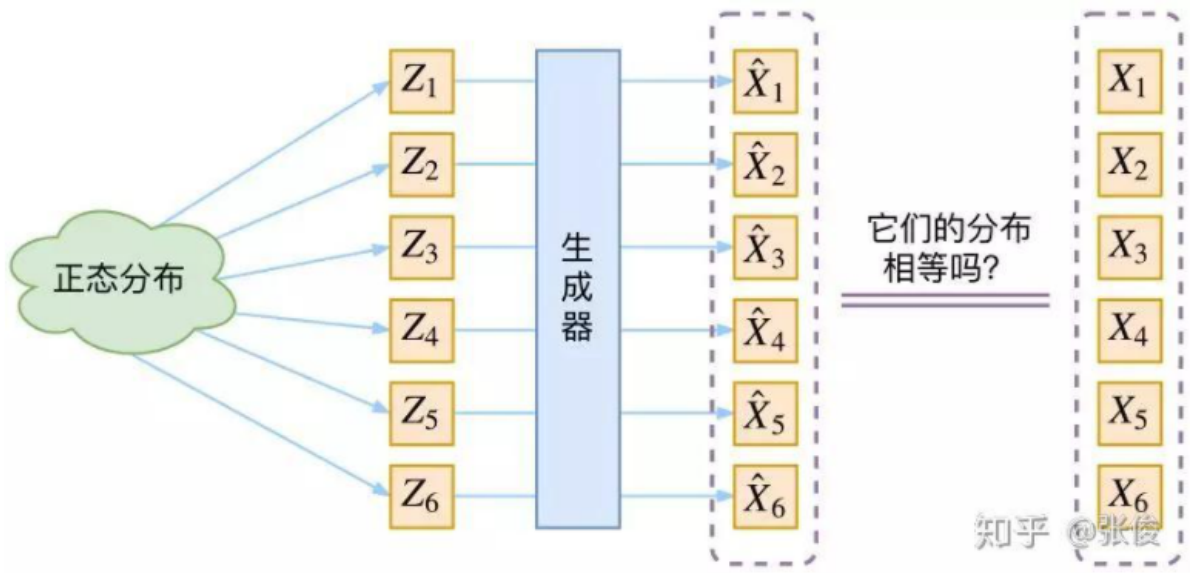
Variational Auto-Encoding (VAE, 变分自编码器)

VAE和GAN (Generative Adversarial Network, 对抗生成网络) 的目标是基本一致的，

即模型希望构建一个从隐变量 Z 生成目标数据 X 的模型，都是属于生成式的模型

以数学的形式来说：

1. 首先假设隐变量 Z 服从某些常见的分布（先验分布），例如
    - 正态分布
    - 均匀分布
2. 训练一个模型 $\hat{X} = g(Z)$
3. 使得该模型能够和原来的概率分布映射倒训练集的概率分布，目标是进行分布之间的变换



而生成式的模型的难题在于如何判断生成分布和真实分布的相似度，因为我们只知道两者采样的结果，而**不知道他们的分布表达式**

我们只有从先验分布中采样并通过模型得到的一批数据 $\hat{X}$ 以及从真实数据中采样而来的一批数据 X，但是我们并不知道他们的分布表达式，所以这里不能使用KL散度来刻画概率分布之间的差异。

但是VAE和GAN采用了不同的策略：

- GAN的思路是，无法找到适合的度量，我们就训练一个神经网络来代替这个度量，这个神经网络就被称为判别器。因此GAN不仅需要训练一个生成器也需要训练一个判别器，所以其训练过程较难，

也有使用Wasserstein GAN (W-GAN)利用Wasserstein 距离的角度来对GAN优化的模型。

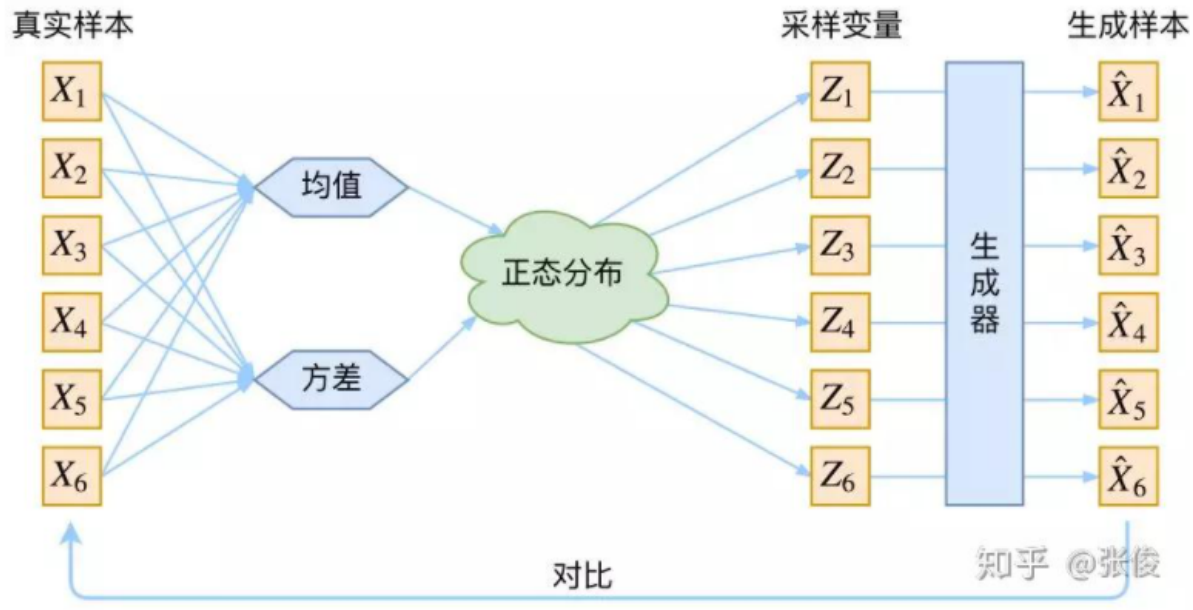
- 而VAE则采用了另外一种技巧，即根据样本获取分布，并且采用了贝叶斯公式

首先，我们进行问题定义，假设我们有一批数据样本$X_1, \ldots, X_n$，整体用X来表示，我们希望利用$X1, .., X_n$得到X的分布$p(X)$，如果可以实现，那便是一个最理想的情况。

在VAE中采用的是另外一种形式，我们首先先利用条件概率公式对$p(X)$进行修改，

$$p(X) = \int p(X|Z)p(Z)dz$$

即利用一个$p(X|Z)$来描述由Z来生成X的模型，而对于$p(Z)$，我们假设Z服从标准正态分布$\mathbb{N}(0, 1)$



但是，我们并不清楚经过重新采样出来的Z_i是否还对应着原来的X_i，所以我们直接最小化它们之间的距离是有问题的。

在VAE的模型中，我们没有使用先验分布是正态分布的假设，我们用的是假设$p(Z|X)$是正态分布：

即，给定一个真实样本X_i，我们假设存在一个**专属**于X_i的分布$p(Z|X_i)$，并且假设该分布是**正态分布**。

为什么?

如果我们假设$p(Z)$是正态分布，然后从$p(Z)$中采样一个Z，那么我们并不知道该Z对应哪个真实的X。而现在我们假设$P(Z|X_i)$是X_i的专属后验分布，则我们有理由说采样出来的Z_i应该要还原到X_k中去

因此，此时每一个X_i都分配上了一个专属的正态分布。但是这样的话，有多少个X就有多少个正态分布了，我们知道正态分布有两组参数

- 均值μ
- 方差σ^2

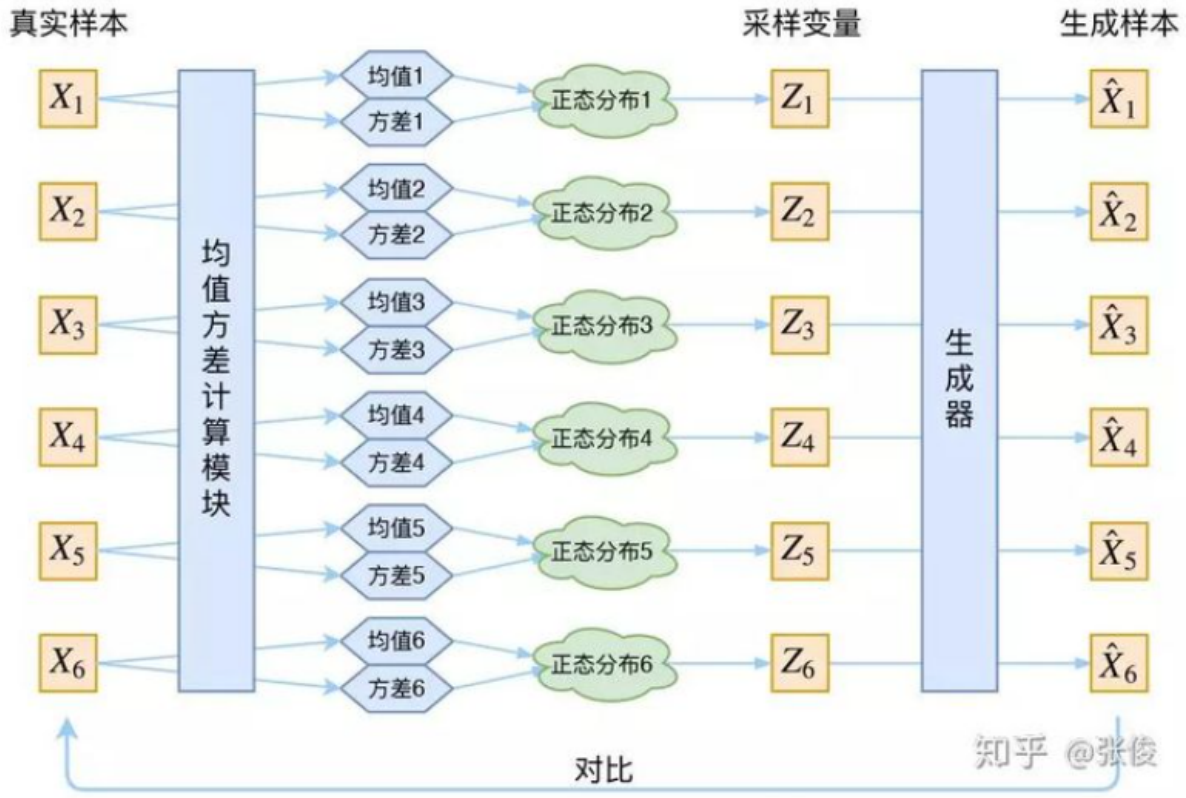
那么，对于每个X_i，我们如何找出其专属正态分布$p(Z|X_i)$的均值和方差呢?

