基于智能聚类的复杂案例匹配研究

Complex Case Retrieval Based on Intelligent Clustering

章 宁 甘仞初

(北京理工大学管理与经济学院 北京100081)

Abstract Processing complex cases is difficult in CBR. A complex case retrieval method based on intelligent clustering is presented in this paper. An object-oriented aggregate representation pattern is designed to represent complex cases. Hierarchy and fuzzy logic is used to measure similarity between complex cases to improve the accuracy of match results. In order to increase the speed of case retrieval, SOM neural network is applied for clustering of complex cases. Generally, several SOM networks are needed for different parts of complex cases to complete the overall clustering. Then the match process is simplified to find matching cases from the previous cases the same kind as the new problem.

Keywords Case-Based Reasoning, Complex case retrieval, Object-oriented, Fuzzy logic, Self-Organizing Map

案例匹配在基于案例的推理(CBR)中非常关键[3],直接影响到CBR系统的有效性和高效性。在某些CBR的应用领域中,案例比较复杂,案例匹配的实现也比较困难,匹配结果的准确性和匹配速度都难以保证。本文提出的基于智能聚类的复杂案例匹配方法,为提高匹配结果的准确性和匹配速度提供了一条良好的途径。

1 复杂案例的集合表示

一般来说,复杂案例可以分解成多个组成部分,各部分又可以分解成更小的组成部分。如果将这种组成关系用层次来描述,那么复杂案例位于层次的第一层,其组成部分位于第二层,更小的组成部分位于层次中的更低层。显然,仅仅用一组特征变量来描述复杂案例是不可能的,为此,我们设计了一种面向对象的复杂案例集合表示模式。下面首先给出集合对象和最小部分对象的定义。

定义1 设对象O由n个对象$O_1,O_2,\cdots,O_n$组成,用这n个对象组成的集合来表示O,即$O=\{O_1,O_2,\cdots,O_n\}$,则称O为集合对象,对象$O_1,O_2,\cdots,O_n$称为O的元素。

定义2 设对象O不能分解成更小的对象,其所属类中定义了n个简单属性$A_1,A_2,\cdots,A_n$,则称O为最小部分对象,并用O中这n个简单属性的值组成的集合来表示,即$O=\{O.A_1,O.A_2,\cdots,O.A_n\}$。

在面向对象的复杂案例集合表示模式中,一个复杂案例对应一个第一层的集合对象,我们给出基本的形式化表示如下:

案例$C=\langle N,O\rangle$

①N:表示案例名称。它是每个案例的唯一标识符。

②O:表示案例对应的第一层集合对象。首先定义O由l个第二层对象组成,即$O=\{O_1,O_2,\cdots,O_l\}$。再对第二层对象进行定义,如定义O_i由m个第三层对象组成,即$O_i=\{O_{i1},O_{i2},\cdots,O_{im}\}$,$i=1,\cdots,l$。以此类推,直到定义的每个对象都是最小部分对象为止。最后再对最小部分对象进行定义,如定义O_x由n个简单属性值组成(x表示该最小部分对象的下标,代表一至几位数字),即$O_x=\{O_x.A_1,O_x.A_2,\cdots,O_x.A_n\}$,其中$A_k$是$O_x$所属类中定义的某个简单属性,$k=1,\cdots,n$。

元组$\langle N,O\rangle$并不能表示案例中所有有用的信息,应该针对不同的领域加入其它不同的项。

2 相似度的计算

案例匹配的主要任务是比较新问题与案例库中旧案例的相似性。相似性的比较方法有很多种,大体可以分为计算二者之间的距离和直接计算二者之间的相似度两类。这里采用直接计算相似度(取值在0~1)的方法。

2.1 对象之间的相似度

定义3 设两个同属一个类的集合对象O和O^*分别由n个对象组成,$O=\{O_1,O_2,\cdots,O_n\}$,$O^*=\{O_1^*,O_2^*,\cdots,O_n^*\}$,则O与$O^*$之间的相似度等于各元素对象之间相似度的加权和,即:

$$SIM(O,O^*)=\sum_{i=1}^{n}W_i\times SIM(O_i,O_i^*)$$

其中:$SIM(O,O^*)$表示集合对象O与O^*之间的相似度;W_i是表示元素对象O_i的重要性程度的权重因子,$0\leqslant W_i\leqslant 1$,且满足$\sum_{i=1}^{n}W_i=1$;$SIM(O_i,O_i^*)$表示元素对象$O_i$与$O_i^*$之间的相似度。

