文章编号:1004-1478(2011)02-0106-05

基于线性判别分析和二分 K 均值的高维数据自适应聚类方法

汪万紫, 裘国永, 张兵权

(陕西师范大学 计算机科学学院, 陕西 西安 710062)

摘要:将线性判别分析和二分 K 均值聚类耦合在一起,提出了一个适合于高维数据聚类的自适应方法:利用线性判别分析将高维数据集变换成低维数据集,然后在低维数据集上执行二分 K 均值聚类,并把得到的聚类结果通过一个簇成员指示矩阵 $\boldsymbol{H}$ 变换到原数据集中.将这样的过程反复进行,直到自适应地得到一个最优结果.基于现实数据集的实验结果证明了该方法的有效性.

关键词:维归约;线性判别分析;二分 K 均值;高维数据自适应聚类方法

中图分类号:TP311.13 **文献标志码**:A

Adaptive clustering method based on linear discriminant analysis and bisecting *K*-means for high dimensional data

WANG Wan-zi, QIU Guo-yong, ZHANG Bing-quan

(*School of Comp. Sci., Shannxi Normal Univ., Xi'an* 710062, *China*)

Abstract: Combining linear discriminant analysis (LDA) and bisecting *K*-means clustering (BKM), an adaptively clustering method was proposed for high dimensional data. The method uses LDA to transform the high dimensional dataset into low dimensional one, applies BKM on the low dimensional dataset, and constructs the clusters in the original high dimensional dataset. The method is adaptively executed to generate the best result. Extensive experimental results on real-world datasets showed the effectiveness of the approach.

Key words: dimension reduction; LDA; bisecting *K*-means; adaptive clustering method for high dimensional data

0 引言

在文本分类、信息检索和生物信息学等数据挖掘的应用领域中,数据的维数往往是很高的.这就引起了所谓的"维灾难"问题——随着维数的增加,高维数据中的样本在其所占据的空间中越来越稀疏;对于分类问题,这可能就意味着没有足够的数据对象来创建类模型[1].比如在对高维数据进行聚类时,如果按照传统的样本点之间的距离计算,每一对样本点之间的距离就几乎看不出来区别[2].维灾难问题导致许多原本对低维数据分析很有效的分类和聚类算法在处理高维数据时遇到困难,导致分类准确率降低,聚类质量下降.比如,像 K 均值这种对维数据聚类效率较高的算法就会受到所谓的

收稿日期:2011-03-23

基金项目:陕西省自然科学基金项目(2010JM8039)

作者简介:汪万紫(1986—),男,浙江省衢州市人,陕西师范大学硕士研究生,主要研究方向为数据挖掘.

局部最优问题的困扰,即随着迭代的进行,常常只能收敛于局部最优,而非全局最优[3]. 为了解决这一问题,迫切需要开发出针对高维数据集的高效聚类方法.

目前已有一些方法可以用来处理维灾难问题. 有的方法是在传统的像 K 均值这样的方法中采用主成分分析(PCA)[4]和奇异值分解(SVD)[5]这样的维归约技术. 其结果是能够保留最重要的特征,去除对正确分类起干扰作用的不相关特征,使 K 均值方法能够在低维数据空间中发挥作用. 这些方法的共同特点是:维归约在数据的预处理阶段进行,维归约过程和聚类过程是分开的,一旦决策子空间在数据的预处理阶段被选定,在聚类过程中就无法对其更改,从而不能保证最后得到最优决策子空间. 为了解决这个问题,可以采用自适应维归约技术[3]. 在这类方法中,先利用某种维归约技术得到一个低维决策子空间,然后在该子空间上运用聚类方法,根据聚类结果再调整原决策子空间,得到一个更优的决策子空间,再在新的决策子空间上进行聚类,如此反复进行下去,直到得到某种形式的收敛结果. 聚类始终是在低维数据空间上进行,从而保证了算法的效率.

这类方法的关键之处有二:一是维归约技术的选择;二是如何利用低维子空间上得到的聚类结果构造出原数据空间上的类. 本文将选择线性判别分析(LDA)来实现维归约. LDA 被广泛地应用于子空间(特征空间)的选择,具有良好的决策能力[6]. 聚类方法采用二分 K 均值聚类[7]. 并特别引入一个簇成员指示矩阵 $\boldsymbol{H}$,目的是能够由低维子空间的聚类结果构造出原数据空间上的簇. 由此提出了耦合 LDA 和二分 K 均值聚类的适合于处理高维数据的自适应聚类方法(LDA - BKM 方法).