在这里，由于我们无法直接求出他的均值和方差，所以我们直接用神经网络来代替。

对于难算的一些东西，我们都倾向于利用神经网络来进行拟合，例如WGAN中的判别器。

于是，我们构建了两个神经网络$\mu_i = f_1(X_i)$，$log\sigma_i^2 = f_2(X_i)$来计算均值和方差，需要强调的是，这里我们选择拟合$\log \sigma_i^2$而不是直接σ^2是因为方差总是非负的，需要利用激活函数处理，而加上log后不需要激活函数，因为他们可正可负。

到此，我们已经知道专属的X_i的均值和方差了，也就是知道它的正态分布长什么样了，然后从这个专属分布中采样一个Z_k出来，然后经过一个生成器得到$\hat{X}_i = g(Z_k)$；这样我们就可以最小化$D(\hat{X}_i, X_i)$了，因为他们是一一对应的。



实际上，VAE就是为每个样本构造专属的正态分布，然后采样来重构

分表标准化的过程：

在重构的过程中会收到噪声的影响，**因为Z_i是通过重新采样过的，不是直接由encode计算得到的。**

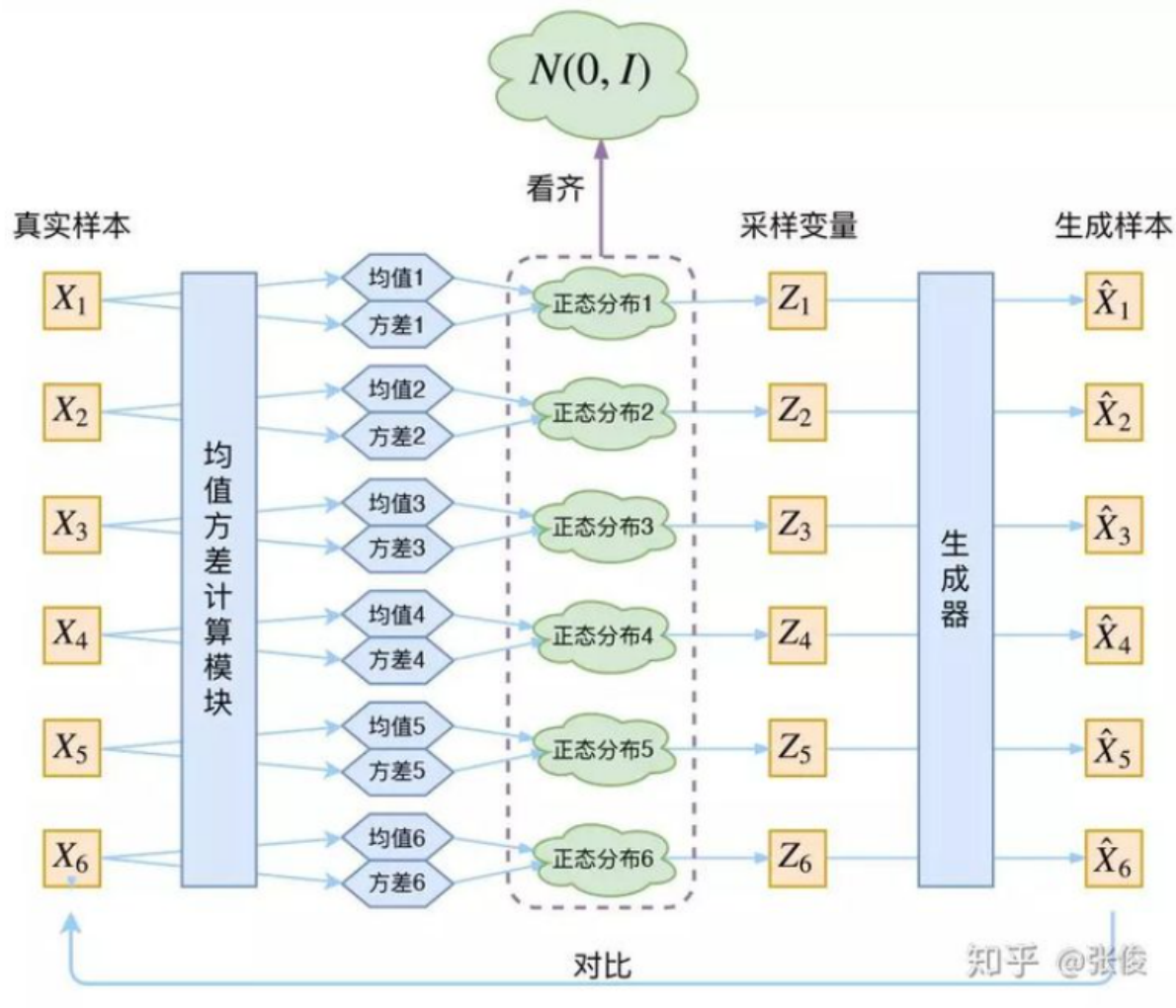
显然噪声会增加重构的难度，但是这个噪声的强度是通过神经网络计算出来的，所以最终模型为了重构的更好，则肯定会想办法让方差为0。然后，方差为0后，就没有随机性了，所以不管如何采样其实都是只能得到确定的结果。

故，VAE还会让所有的后验分布$p(Z|X)$都逼近于标准正态分布，这样就

1. 防止了噪声为0的问题，因为标准正态分布的方差为1
2. 保证了模型的生成能力

根据定义：$p(Z)=\sum_X p(Z|X)P(X)=\sum_X \mathbb{N}(0,1)p(X)=\mathbb{N}(0,1)\sum_X p(X)=\mathbb{N}(0,1)$

所以$p(Z)$是标准正态分布。



那么，如何将后验分布$p(Z|X)$对齐$\mathbb{N}(0,1)$呢?

我们利用KL散度来进行逼近，将后验分布逼近于标准正态分布。

KL散度，又被称为相对熵，可以衡量两个概率密度之间的距离：

信息论：

$X \sim P(x)$

① 概率：$P(x)$

② 信息量：$\frac{1}{P(x)}$

③ 熵：$H(x)=\sum_{x\in X} P(x)\log_2\frac{1}{P(x)}$

$=-\sum_{x\in X} P(x)\log_2 P(x)$

用于衡量不确定性：

熵高，不确定性高；反之相反

熵是平均意义上对随机变量的编码长度。

$H(X)=E\left[\log\frac{1}{P(x)}\right]$

互信息定义：

对于随机变量 X 与 Y，如果其联合分布为 $P(X,Y)$，边缘为 $P(X)$ 与 $P(Y)$，互信息定义为：

$I(X;Y)=\sum_{x\in X}\sum_{y\in Y} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}$

④ 互信息 MI

缩减不确定性

硬币：$H(x)=\frac{1}{2}\log_2\frac{1}{\frac{1}{2}}+\frac{1}{2}\log_2\frac{1}{\frac{1}{2}}=1$

