# 基于最大化交叉互信息的对称 IB 算法

娄铮铮 叶阳东
(郑州大学信息工程学院 郑州 450001)

**摘 要** 对称 IB(Symmetric Information Bottleneck)通过行、列压缩变量之间的相互协作来挖掘数据中的双向压缩模式.由于行、列压缩变量不能完全承载行、列基层变量中所蕴含的特征信息,从而导致对称 IB 所得的数据双向压缩模式与基层变量所蕴含的内在模式之间存在一定的偏离.针对该问题,通过最大化地保存压缩变量与基层变量交叉之间的互信息,将基层变量引入到数据的双向压缩中,使它们协助压缩变量共同来学习联合分布中的双向压缩模式,提出交叉对称 IB: ICSIB(Inter-Correlated Symmetric Information Bottleneck). ICSIB 算法采用交错的顺序"抽取-合并"迭代过程来优化压缩变量与基层变量交叉之间的互信息,可保证得到目标函数的一个局部优解.实验结果表明,在基层特征变量的协助下,ICSIB 算法得到的数据双向压缩模式更接近于数据中真实的内在模式,并可有效地应用于数据的联合聚类中.

**关键词** IB 方法;多变量 IB;对称 IB;双向压缩;联合聚类;数据挖掘
**中图法分类号** TP18 **DOI 号** 10.11897/SP.J.1016.2016.01515

# Symmetric Information Bottleneck Based on Maximization Inter-Correlated Mutual Information

LOU Zheng-Zheng YE Yang-Dong
(*School of Information Engineering, Zhengzhou University, Zhengzhou* 450001)

**Abstract** The symmetric IB aims to extract the double compressing patterns of data via the cooperation between compressed row and column variables. However, the compressed variables cannot completely carry the information resided in original variables, which results that there will be some deviation between the compressing patterns extracted by symmetric IB and the original patterns resided in original features. To solve this problem, this paper proposes an Inter-Correlated Symmetric Information Bottleneck (ICSIB), which aims to maximize the inter-correlated mutual information between compressed variables and original variables, so that original feature variables can be involved in data double compressing process and can be used to help the compressed variables to learn double compressing patterns. The ICSIB algorithm can monotonically increase the objective function by an intertwining "draw-and-merge" sequential iteration procedure, and guarantee to converge to a local maximum of the information. Our experimental results on benchmark data sets have demonstrated the effectiveness of the proposed method in the application of data double compressing and co-clustering.

**Keywords** Information Bottleneck (IB) method; multivariate IB; symmetric IB; double compressing; co-clustering; data mining

收稿日期:2015-01-25;在线出版日期:2016-03-28.本课题得到国家自然科学基金(61170223,61502434)资助.**娄铮铮**,男,1984 年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为机器学习、模式识别、数据挖掘. E-mail: zzlou@zzu.edu.cn. **叶阳东**(通信作者),男,1962 年生,博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为机器学习、知识工程、智能系统等. E-mail: yeyd@zzu.edu.cn.

## 1 引 言

IB方法(Information Bottleneck method)是Tishby等人[1]于1999年提出的一种基于信息论的数据分析方法.该方法在做数据分析时,将数据模式的提取视为一个数据压缩的过程,如图1所示,其中 $X$ 表示待分析的数据对象,$Y$ 表示描述数据对象的特征变量,$\widetilde{X}$ 为 $X$ 的压缩"瓶颈"变量.变量 $X$ 到 $\widetilde{X}$ 的压缩编码 $p(\tilde{x}|x)$ 即为IB方法所获得的数据压缩模式,若某些数据对象被压缩到同一个簇 $\tilde{x}$ 中,那么它们被视为具有相同的模式特征.为使压缩编码 $p(\tilde{x}|x)$ 尽可能真实地反映数据中所蕴含的内在模式,IB方法在对数据进行压缩的同时,要求"瓶颈"变量 $\widetilde{X}$ 尽可能最大化地保存特征变量 $Y$ 中所载有的信息量.变量 $Y$ 客观地描述了数据对象的特征,是IB方法数据压缩的依据.IB方法具备良好的理论基础,在众多领域中均取得了成功的应用[2-13].

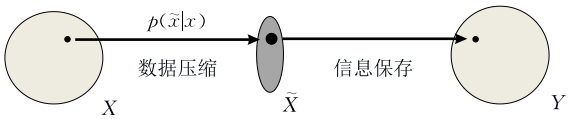



图1 IB方法

多变量IB方法(Multivariate Information Bottleneck)[14-15]是对IB方法的拓展,采用更多的变量来抽象现实问题,让更多的信息参与到数据的压缩中,通过多种信息之间的相互协作来完成更具挑战性的数据分析任务.多变量IB方法为多元数据分析问题提供了理论框架,其协作模型刻画了变量之间的协作关系,为数据分析任务目标函数的确定提供了依据.多变量IB方法在处理多元数据分析问题时具有独特的优势[14-21],然而在应用中需结合具体问题来实体化多变量IB方法的协作模型并设计相应的优化算法.

对称IB[15]是多变量IB方法的一个实体化协作模型,将IB方法的数据单向压缩拓展到双向压缩中,即对联合分布 $p(X,Y)$ 同时做 $X$ 到 $\widetilde{X}$ 及 $Y$ 到 $\widetilde{Y}$ 的压缩.对称IB一方面可获取变量 $X$ 的压缩模式 $p(\tilde{x}|x)$ 及变量 $Y$ 的压缩模式 $p(\tilde{y}|y)$,另一方面可获取联合分布 $p(X,Y)$ 的高度压缩模式 $p(\widetilde{X},\widetilde{Y})$.例如:在对文档数据进行分析时,对称IB可同时对文档和单词进行压缩,将具有相似主题的文档压缩到同一个文档簇中,得到文档的压缩模式,同时也将具有描述相似语义能力的单词压缩到同一个单词簇中,得到单词的压缩模式;另外,对称IB还可获得文档簇与单词簇之间的高层统计规律.若将一个文档簇视为一个主题,一个单词簇视为一个语义单位,那么该高层统计规律则反映出每个主题与每个语义单位之间所对应的关联关系.

对称IB的数据双向压缩可用于解决机器学习领域中的联合聚类问题[22-26].在此值得一提的是,当对称IB将重点放在压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间信息量的保存上时,对称IB的目标函数可简化为Dhillon等人[23]提出的信息论联合聚类(Information-Theoretic Co-Clustering,ITCC)的目标函数,此时信息论联合聚类可视为对称IB的一个特例.本文在2.3节将对此做详细分析;另外在4.3节中的实验结果表明,对称IB的顺序"抽取-合并"优化算法在做联合聚类时性能优于ITCC的"矩阵逼近"优化算法,因此可以说对称IB算法是一个更高效的联合聚类算法.对称IB和ITCC算法所得到的压缩变量之间的统计规律 $p(\widetilde{X},\widetilde{Y})$ 是对联合分布 $p(X,Y)$ 的高度压缩模式,揭示了联合分布 $p(X,Y)$ 中行、列变量压缩模式之间所蕴含的高层特征模式.该高度压缩的模式结构对数据分析同样有着重要的意义,在文档数据分析[27]、图像场景建模[28]、视频数据分析[29]、自适应学习[30-31]中均取得了成功的应用.

对称IB通过行、列压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间的相互协作来学习联合分布 $p(X,Y)$ 中的双向压缩模式,即在迭代学习过程中,$X$ 到 $\widetilde{X}$ 的压缩 $p(\tilde{x}|x)$ 是依据 $\widetilde{Y}$ 所提供的特征信息来进行的,$Y$ 到 $\widetilde{Y}$ 的压缩 $p(\tilde{y}|y)$ 是依据 $\widetilde{X}$ 所提供的信息来进行的.$\widetilde{X}$ 和 $\widetilde{Y}$ 分别是 $X$ 与 $Y$ 被压缩之后的变量,它们并不能完全承载基层变量 $X$ 与 $Y$ 中所固有的特征信息.因此,仅考虑压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间协作关系的对称IB在一定程度上存在着特征信息损失的问题,进而可能导致所得的双向压缩模式偏离数据中所蕴含的内在模式.例如:对称IB在对文档数据进行双向压缩时,文档簇和单词簇分别是单词与文档的压缩依据.尽管一个单词簇中的单词具有描述相同语义的能力,可被视为一个语义单位,但就单词个体而言,它们往往还具备表达出多种语义信息的能力,而作为多个单词合体的单词簇则损失了单词本身的多义性信息.因此,在以单词簇为依据所得的文档压缩模式与以单词为依据所得的文档压缩模式之间可能存在着一定的偏离.另外,一个文档有时可归属于多个主题,因此在以文档簇为依据的单词压缩过程中也存在着同样的上述问题.

针对对称IB在对数据进行双向压缩时存在特征信息损失的问题,在多变量IB方法的基础上,提出

一个交叉对称 IB 协作模型 ICSIB(Inter-Correlated Symmetric IB). 在 ICSIB 的协作模型中，除了关注高层压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间的协作关系外，还将基层变量 $X$ 与 $Y$ 引入到数据的双向压缩中，使得它们协助压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 一起来学习联合分布中的双向压缩模式. ICSIB 算法采用交错的“抽取-合并”顺序迭代过程对目标函数进行优化，具有较低时间和空间复杂度，且可保证得到目标函数的一个局部优化解. 在文档联合聚类及图像无监督模式识别上的实验结果表明，在基层变量的协助下，ICSIB 算法所得数据双向压缩模式更接近于数据中所蕴含的真实内在模式，其性能优于 $k$-means 算法、Normalized Cuts 算法[32]、信息论联合聚类 ITCC 算法[23]、对称 IB 算法[15]、aIBCC 算法[19]及基于非负矩阵分解的 DRCC 联合聚类算法(Dual Regularized Co-clustering)[24]. 另外，在基层特征变量的协助下，ICSIB 算法得到的压缩模式 $p(\widetilde{X},\widetilde{Y})$ 中压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间的关联程度同样优于 ITCC 算法[23]和对称 IB 算法[15].