1　LDA 和二分 K 均值方法(BKM)

设 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)$ 是一个高维(d 维)数据向量集合,可把 $\boldsymbol{X}$ 看成是一个 $d\times n$ 矩阵. 为简单起见,在数据预处理阶段对数据进行处理,使得 $\bar{x}=\sum_i x_i/n=0$.

在线性判别分析[7]中,其目标是找出最优线性变换,将数据从高维空间变换到低维空间,同时保持原数据集的类结构.

为此,假定数据集 $\boldsymbol{X}$ 有 K 类 $C_1,C_2,\cdots,C_K$. 引入总散布(total scatter)、类间散布(between-class scatter)和类内散布(with-in class scatter)矩阵:

$$\boldsymbol{S}_t=\sum_{i=1}^{n}x_ix_i^{\mathrm{T}}$$

$$\boldsymbol{S}_b=\sum_k n_km_km_k^{\mathrm{T}}$$

$$\boldsymbol{S}_w=\sum_k\sum_{i\in C_k}(x_i-m_k)(x_i-m_k)^{\mathrm{T}}$$

其中 $n_k=|C_k|$,m_k 是 C_k 的簇心,m 是整个数据集的中心. 使用 $Tr\boldsymbol{A}$ 表示矩阵 $\boldsymbol{A}$ 的迹,则 $Tr\boldsymbol{S}_w$ 表示的是簇内数据的相似度,$Tr\boldsymbol{S}_b$ 表示簇间分离度,$Tr\boldsymbol{S}_t$ 表示数据集的变差. LDA 就是求解最优化问题

$$\max_{\boldsymbol{U}}Tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{U})\text{和}\min_{\boldsymbol{U}}Tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{U})$$

得到最优变换 $\boldsymbol{U}$.

上述最优化问题等价于下列非零广义特征值问题 $\boldsymbol{S}_{bx}=\lambda\boldsymbol{S}_{wx}$. 若 $\boldsymbol{S}_w$ 非奇异,则相当于矩阵 $\boldsymbol{T}=\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ 的特征问题,而若 $\boldsymbol{S}_b$ 非奇异,则相当于矩阵 $\boldsymbol{T}=\boldsymbol{S}_b^{-1}\boldsymbol{S}_w$ 的特征问题. 如果 $\boldsymbol{S}_w$ 和 $\boldsymbol{S}_b$ 都奇异,则可以引入广义逆来处理相应奇异问题. 其中最大的 $d_1(d_1\ll d)$ 个(广义)特征值对应的(广义)特征向量为变换矩阵 $\boldsymbol{U}$ 的列向量,由此所得到的线性变换 $\boldsymbol{Y}=\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}$ 使得从 d 维数据集 $\boldsymbol{X}$ 变成了 d_1 维数据集 $\boldsymbol{Y}$,从而实现了维归约. 后文只针对 $\boldsymbol{S}_w$ 非奇异的情形进行试论,其他2种情形可类似处理.

二分 K 均值算法[7]可以看做是 K 均值算法的直接扩充,它基于一个简单的想法:为了得到 K 个簇,先将所有样本点的集合利用 K 均值分成2个簇,然后从这些簇中选取1个继续分裂,如此下去,直到得到 K 个簇. 二分 K 均值的算法如下.

二分 K 均值算法:

输入:数据集 $\boldsymbol{X}$,二分次数 b,簇数 K.

输出:簇集 $\boldsymbol{S}=\{C_1^*,C_2^*,\cdots,C_K^*\}$.

1:初始化簇集 $\boldsymbol{S}=\{\boldsymbol{X}\}$ {初始化簇表,使其包含所有样本点组成的簇.}

2:repeat

3:从簇表中取出1个簇 C^*(比如说选出最大的簇).

4:{对 C^* 进行 b 次 K 均值聚类,将它分别划分为 b 对子簇.}

5:for $i=1$ to b do

6:使用基本 K 均值,将 C^* 分成2个子簇 C_1^* 和 C_2^*.

7:end for

8:从 b 对子簇中选择具有最小总 SSE 的1对子簇.