0.9与0.1：

$H(x|y)=-0.9\log 0.9-0.1\log 0.1\approx 0.469$

↓

知道事实后，原硬币的熵，熵减了 0.531 互信息

条件熵：

在一个条件下，随机变量的不确定性

$H(x|y)=H(x,y)-H(y)$

$H(x|y)=\sum_{x,y} P(x,y)\log\frac{1}{P(x|y)}$

信息增益：

在一个条件下，信息不确定性减少的程度

信息增益 = 熵 − 条件熵

$H(x)$ $H(x|y)$

互信息：两个随机变量的相关指标

确定条件后，缩减的不确定性

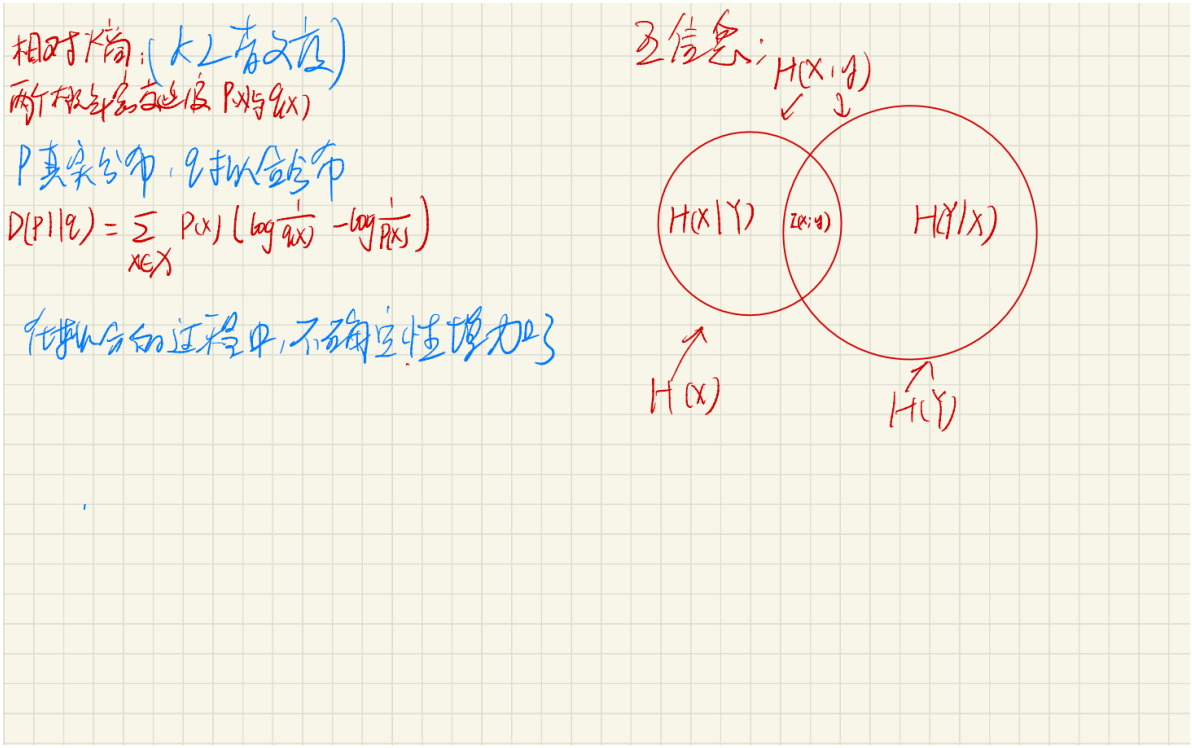
$I(X;Y)=H(X)-H(X|Y)$

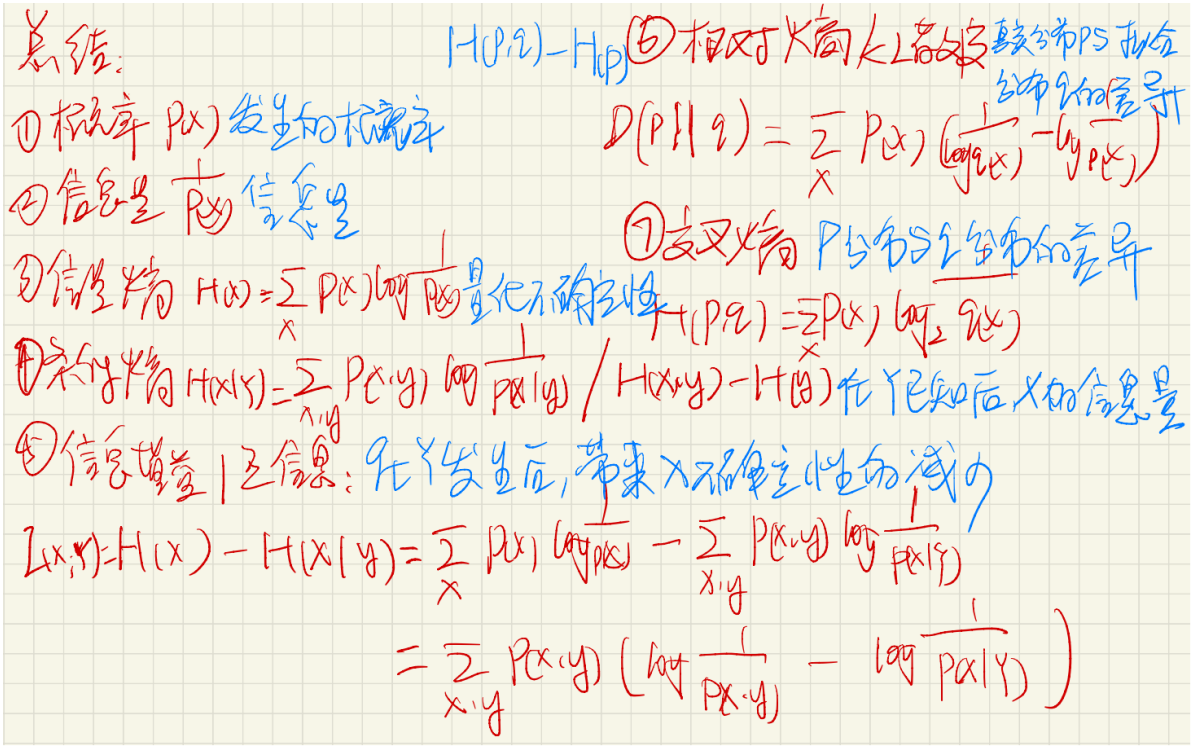
- 条件熵↓ 互信息↑
- 条件熵↑ 互信息↓

交叉熵：设 q 为 p 的估计

$Y\sim P(y)$ $\hat{y}\sim q(y)$

$H(p,q)=\sum_y P(y)\log\frac{1}{q(y)}$





所以，原论文直接算了一般的正态分布与标准正态分布之间的KL散度

$KL(\mathbb{N}(\mu,\sigma^2)||\mathbb{N}(0,1))$作为额外的loss

该上式子展开可以得到：

$$\mathcal{L}_{\mu,\sigma^2} = KL(\mathbb{N}(\mu,\sigma^2)||\mathbb{N}(0,1)) = \int \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} log(\frac{e^{-(x-\mu)^2/2\sigma^2}/\sqrt{2\pi\sigma^2}}{e^{-x^2/2}/\sqrt{2\pi}})dx$$

$$= \int \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} log(\frac{1}{\sigma} e^{\frac{1}{2}[x^2-(x-\mu)^2/\sigma^2]})dx$$

$$= \frac{1}{2}\int \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} log(\frac{1}{\sigma^2} e^{x^2-(x-\mu)^2/\sigma^2})dx$$

$$= \frac{1}{2}\int \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} (-log\sigma^2 + x^2 - (x-\mu)^2/\sigma^2)dx$$

以上的结果可以分为三项：

1. $-log\sigma^2$乘以概率密度的积分，结果为$-log\sigma^2$
2. 正态分布的二阶矩：

   $\mu^2 + \sigma^2$
3. -1

故：

$$\mathcal{L}_{\mu,\sigma^2} = KL(\mathbb{N}(\mu,\sigma^2)||\mathbb{N}(0,1)) = \frac{1}{2}(-log\sigma^2 + \mu^2 + \sigma^2 - 1)$$

重参数技巧：

如果我们需要从$p(Z|X_i)$中采样一个Z_i出来，尽管我们知道了$p(Z|X_i)$是一个正态分布，但是均值和方差都是依赖模型进行计算的，我们需要依赖这个过程反过来优化均值方差的模型，但是采样这个过程是不可导的。

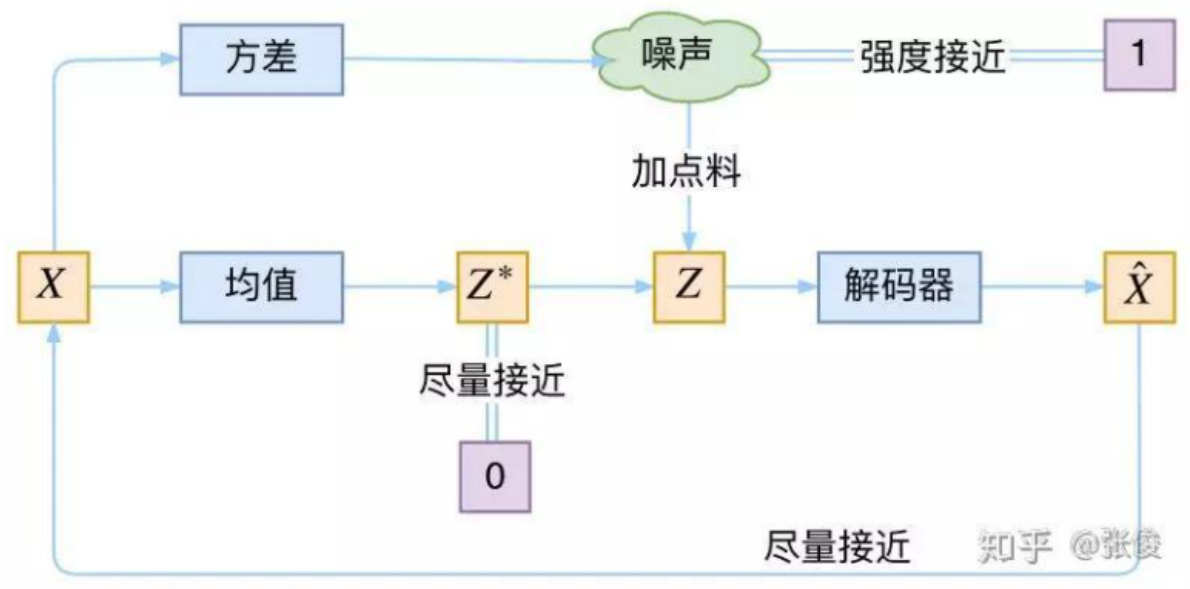
所以我们将从$\mathbb{N}(\mu,\sigma^2)$中采样一个Z，相当于从$\mathbb{N}(0,1)$中采样一个e，然后让$Z = \mu + e\theta$

这样采样的过程就可以参与梯度下降了。

VAE的本质：

在VAE中，它的encoder有两个，一个用来算均值，一个用来算方差
VAE本质就是在常规的自编码器的基础上：

1. 对计算均值encoder的结果加上了高斯噪声，使得decoder对高斯噪声具有鲁棒性
2. KL loss，事实上就是encoder的正则项，希望encoder出来的东西均有0均值
3. 计算方差的encoder动态调节噪声的强度



训练的过程：

- 当 decoder 还没有训练好时（重构误差远大于 KL loss），就会适当降低噪声（KL loss 增加），使得拟合起来容易一些（重构误差开始下降）。
- 当 decoder 训练得还不错时（重构误差小于 KL loss），这时候噪声就会增加（KL loss 减少），使得拟合更加困难了（重构误差又开始增加），这时候 decoder 就要想办法提高它的生成能力

Variational Graph Auto-Encoding (VGAE, 图变分自编码器)

图变分自编码器的变分原理和做法和VAE一样，也是训练两个encoder，一个用于计算均值，一个用于计算方差。

- 用于计算均值的encoder，使得均值尽量接近于0
- 用于计算方差的encoder，使得噪声强度接近于1，给Z加入一些噪声
- decoder对X进行重构

其loss包括两个部分：

- 重构loss，用MSE均方差来计算
- KL散度，将后验概率$p(X|Z)$逼近于标准的正态分布$\mathbb{N}(0,1)$

但是于VAE不同的是，VGAE中的encoder和decoder不仅仅是简单的多层感知机（MLP），而是使用的图卷积。

图卷积：

图卷积操作可以类比于视觉中的卷积操作，都是对局部的信息进行采样。图卷积来源于图卷积神经网络。

给定一个具有 C 类节点的无向图 $G = \{V, E\}$，$V = \{v1, v2, \ldots\ldots, vN\}$和 E 分别是节点集和边集，其中 N 是节点数。 该图的特征在于其属性矩阵 $X \in R^{N \times D}$ 和原始邻接矩阵$A = (aij)^{N \times N}$，其中 D 是节点属性的维度，如果 $(vi, vj) \in E$，则 aij = 1，否则 aij = 0. 对应的度矩阵为 $D = diag(d1, d2, .., dN) \in R^{N \times N}$且$di = P(vi, vj) \in Eaij$。 有了D，可以通过计算$D^{-1/2}(A + I)D^{-1/2}将原始邻接矩阵A归一化

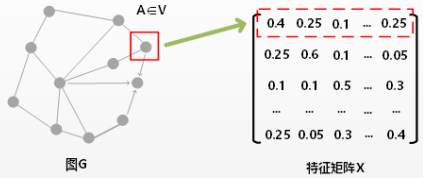
图卷积通常包括三步：

- 变换
- 聚合
- 激活

◆ 图的定义

对于图$G = (V, E)$，V为节点的集合，E为边的集合，对于每个节点i，均有其特征x_i，可以用矩阵$X_{N \times D}$表示。其中N表示节点数，D表示每个节点的特征数，即特征向量的维度。

节点集合V中节点构成特征矩阵



◆ 图相关矩阵的定义

- ✓ 邻接矩阵（Adjacency Matrix）：表示节点间的连接关系，假定为0-1矩阵；
- ✓ 度矩阵（Degree Matrix）：每个节点的度定义为其连接的节点数，度矩阵是一个对角矩阵，对角线元素为 $D_{ii} = \sum_j A_{ij}$
- ✓ 特征矩阵（Content Matrix）：用于表示节点的特征$X_{N \times D}$，这里D是特征的维度；

无向图　有向图　对称邻接矩阵　非对称邻接矩阵

◆ 图神经网络（GNN）

定义：用于处理图数据的神经网络结构的统称。

主要类型：

- ✓ 图卷积网络（Graph Convolution Networks，GCN）
- ✓ 图注意力网络（Graph Attention Networks）
- ✓ 图自编码器（Graph Autoencoders）
- ✓ 图生成网络（Graph Generative Networks）
- ✓ 图时空网络（Graph Spatial-temporal Networks）

图像数据　音频/文本数据

欧几里得结构数据

◆ 图卷积网络（GCN）

本质：学习一个函数映射f，通过该映射图中节点v_i可以**整合**自身特征x_i与其邻居特征x_j，从而生成节点v_i的新表征。

主要类型：

- ✓ 基于谱的图卷积
- ✓ 基于空间的图卷积

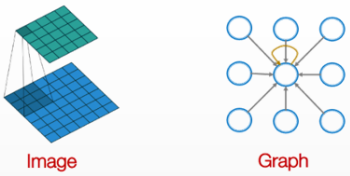
社交网络/知识图谱/生物网络

非欧几里得结构数据

◆ 卷积（CNN）的学习方式：

✓ 对其邻域（$K\times K$的局部空间）的特征进行变换（$w_i x_i$）；

✓ 求和 $\sum_i w_i x_i$（数据通道维度的变化与卷积核的个数有关）

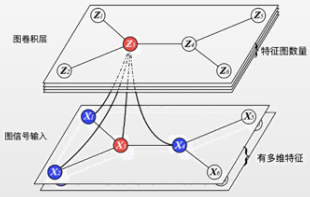


CNN与GCN学习类比

◆ 图卷积（GCN）

✓ 公式表示： $Z^{(l+1)} = f(Z^{(l)}, A \mid W^{(l)}) = \delta\left(AZ^{(l)}W^{(l)}\right)$

