基于 SVC 对古代玻璃制品的分类规律与风化机制研究

摘要

古代玻璃制品是丝绸之路上中西方文化交流的重要的历史见证，对历史研究具有非常重要的价值。本文主要研究不同类别古代玻璃风化过程中化学成分变化规律，并以此预测和评估不同种类玻璃的分类规律与风化机制。研究在文物的保护、修复、风险预测上具有较大的现实意义。

针对问题一，主要分析三个问题，一是分析表面风化与玻璃类型、纹饰、颜色的定类变量关系，我们使用**卡方检验**法检验这三组变量之间是否存在显著差异，结果证明只有表面风化和玻璃类型的类别划分之间存在显著差异。二是分析文物样品表面不同风化程度化学成分含量的统计规律，我们首先对高钾、铅钡玻璃样品表面的化学成分含量进行**描述性统计**，然后使用 ***Shapiro-Wilk* 检验**验证数据分布的正态性；高钾玻璃的数据不满足正态分布，故选用 ***Mann-Whitney U* 检验**法进行非参数检验；铅钡玻璃的数据满足正态分布，故进行**单因素方差分析**；结果发现高钾、铅钡玻璃的化学成分在风化前后整体上发生显著变化，且其中 SiO_2 指标的值很大程度上可以指代样本的风化程度。三是根据风化点检测数据预测风化前化学成分的含量，我们使用 SiO_2 作为指标变量，根据现有文献中的资料建立**函数模型**，分别拟合每个化学成分的数据，使得函数可以预测过去任意时间该化学成分的含量。研究总结了特定化学成分含量的拟合流程并给出了具体的范例。

针对问题二，首先，找到高钾玻璃和铅钡玻璃的分类规律；文章使用以欧氏距离与**线性核函数**为核心的 ***SVC* 模型**，利用附件所给数据进行训练，得到各个化学成分的分类指标。其次，将两类玻璃分别划分为多个亚类；本文根据 ***Critic* 权重法**选择合适的化学成分，使用 ***K-Means* 聚类**算法对两个类别进行亚类划分，我们仅使用 $p<0.01$ 的数据属性，且属性在数据中权重较大，故分类具有较强的合理性。各个亚类之间的化学成分含量区间不重叠，具有较好的分类敏感性。结合问题一的结论对亚类命名为**似风化亚类**和**无风化亚类**，给出两亚类的指标范围，并揭示了部分样本中可能存在肉眼不可见的**隐性风化**现象。

针对问题三，在问题二原有的 *SVC* 模型基础上对未知类别玻璃文物的类型进行鉴定，得出待预测样本 A1~A8 的所属的玻璃类型，*ROC* 曲线右下角面积为 0.49999>0.45，模型敏感度非常高；本文进一步根据 3σ 原理判定 A5 为风化玻璃上的未风化点，且 A2、A6、A7 均非严重风化点；并通过问题二所得到的聚类模型，对对未风化数据（A1、A3、A4、A5、A8）进行亚类的归类。

针对问题四，为研究化学成分之间的关联关系，以及各类别之间的差异性，本文使用 ***Apriori* 算法**。通过样本数据的频繁集，得出数据集幂集元素的支持度、置信度及其关系，结合问题一得到的相关系数进行合理分析。结果发现化学成分 *SrO* 与其他众多元素均具有较强的关联性，但在两种玻璃类型中具有关联性与显著相关性的元素不完全一致。

关键词：　古代玻璃制品　相关性分析　SVC　K-means 聚类　Apriori 算法

一、问题重述

玻璃是古代中西方贸易往来的宝贵物证，促进了中西方的文化交流。玻璃的主要原料是石英砂，主要化学成分是二氧化硅（SiO_2）。与现代玻璃不同，古代玻璃极易受埋藏环境的影响而风化，在风化过程中，内部元素与环境元素进行大量交换导致其成分比例发生变化，从而影响对其类别的正确判断。

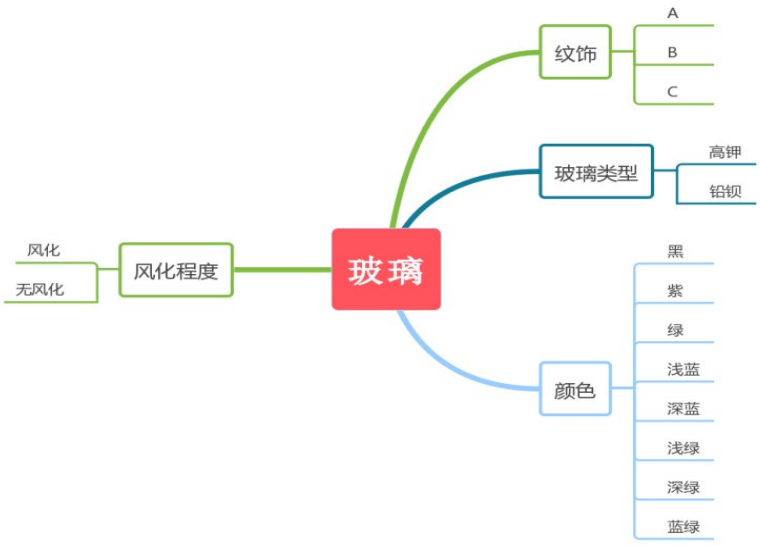
现有一批我国古代玻璃制品的相关数据，专家依据其化学成分，利用相关技术将它们分为高钾玻璃和铅钡玻璃两类。有以下问题需要解决：

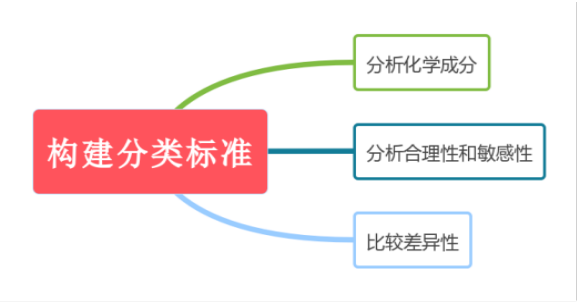
问题一 分析玻璃文物表面风化与其玻璃类型、纹饰和颜色的关系；结合玻璃的类型，分析文物样品表面有无风化化学成分含量的统计规律，并根据风化点检测数据，预测其风化前的化学成分含量。

问题二 依据附件数据分析高钾玻璃、铅钡玻璃的分类规律；对于每个类别选择合适的化学成分对其进行亚类划分，给出具体的划分方法及划分结果，并对分类结果的合理性和敏感性进行分析。