9:将这2个子簇添加到簇表 S 中.

10:until 簇表 S 中包含 K 个簇.

在算法中本文使用误差的平方和 SSE(sum of the squared error)作为度量聚类质量的目标函数. 换言之,本文计算每个样本点的误差,即它到最近簇

心的欧几里得距离,然后计算误差的平方.

二分 K 均值聚类较基本 K 均值聚类来说更不受初始化问题的影响,且每步只有 2 个簇心.

二分 K 均值聚类的每一步即是从 b 对子簇中找出使下列聚类目标函数 SSE 之和最小的 1 对子簇 C_1^* 和 C_2^*:

$$J = \sum_{x_i \in C_1} \| x_i - m_1^* \|^2 + \sum_{x_i \in C_2} \| x_i - m_2^* \|^2$$

其中,m_1^* 和 m_2^* 分别是执行内循环时由 K 均值聚类得到的每对子簇 C_1^* 和 C_2^* 的簇心.

在本文所提出的方法中,二分 K 均值聚类是在是在 d_1 维数据集 Y 上进行的. 执行二分 K 均值的结果可以得到 K 个 d_1 维簇心. 由于这些簇心变换到原 d 维的结果并不唯一,所以不能直接得到 X 的簇心. 为此在执行二分 K 均值聚类时,需要得到 1 个簇成员指示矩阵 $\boldsymbol{H} = (h_{ik})_{n \times K}$ 用于表明数据向量 x_i 属于某个簇,即 C_k 由哪些数据向量组成:如果样本 x_i 属于第 k 个簇,则 $h_{ik} = 1$,否则 $h_{ik} = 0$.

因为二分 K 均值聚类的目标函数 $J = \boldsymbol{S}_t^* = \boldsymbol{S}_w^* + \boldsymbol{S}_b^*$,所以

$$J = Tr\boldsymbol{S}_w^* = Tr(\boldsymbol{S}_t^* - \boldsymbol{S}_b^*)$$

因为 Y 一旦确定,$Tr\boldsymbol{S}_t^*$ 是一个常量,所以,二分 K 均值同样可以最小化类内散布矩阵 $\boldsymbol{S}_w^*$,同时最大化类间散布矩阵 $\boldsymbol{S}_b^*$. 由此可见 LDA 和二分 K 均值聚类有十分相似的特性:最小化类内散布矩阵和/或者最大化类间散布矩阵.

2 耦合 LDA 和 BKM 的高维数据自适应聚类方法

为了将高维数据集 $\boldsymbol{X}$ 变换数成低维数据集进行二分 K 均值聚类,首先需要得到一个变换矩阵 $\boldsymbol{U}$. 这个 $\boldsymbol{U}$ 可以用主成分分析(PCA)得到,也可随机生成. 通过线性变换 $\boldsymbol{Y} = \boldsymbol{U}^T\boldsymbol{X}$ 将 d 维数据集 $\boldsymbol{X}$ 变成 d_1 维数据集 $\boldsymbol{Y}$,在 $\boldsymbol{Y}$ 上实施二分 K 均值聚类得到 $\boldsymbol{Y}$ 的 K 个簇及簇成员指示矩阵 $\boldsymbol{H}$(若 $y_i = \boldsymbol{U}^T x_i$ 属于 C_k^*,则 $h_{ik} = 1$,即 x_i 属于 C_k,否则 $h_{ik} = 0$).

一般而言,由于对应的 d_1 维决策子空间并不一定就是最优的子空间,所以通过 $\boldsymbol{H}$ 在原数据集 $\boldsymbol{X}$ 上构造出的 K 个簇心并不一定就是正确的簇心. 为此在刚刚得到的 K 个簇心的基础上再使用 LDA 重新得到一个变换矩阵 $\boldsymbol{U}$,然后重复上述过程. 这在子空间内聚类的同时也对子空间进行自适应的选择,直到找到最佳的决策子空间.

因此,笔者提出一种耦合 LDA 和 BKM 的针对高维数据的自适应聚类方法,将之简称为 LDA – BKM 算法. 具体的 LDA – BKM 算法细节如下,其中 $d_1 = K - 1$.

LAD – BKM 算法:

输入:数据集 $\boldsymbol{X}$,二分次数 b,簇数 K.

输出:簇集 $\boldsymbol{S} = \{C_1, C_2, \cdots, C_K\}$.