定义4 设两个同属一个类的最小部分对象O和O^*分别由n个简单属性值组成,$O=\{O.A_1,O.A_2,\cdots,O.A_n\}$,$O^*=\{O^*.A_1,O^*.A_2,\cdots,O^*.A_n\}$,则O与$O^*$之间的相似度等于各属性值之间相似度的加权和,即:

$$SIM(O,O^*)=\sum_{i=1}^{n}W_i\times SIM(O.A_i,O^*.A_i)$$

其中:$SIM(O,O^*)$表示最小部分对象O与O^*之间的相似度;W_i是表示简单属性A_i的重要性程度的权重因子,$0\leqslant W_i\leqslant 1$,且满足$\sum_{i=1}^{n}W_i=1$;$SIM(O.A_i,O^*.A_i)$表示简单属性值$O.A_i$与$O^*.A_i$之间的相似度。

根据以上两个定义,可以得到新问题与旧案例之间的相似度计算公式。

设有一个新问题P,对应的第一层集合对象是O,$O=\{O_1,O_2,\cdots,O_l\}$。若案例库中有一个旧案例C,对应的第一层集合对象是O^*,$O^*=\{O_1^*,O_2^*,\cdots,O_l^*\}$。那么根据定义1,O与$O^*$在第一层的相似度计算公式如下:

$$S=SIM(O,O^*)=\sum_{i=1}^{l}W_i\times SIM(O_i,O_i^*) \quad (1)$$

其中S表示O与O*在第一层的相似度。O_i与O_i^*分别是P与C在第二层上包括的某个对象，设$O_i=\{O_{i1},O_{i2},\cdots,O_{im}\}$，$O_i^*=\{O_{i1}^*,O_{i2}^*,\cdots,O_{im}^*\}$。那么根据定义1，$O_i$与$O_i^*$在第二层的相似度计算公式如下：

$$S_i=SIM(O_i,O_i^*)=\sum_{j=1}^{m}W_{ij}\times SIM(O_{ij},O_{ij}^*) \quad (2)$$

其中S_i表示O_i与O_i^*在第二层的相似度。O_{ij}与O_{ij}^*分别是P与C在第三层上包括的某个对象，如果它们是集合对象，那么仍然根据定义1得到它们之间在第三层的相似度计算公式。以此类推，直到要求的是最小部分对象之间的相似度为止。

设O_x与O_x^*分别是P与C中包括的某个最小部分对象(x表示下标)，$O_x=\{O_x\cdot A_1,O_x\cdot A_2,\cdots,O_x\cdot A_n\}$，$O_x^*=\{O_x^*\cdot A_1,O_x^*\cdot A_2,\cdots,O_x^*\cdot A_n\}$。那么根据定义2，$O_x$与$O_x^*$的相似度计算公式如下：

$$S_x=SIM(O_x,O_x^*)=\sum_{k=1}^{n}W_{xk}\times SIM(O_x\cdot A_k,O_x^*\cdot A_k) \quad (3)$$

其中S_x表示O_x与O_x^*的相似度。$O_x\cdot A_k$与$O_x^*\cdot A_k$分别是O_x与O_x^*在属性A_k上的取值。

2.2 属性值之间的相似度

为了提高匹配的准确性，本文运用模糊逻辑来计算属性值之间的相似度。下面简单介绍一下模糊集、隶属函数和模糊关系的概念。

模糊集的基本思想是把经典集合中的绝对隶属关系灵活化，元素对"集合"的隶属度不再局限于0或1，而是可以取从0到1的任一数值[8]。

定义5 设给定论域U，U在闭区间$[0,1]$中的任一映射μ_A：

$\mu_A:U\to[0,1]\quad x\to\mu_A(x),x\in U$

可确定U的一个模糊集A。

$\mu_A(x)$是隶属函数，它在模糊数学中占有很重要的地位，是把模糊性数量化，使事物的不确定性在形式上用经典的数学方法进行表达和运算的桥梁[6]。

定义6 对于集合U、V，其直积$U\times V=\{(x,y)|x\in U,y\in V\}$上的任一子集R，均可称为U与V之间的二元关系，或简称关系。如果R是一个模糊集，则它所刻画的就是U与V之间的模糊关系。