问题三 对附件表单 3 中未知类别玻璃文物的化学成分进行分析，鉴别其所属类型，并对分类结果的敏感性进行分析。

问题四 针对不同类别的玻璃文物样品，分析其化学成分之间的关联关系，并比较不同类别之间的化学成分关联关系的差异性。





二、问题分析与模型假设

2.1 问题一的分析与假设

通过查阅相关文献我们发现，高钾玻璃与铅钡玻璃的风化机制有较大区别[1]。高钾玻璃的风化主要表现为离子态的 K_2O 与空气中的 H_2O 和 CO_2 的反应析出碱流失，风化后质量减少。反应的离子方程式如下：

$$O^{2-}+H_2O=2OH^-$$

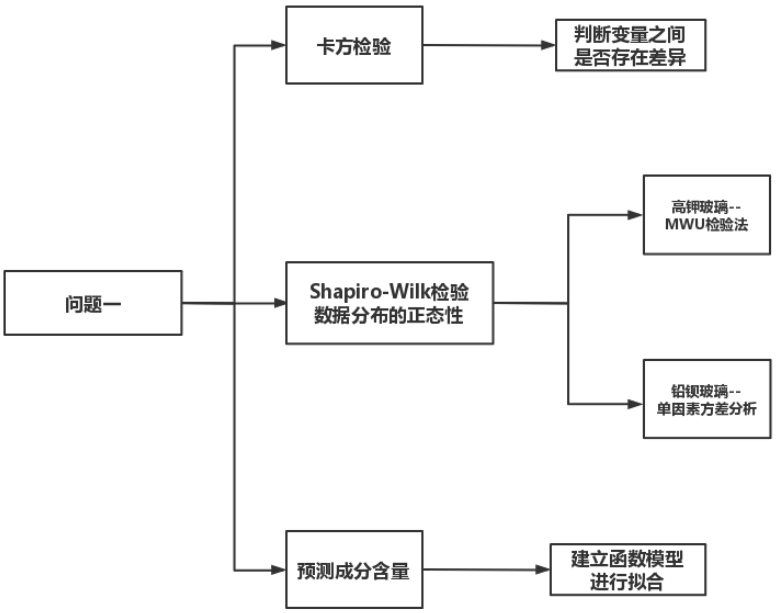
$$O^{2-}+CO_2=2CO_3^{2-}$$

铅钡玻璃的风化主要表现为离子态的 PbO 和 BaO 与空气中 H_2O、CO_2、SO_2 和 O_2 反应生成 $PbCO_3$ 和 $BaSO_4$ 等沉淀物，风化后质量增加。反应的离子方程式如下：

$$Pb^{2+}+O^{2-}+CO_2=PbCO_3\downarrow$$

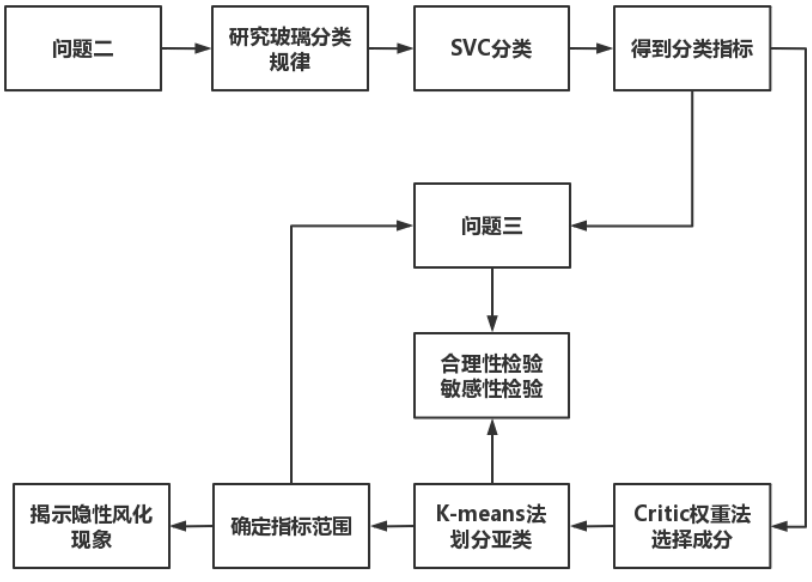
$$Ba^{2+}+SO_3^{2-}+O_2=BaSO_4\downarrow$$

因此，后续应**针对这两种玻璃的不同的风化特点分类开展研究。**



问题一流程图

2.2 问题二与问题三的流程分析



问题二、问题三流程图

2.4 问题四的分析与假设

为研究化学成分之间的关联关系，以及各类别之间的差异性，本文使用 *Apriori* 算法。通过样本数据的频繁集，得出数据集幂集元素的支持度、置信度及

其关系，结合问题一得到的相关系数进行合理分析

为使用 *Apriori* 算法，我们需要定义数据在数据集中“出现”的布尔值：本文定义某一项数据在当前元组中“出现”等价于该项数据标准化后的值大于 0。

三、符号说明

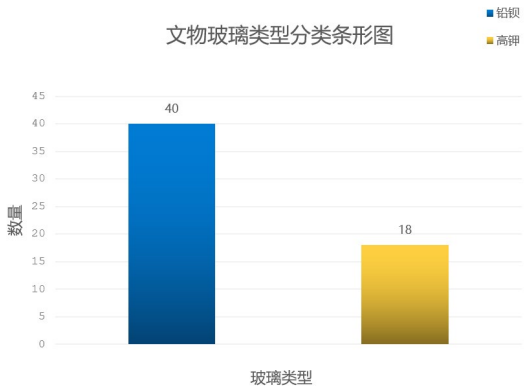
符号	说明
ρ	风化后 SiO_2 占比
ρ	风化前 SiO_2 占比
$\bar{\rho}$	风化后 SiO_2 占比平均值
$\overline{\rho_0}$	风化前 SiO_2 占比平均值
M	风化后 SiO_2 质量
M_0	风化前 SiO_2 质量
Q	某化学成分变化量
t	当前时间
t_0	开始风化的时间
φ	风化后某化学成分占比
φ_0	风化前某化学成分占比
$\bar{\varphi}$	风化后某化学成分占比平均值
$\bar{\varphi}_0$	风化前某化学成分占比平均值
μ	速率常数
k	校正常数
φi	某时刻某化学成分占比
ti	某时刻

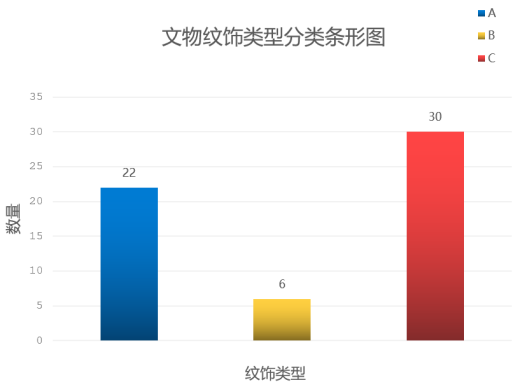
四、问题一模型的建立与求解

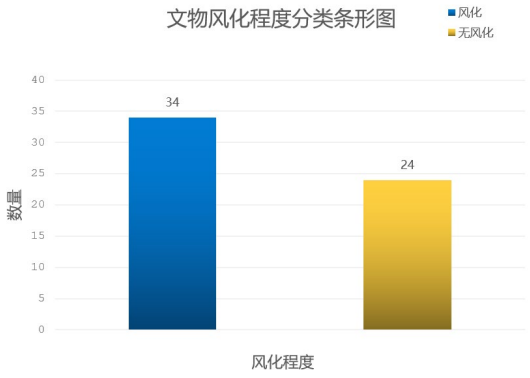
4.1 古代玻璃样本化学成分整体分布情况的描述性统计

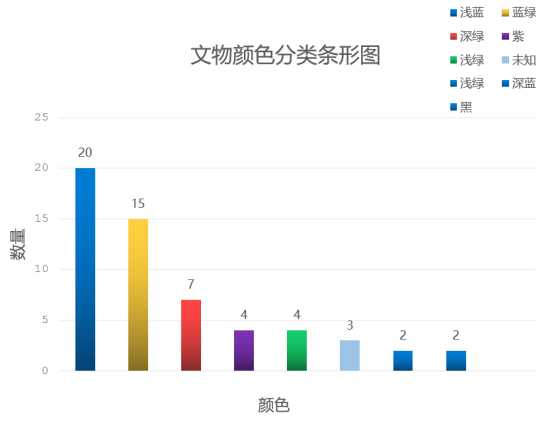
我们首先对不符合条件的数据进行筛除。计算成分比例累加和后，仅有 15 号、17 号样本的总比例不在[85%, 105%]之间，故对这两个成分予以剔除。