本文的主要贡献可总结如下：

(1) 提出一个交叉对称 IB 协作模型 ICSIB. 该模型将压缩之前的基层特征变量 $X$ 与 $Y$ 引入到对称 IB 的双向压缩中，更充分地利用 $X$ 与 $Y$ 所提供的基层特征信息，使其协助压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 更好地学习数据中的双向压缩模式，从而解决对称 IB 中特征信息损失问题.

(2) ICSIB 算法采用交错的“抽取-合并”顺序迭代过程对目标函数进行优化，理论上保证收敛到目标函数的一个局部优解，具有较低时间和空间复杂度.

## 2　背景知识

本文中，符号 $X$、$Y$ 表示离散型随机变量，其值域分别为 $\mathcal{X}=\{x_1,x_2,\cdots,x_m\}$ 和 $\mathcal{Y}=\{y_1,y_2,\cdots,y_n\}$. $\widetilde{X}$、$\widetilde{Y}$ 表示 $X$、$Y$ 的压缩变量，值域分别为 $\widetilde{\mathcal{X}}=\{\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_k\}$ 和 $\widetilde{\mathcal{Y}}=\{\tilde{y}_1,\tilde{y}_2,\cdots,\tilde{y}_l\}$. $\boldsymbol{X}$ 表示一组随机变量的集合，即 $\boldsymbol{X}=\{X_1,\cdots,X_n\}$.

### 2.1　IB 方法

IB 方法[1]起源于著名的率失真理论. 该方法在做数据分析时，将数据模式的提取视为一个数据压缩的过程，即将源变量 $X$ 压缩到一个“瓶颈”变量 $\widetilde{X}$ 中，同时使压缩变量 $\widetilde{X}$ 最大化地保存相关变量 $Y$ 中所蕴含的信息量，其中 $X$ 到 $\widetilde{X}$ 的压缩编码 $p(\tilde{x}|x)$ 便为 IB 方法所得的数据压缩模式. IB 方法可形式化地描述为

$$R(D)=\min_{\{p(\tilde{x}|x):I(\widetilde{X};Y)\geq D\}} I(X;\widetilde{X}) \tag{1}$$

其中，$I(X;\widetilde{X})$ 为变量 $X$ 与 $\widetilde{X}$ 之间的互信息[33]，计算方法为

$$I(X;\widetilde{X})=\sum_x\sum_{\tilde{x}} p(x,\tilde{x})\log\frac{p(x,\tilde{x})}{p(x)p(\tilde{x})} \tag{2}$$

式(1)表明，IB 方法是在满足信息保存限制的条件下，即压缩变量 $\widetilde{X}$ 所保存特征变量 $Y$ 中的信息量 $I(\widetilde{X};Y)$ 满足 $I(\widetilde{X};Y)\geq D$，在所有可能编码方案中选择使压缩信息 $I(X;\widetilde{X})$ 最小的一个编码方案 $p(\tilde{x}|x)$. 为求最优压缩编码方案 $p(\tilde{x}|x)$，文献[1]采用拉格朗日乘子法将式(1)改为如下的 IB 目标函数：

$$\mathcal{L}_{\min}[p(\tilde{x}|x)]=I(\widetilde{X};X)-\beta I(\widetilde{X};Y) \tag{3}$$

其中，$\beta$ 是一个大于等于 0 的拉格朗日因子，用于平衡源信息的压缩和相关信息的保存.

对 IB 方法的目标函数式(3)求关于 $p(\tilde{x}|x)$ 的导数，可得到如下 IB 方法的形式化解，

$$p(\tilde{x}|x)=\frac{p(\tilde{x})}{Z(x,\beta)}\mathrm{e}^{-\beta D_{KL}[p(y|x)\|p(y|\tilde{x})]} \tag{4}$$

$$p(\tilde{x})=\sum_{x,y}p(x,y,\tilde{x})=\sum_x p(x)p(\tilde{x}|x) \tag{5}$$

$$p(y|\tilde{x})=\frac{1}{p(\tilde{x})}\sum_x p(x,y)p(\tilde{x}|x) \tag{6}$$

其中，$D_{KL}[p(y|x)\|p(y|\tilde{x})]$ 是条件分布 $p(y|x)$ 与 $p(y|\tilde{x})$ 之间的 $KL$(Kullback Leibler)距离[33]，$Z(x,\beta)=\sum_{\tilde{x}}p(\tilde{x})\exp(-\beta D_{KL}[p(y|x)\|p(y|\tilde{x})])$ 是归一化函数.

### 2.2　多变量 IB 方法

多变量 IB 方法[14-15]是对 IB 方法的拓展，在处理更复杂的多元数据分析问题时具有独特的优势. 该方法使用一组随机变量 $\boldsymbol{X}=\{X_1,\cdots,X_n\}$ 来表示领域问题，将相关问题的领域知识抽象为一个多元联合分布 $p(\boldsymbol{X})$ 的形式. 给定一组随机变量 $\boldsymbol{X}=\{X_1,\cdots,X_n\}$，多变量 IB 方法力图求解 $\{\boldsymbol{U}_1,\boldsymbol{U}_2,\cdots,\boldsymbol{U}_k\}$($\boldsymbol{U}_j\subset\boldsymbol{X}$)到 $\widetilde{\boldsymbol{X}}=\{\widetilde{X}_1,\cdots,\widetilde{X}_k\}$ 的一组压缩表示，其中 $\widetilde{X}_j$ 是 $\boldsymbol{U}_j$ 的压缩变量. 类似于原 IB 方法，多变量 IB 方法在对变量进行压缩的同时，为使压缩变量有效地获取数据中所蕴含的某种内在模式，压缩变量 $\widetilde{\boldsymbol{X}}=\{\widetilde{X}_1,\cdots,\widetilde{X}_k\}$ 应该最大化地保存变量集合 $\boldsymbol{X}=\{X_1,\cdots,X_n\}$ 中某些变量所提供的相关信息. 变量之间的压缩关系称为期望压缩关系，变量之间的相关模式保存关系称为期望模式保存关系，它们共同组成多变量 IB 方法的协作模型.

多变量 IB 采用贝叶斯网 $G_{\text{in}}$ 来刻画变量之间的期望压缩关系，采用 $G_{\text{out}}$ 来刻画变量之间的期望模式保存关系. 贝叶斯网是一个有向无环图 $G=(V,E)$，

其中 $V$ 为节点集合，由一组随机变量 $\{X_1,\cdots,X_n\}$ 组成，$E$ 为边的集合，描述变量间的相互依赖关系. 如果 $(X_i,X_j)\in E$，则节点 $X_i$ 到 $X_j$ 存在有向边，$X_i$ 为 $X_j$ 的父节点，$X_j$ 为 $X_i$ 的子节点. 记变量节点 $X_i$ 在贝叶斯网 $G$ 中的父节点集为 $\boldsymbol{Pa}_{X_i}^{G}$. 服从贝叶斯网依赖关系的随机变量集合 $\{X_1,\cdots,X_n\}$ 的联合分布可分解为[15]

$$p(X_1,\cdots,X_n)=\prod_i p(X_i\mid \boldsymbol{Pa}_{X_i}^{G}) \tag{7}$$

一旦确定了 $G_{\text{in}}$ 中 $\boldsymbol{X}$ 到 $\widetilde{\boldsymbol{X}}$ 的期望压缩关系，则 $\boldsymbol{X}$ 与 $\widetilde{\boldsymbol{X}}$ 之间的联合分布可由式(8)求得[15]

$$p(\boldsymbol{X},\widetilde{\boldsymbol{X}})=p(\boldsymbol{X})\prod_{j=1}^{k} p(\widetilde{X}_j\mid \boldsymbol{Pa}_{\widetilde{X}_j}^{G_{\text{in}}}) \tag{8}$$

其中 $\boldsymbol{Pa}_{\widetilde{X}_j}^{G_{\text{in}}}=\boldsymbol{U}_j$. 多变量IB方法通过求解 $p(\widetilde{X}_j\mid \boldsymbol{Pa}_{\widetilde{X}_j}^{G_{\text{in}}})$ 来确定变量 $\boldsymbol{U}_j$ 到 $\widetilde{X}_j$ 的压缩关系.

多变量 IB 方法分别采用多信息 $I^{G_{\text{in}}}$ 和 $I^{G_{\text{out}}}$ 来度量 $G_{\text{in}}$ 中变量之间的压缩程度和 $G_{\text{out}}$ 中的相关模式的保存程度. 文献[15]给出如式(9)的多变量 IB 目标函数，其中 $\beta$ 为大于 0 的平衡因子.

$$\mathcal{L}_{\min}=I^{G_{\text{in}}}[p(\boldsymbol{X},\widetilde{\boldsymbol{X}})]-\beta I^{G_{\text{out}}}[p(\boldsymbol{X},\widetilde{\boldsymbol{X}})] \tag{9}$$

为更有效地计算 $G_{\text{in}}$ 和 $G_{\text{out}}$ 中联合分布 $p(\boldsymbol{X},\widetilde{\boldsymbol{X}})$ 内多个变量之间的信息量，文献[15]给出如下定义.

**定义 1.** 如果 $\boldsymbol{X}=\{X_1,\cdots,X_n\}$ 服从联合分布 $p(\boldsymbol{X})$，并且 $\boldsymbol{X}$ 中的节点服从贝叶斯网 $G$ 所刻画的变量之间的依赖关系，则服从贝叶斯网 $G$ 的联合分布 $p(\boldsymbol{X})$ 中的多信息可分解为

$$I^{G}[p(\boldsymbol{X})]=\sum_i I(X_i;\boldsymbol{Pa}_{X_i}^{G}) \tag{10}$$

其中每一个互信息 $I(X_i;\boldsymbol{Pa}_{X_i}^{G})$ 中的联合分布 $p(X_i;\boldsymbol{Pa}_{X_i}^{G})$ 可通过 $p(\boldsymbol{X})$ 的边缘分布计算得到.