相似关系就是一种模糊关系，两个元素之间的相似性不是简单的相似或不相似，而要以相似度来衡量。

当U、V为有限集时，关系R可以用一个矩阵(仍记为R)表示：$R=(r_{ij})_{m\times n}$。这里U有m个元素，V有n个元素，$r_{ij}\in[0,1]$，$i=1,\cdots,m;j=1,\cdots,n$。当R是模糊关系时，称R为模糊关系矩阵。此时，R的元素值也可用下式表示：$r_{ij}=\mu_R(u_i,v_j)$，$\mu_R(u_i,v_j)$是在论域$U\times V$上的隶属函数。

本文将属性值之间的相似关系看作是"相似"模糊集刻画的一种模糊关系，相似度就是隶属于"相似"模糊集的隶属度，因此相似度的计算可以借助于隶属函数。

定义7 设有一个属性的两个取值V_1和V_2，则V_1和V_2之间的相似度等于V_1和V_2隶属于"相似"模糊集的隶属度，即：

$$SIM(V_1,V_2)=\mu_R(V_1,V_2)$$

其中：$SIM(V_1,V_2)$表示属性值V_1与V_2之间的相似度；R表示"相似"模糊集，μ_R表示隶属函数。

简单属性主要有两种类型：数值型和字符型。首先考虑数值型属性相似度的计算。数值型属性的论域元素是连续的，因此，可以选用某些典型函数作为隶属函数。假设新问题中一个数值型属性值的输入代表如下含义：要求旧案例中的属性值最好与输入的值接近。那么我们选取可以描述"接近于"这个概念的隶属函数，比如正态分布的隶属函数。

首先对数值型属性值进行归一化处理。假设原有属性值为V，转换后的属性值为V'，可用如下公式进行变换：

$$V'=V/(V_{max}-V_{min}) \quad (4)$$

其中V_{max}和V_{min}分别代表该属性变量的最大值和最小值。

假设是A_k数值型属性，那么根据定义7，$O_x\cdot A_k$与$O_x^*\cdot A_k$的相似度计算公式如下：

$$SIM(O_x\cdot A_k,O_x^*\cdot A_k)=\mu_R(O_x\cdot A'_k,O_x^*\cdot A'_k) \quad (5)$$

其中R是"相似"模糊集，$O_x\cdot A_k'$和$O_x^*\cdot A_k'$是$O_x\cdot A_k$和$O_x^*\cdot A_k$经过归一化处理后的值。

如果选取正态分布的隶属函数，那么属性值之间的相似度可通过下式计算出来：

$$SIM(O_x\cdot A_k,O_x^*\cdot A_k)=e^{-K(O_x^*\cdot A_k'-O_x\cdot A_k')^2} \quad (6)$$

相似度的变化如图1所示。

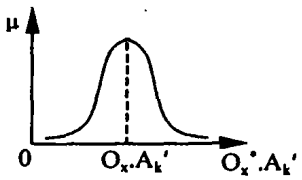

图1 正态分布的隶属函数

我们将字符型属性分为名称变量和顺序变量两种。名称变量的属性值之间只存在"相等"或"不等"的关系。

假设A_k是字符型属性中的名称变量，那么根据定义7，$O_x\cdot A_k$与$O_x^*\cdot A_k$的相似度计算公式如下：

$$SIM(O_x\cdot A_k,O_x^*\cdot A_k)=\begin{cases}0 & \text{if } O_x^*\cdot A_k\neq O_x\cdot A_k\\ 1 & \text{if } O_x^*\cdot A_k=O_x\cdot A_k\end{cases} \quad (7)$$