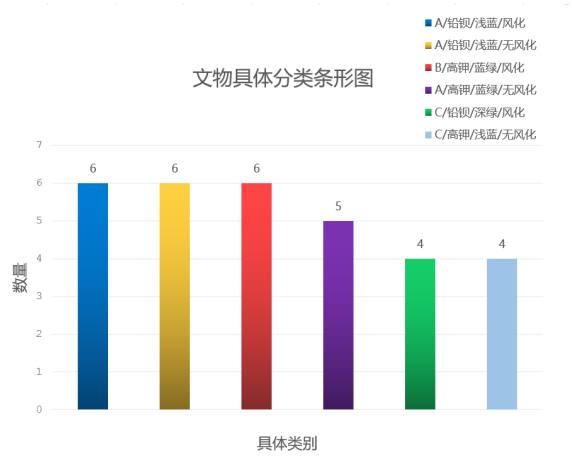
然后，对古代玻璃样本化学成分的整体分布情况进行描述性统计。











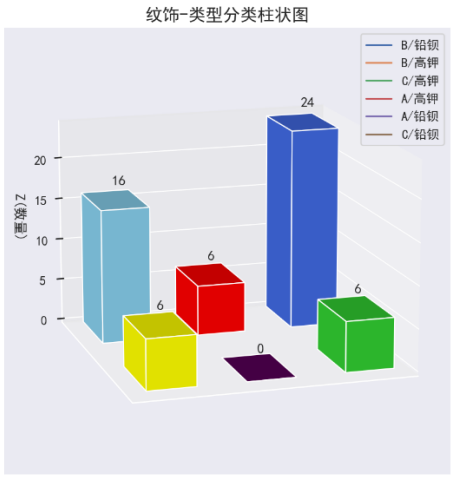


图 1~图 6 玻璃样本特征描述性统计条形图

对这些玻璃文物的表面风化与其玻璃类型、纹饰和颜色的关系进行分析。可以直观地得出： *A* 类纹饰的文物大多表面无风化，*B* 类纹饰的文物表面均为风化，*C* 类纹饰的文物表面风化和无风化约各占一半；风化的铅钡玻璃文物的数量较多，无风化的高钾玻璃文物的数量较多；蓝绿色的文物表面风化和无风化约各占一半，浅蓝色的文物表面多为无风化，深绿色的文物表面多为风化。

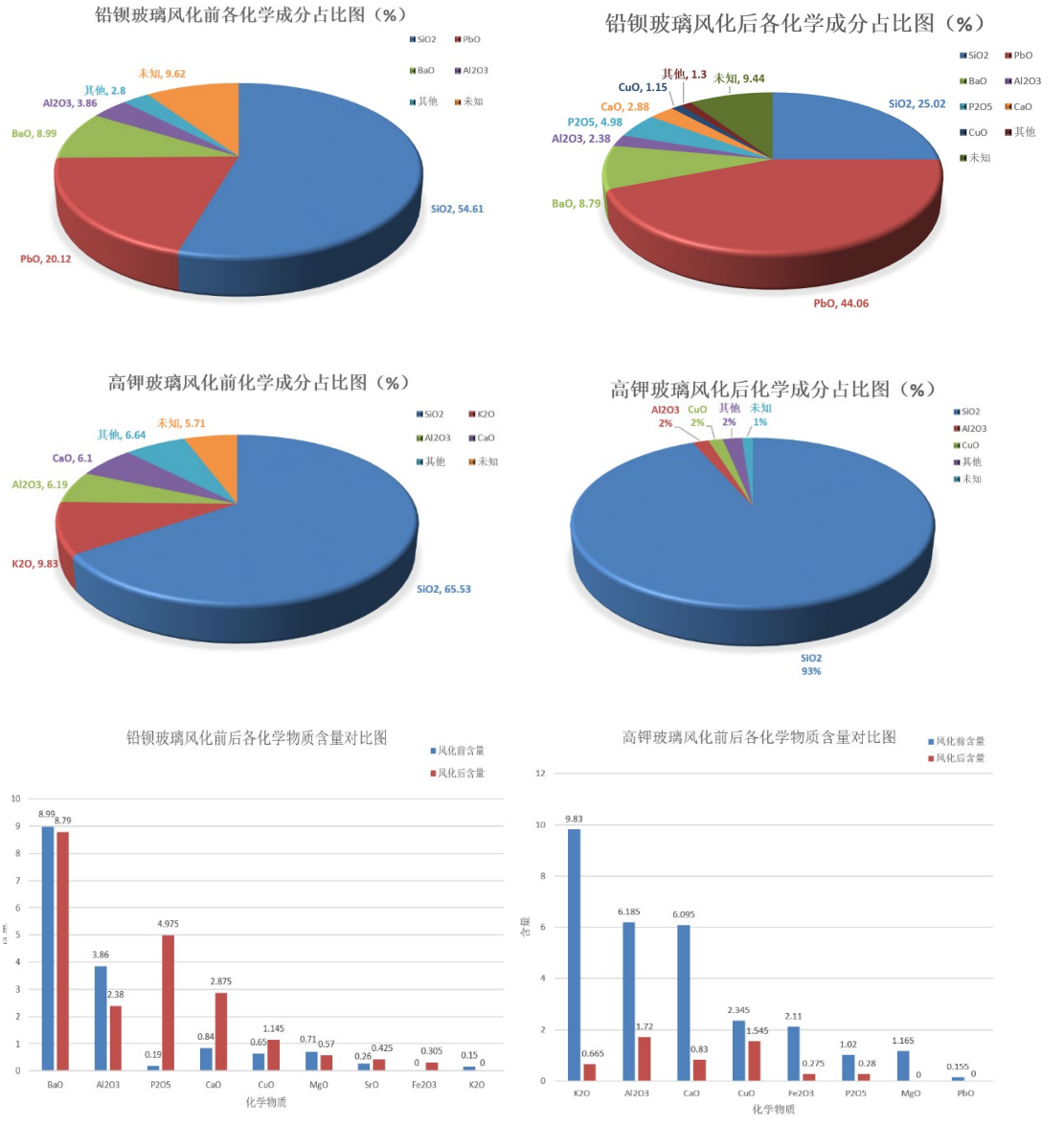


图 7~图 12 高钾、铅钡玻璃风化前后化学成分含量对比图

以下结合文物的玻璃类型，分析风化前后样品表面化学成分含量的变化规律。对于高钾玻璃类型的文物，未风化状态下大部分为蓝绿色，二氧化硅含量高，

受风化程度较低；风化后，其组成成分中氧化物（除二氧化硅）占比显著下降，只有该玻璃类型的文物在风化后有 *B* 类纹饰。

对于铅钡玻璃类型的文物，未风化状态下颜色分布不明显，二氧化硅含量少，受风化程度较高；风化后，其组成成分中氧化铅、氧化钙、氧化铜、五氧化二磷、氧化锶这五种物质占比上升，二氧化硅含量下降。