$W^{(l)}$定义为第l层的学习权重，维度为$M\times N$；$Z^{(l)}$定义为第l层图卷积的输出，当l为0时，$Z^{(0)}=X$；A定义为邻接矩阵；δ定义为激活函数。



图卷积示意图

◆ 图卷积的三步骤

✓ 变换（红框）：对当前节点的特征进行变换学习（乘法规则），改变节点维度；

✓ 聚合（蓝框）：通过聚合邻居节点的特征来更新当前节点的特征（加法规则）；

✓ 激活（绿框）：激活函数（Relu/Tanh），增强信息的非线性表达。

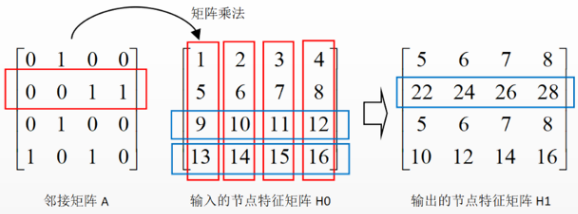
◆ 图卷积的改进

$$Z^{(l+1)} = f(Z^{(l)}, A \mid W^{(l)}) = \delta\left(AZ^{(l)}W^{(l)}\right) \quad (1)$$

上式存在两个问题：

① 计算新特征时没有考虑自身特征；

② 聚合邻居中，节点度数的差异性可能导致梯度爆炸或弥散。



无自环/无归一化的邻居聚合示意图

✓ 针对问题一，给每个节点增加自连接。

$$\tilde{A} = A + I \quad (2)$$

✓ 针对问题二，对邻接矩阵A进行归一化，使A的每行和值为1。

$$Z^{(l+1)} = f(Z^{(l)}, A \mid W^{(l)}) = \delta\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}Z^{(l)}W^{(l)}\right) \quad (3)$$

公式(3)可直观概括为以下4步：

✓ 所有节点增加一条自连接边；$\tilde{A} = A + I$

✓ 对增加自环后的邻接矩阵求各节点入度值；$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$

✓ $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ 相当于对 $\tilde{A}$ 进行归一化[2]；

✓ $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}Z^{(l)}$ 对于每个节点，该节点的特征更新为邻居节点特征相加后的结果。

《Deep Fusion Clustering Network》

基于上述的VAE以及VGAE模型，并且结合了一种自动的融合机制以及三方聚类引导机制，我们组提出了一种新颖深度聚类的模型Deep Fusion Clustering Network (DFCN)。

Abstract

Deep clustering is a fundamental yet challenging task for data analysis. Recently we witness a strong tendency of combining autoencoder and graph neural networks to exploit structure information for clustering performance enhancement. However, we observe that existing literature 1) lacks a dynamic fusion mechanism to selectively integrate and refine the information of graph structure and node attributes for consensus representation learning; 2) fails to extract information from both sides for robust target distribution (i.e., "groundtruth" soft labels) generation. To tackle the above issues, we propose a **D**eep **F**usion **C**lustering **N**etwork (**DFCN**). Specifically, in our network, an interdependency learning-based Structure and Attribute Information Fusion (SAIF) module is proposed to explicitly merge the representations learned by an autoencoder and a graph autoencoder for consensus representation learning. Also, a reliable target distribution generation measure and a triplet self-supervision strategy, which facilitate cross-modality information exploitation, are designed for network training. Extensive experiments on six benchmark datasets have demonstrated that the proposed DFCN consistently outperforms the state-of-the-art deep clustering methods. *Our code is publicly available at https://github.com/WxTu/DFCN.*

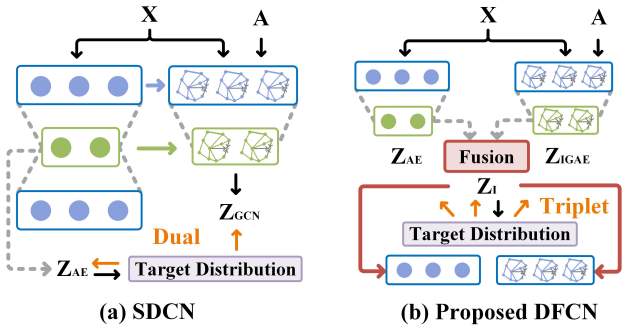


Figure 1: Network structure comparison. Different from the existing structure and attribute information fusion networks (such as SDCN), our proposed method is enhanced with an information fusion module. With this module, 1) both the decoder of AE and IGAE reconstruct the inputs with a learned consensus latent representation. 2) The target distribution is constructed with sufficient negotiation between AE and IGAE. 3) A self-supervised triplet learning strategy is designed.

Introduction

Deep clustering, which aims to train a neural network for learning discriminative feature representations to divide data into several disjoint groups without intense manual guidance, is becoming an increasingly appealing direction to the machine learning researchers. Thanks to the strong representation learning capability of deep learning methods, researches in this field have achieved promising performance in many applications including anomaly detection (Markovitz et al. 2020), social network analysis (Hu, Chan, and He 2017), and face recognition (Wang et al. 2019b). Two important factors, i.e., the optimization objective and the fashion of feature extraction, significantly determine the performance of a deep clustering method. Specifically, in the unsupervised clustering scenario, without the guidance of labels, designing a subtle objective function and an elegant architecture to enable the network to collect more comprehensive and discriminative information for intrinsic structure revealing is extremely crucial and challenging.

According to the network optimization objective, existing deep clustering methods can be roughly grouped into five categories, i.e., subspace clustering-based methods (Zhou et al. 2019a; Ji et al. 2017; Peng et al. 2017), generative adversarial network-based methods (Mukherjee et al. 2019; Ghasedi et al. 2019), spectral clustering-based methods (Yang et al. 2019b; Shaham et al. 2018), Gaussian mixture model-based methods (Yang et al. 2019a; Chen et al. 2019), and self-optimizing-based methods (Xie, Girshick, and Farhadi 2016; Guo et al. 2017). Our method falls into the last category. In the early state, the above deep clustering methods mainly concentrate on exploiting the attribute information in the original feature space of data and have achieved good performance in many circumstances. To further improve the clustering accuracy, recent literature shows a strong tendency in extracting geometrical structure information and then integrates it with attribute information for representation learning. Specifically, Yang et al. design a

*First authors with equal contribution

†Corresponding author

novel stochastic extension of graph embedding to add local data structures into probabilistic deep Gaussian mixture model (GMM) for clustering (Yang et al. 2019a). Distribution preserving subspace clustering (DPSC) first estimates the density distribution of the original data space and the latent embedding space with kernel density estimation. Then it preserves the intrinsic cluster structure within data by minimizing the distribution inconsistency between the two spaces (Zhou et al. 2019a). More recently, graph convolutional networks (GCNs), which aggregate the neighborhood information for better sample representation learning, have attracted the attention of many researchers. The work in deep attentional embedded graph clustering (DAEGC) exploits both graph structure and node attributes with a graph attention encoder. It reconstructs the adjacency matrix by a self-optimizing embedding method (Wang et al. 2019a). Following the setting of DAEGC, adversarially regularized graph autoencoder (ARGA) further develops an adversarial regularizer to guide the learning of latent representations (Pan et al. 2020). After that, structural deep clustering network (SDCN) (Bo et al. 2020) integrates an autoencoder and a graph convolutional network into a unified framework by designing an information passing delivery operator and a dual self-supervised learning mechanism.

Although the former efforts have achieved preferable performance enhancement by leveraging both kinds of information, we find that 1) the existing methods lack an cross-modality dynamic information fusion and processing mechanism. Information from two sources is simply aligned or concatenated, leading to insufficient information interaction and merging; 2) the generation of the target distribution in existing literature has seldom used information from both sources, making the guidance of network training less comprehensive and accurate. As a consequence, the negotiation between two information sources is obstructed, resulting in unsatisfying clustering performance.

To tackle the above issues, we propose a deep fusion clustering network (DFCN). The main idea of our solution is to design a dynamic information fusion module to finely process the attribute and structure information extracted from autoencoder (AE) and graph autoencoder (GAE) for a more comprehensive and accurate representation construction. Specifically, a structure and attribute information fusion (SAIF) module is carefully designed for elaborating both-source information processing. Firstly, we integrate two kinds of sample embeddings in both the perspective of local and global level for consensus representation learning. After that, by estimating the similarity between sample points and pre-calculated cluster centers in the latent embedding space with Students' t-distribution, we acquire more precise target distribution. Finally, we design a triplet self-supervision mechanism which uses the target distribution to provide more dependable guidance for AE, GAE, and the information fusion part simultaneously. Moreover, we develop an improved graph autoencoder (IGAE) with a symmetric structure and reconstruct the adjacency matrix with both the latent representations and the feature representations reconstructed by the graph decoder. The key contributions of this paper are listed as follows:

- We propose a deep fusion clustering network (DFCN). In this network, a structure and attribute information fusion (SAIF) module is designed for better information interaction between AE and GAE. With this module, 1) since both the decoders of AE and GAE reconstruct the inputs using a consensus latent representation, the generalization capacity of the latent embeddings is boosted. 2) The reliability of the generated target distribution is enhanced by integrating the complementary information between AE and GAE. 3) The self-supervised triplet learning mechanism integrates the learning of AE, GAE and the fusion part in a unified and robust system, thus further improves the clustering performance.
- We develop a symmetric graph autoencoder, i.e., improved graph autoencoder (IGAE), to further improve the generalization capability of the proposed method.
- Extensive experiment results on six public benchmark datasets have demonstrated that our method is highly competitive and consistently outperforms the state-of-the-art ones with a preferable margin.

Related Work

Attributed Graph Clustering

Benefiting from the strong representation power of graph convolutional networks (GCNs) (Kipf and Welling 2017), GCN-based clustering methods that jointly learn graph structure and node attributes have been widely studied in recent years (Fan et al. 2020; Cheng et al. 2020; Sun, Lin, and Zhu 2020). Specifically, graph autoencoder (GAE) and variational graph autoencoder (VGAE) are proposed to integrate graph structure into node attributes via iteratively aggregating neighborhood representations around each central node (Kipf and Welling 2016). After that, ARGA (Pan et al. 2020), AGAE (Tao et al. 2019), DAEGC (Wang et al. 2019a), and MinCutPool (Bianchi, Grattarola, and Alippi 2020) improve the performance of the early-stage methods with adversarial training, attention, and graph pooling mechanisms, respectively. Although the performance of the corresponding methods has been improved considerably, the over-smoothing phenomenon of the GCNs still limits the accuracy of these methods. More recently, SDCN (Bo et al. 2020) is proposed to integrate autoencoder and GCN module for better representation learning. Through careful theoretical and experimental analysis, authors find that in their proposed network, autoencoder can help provide complementary attribute information and help relieve the over-smoothing phenomenon of GCN module, while GCN module provides high-order structure information to autoencoder. Although SDCN proves that combining autoencoder and GCN module can boost the clustering performance of both components, in this work, the GCN module acts only as a regularizer of the autoencoder. Thus, the learned features of the GCN module are insufficiently utilized for guiding the self-optimizing network training and the representation learning of the framework lacks the negotiation between the two sub-networks. Differently, in our proposed method, an information fusion module (i.e., SAIF module) is proposed to integrate and refine the features learned by the AE and

IGAE. As a consequence, the complementary information from two sub-networks is finely merged to reach a consensus, and more discriminative representations are learned.