第 1 步:在 $\boldsymbol{X}$ 上执行 PCA,获得初始化的 $\boldsymbol{U}$;

第 2 步:在 $\boldsymbol{Y} = \boldsymbol{U}^T\boldsymbol{X}$ 上执行 BKM 算法获得 $\boldsymbol{H}$ 和簇集 $\{C_1^*, C_2^*, \cdots, C_K^*\}$;

第 3 步:执行 LDA 算法获得 $\boldsymbol{U}$;

第 4 步:重复迭代第 2 步和第 3 步直至收敛.

LDA – BKM 方法自适应地在原数据空间上执行 LDA,而在决策子空间上执行二分 K 均值聚类,两者交替执行.

下面是 BKM 算法的细节:

BKM 算法:

输入:数据集 $\boldsymbol{Y}$,二分次数 b,簇数 K.

输出:$\boldsymbol{H}$ 和簇集 $\boldsymbol{S} = \{C_1^*, C_2^*, \cdots, C_K^*\}$.

1:初始化簇集 $\boldsymbol{S} = \{\boldsymbol{Y}\}$ {初始化簇集,使其包含所有样本点组成的簇.}

2:repeat

3:从簇集中取出 1 个簇 C^*(比如说选出最大的簇).

4:{对 C^* 进行 b 次 K 均值聚类,将它分别划分为 b 对子簇.}

5:for $i = 1$ to b do

6:使用基本 K 均值,将 C^* 分成 2 个子簇 C_1^* 和 C_2^*.

7:end for

8:从 b 对子簇中选择具有最小总 SSE 的 1 对子簇.

9:将这 2 个子簇添加到簇集 S 中.

10:until 簇集 $\boldsymbol{S}$ 中包含 K 个簇.

11:由 $C_1^*, C_2^*, \cdots, C_K^*$ 得到 $\boldsymbol{H}$(若 $y_i = \boldsymbol{U}^T x_i$ 属于 C_k^*,则 $h_{ik} = 1$,否则 $h_{ik} = 0$).

二分 K 均值聚类方法在数据集 $\boldsymbol{Y} = \boldsymbol{U}^T\boldsymbol{X}$ 中进行,并得到 $\boldsymbol{H}$. 利用

$$m_k = \sum_i h_{ik} x_i / n_k \quad k = 1, 2, \cdots, K$$

计算出 $\boldsymbol{X}$ 的 K 个簇心 $M = \{m_1, m_2, \cdots, m_K\}$,以及 $\boldsymbol{S}_w$ 和 $\boldsymbol{S}_b$. 引入 $\boldsymbol{H}$ 后,M,$\boldsymbol{S}_w$ 和 $\boldsymbol{S}_b$ 的计算可以表示成以下公式:

$$M = \boldsymbol{X}\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1} \quad ⑧$$

$$\boldsymbol{S}_w = (\boldsymbol{X} - M\boldsymbol{H}^T)(\boldsymbol{X} - M\boldsymbol{H}^T)^T \quad ⑨$$

$$\boldsymbol{S}_b = M\boldsymbol{H}^T\boldsymbol{H}M^T \quad ⑩$$

下面是 LDA 算法的细节:

LDA 算法:

输入:$\boldsymbol{H}$ 和簇集 $\boldsymbol{S} = \{C_1^*, C_2^*, \cdots, C_K^*\}$.

输出:U.

1:计算 $M=XH(H^{\mathrm{T}}H)^{-1}$.

2:计算 $S_w=(X-MH^{\mathrm{T}})(X-MH^{\mathrm{T}})^{\mathrm{T}}$.

3:计算 $S_b=MH^{\mathrm{T}}HM^{\mathrm{T}}$.

4:求解矩阵 $S_w^{-1}S_b$ 的特征问题.

5:取最大的 d_1 个特征值对应的特征向量为变换矩阵 U 的列向量,得到 U.

3　LDA－BKM 算法的计算复杂度

从算法的执行过程看,它的计算复杂度主要是 $t\times$(执行一次 LDA + 执行一次 BKM 的时间复杂度),其中 $t\leqslant 10$ 是指该算法到收敛所需的迭代次数[8]. 因此,计算复杂度应该是二分 K 均值的 $O(dnt)$ 加上 LDA 的 $O(d^2nt)$,其中 d 是数据的维数,n 是样本点的个数.