### 2.3 对称 IB

对称 IB[15] 将 IB 方法的数据单向压缩拓展到数据的双向压缩中，其协作模型如图 2 所示. 图 2(a) 中的贝叶斯网 $G_{\text{in}}$ 描述了变量之间的期望压缩关系，其中变量 $X$ 到变量 $Y$ 之间的箭头表示变量 $X$ 与 $Y$ 之间服从联合分布 $p(X,Y)$，该箭头也可由变量 $Y$ 到变量 $X$；变量 $X$ 到变量 $\widetilde{X}$、变量 $Y$ 到变量 $\widetilde{Y}$ 之间的箭头表示对称 IB 力图做变量 $X$ 到变量 $\widetilde{X}$、变量 $Y$ 到变量 $\widetilde{Y}$ 的双向压缩，其中 $X$ 为 $\widetilde{X}$ 的父节点、$\widetilde{X}$ 为 $X$ 的子节点，变量 $Y$ 与 $\widetilde{Y}$ 之间也具备类似的关系. 图 2(b) 中的 $G_{\text{out}}$ 描述了压缩变量之间的期望模式保存关系. 从该图中可以看出，对称 IB 力图保存压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间的信息量，变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间的箭头方向可互换. 为求得变量 $X$ 到 $\widetilde{X}$ 的压缩模式 $p(\tilde{x}\mid x)$ 及变量 $Y$ 到 $\widetilde{Y}$ 的压缩模式 $p(\tilde{y}\mid y)$，在期望模式保存关系图 $G_{\text{out}}$ 中，压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 相互为对方提供特征模式，即压缩变量 $\widetilde{Y}$ 中所承载的信息为变量 $X$ 到 $\widetilde{X}$ 的压缩提供了依据，压缩变量 $\widetilde{X}$ 中所承载的信息为变量 $Y$ 到 $\widetilde{Y}$ 的压缩提供了依据. 在学习过程中，变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间相互协作，共同挖掘联合分布 $p(X,Y)$ 中所蕴含的内在模式.

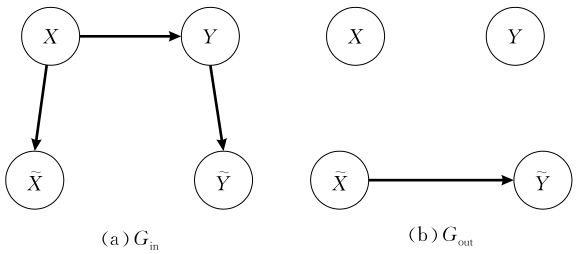



图 2 对称 IB 协作模型

式(11)为对称 $IB$ 的目标函数，其中 $I(X;\widetilde{X})+I(Y;\widetilde{Y})$ 度量了 $G_{\text{in}}$ 中 $X$ 到 $\widetilde{X}$、$Y$ 到 $\widetilde{Y}$ 的压缩程度，而 $I(\widetilde{X};\widetilde{Y})$ 度量了 $G_{\text{out}}$ 期望模式的保留程度，通过 $\beta$ 调节二者的平衡程度.

$$\mathcal{L}_{\max}=I(\widetilde{X};\widetilde{Y})-\beta^{-1}[I(X;\widetilde{X})+I(Y;\widetilde{Y})] \tag{11}$$

对称 IB 可用于解决机器学习领域中的联合聚类问题，这里给出对称 IB 与经典的信息论联合聚类 ITCC[23] 之间的关联. ITCC 在做联合聚类时，力图使压缩前后联合分布 $p(X,Y)$ 和 $p(\widetilde{X},\widetilde{Y})$ 中所包含的信息损失最少，采用互信息来度量聚类前后 $p(X,Y)$ 与 $p(\widetilde{X},\widetilde{Y})$ 内的信息量，其目标函数如式(12)所示. 在该目标函数中，给定联合分布 $p(X,Y)$，$I(X;Y)$ 的值不变，为一个确定的常量. 此时，目标函数(12)等同于式(13). 如果在对称 IB 目标函数(11)中 $\beta$ 取值为 $\infty$(无穷大)时，则对称 IB 的目标函数与信息论联合聚类的目标函数一致. 因此，文献[23]中的信息论联合聚类可视为对称 IB 的一个特例.

$$\mathcal{L}_{\text{mim}}=I(X;Y)-I(\widetilde{X};\widetilde{Y}) \tag{12}$$

$$\mathcal{L}_{\max}=I(\widetilde{X};\widetilde{Y}) \tag{13}$$

文献[19]将 IB 方法应用到联合聚类中. 该文献虽然在期望压缩模式保存关系图中将联合分布 $p(X,Y)$ 中 $X$ 和 $Y$ 之间相互蕴含的基层特征信息引入到联合聚类中，但是该算法采用层次凝聚的方法来优化目标函数，即通过不断合并最相似的两个簇(行变量的簇或列变量的簇)来实现联合聚类. 该优化过程时空复杂度高，且不能保证得到目标函数的一个局部优化解. 另外，文献[19]仅关注双向压缩中的聚类结果，而对双向压缩的另一个结果，压缩变量之间的高层统计规律，却没有做深入的研究，该模式从更高层面反映了簇与簇之间的关联性，具有较高的应用价值[27-31].

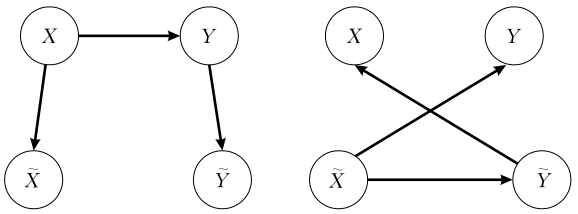
## 3　交叉对称 IB

图 2 中对称 IB 的协作模型 $G_{\text{in}}$ 描述了变量 $X$ 到 $\widetilde{X}$ 与变量 $Y$ 到 $\widetilde{Y}$ 的数据双向压缩关系，$G_{\text{out}}$ 刻画了压缩过程中所期望保存的特征模式. $G_{\text{in}}$ 和 $G_{\text{out}}$ 表明，在压缩学习过程中，$X$ 到 $\widetilde{X}$ 的压缩编码 $p(\tilde{x}|x)$ 是依据 $\widetilde{Y}$ 所提供的特征信息确定的，$Y$ 到 $\widetilde{Y}$ 的压缩 $p(\tilde{y}|y)$ 是依据 $\widetilde{X}$ 所提供的信息确定的. 然而，$\widetilde{X}$ 和 $\widetilde{Y}$ 分别是 $X$ 与 $Y$ 被压缩之后的变量，它们并不能完全承载基层变量 $X$ 与 $Y$ 中所固有特征信息. 因此，仅考虑压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间协作关系的对称 IB 存在着特征信息损失的问题，进而可能导致所得的双向压缩模式偏离数据中所蕴含的内在模式.

本节在多变量 IB 的基础上，重点研究数据的双向压缩，提出一个交叉对称 IB 协作模型 ICSIB. ICSIB 算法在做双向压缩时，不仅关注压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 高层特征之间的关联模式，同时还将变量 $X$ 与 $Y$ 中所蕴含的基层特征信息引入到数据的双向压缩中，使它们协助 $\widetilde{X}$ 与 $\widetilde{Y}$ 共同来挖掘数据中的双向压缩模式，从而更充分地利用数据中所蕴含的有价值信息.

### 3.1　ICSIB 协作模型

ICSIB 的协作模型如图 3 所示，该协作模型的 $G_{\text{in}}$ 描述了 $X$ 到 $\widetilde{X}$ 及 $Y$ 到 $\widetilde{Y}$ 的压缩关系，$G_{\text{out}}$ 描述了压缩变量所期望的相关模式保存关系. 与图 2 中的对称 IB 协作模型相比，在图 3 的 $G_{\text{out}}$ 中多了两条交叉的 $\widetilde{X}\to Y$ 与 $\widetilde{Y}\to X$，以使压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 分别交叉地保存变量 $Y$ 与 $X$ 所提供的基层特征信息，从而使基层特征变量协助压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 共同学习数据中所蕴含的双向压缩模式. 因此，ICSIB 所得的期望压缩编码 $p(\tilde{x}|x)$ 和 $p(\tilde{y}|y)$ 不仅反映了压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间相互蕴含的高层特征信息，同时还交叉地反映了承载在变量 $X$ 与 $Y$ 中数据原有的基层特征信息，从而解决了对称 IB 特征信息损失的问题.




(a) $G_{\text{in}}$　　(b) $G_{\text{out}}$

图 3　交叉对称 IB 协作模型

图 3 中的贝叶斯网 $G_{\text{in}}$ 和 $G_{\text{out}}$ 分别描述了变量之间的期望压缩关系与期望模式保存关系. 在 $G_{\text{in}}$ 中，变量 $X$ 的父节点集合为空，即 $\boldsymbol{Pa}_X^{G_{\text{in}}}=\varnothing$，变量 $\widetilde{X}$ 的父节点集合 $\boldsymbol{Pa}_{\widetilde{X}}^{G_{\text{in}}}=\{X\}$，变量 $Y$ 的父节点集合 $\boldsymbol{Pa}_Y^{G_{\text{in}}}=\{X\}$，变量 $\widetilde{Y}$ 的父节点结合 $\boldsymbol{Pa}_{\widetilde{Y}}^{G_{\text{in}}}=\{Y\}$. 根据定义 1 中的式(10)可将图 3(a)中 $G_{\text{in}}$ 的期望压缩信息 $I^{G_{\text{in}}}$ 分解为

$$I^{G_{\text{in}}}=I(X;\widetilde{X})+I(Y;\widetilde{Y})+I(X;Y) \tag{14}$$

在图 3(b)的 $G_{\text{out}}$ 中，$\boldsymbol{Pa}_X^{G_{\text{in}}}=\{\widetilde{Y}\}$，$\boldsymbol{Pa}_Y^{G_{\text{in}}}=\{\widetilde{X}\}$，$\boldsymbol{Pa}_{\widetilde{X}}^{G_{\text{in}}}=\varnothing$，$\boldsymbol{Pa}_{\widetilde{Y}}^{G_{\text{in}}}=\{\widetilde{X}\}$. 根据定义 1 中的式(10)可将图 3(b)中 $G_{\text{out}}$ 的期望模式保存信息 $I^{G_{\text{out}}}$ 分解为

$$I^{G_{\text{out}}}=I(\widetilde{X};\widetilde{Y})+I(\widetilde{X};Y)+I(X;\widetilde{Y}) \tag{15}$$