顺序变量的属性值之间有顺序关系，"大于"和"小于"的概念有意义。为了形式化地定义相似度计算公式，我们首先作如下定义：

定义8 设有一个顺序变量的两个属性值V_1和V_2，且V_1按顺序排列排在V_2的前面，则称V_1小于V_2，记为$V_1<V_2$。

字符型属性的论域元素是离散的，不能直接通过数值计算方式得到隶属度。假设新问题中一个顺序变量的属性值的输入代表如下含义：要求旧案例中的属性值最好与输入的值接近。那么可以选择根据专家的主观认识和个人经验来给出隶属度的具体数值的方式，得到表示元素两两之间相似度的相似矩阵。

定义9 设R是一个以$U\times U$为论域的模糊关系矩阵。如$\mu_R(u_i,u_i)=1,i=1,\cdots,m$，称R满足自反性；如$\mu_R(u_i,u_j)=\mu_R(u_j,u_i),i=1,\cdots,m,j=1,\cdots,m$，称R满足对称性。如R既满足自反性又满足对称性，则称R是一个相似矩阵。

假设A_k是字符型属性中的顺序变量，且论域U中有n个元素(属性值)$V_i,i=1,\cdots,n$，且$V_1<V_2<\cdots<V_n$，通过专家确定它们两两之间的相似度，得到其n阶相似矩阵为：

$$R=(r_{ij})_{n\times n}=\begin{pmatrix}r_{11} & r_{12} & \cdots & r_{1n}\\ r_{21} & r_{22} & \cdots & r_{2n}\\ \cdots & \cdots & \cdots & \cdots\\ r_{n1} & r_{n2} & \cdots & r_{nn}\end{pmatrix}$$

其中 $r_{ij}\in[0,1]$表示 V_i 和 V_j 之间的相似度，$r_{ij}=\mu_R(V_i,V_j)$，$\mu_R(V_i,V_j)$是在论域 $U\times U$ 上的隶属函数，由专家确定。根据相似矩阵的自反性和对称性可知，只需确定矩阵右上三角中的元素即可。如果从行上考虑，在矩阵的第 i 行中，$i=1,\cdots,n-1$，只需确定 $r_{i(i+1)},r_{i(i+2)},\cdots,r_{in}$即可；如果从列上考虑，在矩阵的第 j 列中，$j=2,\cdots,n$，只需确定 $r_{1j},r_{2j},\cdots,r_{(j-1)j}$即可。

定理1 设 R 是一个描述顺序变量的属性值之间相似关系的相似矩阵，且 $R=(r_{ij})_{n\times n}$，则 R 满足以下约束条件：

①$\forall$ i，有 $r_{ii}=1$；$\forall$ i，$\forall$ j，有 $r_{ij}=r_{ji}$；

②$\forall$ i，$i=1,\cdots,n-1$，有 $r_{i(i+1)}>r_{i(i+2)}>\cdots>r_{in}$；

③$\forall$ j，$j=2,\cdots,n$，有 $r_{1j}<r_{2j}<\cdots<r_{(j-1)j}$。

证明从略。

在确定 A_k 的相似矩阵 R 之后，根据定义7，$O_x\cdot A_k$ 与 $O_x^*\cdot A_k$ 的相似度计算公式如下：

$$SIM(O_x\cdot A_k,O_x^*\cdot A_k)=r_{ij},\quad \text{if } O_x^*\cdot A_k=V_i \text{ and } O_x\cdot A_k=V_j \tag{8}$$

虽然我们定义相似度计算公式的过程是自顶向下的，但实际计算相似度的过程是自底向上的，即先计算各属性值之间的相似度，再计算各最小部分对象的相似度，然后根据组成关系逐层向上计算各集合对象的相似度，最后得到整个案例（第一层集合对象）的相似度。

3 基于智能聚类的复杂案例匹配

复杂案例之间相似度的计算是复杂而耗时的，为了提高匹配速度，本文提出基于智能聚类的复杂案例匹配方法，在利用智能技术对案例库中旧案例进行客观聚类的基础上，将寻找与新问题相似的旧案例的匹配过程分为两个步骤，首先对新问题进行聚类，然后在与新问题同类的旧案例中寻找相似的案例，从而大大节省了匹配时间，同时也进一步保证了检索出的旧案例与新问题的一致性。