4.2 古代玻璃表面风化与各特征关系卡方分析

卡方检验可以验证定类变量之间的差异性。*Pearson* 卡方检验由 *Karl Pearson* 提出，广泛运用于分类变量的比较检验中。对于一个 $R\times C$ 的列联表，首先计算其各项的理论频数：

$$T_{ij}=\frac{\sum_{c=i}^{C}A_{ic}\sum_{r=1}^{R}A_{rj}}{N}$$

Pearson 卡方的计算值如下：

$$\chi^2=\sum_{i=1}^{R}\sum_{j=1}^{C}\frac{(A_{ij}-T_{ij})^2}{T_{ij}}$$

首先将表面风化分别与玻璃类型、纹饰、颜色进行 *Pearson* 卡方检验，以探索表面风化与玻璃类型、纹饰、颜色的数据分类是否存在显著差异。检验结果如下表所示：

题目	名称	表面风化		X^2	P
		无风化	风化		
纹饰	A	14	14	5.722	0.057
	B	0	6		
	C	11	22		
类型	铅钡	13	36	9.066	0.003**
	高钾	12	6		
颜色	浅绿	2	1	11.557	0.116
	浅蓝	6	16		
	深绿	3	4		
	深蓝	3	0		
	紫	2	4		
	绿	1	0		

蓝绿	8	9	
黑	0	4	

表面风化与玻璃类型、纹饰、颜色的数据分类差异的卡方检验

其中*、**、***分别表示在 0.05、0.01、0.001 的水平上显著，下同。

对于风化和颜色，卡方检验的显著性 *P* 值为 0.116，不呈现显著性，因此不能拒绝原假设，即表面风化和颜色两种分类在数据上不存在显著性差异。

对于风化和纹饰，卡方检验的显著性 *P* 值为 0.057，不呈现显著性，因此不能拒绝原假设，即表面风化和纹饰两种分类在数据上不存在显著性差异。

对于风化和类型，卡方检验的显著性 *P* 值为 0.003，呈现显著性，因此拒绝原假设，即表面风化和类型两种分类在数据上存在显著性差异。

4.3 不同风化程度古代玻璃化学成分含量的非参数检验与方差分析

研究中我们发现，由于 SiO_2 基本不参与到玻璃风化的化学反应过程中，所以同一样本中 SiO_2 的质量基本不随时间改变。我们可以将 SiO_2 作为指标量，来估计风化前后玻璃的整体化学成分是否发生变化。

下面对风化、未风化 SiO_2 的样本差异进行检验，以反映总体化学成分的变化。在对样本差异进行检验之前，使用 *Shapiro-Wilk* 检验可以验证小样本数据分布的正态性，进而选择合适的差异检验方法。对高钾玻璃 SiO_2 含量的 *Shapiro-Wilk* 检验中，显著性 *P* 值为 0.000，拒绝原假设，因此数据不满足正态分布，应进行非参数检验。由于样本数量较小，因此对高钾玻璃 SiO_2 含量在风化、未风化两种样本上的差异性选用 *Mann-Whitney U* 检验法进行秩和检验。统计量 *U* 的计算值如下：

$$U_A=n_An_B+\frac{n_A(n_A+1)}{2}-R_A$$

$$U_B=n_An_B+\frac{n_B(n_B+1)}{2}-R_B$$

$$U=min(U_A, U_B)$$

其中 *RA*、*RB* 为 *A*、*B* 两组样本的秩和。经计算，拒绝原假设，说明**高钾玻璃中风化、未风化两种样本在 SiO_2 含量上存在显著差异。**

	个案数	秩平均值	秩的总和	U	显著性
未风化	12	6.500	78.000	0.000	0.000
风化	6	15.500	93.000		

高钾玻璃中 SiO_2 含量关于表面风化两种样本的秩和检验

对铅钡玻璃 SiO_2 含量进行 *Shapiro-Wilk* 检验，显著性 *P* 值为 *0.102*，不呈现显著性，因此数据满足正态分布；方差齐性检验中 *P* 值 *0.315*，不呈现显著性，说明数据满足方差齐性，可以进行方差分析。故对铅钡玻璃 *SiO2* 含量在严重风化、风化、未风化三种样本上的差异性进行单因素方差分析（*F* 检验）。首先计算组间平方和 *SSA* 与组内平方和 *SSE*：

$$SSA = \sum_{i=1}^{r} n_i(\overline{X}_i - \overline{X})^2$$

$$SSE = \sum_{i=1}^{r}\sum_{j=1}^{n_i} (X_{ij} - \overline{X}_i)^2$$

组间方差 *MSA* 和组内方差 *MSE*：

$$MSA = \frac{SSA}{k-1}$$

$$MSE = \frac{SSE}{n-k}$$

方差分析结果 *P* 值为 0.000，因此统计结果显著，说明 SiO_2 含量在不同的程度的风化上存在显著差异。η^2值为 0.702，说明数据的差异有 70.2%是来源于不同组别间的差异。*Cohen's f* 值为 1.535＞0.8，说明故**铅钡玻璃中不同的程度的风化样本之间 SiO_2 含量存在很大程度的差异。**

变量名	变量值	样本量	平均值	标准差	F	P
SiO2	未风化	23	54.66	11.829	54.167	0.000
	风化	23	27.056	9.005		
	严重风化	3	8.48	7.487		

铅钡玻璃中 SiO_2 含量关于表面风化的方差分析

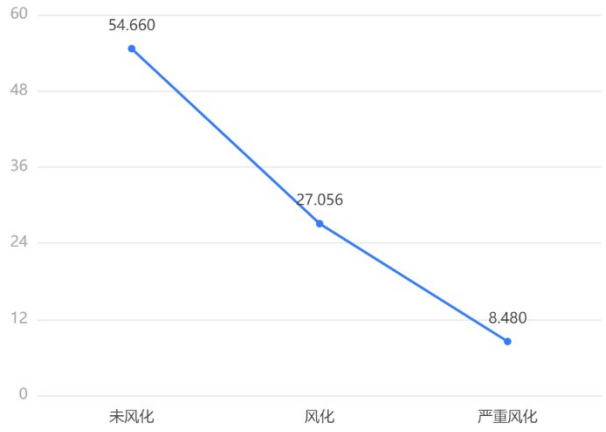


图 13 单因素方差对比分析图

以上分析**证明，高钾、铅钡玻璃的化学成分在风化前后在整体构成上均发生显著变化，其中 SiO_2 指标的值很大程度上可以指代样本的风化程度。**

4.4 古代玻璃不同化学成分含量关系的相关性分析

为了进一步揭示各化学成分含量的关系，本研究对各成分含量进行相关性分析。因为各成分分别取自风化、未风化的点，符合正态分布的可能性较低，故应使用 *Spearman* 等级相关分析。*Spearman* 相关系数 ρ_s 被定义为等级变量之间的 *Pearson* 相关系数：

$$\rho_S = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

相关性分析的结果如下：

	SiO_2	K_2O	Al_2O_3	CaO	CuO	Fe_2O_3	P_2O_5	MgO	PbO	SrO
SiO_2	1									
K_2O	-0.781***	1								
Al_2O_3	-0.831***	0.506*	1							
CaO	-0.752***	0.690**	0.521*	1						
CuO	-0.430	0.179	0.246	0.344	1					

Fe_2O_3	-0.783***	0.457	0.748***	0.537*	0.643**	1				
P_2O_5	-0.530*	0.156	0.518*	0.023	0.466	0.517*	1			
MgO	-0.546*	0.256	0.735**	0.087	0.204	0.569*	0.643**	1		
PbO	-0.443	0.241	0.602**	0.256	0.047	0.323	0.121	0.295	1	
SrO	-0.524*	0.404	0.511*	-0.052	0.141	0.409	0.500*	0.666**	0.368	1