将式(14)与式(15)中的 $I^{G_{\text{in}}}$ 和 $I^{G_{\text{out}}}$ 代入多变量 IB 目标函数(9)中，并在等式两边同时除以 $-\beta$，可得式(16)所示交叉对称 IB 的目标函数. 由于 $p(X,Y)$ 已知，在式(14)中的 $I(X;Y)$ 为一个固定的值. 在不影响数据分析性能的情况下，为简化目标函数，在式(16)中省去 $I(X;Y)$ 这一项.

$$\begin{aligned}\mathcal{L}_{\max}[p(\tilde{x}|x),p(\tilde{y}|y)]=&\\ I(\widetilde{X};\widetilde{Y})+I(\widetilde{X};Y)+I(X;\widetilde{Y})-&\\ \beta^{-1}[I(X;\widetilde{X})+I(Y;\widetilde{Y})]&\end{aligned} \tag{16}$$

下面给出目标函数(16)中联合分布 $p(x,\tilde{x})$、$p(y,\tilde{y})$、$p(\tilde{x},\tilde{y})$、$p(\tilde{x},y)$ 及 $p(x,\tilde{y})$ 的计算公式. 首先，根据式(8)可对 $G_{\text{in}}$ 中变量之间的联合分布进行分解，得到

$$p(X,Y,\widetilde{X},\widetilde{Y})=p(\widetilde{X}|X)p(\widetilde{Y}|Y)p(X,Y) \tag{17}$$

其中 $p(X,Y)$ 已知，由此可得

$$\begin{aligned}&p(x,\tilde{x})=p(x)p(\tilde{x}|x),\ p(y,\tilde{y})=p(y)p(\tilde{y}|y),\\ &p(\tilde{x},\tilde{y})=\sum_x\sum_y p(x,y,\tilde{x},\tilde{y})\\ &\qquad=\sum_x\sum_y p(\tilde{x}|x)p(\tilde{y}|y)p(x,y),\\ &p(\tilde{x},y)=\sum_x\sum_{\tilde{y}} p(x,y,\tilde{x},\tilde{y})\\ &\qquad=\sum_x\sum_{\tilde{y}} p(\tilde{x}|x)p(\tilde{y}|y)p(x,y)\\ &\qquad=\sum_x p(\tilde{x}|x)p(x,y),\\ &p(x,\tilde{y})=\sum_y p(\tilde{y}|y)p(x,y).\end{aligned}$$

在双向压缩的应用中，由于 $|\widetilde{X}|\ll|X|$，$|\widetilde{Y}|\ll|Y|$，$X$ 到 $\widetilde{X}$ 及 $Y$ 到 $\widetilde{Y}$ 本身就是很大的压缩. 此时我们可将重点放在相关信息的保存上，$\beta$ 可取值为无穷大，交叉对称 IB 的目标函数可简化为

$$\mathcal{L}_{\max}[p(\tilde{x}|x),p(\tilde{y}|y)]=I(\widetilde{X};\widetilde{Y})+I(\widetilde{X};Y)+I(X;\widetilde{Y}) \tag{18}$$

在此，我们将其称为基于最大化交叉互信息的对称 IB.

### 3.2 ICSIB 算法

ICSIB 算法在做双向压缩时，采用交错的“抽取-合并”顺序迭代过程对目标函数(18)进行优化，来求解 $X$、$Y$ 到 $\widetilde{X}$、$\widetilde{Y}$ 的最优压缩表示 $p(\tilde{x}|x)$ 和 $p(\tilde{y}|y)$. 在寻求 $X$ 到 $\widetilde{X}$ 的最优压缩 $p(\tilde{x}|x)$ 时，$p(\tilde{y}|y)$ 保持不变，而在求 $p(\tilde{y}|y)$ 时，$p(\tilde{x}|x)$ 保持不变. 通过交错地“抽取-合并”顺序迭代过程来优化 ICSIB 目标函数(18). 本文仅考虑“硬划分”，即 $p(\tilde{x}|x)$ 和 $p(\tilde{y}|y)$ 取值仅为 0 或 1，此时，双向压缩即为将 $\mathcal{X}$ 划分为 $|\widetilde{X}|$ 个簇，将 $\mathcal{Y}$ 划分为 $|\widetilde{Y}|$ 个簇. 在 $p(\tilde{y}|y)$ 保持不变的情况下，首先介绍“抽取-合并”顺序优化 $p(\tilde{x}|x)$ 的过程. 在已知 $\mathcal{X}$ 的某一划分 $p(\tilde{x}|x)$ 上，记为 $\widetilde{X}^{\text{old}}$，迭代的将 $\mathcal{X}$ 中的每个元素 $x$ 顺序地从其所在的簇 $\tilde{x}^{\text{old}}: p(\tilde{x}^{\text{old}}|x)=1$ 中“抽取”出来，将其视为单独的一个簇，此时共有 $|\widetilde{X}|+1$ 个簇，我们将此时的 $|\widetilde{X}|+1$ 个簇表示为 $\widetilde{X}^{\text{med}}$. 为确保最终结果为 $|\widetilde{X}|$ 个簇，再将 $x$“合并”到满足 $\tilde{x}^{\text{new}}=\arg\min_{\tilde{x}} d_L(\{x\},\tilde{x})$ 的新簇 $\tilde{x}^{\text{new}}$ 中，其中 $d_L$ 表示将 $x$ 合并到 $\tilde{x}$ 时目标函数的减少量. 在 $p(\tilde{x}|x)$ 保持不变的情况下，“抽取-合并”顺序优化 $p(\tilde{y}|y)$ 的过程同上. 整个 ICSIB 算法交错“抽取-合并”顺序迭代的过程如算法 1 所示.

**算法 1.** ICSIB 算法.

输入：联合分布 $p(X,Y)$；$\mathcal{X}$ 的划分数目 $k$；$\mathcal{Y}$ 的划分数目 $l$

输出：$X$ 到 $\widetilde{X}$ 的压缩表示 $p(\tilde{x}|x)$；$Y$ 到 $\widetilde{Y}$ 的压缩表示 $p(\tilde{y}|y)$；压缩表示之间的高层统计规律 $p(\widetilde{X},\widetilde{Y})$

1. $p(\tilde{x}|x)\leftarrow X$ 的初始划分；
2. $p(\tilde{y}|y)\leftarrow Y$ 的初始划分；
3. REPEAT
4. FOR every $x\in\mathcal{X}$
5. 将 $x$ 从当前簇 $\tilde{x}(p(\tilde{x}|x)=1)$ 中抽取出来；
6. 将 $\{x\}$ 合并到簇 $\tilde{x}^{\text{new}}$ 中，其中 $\tilde{x}^{\text{new}}=\arg\min_{\tilde{x}} d_L(\{x\},\tilde{x})$，$d_L(\{x\},\tilde{x})$ 的计算如式(22)；
7. END
8. FOR every $x\in\mathcal{Y}$
9. 将 $y$ 从当前簇 $\tilde{y}(p(\tilde{y}|y)=1)$ 中抽取出来；
10. 将 $\{y\}$ 合并到簇 $\tilde{y}^{\text{new}}$ 中，其中 $\tilde{y}^{\text{new}}=\arg\min_{\tilde{y}} d_L(\{y\},\tilde{y})$ $d_L(\{y\},\tilde{y})$ 的计算如式(23)；
11. END
12. UNTIL 划分 $p(\tilde{x}|x)$ 与 $p(\tilde{y}|y)$ 不再有新的变化
13. 根据公式 $p(\tilde{x},\tilde{y})=\sum_x\sum_y p(\tilde{x}|x)p(\tilde{y}|y)p(x,y)$ 计算 $p(\widetilde{X},\widetilde{Y})$

为使 ICSIB 算法快速收敛到目标函数的一个局部优值，在每次合并时都要确保目标函数中信息量损失最少，即新簇 $\tilde{x}^{\text{new}}$ 和 $\tilde{y}^{\text{new}}$ 应满足 $\tilde{x}^{\text{new}}=\arg\min_{\tilde{x}} d_L(\{x\},\tilde{x})$ 和 $\tilde{y}^{\text{new}}=\arg\min_{\tilde{y}} d_L(\{y\},\tilde{y})$. 假设 $p(\tilde{y}|y)$ 固定不变，将 $x$ 从 $\tilde{x}^{\text{old}}: p(\tilde{x}^{\text{old}}|x)=1$ 中“抽取”出来，将其视为单独的一个簇 $\{x\}$，并将其合并到簇 $\tilde{x}$ 中，形成新簇 $\tilde{x}^{\text{new}}$，则

$$p(\tilde{x}^{\text{new}})=p(x)+p(\tilde{x}) \tag{19}$$

$$p(y|\tilde{x}^{\text{new}})=\frac{p(x)}{p(\tilde{x}^{\text{new}})}p(y|x)+\frac{p(\tilde{x})}{p(\tilde{x}^{\text{new}})}p(y|\tilde{x}) \tag{20}$$

$$p(\tilde{y}|\tilde{x}^{\text{new}})=\frac{p(x)}{p(\tilde{x}^{\text{new}})}p(\tilde{y}|x)+\frac{p(\tilde{x})}{p(\tilde{x}^{\text{new}})}p(\tilde{y}|\tilde{x}) \tag{21}$$

当 $p(\tilde{y}|y)$ 固定不变时，$p(\tilde{x}|x)$ 的优化过程仅涉及到目标函数(18)中 $I(\widetilde{X};Y)$ 与 $I(\widetilde{X};\widetilde{Y})$ 两项的值，而 $I(X;\widetilde{Y})$ 不变. 因此，目标函数的减少量 $d_L(\{x\},\tilde{x})$ 可计算为

$$\begin{aligned}d_L((x),\tilde{x})&=\Delta\mathcal{L}=\mathcal{L}^{\text{med}}-\mathcal{L}^{\text{new}}\\&=I(\widetilde{X}^{\text{med}};Y)+I(\widetilde{X}^{\text{med}};\widetilde{Y})-\\&\quad[I(\widetilde{X}^{\text{new}};Y)+I(\widetilde{X}^{\text{new}};\widetilde{Y})]\\&=[I(\widetilde{X}^{\text{med}};Y)-I(\widetilde{X}^{\text{new}};Y)]+\\&\quad[I(\widetilde{X}^{\text{med}};\widetilde{Y})-I(\widetilde{X}^{\text{new}};\widetilde{Y})]\\&=\Delta I_1+\Delta I_2,\end{aligned}$$