Target Distribution Generation

Since reliable guidance is missing in clustering network training, many deep clustering methods seek to generate the target distribution (i.e., "groundtruth" soft labels) for discriminative representation learning in a self-optimizing manner (Ren et al. 2019; Xu et al. 2019; Li et al. 2019). The early method (DEC) in this category first trains an encoder, and then with the pre-trained network, it further defines a target distribution based on the Student's *t*-distribution and fine-tunes the network with stronger guidance (Xie, Girshick, and Farhadi 2016). To increase the accuracy of the target distribution, IDEC jointly optimizes the cluster assignment and learns features that are suitable for clustering with local structure preservation (Guo et al. 2017). After that, to better train the autoencoder and GCN module integrated network, SDCN designs a dual self-supervised learning mechanism which conducts target distribution refinement and subnetwork training in a unified system (Bo et al. 2020). Despite their success, existing methods generate the target distribution with only the information of autoencoder or GCN module. None of them considers combining the information from both sides and then comes up with a more robust guidance, thus the generated target distribution could be less comprehensive. In contrast, in our method, as the information fusion module allows the information from the two sub-networks to adequately interact with each other, the resultant target distribution has the potential to be more reliable and robust than that of the single-source counterparts.

The Proposed Method

Our proposed method mainly consists of four parts, i.e., an autoencoder, an improved graph autoencoder, a fusion module, and the optimization targets (please check Fig. 1 for the diagram of our network structure). The encoder part of both AE and IGAE are similar with that of the existing literature. In the following sections, we will first introduce the basic notations and then introduce the decoder of both subnetworks, the fusion module, and the optimization targets in detail.

Notations

Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with K cluster centers, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and E are the node set and the edge set, respectively, where N is the number of samples. The graph is characterized by its attribute matrix $\mathbf{X} \in \mathbb{R}^{N\times d}$ and original adjacency matrix $\mathbf{A} = (a_{ij})_{N\times N} \in \mathbb{R}^{N\times N}$. Here, d is the attribute dimension and $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, otherwise $a_{ij} = 0$. The corresponding degree matrix is $\mathbf{D} = diag(d_1, d_2, ..., d_N) \in \mathbb{R}^{N\times N}$ and $d_i = \sum_{v_j\in V} a_{ij}$. With $\mathbf{D}$, the original adjacency matrix is further normalized as $\widetilde{\mathbf{A}} \in \mathbb{R}^{N\times N}$ through calculating $\mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I} \in \mathbb{R}^{N\times N}$ indicates that each node in $\mathcal{V}$ is linked with a self-loop structure. All notations are summarized in Table 1.

Notations	Meaning
$\mathbf{X} \in \mathbb{R}^{N\times d}$	Attribute matrix
$\mathbf{A} \in \mathbb{R}^{N\times N}$	Original adjacency matrix
$\mathbf{I} \in \mathbb{R}^{N\times N}$	Identity matrix
$\widetilde{\mathbf{A}} \in \mathbb{R}^{N\times N}$	Normalized adjacency matrix
$\mathbf{D} \in \mathbb{R}^{N\times N}$	Degree matrix
$\widehat{\mathbf{Z}} \in \mathbb{R}^{N\times d}$	Reconstructed weighted attribute matrix
$\widehat{\mathbf{A}} \in \mathbb{R}^{N\times N}$	Reconstructed adjacency matrix
$\mathbf{Z}_{AE} \in \mathbb{R}^{N\times d'}$	Latent embedding of AE
$\mathbf{Z}_{IGAE} \in \mathbb{R}^{N\times d'}$	Latent embedding of IGAE
$\mathbf{Z}_I \in \mathbb{R}^{N\times d'}$	Initial fused embedding
$\mathbf{Z}_L \in \mathbb{R}^{N\times d'}$	Local structure enhanced $\mathbf{Z}_I$
$\mathbf{S} \in \mathbb{R}^{N\times N}$	Normalized self-correlation matrix
$\mathbf{Z}_G \in \mathbb{R}^{N\times d'}$	Global structure enhanced $\mathbf{Z}_L$
$\widetilde{\mathbf{Z}} \in \mathbb{R}^{N\times d'}$	Clustering embedding
$\mathbf{Q} \in \mathbb{R}^{N\times K}$	Soft assignment distribution
$\mathbf{P} \in \mathbb{R}^{N\times K}$	Target distribution

Table 1: Basic notations for the proposed DFCN

Fusion-based Autoencoders

Input of the Decoder. Most of the existing autoencoders, either classic autoencoder or graph autoencoder, reconstruct the inputs with only its own latent representations. However, in our proposed method, with the compressed representations of AE and GAE, we first integrate the information from both sources for a consensus latent representation. Then, with this embedding as an input, both the decoders of AE and GAE reconstruct the inputs of two sub-networks. This is very different from the existing methods that our proposed method fuses heterogeneous structure and attribute information with a carefully designed fusion module and then reconstructs the inputs of both sub-networks with the consensus latent representation. Detailed information about the fusion module will be introduced in the Structure and Attribute Information Fusion section.

Improved Graph Autoencoder. In the existing literature, the classic autoencoders are usually symmetric, while graph convolutional networks are usually asymmetric (Kipf and Welling 2016; Wang et al. 2019a; Tao et al. 2019). They require only the latent representation to reconstruct the adjacency information and overlook that the structure-based attribute information can also be exploited for improving the generalization capability of the corresponding network. To better make use of both the adjacency information and the attribute information, we design a symmetric improved graph autoencoder (IGAE). This network requires to reconstruct both the weighted attribute matrix and the adjacency matrix simultaneously. In the proposed IGAE, a layer in the encoder and decoder is formulated as:

$$\mathbf{Z}^{(l)} = \sigma(\widetilde{\mathbf{A}}\mathbf{Z}^{(l-1)}\mathbf{W}^{(l)}), \tag{1}$$

$$\widehat{\mathbf{Z}}^{(h)} = \sigma(\widetilde{\mathbf{A}}\widehat{\mathbf{Z}}^{(h-1)}\widehat{\mathbf{W}}^{(h)}), \tag{2}$$

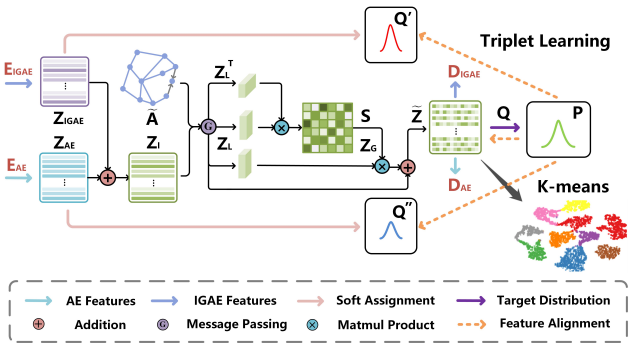


Figure 2: Illustration of the Structure and Attribute Information Fusion (SAIF) module.

where $\mathbf{W}^{(l)}$ and $\widehat{\mathbf{W}}^{(h)}$ denote the learnable parameters of the l-th encoder layer and h-th decoder layer. σ is a non-linear activation function, such as ReLU or Tanh. To minimize both the reconstruction loss functions over the weighted attribute matrix and the adjacency matrix, our IGAE is designed to minimize a hybrid loss function:

$$L_{IGAE} = L_w + \gamma L_a. \tag{3}$$

In Eq.(3), γ is a pre-defined hyper-parameter that balances the weight of the two reconstruction loss functions. Specially, L_w and L_a are defined as follows:

$$L_w = \frac{1}{2N}\|\widetilde{\mathbf{A}}\mathbf{X} - \widehat{\mathbf{Z}}\|_F^2, \tag{4}$$

$$L_a = \frac{1}{2N}\|\widetilde{\mathbf{A}} - \widehat{\mathbf{A}}\|_F^2. \tag{5}$$

In Eq.(4), $\widehat{\mathbf{Z}} \in \mathbb{R}^{N\times d}$ is the reconstructed weighted attribute matrix. In Eq.(5), $\widehat{\mathbf{A}} \in \mathbb{R}^{N\times N}$ is the reconstructed adjacency matrix generated by an inner product operation with multi-level representations of the network. By minimizing both Eq.(4) and Eq.(5), the proposed IGAE is termed to minimize the reconstruction loss over the weighted attribute matrix and the adjacency matrix at the same time. Experimental results in the following parts validate the effectiveness of this setting.

Structure and Attribute Information Fusion

To sufficiently explore the graph structure and node attributes information extracted by the AE and IGAE, we propose a structure and attribute information fusion (SAIF) module. This module consists of two parts, i.e., a cross-modality dynamic fusion mechanism and a triplet self-supervised strategy. The overall structure of SAIF is illustrated in Fig. 2.

Cross-modality Dynamic Fusion Mechanism. The information integration within our fusion module includes four steps. First, we combine the latent embedding of AE ($\mathbf{Z}_{AE} \in \mathbb{R}^{N\times d'}$) and IGAE ($\mathbf{Z}_{IGAE} \in \mathbb{R}^{N\times d'}$) with a linear combination operation:

$$\mathbf{Z}_I = \alpha\mathbf{Z}_{AE} + (1-\alpha)\mathbf{Z}_{IGAE}, \tag{6}$$

where d' is the latent embedding dimension, and α is a learnable coefficient which selectively determines the importance of two information sources according to the property of the corresponding dataset. In our paper, α is initialized as 0.5 and then tuned automatically with a gradient decent method.

Then, we process the combined information with a graph convolution-like operation (i.e., message passing operation). With this operation, we enhance the initial fused embedding $\mathbf{Z}_I \in \mathbb{R}^{N\times d'}$ by considering the local structure within data:

$$\mathbf{Z}_L = \widetilde{\mathbf{A}}\mathbf{Z}_I. \tag{7}$$

In Eq.(7), $\mathbf{Z}_L \in \mathbb{R}^{N\times d'}$ denotes the local structure enhanced $\mathbf{Z}_I$.

After that, we further introduce a self-correlated learning mechanism to exploit the non-local relationship in the preliminary information fusion space among samples. Specifically, we first calculate the normalized self-correlation matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ through Eq.(8):

$$\mathbf{S}_{ij} = \frac{e^{(\mathbf{Z}_L\mathbf{Z}_L^{\mathrm{T}})_{ij}}}{\sum_{k=1}^{N} e^{(\mathbf{Z}_L\mathbf{Z}_L^{\mathrm{T}})_{ik}}}. \tag{8}$$

With $\mathbf{S}$ as coefficients, we recombine $\mathbf{Z}_L$ by considering the global correlation among samples: $\mathbf{Z}_G = \mathbf{S}\mathbf{Z}_L$.

Finally, we adopt a skip connection to encourage information to pass smoothly within the fusion mechanism:

$$\widetilde{\mathbf{Z}} = \beta\mathbf{Z}_G + \mathbf{Z}_L, \tag{9}$$

where β is a scale parameter. Following the setting in (Fu et al. 2019), we initialize it as 0 and learn its weight while training the network. Technically, our cross-modality dynamic fusion mechanism considers the sample correlation in both the perspective of the local and global level. Thus, it has potential benefit on finely fusing and refining the information from both AE and IGAE for learning consensus latent representations.