4　实验与结果分析

4.1　实验数据集

在实验中,使用了范围较大的数据集(见表1).

在采用的数据集中,样本数从47到8 280,维度从4维到1 000维,类别数从2类到20类. 共采用的14个数据集,其中8个数据集分别是来自 UCI 数据库的 Gigits, Iris, Glass, Soybean, Protein, Zoo, Ionosphere 和 Wine,此外,还有 CSTR, Log, Reuters, WebACE, WebKB4, WebKB 等6个经常在文本聚类中使用的标准文本数据集,这6个数据集中的文本通过向量空间模型被表达成词向量,且这些文本数据集均经过预处理.

表1　实验数据集

数据集	样本数	维数	类别数
Gigits	7 494	16	10
Iris	150	4	3
Glass	214	9	7
Soybean	47	35	4
Protein	116	20	6
Zoo	101	18	7
Ionosphere	351	34	2
Wine	178	13	3
CSTR	475	1 000	4
Log	1 367	200	8
Reuters	2 900	1 000	10
WebACE	2 340	1 000	20
WebKB4	4 199	1 000	4
WebKB	8 280	1 000	7

4.2　结果分析

使用准确率来衡量聚类的性能. 准确率可以发现簇与类别之间一对一的关系以及哪个簇包含来自相对应类的数据. 准确率可以用如下公式表示:

$$Accuracy=Max\left(\sum_{C_k,L_m}T(C_k,L_m)\right)/n$$

其中,n 是样本个数,C_k 是指第 k 个簇,L_m 是指第 m 个类. $T(C_k,L_m)$ 表示属于第 m 个类的样本被聚类到第 k 个簇的样本个数. 因此,准确率可以被看作是最大化所有簇与类别对的 $T(C_k,L_m)$ 之和,其中每一对簇与类别对之间互不重复.

基于8个来自 UCI 数据库的数据集,将 LAD－BKM 算法与二分 K 均值算法,PCA－Kmeans 算法进行比较,PCA－Kmeans 表示基于 PCA 的 K 均值聚类算法,即先用 PCA 降低数据的维度,再接着使用 K 均值聚类. 表2为在 UCI 数据集上得到的聚类准确率,该结果是5次实验的平均值.

表2　UCI 数据集的聚类准确率

数据集	二分 K 均值	PCA－Kmeans	LDA－BKM
Gigits	0.719	0.769	0.785
Iris	0.896	0.886	0.985
Glass	0.477	0.451	0.521
Soybean	0.684	0.723	0.767
Protein	0.485	0.525	0.597
Zoo	0.775	0.793	0.854
Ionosphere	0.721	0.721	0.723
Wine	0.706	0.707	0.839

对于文本数据集,将 LDA－BKM 与标准的 K 均值算法和自适应子空间聚类(ASI)进行比较. 结果显示在图1中.

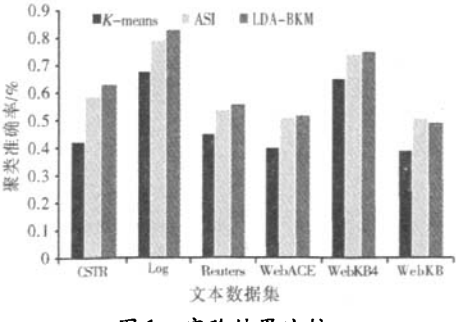

图1　实验结果比较

从表2中可见,在本文所采用的 UCI 数据集上,LDA－BKM 算法拥有最佳的准确率,并且在某些数据集上(如 Iris, Glass, Soybean, Protein, Zoo, Wine),

LDA - BKM 的准确率较二分 K 均值与 PCA - Kmeans 有较大的提高. 对于文本数据集,LDA - BKM 在 CSTR,Log,Reuters,WebACE,WebKB4 等数据集上具有最高的准确率. 总的来说,LDA - BKM 算法的效率是最佳的或者接近最佳. 因此,LDA - BKM 算法是可行的,并且是具有竞争力的.