其中

$$\begin{aligned}\Delta I_1&=I(\widetilde{X}^{\text{med}};Y)-I(\widetilde{X}^{\text{new}};Y)\\&=p(x)\sum_y p(y|x)\log\frac{p(y|x)}{p(y)}+\\&\quad p(\tilde{x})\sum_y p(y|\tilde{x})\log\frac{p(y|\tilde{x})}{p(y)}-\\&\quad p(\tilde{x}^{\text{new}})\sum_y p(y|\tilde{x}^{\text{new}})\log\frac{p(y|\tilde{x}^{\text{new}})}{p(y)},\end{aligned}$$

将式(19)、(20)代入上式得

$$\begin{aligned}\Delta I_1&=p(x)\sum_y p(y|x)\log\frac{p(y|x)}{p(y)}+\\&\quad p(\tilde{x})\sum_y p(y|\tilde{x})\log\frac{p(y|\tilde{x})}{p(y)}-\\&\quad\sum_y p(x)p(y|x)\log\frac{p(y|\tilde{x}^{\text{new}})}{p(y)}-\\&\quad\sum_y p(\tilde{x})p(y|\tilde{x})\log\frac{p(y|\tilde{x}^{\text{new}})}{p(y)}\\&=p(x)\sum_y p(y|x)\log\frac{p(y|x)}{p(y|\tilde{x}^{\text{new}})}+\\&\quad p(\tilde{x})\sum_y p(y|\tilde{x})\log\frac{p(y|\tilde{x})}{p(y|\tilde{x}^{\text{new}})}\\&=p(x)D_{KL}[p(Y|x)\|p(Y|\tilde{x}^{\text{new}})]+\\&\quad p(\tilde{x})D_{KL}[p(Y|\tilde{x})\|p(Y|\tilde{x}^{\text{new}})]\\&=[p(x)+p(\tilde{x})]\cdot JS[p(Y|x),p(Y|\tilde{x})],\end{aligned}$$

其中 $JS[p(Y|x),p(Y|\tilde{x})]$ 为条件分布 $p(Y|x)$ 与 $p(Y|\tilde{x})$ 之间的 $JS$(Jensen Shannon)距离[33]. 由此类推可得

$$\Delta I_2=[p(x)+p(\tilde{x})]\cdot JS[p(\tilde{Y}|x),p(\tilde{Y}|\tilde{x})].$$

因此,在将 $\{x\}$ 合并到簇 $\tilde{x}$ 中时,目标函数的信息损失量 $d_L(\{x\},\tilde{x})$ 为

$$d_L(\{x\},\tilde{x})=[p(x)+p(\tilde{x})]\cdot\{JS[p(Y|x),p(Y|\tilde{x})]+JS[p(\tilde{Y}|x),p(\tilde{Y}|\tilde{x})]\} \tag{22}$$

同理,当 $p(\tilde{x}|x)$ 固定不变时,在优化 $p(\tilde{y}|y)$ 的过程中,将 $y$ 从 $\tilde{y}^{\text{old}}:p(\tilde{y}^{\text{old}}|y)=1$ 中"抽取"出来,将其视为单独的一个簇 $\{y\}$,并将其合并到簇 $\tilde{y}$ 中,形成新簇 $\tilde{y}^{\text{new}}$,此时,目标函数的信息损失量 $d_L(\{y\},\tilde{y})$ 为

$$d_L(\{y\},\tilde{y})=[p(y)+p(\tilde{y})]\cdot\{JS[p(X|y),p(X|\tilde{y})]+JS[p(\tilde{X}|y),p(\tilde{X}|\tilde{y})]\} \tag{23}$$

### 3.3　算法分析

3.3.1　算法收敛性分析

**定理 1.**　在 ICSIB 算法中,采用 $\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}})$ 表示将一个元素从当前簇中"抽取"出来之前的目标函数值,采用 $\mathcal{L}(\tilde{X}^{\text{new}},\tilde{Y}^{\text{new}})$ 表示将"抽取"出来的元素重新"合并"到某个簇中的目标函数值,则有

$$\mathcal{L}(\tilde{X}^{\text{new}},\tilde{Y}^{\text{new}})\geq\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}}) \tag{24}$$

证明.　首先证明当 $p(\tilde{y}|y)$ 固定不变时,将 $x$ 从当前簇 $\tilde{x}^{\text{old}}:p(\tilde{x}^{\text{old}}|x)=1$ 中"抽取"出来,然后将其"合并"到簇 $\tilde{x}^{\text{new}}=\arg\min_{\tilde{x}} d_L(\{x\},\tilde{x})$ 中,此"抽取-合并"前后的目标函数值满足 $\mathcal{L}(\tilde{X}^{\text{new}},\tilde{Y}^{\text{old}})\geq\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}})$.

记将 $x$ 从当前簇 $\tilde{x}^{\text{old}}:p(\tilde{x}^{\text{old}}|x)=1$ 中"抽取"出来后的目标函数值为 $\mathcal{L}(\tilde{X}^{\text{med}},\tilde{Y}^{\text{old}})$. 在"合并"过程中,当 $\tilde{x}^{\text{new}}=\tilde{x}^{\text{old}}$ 时,$p(\tilde{x}|x)$ 没有改变,则有 $\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}})=\mathcal{L}(\tilde{X}^{\text{new}},\tilde{Y}^{\text{old}})$;当 $\tilde{x}^{\text{new}}\neq\tilde{x}$ 时,由于 $\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}})=\mathcal{L}(\tilde{X}^{\text{med}},\tilde{Y}^{\text{old}})-d_L(\{x\},\tilde{x}^{\text{old}})$,$\mathcal{L}(\tilde{X}^{\text{new}},\tilde{Y}^{\text{old}})=\mathcal{L}(\tilde{X}^{\text{med}},\tilde{Y}^{\text{old}})-d_L(\{x\},\tilde{x}^{\text{new}})$,而 $\tilde{x}^{\text{new}}=\arg\min_{\tilde{x}} d_L(\{x\},\tilde{x})$,则 $d_L(\{x\},\tilde{x}^{\text{old}})>d_L(\{x\},\tilde{x}^{\text{new}})$,从而 $\mathcal{L}(\tilde{X}^{\text{new}},\tilde{Y}^{\text{old}})>\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}})$. 因此当 $p(\tilde{y}|y)$ 固定不变时每次"抽取-合并"前后的目标函数值满足 $\mathcal{L}(\tilde{X}^{\text{new}},\tilde{Y}^{\text{old}})\geq\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}})$.

同理可得,当 $p(\tilde{x}|x)$ 固定不变,将 $y$ 从当前簇 $\tilde{y}^{\text{old}}:p(\tilde{y}^{\text{old}}|y)=1$ 中"抽取"出来,并将其合并到簇 $\tilde{y}^{\text{new}}=\arg\min_{\tilde{y}} d_L(\{y\},\tilde{y})$ 时,$\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{new}})\geq\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}})$.

因此,在 ICSIB 算法的每一步"抽取"前与"合并"后,都有 $\mathcal{L}(\tilde{X}^{\text{new}},\tilde{Y}^{\text{new}})\geq\mathcal{L}(\tilde{X}^{\text{old}},\tilde{Y}^{\text{old}})$.　证毕.

**推论 1.**　ICSIB 算法可在有限步骤内收敛到交叉对称 IB 目标函数(18)的一个局部优解.

证明.　由定理 1 可知,ICSIB 算法的每一步"抽取-合并"过程都能增加交叉对称 IB 目标函数(18)的值,而 $I(\tilde{X};\tilde{Y})\leq I(X;Y)$,$I(\tilde{X};Y)\leq I(X;Y)$,$I(X;\tilde{Y})\leq I(X;Y)$,因此该目标函数是有界的,即 $\mathcal{L}\leq 3\cdot I(X;Y)$. 由此可得出,ICSIB 算法可在有限步骤内收敛到交叉对称 IB 目标函数的一个局部优解.　证毕.

3.3.2　算法时间复杂度分析

ICSIB 算法的第 5 步将数据元素从当前簇中抽取出来,该过程可在单位时间内完成,其时间复杂度为 $O(1)$;第 6 步为算法的优化合并过程,该过程需要计算当前数据对象到每一簇质心分布的 $JS$ 的距离,该步骤的时间复杂度分别为 $O(k|\mathcal{Y}|)$,因此第 4 步到第 7 步的时间复杂度为 $O(k|\mathcal{X}||\mathcal{Y}|)$. 类似地,第 8 步到第 11 步的时间复杂度为 $O(l|\mathcal{X}||\mathcal{Y}|)$. 整个算法的时间复杂度为 $O(L(k+l)|\mathcal{X}||\mathcal{Y}|)$,其中 $L$ 是 ICSIB 算法找到局部优化解迭代循环的次数. 综上所述,ICSIB 算法的时间复杂度和数据的规模线性相关.

ICSIB 算法是对对称 IB 算法[15]及 ITCC 算法[23]的拓展,后两者的时间复杂同为 $O(L(k+l)|\mathcal{X}||\mathcal{Y}|)$. 由此可见,增加基层特征变量之间的交叉项并没有增加算法的时间复杂度. aIBCC 算法在做联合聚类时同样引入基层特征变量之间所蕴含的特征信息,但该算法采用层次凝聚的过程来优化目标函数,其时间复杂度为 $O(|\mathcal{X}||\mathcal{Y}|(|\mathcal{X}|^2+|\mathcal{Y}|^2))$. 由此可见,ICSIB 算法的时间复杂度远低于 aIBCC 算法.

## 4　实验与性能分析

本文从文档联合聚类、图像无监督分类和高层模式提取 3 个方面来评估 ICSIB 算法数据分析的性能.

### 4.1　数据集选择及预处理

(1) 文档数据

20 Newsgroups① 包含大约 20000 篇新闻文档,可划分为 20 个新闻话题组. 本文从该数据集中抽取 9 个数据子集作为实验数据,其中每个数据集由 500 篇文档组成,它们是随机地从 20 Newsgroups 中的一些新闻话题文档中选出的.