3.1 自组织神经网络简介

自组织网络采用没有指导的学习过程，不必给定应有的输出，网络只靠输入模式本身的特征，根据一定的判断标准自行修改单元连接的强度（权重），使权矢量在输入向量空间中的分布近似于样本的分布。这也是本文采用自组织神经网络来解决案例聚类问题的原因和依据，因为在对案例进行聚类前，可以分成什么样的类以及一个案例归属的类别往往是未知的。

Kohonen 网络是自组织映射（SOM）神经网络[7]，由输入层和输出层构成，如图2所示。

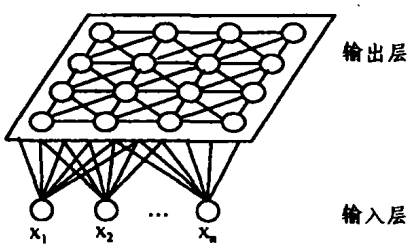

图2 Kohonen 自组织映射网络结构

输入层中的每个单元都通过不同的连接权与输出层的每个单元相连。输出层的处理单元一般是以二维形式排列的，每个单元都是输入样本的“映象”。在输出层中竞争是这样进行的：对于“赢”的那个单元 c，在其周围 Nc 区域内的单元在不同程度上得到兴奋，而在 Nc 区域以外的单元都被抑制。Nc 是时间 t 的函数，随着 t 的增加，Nc 的面积成比例地缩小，最后只剩下一个单元，也可能是一组单元，它们反映了一类样本的属性。

利用该网络进行聚类的结果是将输入的样本在指定的相似测度下，按样本间的相似程度，将其映射到输出层的某个节点中。

3.2 复杂案例的 SOM 网络实现

复杂案例具有层次性，从理论上讲，应该对第一层集合对象进行聚类分析。在新问题的匹配过程中，首先对新问题进行第一层的聚类，然后计算与新问题同属一类的旧案例和新问题之间在第一层的相似度。然而，越是高层的集合对象，越具有任务复杂、属性众多、分类多等特点，相应的 SOM 网络实现越困难，主要包括：

①网络结构难以确定，即使确定，网络的节点会很多，结构庞大，造成训练困难。

②由于节点数众多，要保证网络的性能和质量，就要学习大量的样本，而对于复杂案例来说，大量案例的搜集很困难。

③如果出现新的类别的样本，网络必须抹去全部记忆，重新学习，对于一个结构庞大、样本数众多的网络，每一次的学习都是困难和费时的。

因此，在实践中要根据样本数的多少和对象中属性的多少来决定是否对该对象进行聚类分析，但要尽量选择高层对象。一般来说，至少可以实现对每个最小部分对象进行聚类分析，从而保证聚类的作用覆盖整个案例。

下面讨论各 SOM 网络的实现，包括输入层、输出层和学习算法的确定。设要进行聚类分析的对象下标是 z（代表零至几位数字），输入网络的样本数是 N。

·输入层 代表对象的属性。集合对象的属性包括所有和它具有组成关系的最小部分对象的属性。为了简化网络结构、减少节点数，对于每一个属性只用一个输入节点表示，因此输入节点数就等于属性的个数，设为 n 个。输入样本向量为 X，由变量 $x_1,x_2,\cdots,x_n$ 组成，如图2中的输入层所示。由于神经网络只能处理数字输入数据，因此，对于其它各种输入量都必须以一定的方式转换为数字。

对于数值型属性值，用式(4)进行变换。对于字符型属性中的顺序变量，每个值都按顺序赋以百分数，0为百分之0，1为百分之百。对于字符型属性中的名称变量，也可以将每个值转换成0～1之间的值，当然这种赋值并不具有顺序含义。

·输出层 其每个节点都代表了一类样本。一般情况下，希望在输出二维平面上的节点数足够多，设为 m 个，$m>>n$。这里以4行4列为例，节点数为 $4*4=16$ 个，相邻节点按正方形排列，如图2所示。