高钾玻璃中各化学成分含量的相关性

	SiO_2	PbO	BaO	Al_2O_3	P_2O_5	CaO	CuO	MgO	SrO	Fe_2O_3	K_2O
SiO_2	1										
PbO	-0.757***	1									
BaO	-0.28	-0.100	1								
Al_2O_3	0.425**	-0.389**	-0.369**	1							
P_2O_5	-0.538***	0.361*	-0.169	-0.019	1						
CaO	-0.490***	0.351*	-0.157	0.123	0.502***	1					
CuO	0.447**	0.094	0.495***	-0.325*	0.221	-0.037	1				
MgO	0.123	0.076	-0.435**	0.674***	0.153	0.325*	-0.266	1			
SrO	-0.552**	0.362*	0.182	-0.197	0.255	0.308*	0.210	0.611	1		
Fe_2O_3	0.044	0.081	-0.443**	0.387**	0.225	0.411**	-0.390**	0.074	0.427	1	
K_2O	0.308*	-0.336*	-0.001	0.512**	-0.140	-0.035	-0.198	0.362*	-0.167	0.149	1

铅钡玻璃中各化学成分含量的相关性

在此仅展示玻璃中含量中位数大于 0%的化学成分。

4.5 古代玻璃风化前化学成分含量的预测模型

假定风化的过程中单个采样点 SiO_2 的质量固定不变，风化前总质量为 M_0，风化后为 M；风化前各 SiO_2 占比为 ρ_0，风化后占比为 ρ，可得：

$$\rho M=\rho_0 M_0$$

王承遇分析了硅酸盐玻璃风化的影响因素[2]，指出在温度湿度恒定的情况下，碱的变化量 Q 与时间的关系为如下。其中，风化开始时间点为 t_0，当前时间

点为 t。μ 为速率常数，不同玻璃的不同化学成分中速率常数均不相同。

$$Q = M\sqrt{t - t_0}$$

设风化前某个非 SiO_2 成分占比为 φ_0，风化后占比为 φ，有：

$$Q = M_0\varphi_0 - M\varphi = \left(\frac{\rho}{\rho_0}\varphi_0 - \varphi\right)\mu$$

用未风化的采样点的状态近似预测风化点风化前的化学成分含量。某种玻璃中每个化学成分的 μ 值，均可以由未风化样本点 SiO_2 的平均含量 ρ_0 和已风化的平均含量 ρ、未风化样本点非 SiO_2 的平均含量 φ_0 和已风化的平均含量 φ 进行估计：

$$\mu = m\frac{\left(\frac{\bar{\rho}}{\bar{\rho}_0}\bar{\varphi}_0 - \bar{\varphi}\right)\mathrm{M}}{\sqrt{t - t_0}}$$

考虑到高钾玻璃和铅钡玻璃的风化作用机制有很大不同，因此需要对二者分开进行处理。由于前面采用了均值估计，所以函数图像很有可能不经过点(t, φ)，故此处应对 φ_0 对 φ 的函数加上一个常数 k 使其经过(t, φ)：

$$\varphi_0 = \frac{\bar{\rho}_0}{\bar{\rho}}\left(\frac{M\sqrt{t - t_0}}{M} + \varphi\right) + k$$

其中：

$$k = \left(1 - \frac{\bar{\rho}_0}{\bar{\rho}}\right)\varphi$$

可得：

$$\varphi_0 = \varphi + \frac{\bar{\rho}_0 M\sqrt{t - t_0}}{\bar{\rho}M}$$

更广泛地，对于任意观测时间 t_i，其在该化学成分的成分含量的估计值为：

$$\varphi_i = \varphi_0 - \frac{\rho_0 M\sqrt{t - t_0}}{\rho M}$$

在已知某种文物发掘时间的情况下，重复以下的步骤，即可预测**所有风化样本在任意时间的所有化学成分含量**，如下流程：

STEP1: 对于某种玻璃类型，计算其 ρ_0、ρ、φ_0、φ。

STEP2: 根据该种玻璃类型中特定化学成分的 ρ_0、ρ、φ_0、φ 的值估计这种化学成分在此玻璃类型中的 μ 值。

STEP3: 根据特定样本 φ 的观测值与时间点 t_0、t 估计 φ_0。

STEP4: 根据 φ_0 估计任意观测时间 t_i 时 φ_i 的值。

示例：

假设所有高钾玻璃样本均为 2000 年前制造，要求对于 10 号样本的 K_2O 含量进行预测。不难得出在高钾玻璃中 ρ_0=67.984，ρ=93.963，φ_0=9.331，φ=0.543 对每个取样点均以 1mg SiO_2 为标准进行采样，μ 值估计为：μ=0.276(*mg*/年 $^{1/2}$)。已知风化后 K_2O 的成分含量为 0.92%，由于前面采用了均值估计，所以函数图像不经过(2000,0.92)，故此处应对函数加上一个常数 k 使其经过(2000,0.92)。得到 10 号样本的 $\varphi 0$ 估计值是 9.851，即 10 号样本 2000 年前 K_2O 的含量估计值为 9.851%。由该曲线我们也可以得到，10 号样本 1000 年前 K_2O 的含量估计值为 3.536%。

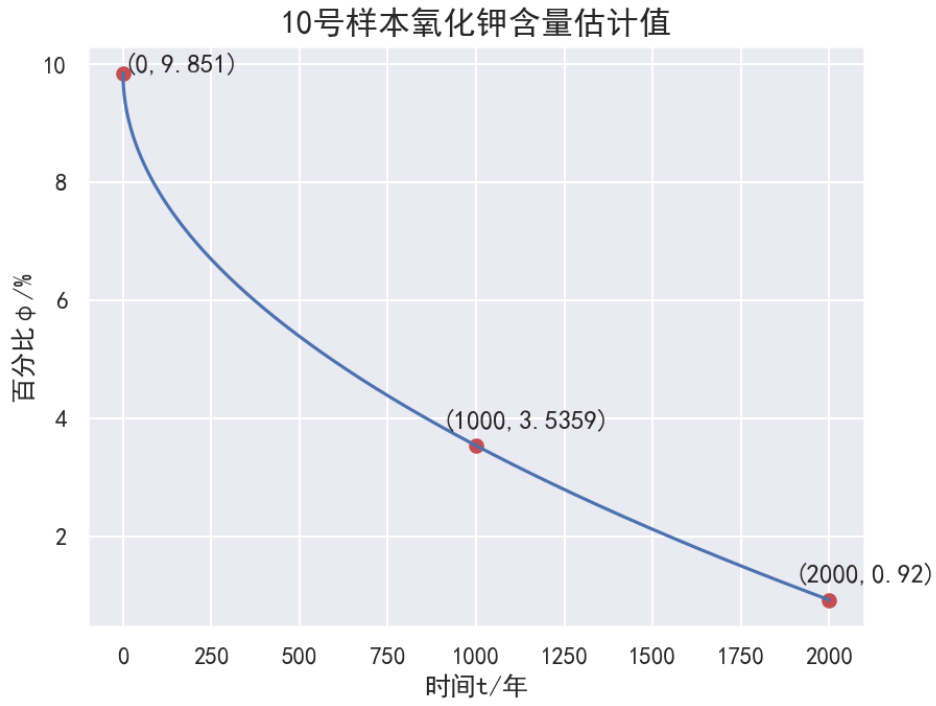


图 14 10 号样本氧化钾含量估计值与风化时间的关系

五、问题二模型的建立与求解

5.1 基于 *SVC* 的玻璃分类规律探索

为了发现高钾、铅钡两种玻璃的分类规律，本研究采用了支持向量分类智能算法。支持向量分类（*SVC*）是一种基于边界的分类算法。同时该算法能够处理异常值，并且不需要知道簇的数量。[3]