对选出的文档进行预处理,主要包括大写字母转化为小写字母;所有阿拉伯数字合并为单一的数字符号;删除非希腊数字符号、停用词(stop words)

① http://people.csail.mit.edu/jrennie/20Newsgroups/

和仅出现一次的单词;选择对互信息贡献最大的2000个单词作为文档的相关变量集[2].

(2)图像数据

本文采用文献[34]中所使用的amazon,dslr,webcam① 三个图像数据集来验证ICSIB算法对图像无监督模式识别的性能,其中amazon中的图像是网络上的商标图像,dslr中的图像是由数码相机拍摄得到,webcam域中的图像是由网络摄像头拍摄得到,每个图像数据集均包含有31个类别的图像.

原始的图像数据以像素为单位,故而学习算法无法直接对其进行模式分析,因此需要事先对图像数据进行一些预处理,将其转换为算法1可分析的数据形式.本文采用Bag-of-Words的方法[11,34-35]将图像数据集转化为共现矩阵.该方法主要分为4个步骤:(1)采用SURF算法[36]从每幅图像中抽取局部特征,并用64维的SURF描述符来描述每一个局部特征;(2)采用$k$-means算法量化SURF描述符,构建一个规模为800的码本,每一个码本可视为一个视觉单词(visual word);(3)把第一步从图像中抽取的每一个SURF描述符映射到上述码本中,将局部特征转化为视觉单词;(4)统计每个视觉单词在每幅图像中出现的次数,将图像数据集转化为共现矩阵.

本文所使用的文档和图像数据集具体描述如表1所示.

**表1 数据集描述**

| 数据集名称 | 规模($\|X\|$) | 特征数目($\|Y\|$) | 真实类别数 |
|---|---|---|---|
| Binary_1,2,3 | 500 | 2000 | 2 |
| Multi5_1,2,3 | 500 | 2000 | 5 |
| Multi10_1,2,3 | 500 | 2000 | 10 |
| dslr | 498 | 800 | 31 |
| webcam | 795 | 800 | 31 |
| amazon | 2813 | 800 | 31 |

ICSIB算法、对称IB算法[15]、ITCC算法[23]、aIBCC算法[19]均需要事先给出数据的联合分布.通过上述方法对文档及图像数据集预处理之后,可得到文档与单词、图像与视觉单词之间的共现矩阵.通过式(25)便可得到上述数据集的联合分布,其中$n(x,y)$为单词(视觉单词)$y$在文档(图像)$x$中出现的次数.

$$p(x,y)=\frac{n(x,y)}{\sum_x\sum_y n(x,y)} \tag{25}$$

## 4.2 实验设计

### 4.2.1 对比方法

为验证ICSIB算法对数据进行双向压缩的性能,本文给出该算法与以下算法的对比实验结果.

(1)$k$-means算法.首先对数据集进行L2规范化(L2 norm)预处理,然后采用$k$-means算法对数据对象进行聚类.

(2)Normalized Cuts(NCuts)算法[32].该算法为基于图分割的聚类算法,以数据间的相似度矩阵作为输入数据.实验中首先对数据集进行L2规范化预处理,数据间的相似度矩阵为数据对象间欧几里得距离的负值.

(3)对称IB(Sy_IB)[15].采用顺序迭代的方法对对称IB目标函数进行优化.

(4)信息论联合聚类算法(ITCC)[23].采用矩阵逼近(Matrix Approximations)的方法对目标函数进行优化.

(5)凝聚IB联合聚类算法(aIBCC)[19].数据对象与特征之间的联合分布作为输入数据,采用凝聚迭代的过程对目标函数进行优化.

(6)DRCC(Dual Regularized Co-Clustering)联合聚类算法[24].原始数据作为输入数据.

$k$-means算法和Normalized Cuts算法是传统的聚类算法,仅对数据对象进行聚类.ICSIB算法、ITCC算法、对称IB算法、aIBCC算法及DRCC算法则同时对矩阵的行和列进行聚类.

### 4.2.2 实验评估方法

本文采用准确度(Accuracy)[25]和标准化互信息(Normalized mutual information)[25]来度量上述算法所得的数据模式与已知数据类标签的拟合程度,从而来评估不同算法的性能.对于数据对象$x_i$,分别采用$\tilde{x}_i$和$c_i$表示$x_i$的簇标签和$x_i$的已知类标签,则准确度可定义如下:

$$AC=\frac{\sum_{i=1}^{m}\delta(c_i,\mathrm{map}(\tilde{x}_i))}{m} \tag{26}$$

其中$m$是数据集中数据对象的总数目;$\delta(x,y)$为狄垃克函数,当$x=y$时该值为1,否则该值为0;$map(\tilde{x}_i)$为将簇标签$\tilde{x}_i$映射到等价真实类标签的映射函数,最佳的映射关系可通过Kuhn-Munkres算法[25]得到.

采用$\widetilde{X}$和$C$分别表示簇标签和真实类标签,则压缩结果和已知类标签之间的互信息$I(\widetilde{X};C)$可由式(2)求得.本文采用标准化互信息来度量不同算法所得结果和已知簇标签之间的拟合程度,其定义为

① http://www.icsi.berkeley.edu/~saenko/projects.html#data

$$NMI=\frac{I(\tilde{X};C)}{\max(H(\tilde{X}),H(C))} \tag{27}$$

4.2.3　实验细节

上述 6 种对比算法及本文所提出的 ICSIB 算法在运行时均需要事先指定相应的参数. 表 2 给出了这些算法在运行时参数的选择，随机初始化及最终结果选择方法的详细信息，其中 NTCs(the Number of True Clusters)表示数据中集固有的簇数目，D 代表文档，I 代表图像. 另外，在本文实验中，行代表数据对象，列代表特征变量.

**表 2　算法的参数设置、随机初始化及最终结果选择方法**

| | 行簇数 | 列簇数 | 初始化 | 运行次数 | 最终结果 |
|---|---|---|---|---|---|
| $k$-means | NTCs | × | 随机 | 10 | 最好结果 |
| NCuts | NTCs | × | 随机 | 10 | 最好结果 |
| DRCC | NTCs | NTCs | $k$-means | 60 | 最好结果 |
| aIBCC | NTCs | 自动确定 | × | 1 | × |
| 对称 IB | NTCs | D:100 I:50 | | 10 | 最优目标函数 |
| ITCC | NTCs | D:100 I:50 | 相同的随机初始化 | 10 | 最优目标函数 |
| ICSIB | NTCs | D:100 I:50 | | 10 | 最优目标函数 |

为了发现数据中所蕴含的模式结构，所有的算法在运行时均需要事先指定数据中的簇数目. 本文实验中，行簇数设置为数据集内所固有的类别数. $k$-means 算法与 Normalized Cuts 算法均为单向聚类算法，仅需指定行簇数目即可. 这两个算法均运行 10 次，从中选择准确度最高的作为最终结果.

ICSIB 算法、ITCC 算法[23]、对称 IB 算法[15]这 3 个双向压缩算法在文档数据集上的列簇数目为 100，在图像数据集上列簇数目为 50. 另外这 3 个算法均需要事先给定一个初始划分，不同的初始划分可得到不同的聚类结果. 为了确保实验对比的公平性，避免有些算法在较好的初始划分上得到较好的聚类结果，在本文的实验中，首先对数据做 10 次随机初始划分，然后将这 10 个初始划分作为这 3 个算法的初始划分，分别运行 10 次，每个算法得到 10 组不同的压缩结果，从中选择目标函数最优的作为最终结果.

本文按照文献[24]中的参数选取方法来运行 DRCC 算法，列聚类簇数目与行聚类簇数目相同，均为数据集中真实类别数，并且采用 $k$-means 算法来初始行与列的划分. 另外该算法需要指定近邻的数目 $k$ 及规则化参数 $\lambda$ 和 $\mu$. 为了选取适合的参数值，实验中测试不同的 $k$ 取值及 $\lambda$ 和 $\mu$ 的取值来运行 DRCC 算法，其中 $k$ 的取值范围为{1,2,3,…,10}，$\lambda$ 和 $\mu$ 的取值范围为{0.1,1,10,100,500,1000}，且 $\lambda=\mu$. 因此针对不同的参数，在同一个数据集上 DRCC 算法运行 60 次，从中选择准确度最大的作为最终结果.

aIBCC 算法[19]为凝聚层次联合聚类算法，不需要事先给定初始划分. 在该算法中，仅需给出平衡参数 $\gamma$ 即可，实验中该参数取值为 1. 另外对称 IB 算法[15]与 ICSIB 算法中的平衡参数 $\beta=\infty$，此时对称 IB 算法的目标函数等同于 ITCC 算法[23]的目标函数.

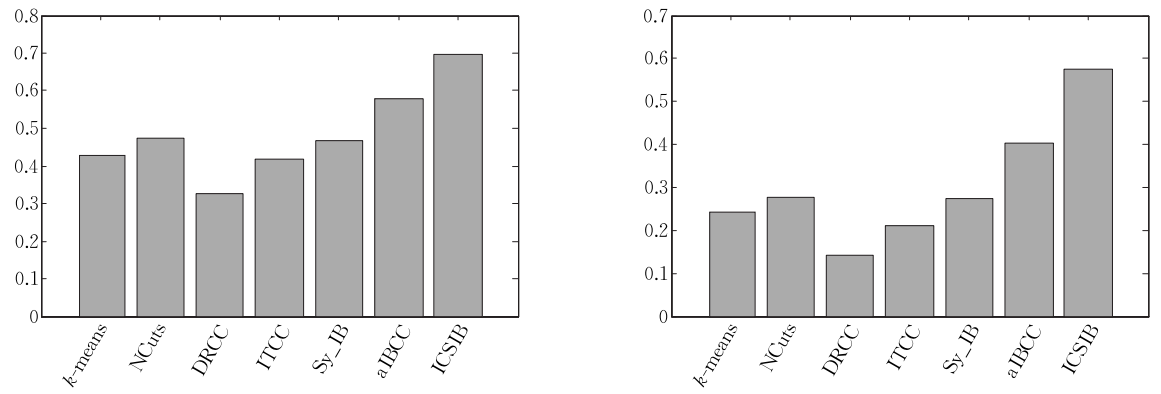
### 4.3　实验结果及性能分析

4.3.1　文档联合聚类

在文档数据集中，行数据代表文档，列代表单词. 由于实验中仅知道每篇文档所归属的主题类别，而无法确定单词的分类模式. 因此，本文实验也仅给出文档，即行数据的实验评估结果，如表 3 所示. 图 4 中的(a)和(b)分别给出了算法在所有文档数据集上联合聚类的准确度和标准化互信息的平均值.