Triplet Self-supervised Strategy. To generate more reliable guidance for clustering network training, we first adopt the more robust clustering embedding $\widetilde{\mathbf{Z}} \in \mathbb{R}^{N\times d'}$ which has integrated the information from both AE and IGAE for target distribution generation. As shown in Eq.(10) and Eq.(11), the generation process includes two steps:

$$q_{ij} = \frac{(1+\|\tilde{z}_i - u_j\|^2/v)^{-\frac{v+1}{2}}}{\sum_{j'}(1+\|\tilde{z}_i - u_{j'}\|^2/v)^{-\frac{v+1}{2}}}, \tag{10}$$

$$p_{ij} = \frac{q_{ij}^2/\sum_i q_{ij}}{\sum_{j'}(q_{ij'}^2/\sum_i q_{ij'})}. \tag{11}$$

In the first step (corresponding to Eq.(10)), we calculate the similarity between the i-th sample ($\tilde{z}_i$) and the j-th pre-calculated clustering center (u_j) in the fused embedding space using Student's t-distribution as kernel. In Eq.(10), v is the degree of freedom for Student's t-distribution and q_{ij} indicates the probability of assigning the i-th node to the j-th center (i.e., a soft assignment). The soft assignment matrix $\mathbf{Q} \in \mathbb{R}^{N\times K}$ reflects the distribution of all samples. In

the second step, to increase the confidence of cluster assignment, we introduce Eq.(11) to drive all samples to get closer to cluster centers. Specifically, $0 \leq p_{ij} \leq 1$ is an element of the generated target distribution $\mathbf{P} \in \mathbb{R}^{N \times K}$, which indicates the probability of the i-th sample belongs to the j-th cluster center.

With the iteratively generated target distribution, we then calculate the soft assignment distribution of AE and IGAE by using Eq.(10) over the latent embeddings of two sub-networks, respectively. We denote the soft assignment distribution of IGAE and AE as $\mathbf{Q}^{'}$ and $\mathbf{Q}^{''}$.

To train the network in a unified framework and improve the representative capability of each component, we design a triplet clustering loss by adapting the KL-divergence in the following form:

$$L_{KL} = \sum_{i} \sum_{j} p_{ij} log \frac{p_{ij}}{(q_{ij} + q_{ij}^{'} + q_{ij}^{''})/3}. \quad (12)$$

In this formulation, the summation of soft assignment distribution of AE, IGAE, and the fused representations are aligned with the robust target distribution simultaneously. Since the target distribution is generated without human guidance, we name the loss function triplet clustering loss and the corresponding training mechanism as triplet self-supervised strategy.

Algorithm 1 Deep Fusion Clustering Network

Input: Attribute matrix $\mathbf{X}$; Adjacency matrix $\mathbf{A}$; Target distribution update interval T; Iteration number I; Cluster number K; Hyper-parameters γ, λ.
Output: Clustering results $\mathbf{O}$.

```
Initialize the parameters of AE, IGAE, and the fusion part to obtain Z_AE, Z_IGAE, and Z̃;
Initialize the clustering centers u with K-means based on Z̃;
for i = 1 to I do
    Update Z_I and Z_L by Eq.(6) and Eq.(7);
    Update the normalized self-correlation matrix S and the deep clustering embedding Z̃ by Eq.(8) and Eq.(9), respectively;
    Calculate soft assignment distributions Q, Q', and Q'' based on Z̃, Z_IGAE, and Z_AE by Eq.(10);
    if i%T == 0 then
        Calculate the target distribution P derived from Q by Eq.(11);
    end if
    Utilize P to refine Q, Q', and Q'' in turn by Eq.(12);
    Calculate L_AE, L_IGAE, and L_KL, respectively.
    Update the whole network by minimizing Eq.(13);
end for
Obtain the clustering results O with the final Z̃ by K-means.
return O
```

Joint loss and Optimization

The overall learning objective consists of two main parts, i.e., the reconstruction loss of AE and IGAE, and the clustering loss which is correlated with the target distribution:

$$L = \underbrace{L_{AE} + L_{IGAE}}_{Reconstruction} + \underbrace{\lambda L_{KL}}_{Clustering}. \quad (13)$$

Dataset	Type	Samples	Classes	Dimension
USPS	Image	9298	10	256
HHAR	Record	10299	6	561
REUT	Text	10000	4	2000
ACM	Graph	3025	3	1870
DBLP	Graph	4058	4	334
CITE	Graph	3327	6	3703

Table 2: Dataset summary

In Eq.(13), L_{AE} is the mean square error (MSE) reconstruction loss of AE. Different from SDCN, the proposed DFCN reconstructs the inputs of both sub-networks with the consensus latent representation. λ is a pre-defined hyper-parameter which balances the importance of reconstruction and clustering. The detailed learning procedure of the proposed DFCN is shown in Algorithm 1.

Experiments

Benchmark Datasets

We evaluate the proposed DFCN on six popular public datasets, including three graph datasets (ACM[1], DBLP[2], and CITE[3]) and three non-graph datasets (USPS (LeCun et al. 1990), HHAR (Lewis et al. 2004), and REUT (Stisen et al. 2015)). Table 2 summarizes the brief information of these datasets. For the dataset (like USPS, HHAR, and REUT) whose affinity matrix is absent, we follow (Bo et al. 2020) and construct the matrix with heat kernel method.

Experiment Setup

Training Procedure Our method is implemented with PyTorch platform and a NVIDIA 2080TI GPU. The training of the proposed DFCN includes three steps. First, we pre-train the AE and IGAE independently for 30 iterations by minimizing the reconstruction loss functions. Then, both sub-networks are integrated into a united framework for another 100 iterations. Finally, with the learned centers of different clusters and under the guidance of the triplet self-supervised strategy, we train the whole network for at least 200 iterations until convergence. The cluster ID is acquired by performing K-means algorithm over the consensus clustering embedding $\widetilde{\mathbf{Z}}$. Following all the compared methods, to alleviate the adverse influence of randomness, we repeat each experiment for 10 times and report the average values and the corresponding standard deviations.

Parameters Setting For ARGA (Pan et al. 2020), we set the parameters of the method by following the setting of the original paper. For other compared methods, we report the results listed in the paper SDCN (Bo et al. 2020) directly. For our method, we adopt the original code and data of SDCN for data pre-processing and testing. All ablation studies are trained with the Adam optimizer. The optimization stops when the validation loss comes to a plateau. The

[1] http://dl.acm.org/
[2] https://dblp.uni-trier.de
[3] http://citeseerx.ist.psu.edu/index

Data	Metric	K-means	AE	DEC	IDEC	GAE	VGAE	ARGA	DAEGC	SDCN$_Q$	SDCN	DFCN
USPS	ACC	66.8±0.0	71.0±0.0	73.3±0.2	76.2±0.1	63.1±0.3	56.2±0.7	66.8±0.7	73.6±0.4	77.1±0.2	78.1±0.2	79.5±0.2
	NMI	62.6±0.0	67.5±0.0	70.6±0.3	75.6±0.1	60.7±0.6	51.1±0.4	61.6±0.3	71.1±0.2	77.7±0.2	79.5±0.3	82.8±0.3
	ARI	54.6±0.0	58.8±0.1	63.7±0.3	67.9±0.1	50.3±0.6	41.0±0.6	51.1±0.6	63.3±0.3	70.2±0.2	71.8±0.2	75.3±0.2
	F1	64.8±0.0	69.7±0.0	71.8±0.2	74.6±0.1	61.8±0.4	53.6±1.1	66.1±1.2	72.5±0.5	75.9±0.2	77.0±0.2	78.3±0.2
HHAR	ACC	60.0±0.0	68.7±0.3	69.4±0.3	71.1±0.4	62.3±1.0	71.3±0.4	63.3±0.8	76.5±2.2	83.5±0.2	84.3±0.2	87.1±0.1
	NMI	58.9±0.0	71.4±1.0	72.9±0.4	74.2±0.4	55.1±1.4	63.0±0.4	57.1±1.4	69.1±2.3	78.8±0.3	79.9±0.1	82.2±0.1
	ARI	46.1±0.0	60.4±0.9	61.3±0.5	62.8±0.5	42.6±1.6	51.5±0.7	44.7±1.0	60.4±2.2	71.8±0.2	72.8±0.1	76.4±0.1
	F1	58.3±0.0	66.4±0.3	67.3±0.3	68.6±0.3	62.6±1.0	71.6±0.3	61.1±0.9	76.9±2.2	81.5±0.1	82.6±0.1	87.3±0.1
REUT	ACC	54.0±0.0	74.9±0.2	73.6±0.1	75.4±0.1	54.4±0.3	60.9±0.2	56.2±0.2	65.6±0.1	79.3±0.1	77.2±0.2	77.7±0.2
	NMI	41.5±0.5	49.7±0.3	47.5±0.3	50.3±0.2	25.9±0.4	25.5±0.2	28.7±0.3	30.6±0.3	56.9±0.3	50.8±0.2	59.9±0.4
	ARI	28.0±0.4	49.6±0.4	48.4±0.1	51.3±0.2	19.6±0.2	26.2±0.4	24.5±0.4	31.1±0.2	59.6±0.3	55.4±0.4	59.8±0.4
	F1	41.3±2.4	61.0±0.2	64.3±0.2	63.2±0.1	43.5±0.4	57.1±0.2	51.1±0.2	61.8±0.1	66.2±0.2	65.5±0.1	69.6±0.1
ACM	ACC	67.3±0.7	81.8±0.1	84.3±0.8	85.1±0.5	84.5±1.4	84.1±0.2	86.1±1.2	86.9±2.8	87.0±0.1	90.5±0.2	90.9±0.2
	NMI	32.4±0.5	49.3±0.2	54.5±1.5	56.6±1.2	55.4±1.9	53.2±0.5	55.7±1.4	56.2±4.2	58.9±0.2	68.3±0.3	69.4±0.4
	ARI	30.6±0.7	54.6±0.2	60.6±1.9	62.2±1.5	59.5±3.1	57.7±0.7	62.9±2.1	59.4±3.9	65.3±0.2	73.9±0.4	74.9±0.4
	F1	67.6±0.7	82.0±0.1	84.5±0.7	85.1±0.5	84.7±1.3	84.2±0.2	86.1±1.2	87.1±2.8	86.8±0.1	90.4±0.2	90.8±0.2
DBLP	ACC	38.7±0.7	51.4±0.4	58.2±0.6	60.3±0.6	61.2±1.2	58.6±0.1	61.6±1.0	62.1±0.5	65.7±1.3	68.1±1.8	76.0±0.8
	NMI	11.5±0.4	25.4±0.2	29.5±0.3	31.2±0.5	30.8±0.9	26.9±0.1	26.8±1.0	32.5±0.5	35.1±1.1	39.5±1.3	43.7±1.0
	ARI	7.0±0.4	12.2±0.4	23.9±0.4	25.4±0.6	22.0±1.4	17.9±0.1	22.7±0.3	21.0±0.5	34.0±1.8	39.2±2.0	47.0±1.5
	F1	31.9±0.3	52.5±0.4	59.4±0.5	61.3±0.6	61.4±2.2	58.7±0.1	61.8±0.9	61.8±0.7	65.8±1.2	67.7±1.5	75.7±0.8
CITE	ACC	39.3±3.2	57.1±0.1	55.9±0.2	60.5±1.4	61.4±0.8	61.0±0.4	56.9±0.7	64.5±1.4	61.7±1.1	66.0±0.3	69.5±0.2
	NMI	16.9±3.2	27.6±0.1	28.3±0.3	27.2±2.4	34.6±0.7	32.7±0.3	34.5±0.8	36.4±0.9	34.4±1.2	38.7±0.3	43.9±0.2
	ARI	13.4±3.0	29.3±0.1	28.1±0.4	25.7±2.7	33.6±1.2	33.1±0.5	33.4±1.5	37.8±1.2	35.5±1.5	40.2±0.4	45.5±0.3
	F1	36.1±3.5	53.8±0.1	52.6±0.2	61.6±1.4	57.4±0.8	57.7±0.5	54.8±0.8	62.2±1.3	57.8±1.0	63.6±0.2	64.3±0.2

Table 3: Clustering performance on six datasets (mean±std). The red and blue values indicate the best and the runner-up results, respectively.

learning rate is set to 1e-3 for USPS, HHAR, 1e-4 for REUT, DBLP, and CITE, and 5e-5 for ACM. The training batch size is set to 256 and we adopt an early stop strategy to avoid over-fitting. According to the results of parameter sensitivity testing, we fix two balanced hyper-parameters γ and λ to 0.1 and 10, respectively. Moreover, we set the nearest neighbors number of each node as 5 for all non-graph datasets.