各节点的输出变量为 $y_j(j=1,\cdots,m)$，输出值为输入样本向量 X 与该节点对应的权矢量 ω_j 匹配的程度（相似度）。通常 SOM 网络用欧氏距离作为相似测度，本文使用上一节中的相似度计算方法来计算 y_j 的值。

$$y_j=SIM(X,\omega_j)=\sum_{i=1}^{n}U_i\times SIM(x_i,\omega_{ij}) \tag{9}$$

其中：$SIM(X,\omega_j)$表示输入向量 X 与权矢量 ω_j 的相似度；$SIM(x_i,\omega_{ij})$表示输入变量 x_i 与权值 ω_{ij} 的相似度；U_i 是表示 x_i 的重要性程度的权重因子，$0\leqslant U_i\leqslant 1$，且满足 $\sum_{i=1}^{n}U_i=1$。注意 U_i 和 ω_{ij} 虽然都是表示权重，但两者有着截然不同的意义。

$SIM(x_i,\omega_{ij})$的计算方法与上一节中计算属性值之间相似

度的方法相同。对于最小部分对象来说，U_i 就是该对象中第 i 个属性的权重，即：

$$U_i = W_n \tag{10}$$

对于集合对象来说，设 x_i 是下标为 z′（代表几位数字，前几位是 z）的最小部分对象中的第 k 个属性，z′相对于 z 的权重为 $V_{z'-z}$（等于两个对象之间各层权重之积）。U_i 的计算公式如下：

$$U_i = V_{z'-z} \times W_{z'k} \tag{11}$$

·学习算法　如果在输出层有一个节点与输入 X 匹配最好，记为 c，则

$$y_c = SIM(X, \omega_c) = \max_j y_j = \max_j SIM(X, \omega_j)$$

权的修正是对 ω_c 和 $j \in N_c$（c 的邻域）中的 ω_j 进行的。具体学习算法如下：

①将 $\omega_{ij}(0)$ 赋予[0,1]的随机值，i=1,…,n，j=1,…,m，确定学习率 $\alpha(0)$，邻域大小 $N_c(0)$ 及总学习次数 T；

②在样本 $X^1, X^2, \cdots, X^N$ 中，取一个样本 $X^p(t)$ 作为网络的输入；

③根据式(9)计算各节点输出值 y_j，j=1,…,m，取其中输出值最大的节点 c，作为竞争得胜的节点；

④对权 ω_{ij} 进行修正

$$\begin{cases} \omega_{ij}(t+1) = \omega_{ij}(t) + \alpha(t)(x_i^p - \omega_{ij}(t)) & j \in N_c(t) \\ \omega_{ij}(t+1) = \omega_{ij}(t) & j \notin N_c(t) \end{cases}$$
$$i = 1, 2, \cdots, n \tag{12}$$

⑤更新 $\alpha(t)$ 和 $N_c(t)$

$$\alpha(t) = \alpha(0)\left(1 - \frac{t}{T}\right) \tag{13}$$

$$N_c(t) = INT\left[N_c(0)\left(1 - \frac{t}{T}\right)\right] \tag{14}$$

其中 INT 表示求整；

⑥回到步骤②，继续上述过程，直到迭代了 T 次为止。网络的权值说明了对应类中样本的中心。

学习结束后，就可以对各样本进行分类了。

3.3　基于智能聚类的复杂案例匹配过程

为了形式化地描述基于智能聚类的复杂案例匹配过程，本文将对下标为 z 的对象进行聚类分析的 SOM 网络记为 K_z。

定义10　对于包含 p 个 SOM 网络的集合：$\{K_{z(1)}, K_{z(2)}, \cdots, K_{z(p)}\}$，其中 z(i)为对象的下标，i=1,…,p，如果它们对应的对象能够组成整个案例，并且相互独立（没有组成关系），则称这个集合为 SOM 完备集，记为 I^K。

复杂案例的 SOM 网络实现就是要得到一个 SOM 完备集 I^K。设新问题 P=〈N,O〉，基于智能聚类的匹配过程如下：

·根据 I^K 对新问题进行聚类匹配　首先根据 $I^K = \{K_{z(1)}, K_{z(2)}, \cdots, K_{z(p)}\}$ 对新问题中的对象 $O_{z(i)}$（i=1,…,p）进行聚类。然后根据聚类的结果计算 $O_{z(i)}$ 与同类的旧案例之间的相似度，并将相似度大于 ε（预先给定的阈值）的案例作为匹配结果返回，存入集合 $G_{z(i)}$ 中。如果没有找到与 $O_{z(i)}$ 匹配的案例，则 $G_{z(i)} = \Phi$。