**表 3　文档数据的实验结果**

| Data Sets | 准确度 $AC$/% | | | | | | |
|---|---|---|---|---|---|---|---|
| | $k$-means | NCuts | DRCC | ITCC | Sy-IB | aIBCC | ICSIB |
| Binary_1 | 59.0 | 76.4 | 52.4 | 80.8 | 81.4 | 79.4 | **92.4** |
| Binary_2 | 55.6 | 74.6 | 53.0 | 69.4 | 76.0 | 85.6 | **88.8** |
| Binary_3 | 60.8 | 84.0 | 54.6 | 74.8 | 85.0 | 84.2 | **91.4** |
| Multi5_1 | 47.2 | 44.0 | 30.0 | 31.6 | 38.8 | 56.6 | **90.2** |
| Multi5_2 | 47.2 | 47.0 | 27.0 | 34.0 | 42.6 | 73.4 | **89.6** |
| Multi5_3 | 49.4 | 42.2 | 31.4 | 41.0 | 45.8 | 70.8 | **94.0** |
| Multi10_1 | 33.8 | 36.8 | 18.4 | 24.2 | 23.8 | 43.6 | **56.0** |
| Multi10_2 | 29.6 | 36.4 | 19.8 | 22.2 | 27.2 | 43.0 | **54.6** |
| Multi10_3 | 31.4 | 31.6 | 14.0 | 23.6 | 26.8 | 41.2 | **59.8** |
| | 标准化互信息 $NMI$/% | | | | | | |
| | $k$-means | NCuts | DRCC | ITCC | Sy-IB | aIBCC | ICSIB |
| Binary_1 | 5.6 | 23.9 | 0.5 | 29.7 | 30.7 | 32.2 | **61.3** |
| Binary_2 | 1.4 | 18.2 | 0.9 | 11.2 | 20.9 | 40.7 | **49.4** |
| Binary_3 | 6.2 | 36.7 | 1.5 | 19.0 | 39.7 | 37.7 | **58.1** |
| Multi5_1 | 24.2 | 18.0 | 6.2 | 6.7 | 12.7 | 34.9 | **74.2** |
| Multi5_2 | 20.8 | 20.9 | 3.6 | 7.7 | 15.0 | 44.6 | **73.1** |
| Multi5_3 | 24.6 | 14.2 | 11.5 | 16.6 | 19.2 | 44.3 | **83.2** |
| Multi10_1 | 23.9 | 24.3 | 8.7 | 9.9 | 13.5 | 31.1 | **46.2** |
| Multi10_2 | 23.2 | 23.7 | 8.9 | 8.8 | 15.1 | 31.9 | **45.1** |
| Multi10_3 | 23.1 | 20.5 | 4.6 | 10.2 | 16.0 | 33.4 | **47.2** |




(a) 准确度 (b) 标准化互信息

图 4 文档联合聚类准确度及标准化互信息对比结果

表 3 和图 4 中的实验结果表明：

(1) 本文所提出的 ICSIB 算法在做文档联合聚类时，性能远优于单向聚类的 $k$-means 算法和 Normalized Cuts 算法.

(2) ITCC 算法和对称 IB 算法在做联合聚类时，仅关注行、列压缩变量之间的协作关系，却忽略了行、列基层变量之间相互为对方提供的特征信息，而 ICSIB 算法则从更多角度更充分地利用数据中所蕴含的有价值信息，其性能明显优于上述两种算法.

(3) aIBCC 算法的层次凝聚优化过程不能充分地优化目标函数，而 ICSIB 算法的交错的顺序迭代方法则更充分地优化了目标函数，其性能优于 aIBCC 算法.

### 4.3.2 图像无监督模式识别

表 4 给出了各算法在 dslr、webcam、amazon 这 3 个图像数据集上的实验结果. 该组实验结果表明：相对于其他 6 个对比算法，ICSIB 算法在图像模式识别中同样具有较强的识别能力.

**表 4 图像模式识别的实验结果**

| Data Sets | 准确度 $AC$/% | | | | | | |
|---|---|---|---|---|---|---|---|
| | $k$-means | NCuts | DRCC | ITCC | Sy-IB | aIBCC | ICSIB |
| dslr | 38.0 | 37.1 | 38.8 | 38.2 | 45.8 | 45.6 | **46.8** |
| webcam | 40.3 | 35.7 | 34.7 | 39.7 | 43.8 | 47.3 | **48.2** |
| amazon | 23.1 | 22.2 | 18.8 | 20.3 | 21.9 | 23.2 | **25.6** |
| | 标准化互信息 $NMI$/% | | | | | | |
| | $k$-means | NCuts | DRCC | ITCC | Sy-IB | aIBCC | ICSIB |
| dslr | 56.3 | 56.0 | 52.7 | 53.0 | 60.9 | 60.9 | **61.4** |
| webcam | 55.6 | 52.0 | 50.5 | 55.7 | 59.7 | 62.8 | **63.6** |
| amazon | 26.1 | 24.6 | 22.5 | 24.5 | 26.6 | 27.7 | **28.4** |

### 4.3.3 高层模式提取

双向压缩除了得到 $X$ 到 $\widetilde{X}$ 压缩表示 $p(\tilde{x}|x)$ 及 $Y$ 到 $\widetilde{Y}$ 的压缩表示 $p(\tilde{y}|y)$ 外，也可获取 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间的高层统计规律 $p(\widetilde{X},\widetilde{Y})$，而传统的单向聚类算法是不能获得该高层统计规律的. 在 $p(\widetilde{X},\widetilde{Y})$ 中，压缩变量 $\widetilde{X}$ 与 $\widetilde{Y}$ 之间的关联程度同样反映了双向压缩算法性能的优越程度. ITCC 算法和对称 IB 算法在做联合聚类时目标一致，均力图使行簇与列簇之间高度关联，并采用互信息 $I(\widetilde{X};\widetilde{Y})$ 作为双向压缩的目标函数. 本文提出的 ICSIB 算法不单力图使压缩变量之间高度关联，还力图使压缩变量中保留数据集内原有的模式结构. 为评估压缩变量之间的关联程度，本文采用互信息 $I(\widetilde{X};\widetilde{Y})$ 作为评估标准.

表 5 给出了 ICSIB、ITCC 及对称 IB 聚类算法所得双向压缩结果 $p(\widetilde{X},\widetilde{Y})$ 中互信息 $I(\widetilde{X};\widetilde{Y})$ 的值，该值度量了簇与簇之间的关联程度. 从该表中可以看出，比较这 3 个算法得到的簇与簇之间的关联程度，ICSIB 算法最优，其次为对称 IB 算法. 该组实验结果表明：

(1) 同样的目标函数，对称 IB 算法中的顺序迭代方法所得到的目标函数值优于 ITCC 算法中的矩阵逼近方法所得的目标函数值.

(2) ICSIB 算法在 12 个数据集中的 8 个数据集上取得了最优结果，并且 ICSIB 算法所得 $I(\widetilde{X};\widetilde{Y})$ 的平均值最优.

(3) ICSIB 算法在对称 IB 算法的目标函数的基

**表 5 互信息 $I(\widetilde{X};\widetilde{Y})$ 的对比结果**

| Data Sets | ITCC | Sy_IB | ICSIB |
|---|---|---|---|
| Binary_1 | 0.1848 | 0.2042 | **0.2044** |
| Binary_2 | 0.1688 | **0.1883** | 0.1791 |
| Binary_3 | 0.1835 | **0.2123** | 0.1936 |
| Multi5_1 | 0.4601 | 0.5318 | **0.6189** |
| Multi5_2 | 0.4681 | 0.5264 | **0.6199** |
| Multi5_3 | 0.4626 | 0.5352 | **0.6234** |
| Multi10_1 | 0.6137 | 0.7242 | **0.7748** |
| Multi10_2 | 0.6289 | 0.7270 | **0.7785** |
| Multi10_3 | 0.6092 | 0.7216 | **0.7810** |
| dslr | 0.4151 | **0.4509** | 0.4470 |
| webcam | 0.3321 | 0.3537 | **0.3576** |
| amazon | 0.5234 | **0.5483** | 0.5354 |
| average | 0.4209 | 0.4770 | **0.5095** |

础上，将数据中的原模式引入双向压缩中，在关注数据中原有模式的同时，同样可确保压缩变量之间高度关联.

4.3.4　算法收敛性实验

图 5 给出了 ICSIB 算法在 Multi10_1 数据集上每次迭代时 ICSIB 的目标函数值. 该图表明，ICSIB 算法的每次迭代都确保了目标函数值的增加，最终得到目标函数的一个局部优值. 在 Multi10_1 数据集上，ICSIB 算法经过 19 次迭代便得到了目标函数的一个局部优值，该算法具有很好的收敛性. 在本文其他数据集上的实验中，ICSIB 算法同样表现出很好的收敛性，本文在此并不全部给出.