5 结论

本文分析了 LDA 与二分 K 均值聚类的特点,然后将两者耦合在一起,并提出了一种针对高维数据集的自适应聚类方法 LDA - BKM 算法. 在这个算法中,利用 LDA 来实现维归约,将高维数据集变换成低维数据集,在低维数据集上执行二分 K 均值聚类方法来把数据分成簇,引入簇成员指示矩阵得到原数据集上的簇. 然后在此基础上再利用 LDA 进行维归约,这个过程反复进行下去,不断地修正前面得到的聚类结果,直到全局最优. 首先,该方法能够避免维灾难问题,同时它也充分发挥了二分 K 均值聚类方法在低维数据集上的效率. 其次,与先前的一些算法比较,LDA - BKM 改进了它们的一些缺点. 比如,自适应维归约算法只是最优化类间散度,自适应子空间迭代算法只是最优化类内散度,而 LDA - BKM则不仅最小化类内散布矩阵,而且最大化类间散布矩阵. 对一些通用高维数据集的实验结果也证明了 LDA - BKM 算法的有效性.

参考文献:

[1] 贺玲,蔡益朝,杨征. 高维数据聚类方法综述[J]. 计算机应用研究,2010,27(1):23.

[2] 余元辉,邓莹. 一种新的高维数据聚类自适应算法的研究[J]. 沈阳化工大学学报,2010,24(2):165.

[3] Ding C,He X,Zha H,et al. Adaptive dimension reduction for clustering high dimensional data[C]//Proc IEEE Int'l Conf Data Mining,Washington DC:IEEE Computer Society,2002.

[4] 唐懿芳,钟达夫. 主成分分析方法对数据进行预处理[J]. 广西师范大学学报:哲学社会科学版,2002(S1):223.

[5] Berry M W. Large scale singular value computations[J]. Int J of Supercomp Appli,1992,6(1):13.

[6] Loris Nanni, Alessandra Lumini. Orthogonal linear discriminant analysis and feature selection for micro-array data classification[J]. Expert Syst with Appli,2010,37(10):7132.

[7] Steinbach M,Karypis G,Kumarv. A comparison of document clustering techniques[C]//Proc of the Sixth ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining (KDD 2000),Boston:MA,2000:20 - 23.

[8] Li T, Ma S. IFD: Iterative feature and data clustering[C]//Proc SIAM Int'l Conf on Data Mining (SDM 2004),Florida:Society of Industrial and Applied Mathematics,2004:472 - 476.

(上接第100页)

核剂 $Na_2HPO_4 \cdot 12H_2O$ 和增稠剂明胶做添加剂后,三水醋酸钠的过冷度得到了很好的抑制,但是应用时比较麻烦. 相比较十水硫酸钠而言,不具有循环利用热量的优点. 因此本文选用十水硫酸钠作为大棚内的储热材料,更具有应用价值.

4 结语

在晴天情况下,利用 $Na_2SO_4 \cdot 10H_2O$ 作为储热材料储存白天多余的太阳能,可使塑料大棚内日平均气温提高 2.4 ℃. 夜间气温平均提高 5.4 ℃. 因此,储热材料具有明显的白天蓄热、夜间放热的效果,并且还具有长期蓄热的潜力,可满足农作物在连续阴天时的保温需要.

参考文献:

[1] 王克振,郭长华. 高温相变储热铝合金材料的研究现状及展望[J]. 兰州工业高等专科学校学报,2010,17(2):53.

[2] 刘成楼,陈学联. 相变储能保温隔热砂浆的研究与应用[J]. 中国涂料,2010,25(5):36.

[3] 黄金. 高温复合相变蓄热材料及其在工业窑炉蓄热式燃烧系统中的应用[J]. 冶金能源,2006,25(6):31.

[4] 张正国,庄秋虹,张毓芳,等. 硬脂酸丁酯/膨润土复合相变材料的制备及其在储热建筑材料中的应用[J]. 现代化工,2006,26:131.

[5] 王志强,曹明礼,龚安华,等. 相变储热材料的种类、应用及展望[J]. 安徽化工,2005(2):8.

[6] 郭成州,黎锦清. 太阳能热利用储能材料的研究[J]. 国外建材科技,2007,28(5):19.

[7] 邓安仲,李胜波,沈小东,等. 相变温控混凝土相变储热性能试验研究[J]. 后勤工程学院学报,2007(2):88.

[8] 李金田,茅靳丰,李伟华,等. 三水醋酸钠的过冷机理与实验研究[J]. 制冷学报,2009(5):35.