在进行支持向量机分类之前，首先对数据进行 *Z-Score* 标准化，首先计算均值：

$$\mu=\frac{1}{N}\Sigma_{i=1}^{N}x_i$$

标准差：

$$\sigma=\sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(x_i-\mu)^2}$$

标准化后的值：

$$z=\frac{x-\mu}{\sigma}$$

进行数据标准化后，我们使用 *SVC* 探索化学成分的分类规律，*SVC* 分类的原理是找到超平面使得各类数据点到超平面之间的距离最远。这里使用的距离为欧氏距离：

$$d=\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2}$$

其中的最优化函数为：

$$min_{w,b,\xi^*,\xi,\epsilon}=\left(\frac{1}{2}w^Tw+\upsilon\varepsilon+C\Sigma_{i=1}^{1}\left(\xi_i+\xi_i^*\right)\right)$$

其中 w 为数据所代表的向量，w^T^为其转置向量，为允许的软间隔（又称惩罚变量），ξ_i与ξ_i^*为数据的松弛变量。惩罚参数 C 用于调整的软间隔，惩罚参数越小，构造出的超平面对数据落在间隔带中的兼容性就更强；由于此处的数据量较

小，我们将 C 选择为默认值 1。

线性核函数主要用于数据线性可分的情况，对于线性可分数据，其分类效果很理想，因此我们通常首先用线性核函数来进行尝试，观察效果如何；这里划分得到的数据得分为 $1>0.8$，证明数据是线性可分的，因此我们认为线性核函数适用于这次划分。线性核函数计算如下：

$$K(x, x_i)=x\cdot x_i$$

然后运行线性核函数简化的 *SVC* 智能算法，使用平行坐标图展示多维度分类数据的结果。

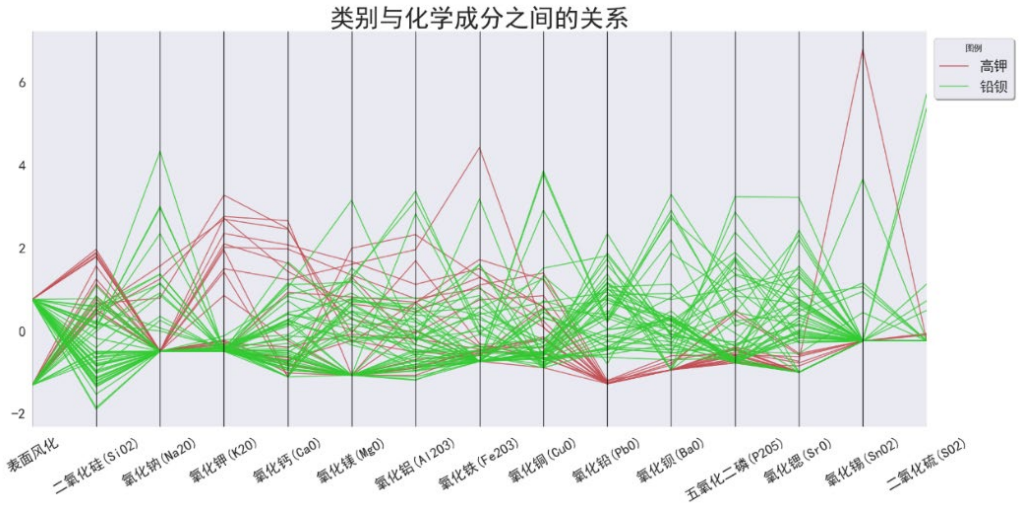


图 15 类别与化学成分之间的关系图

图中可以得出，高钾玻璃的二氧化硅、氧化钾含量较高；铅钡玻璃的氧化钠、氧化铅、氧化钡、五氧化二磷、氧化锶的含量较高。

5.2 基于 K-means 聚类算法的古代玻璃亚类划分

为了对每个类别选择合适的化学成分对其进行亚类划分，我们需要先了解每个类别中起到决定因素的化学成分，我们需要获取各个化学成分在两个类别中所占的权重。

在 *critic* 权重法和熵权法中，*critic* 权重法对数据量小的集合具有明显的优势，故在此我们使用原始数据进行 *critic* 权重计算。*critic* 权重法引入了考虑了指标的对比强度与冲突性，且无需对数据进行标准化，数据的标准差越大说明数据波动越大，即对比强度大，会导致数据的权重增大；而冲突性以数据的相关系数进行表示，指标间的相关系数越大，说明冲突性越小，那么其权重就更低。计算 *critic* 权重的具体方法如下：

原始数据矩阵（有 m 个元组，每一个元组有 n 个属性）：

$$X=\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

首先进行数据标准化，对于正向指标：

$$x_{ij}=\frac{x_{ij}\text{-}min(x_j)}{max(x_j)\text{-}min(x_j)}$$

对于逆向指标：

$$x_{ij}^{'}=\frac{max(x_j)-x_{ij}}{max(x_j)-min(x_j)}$$

计算信息承载量，其中波动性：

$$S_j=\sqrt{\frac{\sum_{i=1}^{m}(x_{ij}-\bar{x}_j)^2}{n-1}}$$

r_{ij} 是第i个指标与第j个指标的相关系数，则冲突性：

$$A_j=\sum_{i=1}^{n}(1-r_{ij})$$

信息量：

$$C_j=S_j\times A_j$$

计算权重：

$$W_j=\frac{C_j}{\sum_{j=1}^{n}C_j}$$

得到如下的权重直方图：

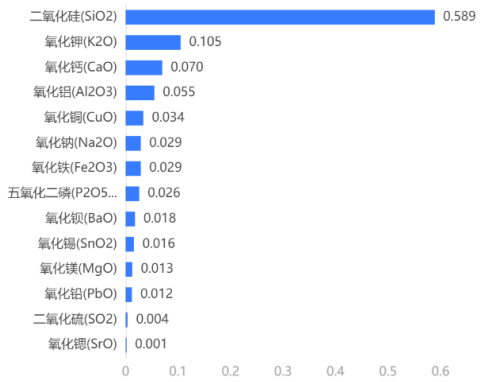


图 16 高钾玻璃指标重要度直方图

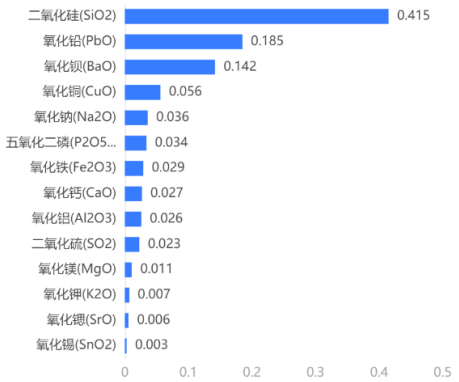


图 17 铅钡玻璃指标重要度直方图

由此我们得出在高钾玻璃中氧化钾，氧化钙，氧化铝，二氧化硅的权重较大；铅钡玻璃中氧化铅、氧化钡、氧化铜、二氧化硅的权重较大。我们使用上述权重占比较大的化学成分进行 *k-means* 聚类分析，并将不显著区分类别的化学成分移除（由于未风化和风化的数据项的化学成分含量具有显著的差异，在这里我们只使用未风化的高钾玻璃和铅钡玻璃进行探究），得到高钾玻璃的聚类结果如下：

	聚类类别（平均值±标准差）		F	P
	类别 1(n=9)	类别 2(n=3)		
氧化钾(K_2O)	10.818±2.37	4.87±4.718	8.897	0.014*
氧化钙(CaO)	6.363±2.64	2.24±2.363	5.714	0.038*
氧化铝(Al_2O_3)	7.349±2.346	4.433±1.603	3.891	0.077
二氧化硅(SiO_2)	63.624±3.558	81.063±5.368	43.059	0.000***