Evaluation Metric The clustering performance of all methods is evaluated by four metrics: Accuracy (ACC), Normalized Mutual Information (NMI), Average Rand Index (ARI), and macro F1-score (F1) (Zhou et al. 2020, 2019b; Liu et al. 2020a,b, 2019). The best map between cluster ID and class ID is found by using the Kuhn-Munkres algorithm (Lovász and Plummer 1986).

Comparison with the State-of-the-art Methods

In this part, we compare our proposed method with ten state-of-the-art clustering methods to illustrate its effectiveness. Among them, K-means (Hartigan and Wong 1979) is the representative one of classic shallow clustering methods. AE (Hinton and Salakhutdinov 2006), DEC (Xie, Girshick, and Farhadi 2016), and IDEC (Guo et al. 2017) represent the autoencoder-based clustering methods which learn the representations for clustering through training an autoencoder. GAE/VGAE (Kipf and Welling 2016), ARGA (Pan et al. 2020), and DAEGC (Wang et al. 2019a) are typical methods of graph convolutional network-based methods. In these methods, the clustering representation is embedded with structure information by GCN. SDCN$_Q$ and SDCN (Bo et al. 2020) are representatives of hybrid methods which take advantage of both AE and GCN module for clustering.

The clustering performance of our method and 10 baseline methods on six benchmark datasets are summarized in Table 3. Based on the results, we have the following observations:

1) DFCN shows superior performance against the compared methods in most circumstances. Specifically, K-means performs clustering on raw data. AE, DEC, and IDEC merely exploit node attribute representations for clustering. These methods seldom take structure information into account, leading to sub-optimal performance. In contrast, DFCN successfully leverages available data by selectively integrating the information of graph structure and node attributes, which complements each other for consensus representation learning and greatly improves clustering perfor-

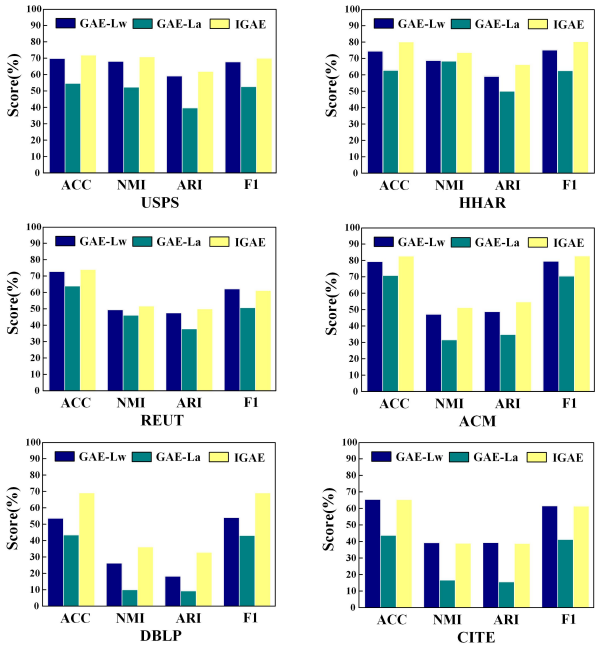


Figure 3: Clustering results of the graph autoencoder with different reconstruction strategy. GAE-L$_w$, GAE-L$_a$, and IGAE correspond to the reconstruction of weighted attribute matrix, adjacency matrix, and both.

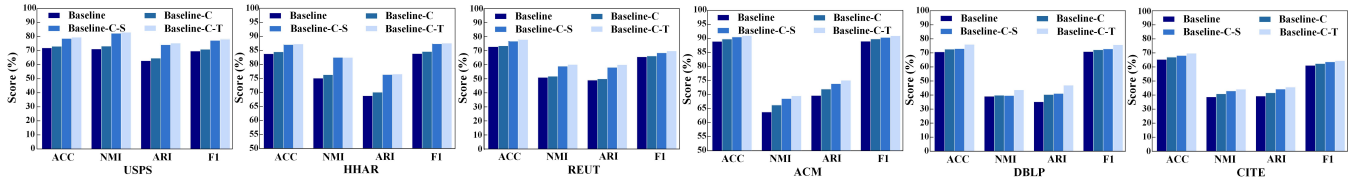


Figure 4: Ablation comparisons of cross-modality dynamic fusion mechanism and triplet self-supervised strategy in SAIF. The baseline refers to a naive united framework consisting of AE and IGAE. -C, -S, and -T indicate that the baseline utilizes the cross-modality dynamic fusion mechanism, single or triplet self-supervised strategy, respectively.

Dataset	Model	ACC	NMI	ARI	F1
USPS	+AE	78.3±0.3	81.3±0.1	73.6±0.3	76.8±0.3
	+IGAE	76.9±0.4	77.1±0.4	68.8±0.6	74.8±0.5
	DFCN	**79.5±0.2**	**82.8±0.3**	**75.3±0.2**	**78.3±0.2**
HHAR	+AE	75.2±1.4	**82.8±1.0**	71.7±1.2	72.6±0.9
	+IGAE	82.8±0.1	79.6±0.1	72.3±0.1	83.4±0.1
	DFCN	**87.1±0.1**	82.2±0.1	**76.4±0.1**	**87.3±0.1**
REUT	+AE	69.3±0.8	48.5±1.6	44.6±1.1	58.3±0.6
	+IGAE	71.4±1.7	52.5±1.0	49.1±2.2	61.5±2.9
	DFCN	**77.7±0.2**	**59.9±0.4**	**59.8±0.4**	**69.6±0.1**
ACM	+AE	90.2±0.3	67.5±0.8	73.2±0.8	90.2±0.3
	+IGAE	89.6±0.2	65.6±0.4	71.8±0.4	89.6±0.2
	DFCN	**90.9±0.2**	**69.4±0.4**	**74.9±0.4**	**90.8±0.2**
DBLP	+AE	64.2±2.9	30.2±3.2	29.4±3.4	64.6±2.8
	+IGAE	67.5±1.0	34.2±1.1	31.5±1.1	67.6±1.0
	DFCN	**76.0±0.8**	**43.7±1.0**	**47.0±1.5**	**75.7±0.8**
CITE	+AE	69.3±0.3	42.9±0.4	44.7±0.4	**64.4±0.3**
	+IGAE	67.9±0.9	41.8±1.0	43.0±1.4	63.7±0.7
	DFCN	**69.5±0.2**	**43.9±0.2**	**45.5±0.3**	64.3±0.2

Table 4: Ablation comparisons of the target distribution generation with signle- or both-source information.

mance.

2) It is obvious that GCN-based methods such as GAE, VGAE, ARGA, and DAEGC are not comparable to ours, because these methods under-utilize abundant information from data itself and might be limited to the over-smoothing phenomenon. Differently, DFCN incorporates attribute-based representations learned by AE into the whole clustering framework, and mutually explores graph structure and node attributes with a fusion module for consensus representation learning. As a result, the proposed DFCN improves the clustering performance of the existing GCN-based methods with a preferable gap.

3) DFCN achieves better clustering results than the strongest baseline methods SDCN$_Q$ and SDCN in the majority of cases, especially on HHAR, DBLP, and CITE datasets. On DBLP dataset for instance, our method achieves a 7.9%, 4.2%, 7.8%, and 8.0% increment with respect to ACC, NMI, ARI and F1 against SDCN. This is because DFCN not only achieves a dynamic interaction between graph structure and node attributes to reveal the intrinsic clustering structure, but also adopts a triplet self-supervised strategy to provide precise network training guidance.

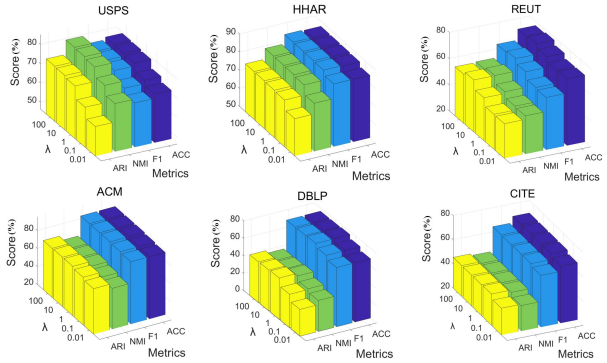


Figure 5: The sensitivity of DFCN with the variation of λ on six datasets.

Ablation Studies

Effectiveness of IGAE We further conduct ablation studies to verify the effectiveness of IGAE and report the results in Fig. 3. GAE-L$_w$ or GAE-L$_a$ denotes the method optimized by the reconstruction loss function of weighted attribute matrix or adjacency matrix only. We can find out that GAE-L$_w$ consistently performs better than GAE-L$_a$ on six datasets. Besides, IGAE clearly improves the clustering performance over the method which constructs the adjacency matrix only. Both observations illustrate that our proposed reconstruction measure is able to exploit more comprehensive information for improving the generalization capability of the deep clustering network. By this means, the latent embedding inherits more properties from the attribute space of the original graph, preserving representative features that generate better clustering decisions.

Analysis of the SAIF Module In this part, we conduct several experiments to verify the effectiveness of the SAIF module. As summarized in Fig. 4, we observe that 1) compared with the baseline, Baseline-C method has about 0.5% to 5.0% performance improvements, indicating that exploring graph structure and node attributes in both the perspective of the local and global level is helpful to learn consensus latent representations for better clustering; 2) the performance of Baseline-C-T method is consistently better than that of Baseline-C-S method on all datasets. The reason is that our triplet self-supervised strategy successfully generates more reliable guidance for the training of AE, IGAE, and the fusion part, making them benefit from each other.