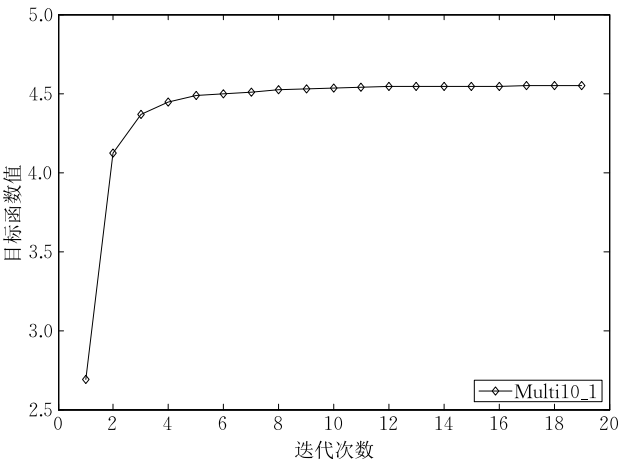



图 5　ICSIB 算法收敛性实验

## 5　总　结

对称 IB 在做双向压缩时，仅关注行、列压缩变量之间相互蕴含的特征模式，却忽略了压缩之前行、列基层变量相互为对方提供的特征信息. 行、列压缩变量相互为对方提供的特征信息是被压缩之后的信息，伴随着变量的压缩，数据中必有信息的损失，因此它们并不能完全承载数据中原有的特征信息. 针对该问题，在多变量 IB 方法的基础上，提出交叉对称 IB：ICSIB. ICSIB 除了关注压缩变量之间相互蕴含的特征模式外，还将压缩之前的原变量引入到对称 IB 中，使它们协助压缩变量来学习联合分布中的双向压缩模式. ICSIB 算法采用交错的“抽取-合并”顺序迭代过程对目标函数进行优化，理论上保证收敛到目标函数的一个局部优解，具有较低时间复杂度. 实验结果表明，ICSIB 算法得到的数据双向压缩模式更接近于数据中真实的内在模式，并可有效地应用于数据的联合聚类中.

## 参　考　文　献

[1] Tishby N, Pereira F, Bialek W. The information bottleneck method//Proceedings of the Allerton Conference on Communication, Control and Computing. Illinois, USA, 1999: 368-377

[2] Slonim N, Friedman N, Tishby N. Unsupervised document classification using sequential information maximization//Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval. Tampere, Finland, 2002: 129-136

[3] Thirion B, Faugeras O. Feature characterization in fMRI data: The information bottleneck approach. Medical Image Analysis, 2004, 8(4): 403-419

[4] Andritsos P, Tzerpos V. Information-theoretic software clustering. IEEE Transactions on Software Engineering, 2005, 31(2): 150-165

[5] Goldberger J, Gordon S, Greenspan H. Unsupervised image-set clustering using an information theoretic framework. IEEE Transactions on Image Processing, 2006, 15(2): 449-458

[6] Ye Yang-Dong, Liu Dong, Jia Li-Min, Li Gang. An sIB algorithm for automatically determining parameter. Chinese Journal of Computers, 2007, 30(6): 969-978(in Chinese)
(叶阳东，刘东，贾利民，Li Gang. 一种自动确定参数的 sIB 算法. 计算机学报，2007, 30(6): 969-978)

[7] Shen Hua-Wei, Cheng Xue-Qi, Chen Hai-Qiang, Liu Yue. Information bottleneck based community detection in network. Chinese Journal of Computers, 2008, 31(4): 677-686 (in Chinese)
(沈华伟，程学琪，陈海强，刘悦. 基于信息瓶颈的社区发现. 计算机学报，2008, 31(4): 677-686)

[8] Ling X, Xue G, Dai W, et al. Can Chinese web pages be classified with English data source?//Proceedings of the International Conference on World Wide Web. Beijing, China, 2008: 969-978

[9] Bardera A, Rigau J, Baoda I, et al. Image segmentation using information bottleneck method. IEEE Transactions on Image Processing, 2009, 18(7): 1601-1612

[10] Lazebnik S, Raginsky M. Supervised learning of quantizer codebooks by information loss minimization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(7): 1294-1309

[11] Lou Z, Ye Y, Liu D. Unsupervised object category discovery via information bottleneck method//Proceedings of the ACM International Conference on Multimedia. Firenze, Italy, 2010: 863-866

[12] Hecht R M, Noor E, Dobry G, et al. Effective model representation by information bottleneck principle. IEEE Transactions on Audio, Speech, and Language processing, 2013, 21(8): 1755-1759

[13] Lou Zheng-Zheng, Ye Yang-Dong, Liu Rui-Na. Non-redundant multi-view clustering based on information bottleneck. Journal

of Computer Research and Development, 2013, 50(9): 1865-1875(in Chinese)
(娄铮铮, 叶阳东, 刘瑞娜. 基于 IB 方法的无冗余多视角聚类. 计算机研究与发展, 2013, 50(9): 1865-1875)

[14] Friedman N, Mosenzon O, Slonim N, Tishby N. Multivariate information bottleneck//Proceedings of the Conference on Uncertainty in Artificial Intelligence. Washington, USA, 2001: 152-161

[15] Slonim N, Friedman N, Tishby N. Multivariate information bottleneck. Neural Computation, 2006, 18(8): 1739-1789

[16] Chechik G, Tishby N. Extracting relevant structures with side information//Proceedings of the Neural Information Processing Systems. Vancouver, Canada, 2002: 857-864

[17] Elidan G, Friedman N, Chickering D M. Learning hidden variable networks: The information bottleneck approach. Journal of Machine Learning Research, 2005, 6(1): 81-127

[18] Seldin Y, Slonim N, Tishby N. Information bottleneck for non co-occurrence data//Proceedings of the Neural Information Processing Systems. Vancouver, Canada, 2006: 1241-1248

[19] Wang P, Domeniconi C, Laskey K B. Information bottleneck co-clustering//Proceedings of the Text Mining Workshop, SIAM International Conference on Data Mining. Columbus, USA, 2010

[20] Riguzzi F, Di M N. Applying the information bottleneck to statistical relational learning. Machine Learning, 2012, 86(1): 89-114

[21] Lou Z, Ye Y, Yan X. The multi-feature information bottleneck with application to unsupervised image categorization//Proceedings of the 23rd International Joint Conference on Artificial Intelligence. Beijing, China, 2013: 1508-1515

[22] Dhillon I S. Co-clustering documents and words using bipartite spectral graph partitioning//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2001: 269-274

[23] Dhillon I S, Mallela S, Modha D S. Information-theoretic co-clustering//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, USA, 2003: 89-98

[24] Gu Q, Zhou J. Co-clustering on manifolds//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France, 2009: 359-368

[25] Zhang L, Chen C, Bu J, et al. Locally discriminative coclustering. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(6): 1025-1035

[26] Du L, Shen Y D. Towards robust co-clustering//Proceedings of the International Joint Conference on Artificial Intelligence. Beijing, China, 2013: 1317-1322

[27] Li Xiao-Guang, Yu Ge, Wang Da-Ling, Bao Yu-Bin. Latent concept extraction and text clustering based on information theory. Journal of Software, 2008, 19(9): 2276-2284(in Chinese)
(李晓光, 于戈, 王大玲, 鲍玉斌. 基于信息论的潜在概念获取与文本聚类. 软件学报, 2008, 19(9): 2276-2284)

[28] Liu J, Shah M. Scene modeling using co-clustering//Proceedings of the IEEE International Conference on Computer Vision. Rio de Janeiro, Brazil, 2007: 1-7

[29] Wang Peng, Yang Shi-Qiang, Liu Zhi-Qiang. Information-theoretic co-clustering for video shot categorization. Chinese Journal of Computers, 2005, 28(10): 1692-1699(in Chinese)
(王鹏, 杨士强, 刘志强. 信息论联合聚类算法及其在视频镜头聚类中的应用. 计算机学报, 2005, 28(10): 1692-1699)

[30] Dai W, Xue G R, Yang Q, Yu Y. Co-clustering based classification for out-of-domain documents//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. California, USA, 2007: 210-219

[31] Dai W, Yang Q, Xue G R, Yu Y. Self-taught clustering//Proceedings of the International Conference on Machine Learning. Helsinki, Finland, 2008: 200-207

[32] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905

[33] Cover T M, Thomas J A. Elements of Information Theory. New York: John Wiley and Sons, 1991

[34] Saenko K, Kulis B, Fritz M, Darrell T. Adapting visual category models to new domains//Proceedings of the European Conference on Computer Vision. Heraklion, Greece, 2010: 213-226

[35] Sivic J, Zisserman A. Video Google: A text retrieval approach to object matching in videos//Proceedings of the International Conference on Computer Vision. Nice, France, 2003: 1470-1477

[36] Bay H, Ess A T, Tuytelaars L J, Van Gool. Speeded-up robust features (SURF). Computer Vision and Image Understanding, 2008, 110(3): 346-359

**LOU Zheng-Zheng**, born in 1984, Ph. D., lecturer. His main research interests include machine learning, pattern recognition and data mining.

**YE Yang-Dong**, born in 1962, Ph. D., professor. His main research interests include machine learning, knowledge engineering and intelligent systems.

**Background**

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61170223, 61502434.

The Information Bottleneck (IB) method is an unsupervised data organization technique, which extracts the data patterns by compressing them into a bottleneck variable and has been widely used in many fields, such as machine learning and pattern recognition. The multivariate IB method is a general principled framework for multivariate extensions of the IB method, which aims to deal with data analysis problems that are more challenging. In the Multivariate IB framework, the domain knowledge is characterized by multiple variables and more information is employed into the data compressing process. The multivariate IB method provides a theoretical framework for multivariate data analysis problems, where the collaborative model describes the cooperative relationship between variables and provides the foundation for defining objective function about data analysis task. A good collaborative model can make full use of the valuable information. Algorithm is used to optimize the objective function, which is related to the efficiency of the data analysis task. How to construct the collaborative model and how to optimize the objective function are two core problems of multivariate IB method.

Our team has been working on the research of multivariate IB method and has achieved certain results, such as the IB algorithms for improving the clustering accuracy; the application of IB method to unsupervised image categorization; the multi-feature information bottleneck and non-redundant multi-view clustering.

The symmetric IB is one of applications of multivariate IB, which aims to extract two systems of compressed variables that are information about each other. However, the symmetric IB only concentrates the relationship between the compressed variables, and the information resided in the original variable is ignored, which will lead to the fact that the compressed results deviate from the patterns resided in the original features. To solve this problem, this paper proposes an Inter-Correlated Symmetric Information Bottleneck (ICSIB) algorithm, which is an extension of the multivariate IB method. And the experimental results have demonstrated the effectiveness of ICSIB algorithm in the application of data double compressing and co-clustering.