如果 $I^K = \{K\}$，即实现了第一层集合对象的 SOM 网络，那么这个聚类匹配过程就是整个新问题的匹配过程。如果 $I^K \neq \{K\}$，那么这个聚类匹配过程并没有从第一层上寻找与新问题匹配的案例，因此，还要根据聚类匹配的返回结果去寻找与整个新问题匹配的案例。

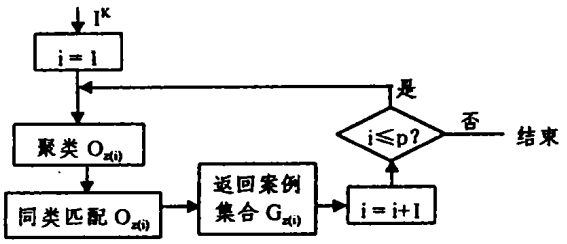

图3　根据 I^K 进行聚类匹配

·第一层集合对象的匹配　这个过程只有在 $I^K_{max} \neq \{K\}$ 时才进行。首先给出如下定理。

定理2　对于新问题 P=〈N,O〉，在根据 $I^K = \{K_{z(1)}, K_{z(2)}, \cdots, K_{z(p)}\}$ 进行聚类匹配后，能够找到与 O 匹配的旧案例的必要条件是：$\exists\, G_{z(i)} \neq \Phi$，i=1,…,p。如果以上条件成立，那么任何一个与 O 匹配的案例 $C \in G_{z(i)}$，$G_{z(i)} \neq \Phi$，i=1,…,p。

证明从略。

根据定理2可知，只需在 $G_{z(1)} \cup \cdots \cup G_{z(p)}$ 的案例集合中寻找与新问题第一层集合对象 O 匹配的案例，一般情况下，这个集合中的案例数 N′<<总案例数 N。如果找到相似度大于 ε 的案例，存入 G 中。如果 G=Φ，表示没有找到与 O 匹配的案例，需要对问题进行层次分解，本文不再详述。

结束语　将基于案例的推理(CBR)应用到复杂领域是 CBR 理论研究的一个难点。本文提出的基于智能聚类的复杂案例匹配方法，为提高匹配结果的准确性和匹配速度提供了一条良好的途径。它具有如下特点：

(1)面向对象的复杂案例集合表示模式能够表示多种复杂案例，而且便于案例推理过程的实现和形式化表达；

(2)相似度的计算运用了集合组合的层次关系和模糊逻辑，提高了匹配结果的准确性；

(3)一到多个自组织映射神经网络的实现对案例库中的旧案例和新问题进行聚类，使得新问题的匹配过程只需在与新问题同类的旧案例中进行，减少了匹配时间；

(4)基于智能聚类的复杂案例匹配过程尽量寻找与新问题整体匹配的旧案例。

参考文献

1　甘仞初，章宁，赵瑞雪. Architecture Design of Information System Using Case Based Reasoning. 见第16届世界计算机会议论文集. 2000.8

2　甘仞初. 基于案例的系统. 中国管理信息系统研究与实践新进展. 湖南大学出版社，1995. 327～332

3　Aamodt A, Plaza E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, 1994, 7(1): 39～59

4　Watson I. Case-Based Reasoning is a Methodology not a Technology. Knowledge-Based Systems, 1999, 12

5　Ricci F, Senter L. Structured Cases, Trees and Efficient Retrieval. Advances in Case-Based Reasoning: 4th European Workshop, EWCBR'98

6　虞和济，陈长征，张省，周建南著. 基于神经网络的智能诊断. 冶金工业出版社，2000.5

7　王永骥，涂健编著. 神经元网络控制. 机械工业出版社，1998,2

8　何新贵著. 模糊知识处理的理论与技术. 国防工业出版社，1998