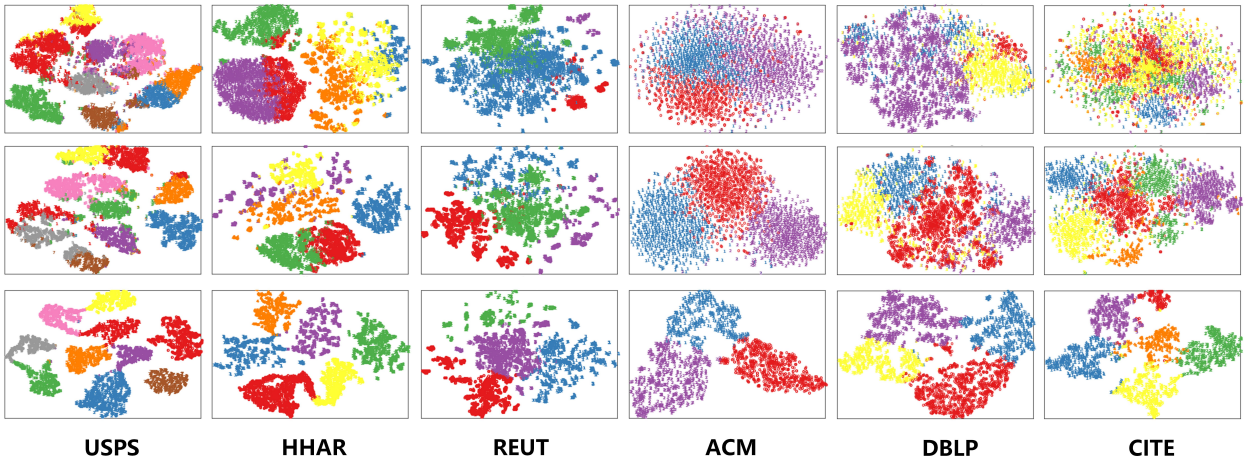


Figure 6: 2D visualization on six datasets. The first, second, and last row correspond to the distribution of raw data, baseline and DFCN (baseline + SAIF), respectively.

According to these observations, the superiority of the SAIF module has clearly been demonstrated over the baseline.

Influence of Exploiting Both-source Information We compare our method with two variants to validate the effectiveness of complementary two-modality (structure and attribute) information learning for target distribution generation. As reported in Table 4, +AE or +IGAE refers to the DFCN with only AE or IGAE part, respectively. On one hand, as +AE and +IGRE achieve better performance on different datasets, it indicates that information from either AE or IGAE cannot consistently outperform that of their counterparts, combining the both-source information can potentially improve the robustness of the hybrid method. On the other hand, DFCN encodes both DNN- and GCN-based representations and consistently outperforms the single-source methods. This shows that 1) both-source information is equally essential for the performance improvement of DFCN; 2) DFCN can facilitate the complementary two-modality information to make the target distribution more reliable and robust for better clustering.

Analysis of Hyper-parameter λ

As can be seen in Eq.(13), DFCN introduces a hyper-parameter λ to make a trade-off between the reconstruction and clustering. We conduct experiments to show the effect of this parameter on all datasets. Fig. 5 illustrates the performance variation of DFCN when λ varies from 0.01 to 100. From these figures, we observe that 1) the hyper-parameter λ is effective in improving the clustering performance; 2) the performance of the method is stable in a wide range of λ; 3) DFCN tends to perform well by setting λ to 10 across all datasets.

Visualization of Clustering Results

To intuitively verify the effectiveness of DFCN, we visualize the distribution of the learned clustering embedding $\widetilde{\mathbf{Z}}$ in two-dimensional space by employing t-SNE algorithm (Maaten and Hinton 2008). As illustrated in Fig. 6, DFCN can better reveal the intrinsic clustering structure among data.

Conclusion

In this paper, we propose a novel neural network-based clustering method termed Deep Fusion Clustering Network (DFCN). In our method, the core component SAIF module leverages both graph structure and node attributes via a dynamic cross-modality fusion mechanism and a triplet self-supervised strategy. In this way, more consensus and discriminative information from both sides is encoded to construct the robust target distribution, which effectively provides the precise network training guidance. Moreover, the proposed IGAE is able to assist in improving the generalization capability of the proposed method. Experiments on six benchmark datasets show that DFCN consistently outperforms state-of-the-art baseline methods. In the future, we plan to further improve our method to adapt it to multi-view graph clustering and incomplete multi-view graph clustering applications.

Acknowledgments

This work is supported by the National Key R & D Program of China (Grant 2018YFB1800202, 2020AAA0107100, 2020YFC2003400), the National Natural Science Foundation of China (Grant 61762033, 62006237, 62072465), the Hainan Province Key R & D Plan Project (Grant ZDYF2020040), the Hainan Provincial Natural Science Foundation of China (Grant 2019RC041, 2019RC098), and the Opening Project of Shanghai Trusted Industrial Control Platform (Grant TICPSH202003005-ZC).

References

Bianchi, F. M.; Grattarola, D.; and Alippi, C. 2020. Spectral Clustering with Graph Neural Networks for Graph Pooling. In *ICML*, 2729–2738.

Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; and Cui, P. 2020. Structural Deep Clustering Network. In *WWW*, 1400–1410.

Chen, J.; Milot, L.; Cheung, H. M. C.; and Martel, A. L. 2019. Unsupervised Clustering of Quantitative Imaging Phenotypes Using Autoencoder and Gaussian Mixture Model. In *MICCAI*, 575–582.

Cheng, J.; Wang, Q.; Tao, Z.; Xie, D.; and Gao, Q. 2020. Multi-View Attribute Graph Convolution Networks for Clustering. In *IJCAI*, 2973–2979.

Fan, S.; Wang, X.; Shi, C.; Lu, E.; Lin, K.; and Wang, B. 2020. One2Multi Graph Autoencoder for Multi-view Graph Clustering. In *WWW*, 3070–3076.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual Attention Network for Scene Segmentation. In *CVPR*, 3146–3154.

Ghasedi, K.; Wang, X.; Deng, C.; and Huang, H. 2019. Balanced Self-Paced Learning for Generative Adversarial Clustering Network. In *CVPR*, 4391–4400.

Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017. Improved Deep Embedded Clustering with Local Structure Preservation. In *IJCAI*, 1753–1759.

Hartigan, J. A.; and Wong, M. A. 1979. A K-Means Clustering Algorithm. *Applied Stats* 28(1): 100–108.

Hinton, G.; and Salakhutdinov, R. R. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313: 504–507.

Hu, P.; Chan, K. C. C.; and He, T. 2017. Deep Graph Clustering in Social Network. In *WWW*, 1425–1426.

Ji, P.; Zhang, T.; Li, H.; Salzmann, M.; and Reid, I. D. 2017. Deep Subspace Clustering Networks. In *NIPS*, 24–33.

Kipf, T. N.; and Welling, M. 2016. Variational Graph Auto-Encoders. *ArXiv abs/1611.07308* .

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 14.

LeCun, Y.; Matan, O.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. E.; Jacket, L. D.; and Baird, H. S. 1990. Handwritten Zip Code Recognition with Multilayer Networks. In *ICPR*, 36–40.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research* 5(2): 361–397.

Li, Z.; Wang, Q.; Tao, Z.; Gao, Q.; and Yang, Z. 2019. Deep Adversarial Multi-view Clustering Network. In *IJCAI*, 2952–2958.

Liu, X.; Wang, L.; Zhu, X.; Li, M.; Zhu, E.; Liu, T.; Liu, L.; Dou, Y.; and Yin, J. 2020a. Absent Multiple Kernel Learning Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42(6): 1303–1316.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Tang, C.; Yin, J.; Shen, D.; Wang, H.; and Gao, W. 2019. Late Fusion Incomplete Multi-View Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(10): 2410–2423.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Zhu, E.; Liu, T.; Kloft, M.; Shen, D.; Yin, J.; and Gao, W. 2020b. Multiple kernel k-means with incomplete kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42(5): 1191–1204.

Lovász, L.; and Plummer, M. 1986. Matching Theory .

Maaten, L. V. D.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9(2605): 2579–2605.

Markovitz, A.; Sharir, G.; Friedman, I.; Zelnik-Manor, L.; and Avidan, S. 2020. Graph Embedded Pose Clustering for Anomaly Detection. In *CVPR*, 10536–10544.

Mukherjee, S.; Asnani, H.; Lin, E.; and Kannan, S. 2019. ClusterGAN: Latent Space Clustering in Generative Adversarial Networks. In *AAAI*, 1965–1972.

Pan, S.; Hu, R.; Fung, S.-F.; Long, G.; Jiang, J.; and Zhang, C. 2020. Learning Graph Embedding with Adversarial Training Methods. *IEEE Transactions on Cybernetics* 50(6): 2475–2487.

Peng, X.; Feng, J.; Lu, J.; Yau, W.; and Yi, Z. 2017. Cascade Subspace Clusterings. In *AAAI*, 2478–2484.

Ren, Y.; Hu, K.; Dai, X.; Pan, L.; Hoi, S. C. H.; and Xu, Z. 2019. Semi-supervised Deep Embedded Clustering. *Neurocomputing* 325(1): 121–130.

Shaham, U.; Stanton, K. P.; Li, H.; Basri, R.; Nadler, B.; and Kluger, Y. 2018. SpectralNet: Spectral Clustering using Deep Neural Networks. In *ICLR*.

Stisen, A.; Blunck, H.; Bhattacharya, S.; Prentow, T. S.; Kjærgaard, M. B.; Dey, A.; Sonne, T.; and Jensen, M. M. 2015. Smart Devices Are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition. In *SENSYS*, 127–140.

Sun, K.; Lin, Z.; and Zhu, Z. 2020. Multi-Stage Self-Supervised Learning for Graph Convolutional Networks on Graphs with Few Labeled Nodes. In *AAAI*, 5892–5899.

Tao, Z.; Liu, H.; Li, J.; Wang, Z.; and Fu, Y. 2019. Adversarial Graph Embedding for Ensemble Clustering. In *IJCAI*, 3562–3568.

Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; and Zhang, C. 2019a. Attributed Graph Clustering: A Deep Attentional Embedding Approach. In *IJCAI*, 3670–3676.

Wang, Z.; Zheng, L.; Li, Y.; and Wang, S. 2019b. Linkage Based Face Clustering via Graph Convolution Network. In *CVPR*, 1117–1125.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised Deep Embedding for Clustering Analysis. In *ICML*, 478–487.

Xu, C.; Guan, Z.; Zhao, W.; Wu, H.; Niu, Y.; and Ling, B. 2019. Adversarial Incomplete Multi-view Clustering. In *IJCAI*, 3933–3939.

Yang, L.; Cheung, N.-M.; Li, J.; and Fang, J. 2019a. Deep Clustering by Gaussian Mixture Variational Autoencoders with Graph Embedding. In *ICCV*, 6440–6449.

Yang, X.; Deng, C.; Zheng, F.; Yan, J.; and Liu, W. 2019b. Deep Spectral Clustering Using Dual Autoencoder Network. In *CVPR*, 4066–4075.

Zhou, L.; Bai, X.; Wang, D.; Liu, X.; Zhou, J.; and Hancock, E. 2019a. Latent Distribution Preserving Deep Subspace Clustering. In *IJCAI*, 4440–4446.

Zhou, S.; Liu, X.; Li, M.; Zhu, E.; Liu, L.; Zhang, C.; and Yin, J. 2019b. Multiple Kernel Clustering with Neighbor-kernel Subspace Segmentation. *IEEE transactions on neural networks and learning systems* 31(4): 1351–1362.

Zhou, S.; Zhu, E.; Liu, X.; Zheng, T.; Liu, Q.; Xia, J.; and Yin, J. 2020. Subspace Segmentation-based Robust Multiple Kernel Clustering. *Information Fusion* 53: 145–154.