	聚类类别（平均值±标准差）		F	P
	类别 1(n=28)	类别 2(n=21)		
二氧化硅(SiO_2)	24.915±9.204	57.49±9.133	151.314	0.000***
氧化铅(PbO)	43.448±10.93	19.884±6.462	77.121	0.000***
氧化钡(BaO)	12.377±9.949	7.975±4.621	3.526	0.067
氧化铜(CuO)	2.415±2.992	1.165±1.273	3.215	0.079

高钾玻璃中二氧化硅、氧化钾、氧化钙对亚类的区分显著。
铅钡玻璃中二氧化硅和氧化铅对亚类的区分显著。

前面已经证明了 SiO_2 可以用于判断玻璃的风化程度，我们可以将高钾、铅

钡玻璃的未风化样本进一步分为无风化亚类和似风化亚类。无风化亚类中，玻璃化学成分自始至终基本没有发生任何变化；似风化亚类中，**虽然可能并不能通过肉眼直观发现玻璃的风化，但事实上样本内部的化学元素已经发生显著变化**，我们把这种变化称为隐性风化，这样的样本存在进一步风化的可能，需要采取合理的保护措施。

根据两种玻璃的不同性质，亚类划分的标准应该各不相同。

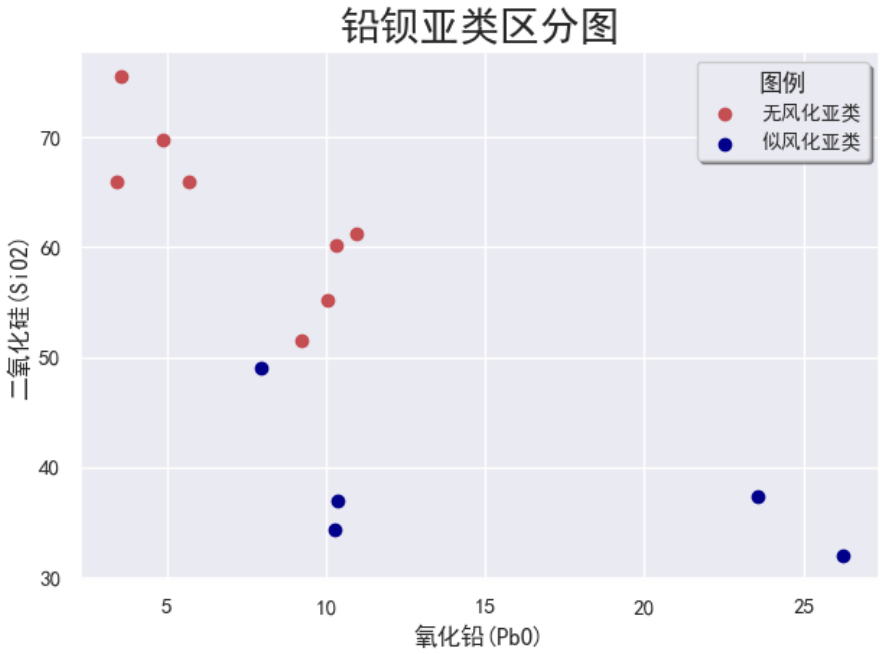


图 18 铅钡玻璃亚类区分图

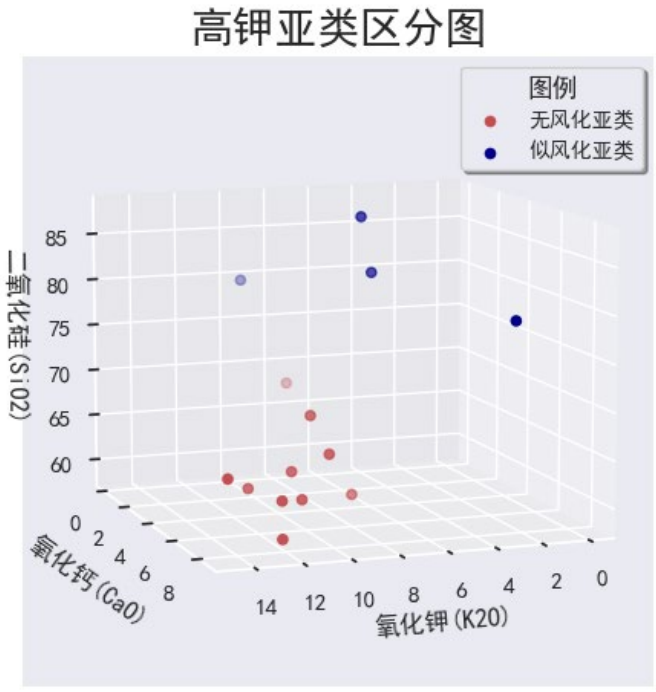


图 19 高钾玻璃亚类区分图

高钾玻璃的无风化亚类，而二氧化硅含量较低，氧化钾与氧化钙含量高；似风化亚类：而二氧化硅含量高，氧化钾与氧化钙含量低。

铅钡玻璃无风化亚类中，二氧化硅含量高，氧化铅含量低；似风化亚类：二氧化硅含量低，氧化铅含量高。

鉴于以上结论，我们可以选择无风化亚类的样本来预测问题一中某个化学成分风化之前的含量，以获得更加精确的结果。

以上分析中，我们只使用了 p 值<0.01 的数据属性，且属性在数据中权重较大，具有较强的合理性；根据聚类类别中各个具有较强显著性的化学成分的区间，我们可以看出，各个亚类之间的区间不重叠，具有较好的分类敏感性。

六、问题三模型的建立与求解

6.1 基于 SVC 的待测数据古代玻璃分类鉴别

根据问题二中使用 *SVC* 得到的超平面模型，我们可以对第三题的待测数据进行分类，得到以下结果：

数据项	高钾预测得分	铅钡预测得分	分类类别
A1	0.888	0.112	高钾
A2	0.003	0.997	铅钡
A3	0.001	0.999	铅钡
A4	0.012	0.988	铅钡
A5	0.201	0.799	铅钡
A6	0.936	0.064	高钾
A7	0.920	0.080	高钾
A8	0.036	0.964	铅钡

由该表可以看出，待测集数据被分配到类别的预测得分大多数大于 0.9，且全部都大于 0.7 的阈值，说明 *SVC* 对该待测集的分类具有较高的合理性。

为合理地判断当前 *SVC* 模型的整体敏感度，我们需要了解该模型召回率 *Recall* 和综合精度 *Precision* 之间的关系，由此绘制该模型的 *ROC* 图。召回率 *Recall* 表示真实数据中被我们成功预测的数据比率。

$$Recall(A)=\frac{Sum(A\_Success)}{Sum(A)}$$

ROC 图使用了假正率（False Position Rate）来代替综合精度，让我们得以衡量模型在尽量评判少数权重较小的属性时，导致权重更大的属性受到影响的变化。

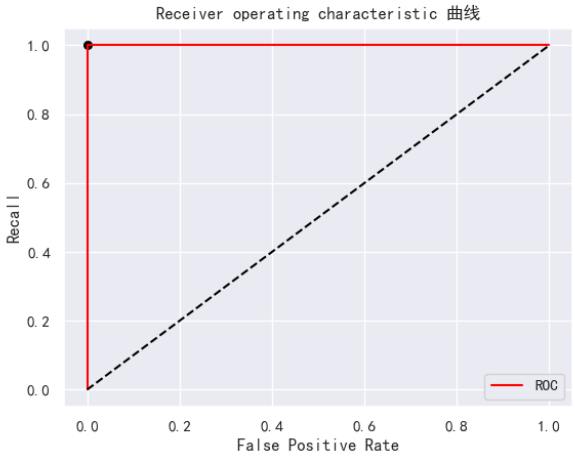


图 20 *ROC* 曲线

由上述定义可知，曲线右下角面积越大（红线与虚线之间的面积），模型的敏感性越强。图片说明，当假正率处于较小的值时，召回率已经达到 100%，模型 *ROC* 曲线与 $y=x$ 的面积为 0.49999 > 0.45，说明模型敏感度非常高。

为进一步的分析数据的类别，我们使用问题三中未风化点的数据，与问题二中聚类得到的模型进行结合，分析其所属的亚类。

经过与聚类类别之间的对比，我们得到如下的亚类区分结果：

数据项	表面风化	采样点	类型	亚类
A1	无风化	未风化点	高钾	似风化亚类
A2	风化	风化点（非严重风化）	铅钡	—
A3	无风化	未风化点	铅钡	似风化亚类
A4	无风化	未风化点	铅钡	似风化亚类
A5	风化	未风化点	铅钡	无风化亚类
A6	风化	风化点（非严重风化）	高钾	—
A7	风化	风化点（非严重风化）	高钾	—
A8	无风化	未风化点	铅钡	无风化亚类

我们发现，A5 中 SiO_2 含量的值为 64.29 > 27.056+3×9.005 $= \bar{X} + 3\sigma$，基本

排除 A5 属于风化点的可能性，因此该样本很有可能采样自风化样品的未风化点。

七、问题四模型的建立与求解

7.1 基于 Apriori 算法的古代玻璃化学成分关联关系分析

Apriori 算法是一个经典的关联规则挖掘算法。它不仅可以发现频繁项集，还可以挖掘项之间的关联规则。*Apriori* 算法使用支持度和置信度分别量化频繁项集和关联规则，本研究中我们使用 *Apriori* 算法进行各个化学成分的关联关系。[4]

Apriori 算法定义的置信度的公式为：

$$Confidence(A \rightarrow B) = \frac{Support(A,B)}{Support(A)} = P(B|A)$$

其中的$Support$是支持率，即数据库中某集合“出现”的概率，$Confidence$是置信率，即当集合*“A”*出现时，*“B”*也出现的概率。

为使用该算法，我们需要定义这里数据在数据集中“出现”的内涵（即如何将定量的数据转换为可以被算法利用的布尔值）：我们定义某一项数据在当前元组中“出现”等价于该项数据标准化后的值大于 0（这里使用的标准化方法仍然是*zscore*）

为筛选出具有较高关联性的化学成分，我们需要合理的调整最小支持率，以筛选出支持率较小的“关系”，由于数据量较大的关系，我们没有使用默认的*min_support*=0.2，而是提高到了 0.3，避免导致数据冗余或较难看出数据之间的关联，并将 *min_confidence*=0.65 使得得到的数据置信度都大于等于 65%（即 A，*B* 出现的概率均大于 20%，且当 *A* 出现时，*B* 出现的概率大于等于 65%）

按照化学成分全集的幂集元素的支持率与 *min_support*，我们可以计算出响应的频繁集（支持率大于 *min_support* 的所有集合），并在得到的频繁集中限制 *min_confidence*，从而得到相关联的数据报告。

经过数据报告我们可以得到关联度较高的化学成分的置信度（数据报告存放于：高钾化学元素关联分析.*xlsx*、铅钡化学元素关联分析.*xlsx*）。

高钾玻璃的置信度为：

前情(A)	后果(B)	置信度(Confidence)
氧化钙	氧化钾	0.875
氧化钾	氧化钙	0.875
氧化铁	氧化钙	0.714
氧化铁	氧化镁	0.857
氧化镁	氧化铁	0.857
氧化锶	氧化镁	1.000
氧化镁	氧化锶	0.714
氧化铁	氧化铜	0.714
氧化铜	氧化铁	0.833

铅钡玻璃的置信度为：

前情(A)	后果(B)	置信度(Confidence)
氧化锶	氧化镁	0.714
氧化镁	氧化锶	0.714
氧化铅	氧化锶	0.833
氧化锶	氧化铅	0.714

为了直观体现各个相关联概率较高的化学成分的具体相关系数，我们使用问题一中 Spearman 相关性分析的结果，生成相关系数热力图。

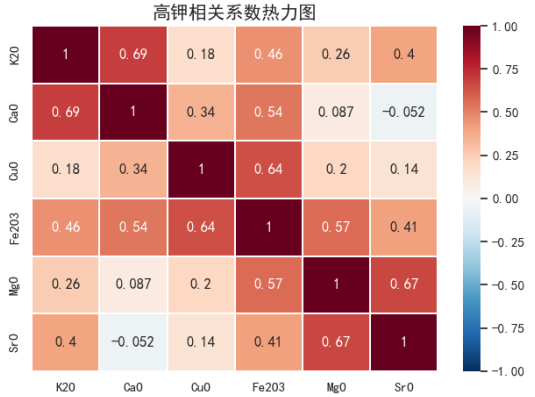


图 21 高钾玻璃相关系数热力图

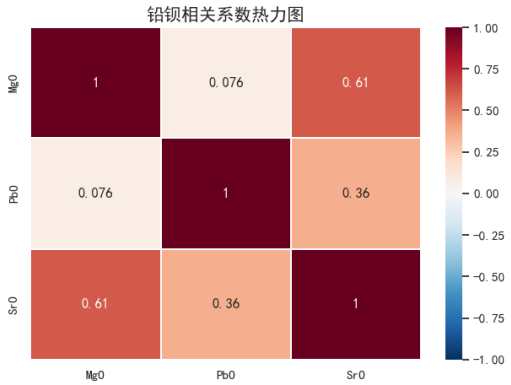


图 22 铅钡玻璃相关系数热力图

根据上述热力图与数据报告，对高钾玻璃中相关性较高的化学成分进行分析，

氧化锶与氧化镁具有较强的正相关关系，并且当氧化锶的含量大于平均值时，氧化镁的含量大于平均值的概率为 *1*，并且氧化锶与氧化钙有较强的负相关关系，氧化铁与氧化铜、氧化镁、氧化锶均具有较强的正相关关系。

铅钡玻璃中相关性较高的化学成分进行分析，氧化锶与氧化铅具有较强的正相关关系，并且当氧化铅的含量大于平均值时，氧化锶的含量大于平均值的概率为 *0.833*；反之，当氧化锶的含量大于平均值时，氧化铅的含量大于平均值的概率为 *0.714*。

下面分析高钾与铅钡玻璃化学成分关联关系的异同性。在高钾玻璃与铅钡玻璃中，氧化锶与其他众多元素均具有较强的关联性，但具有关联性的元素不完全一致，且具有明确线性相关关系的元素并不相同；并且，在高钾玻璃中，氧化镁会与部分元素具有明确线性相关关系，但在铅钡玻璃中，氧化镁与其他元素的相关关系并不显著。

八、模型的评价与推广

8.1 模型评价

模型优点：

1. 根据数据量及分布的正态性，选择差异性检验方法，方法使用准确；
2. 在 *SVC* 支持向量机分类中，对分类具有决定性因素的是少量的支持向量，而不是数据的维度，极大了降低了样本空间维数对时间效率的影响；
3. *SVC* 学习问题是一种凸优化问题，用该模型所得的解是全局最优解而不是局部最优解；
4. *Critic* 权重法更倾向于使用数值差别大的指标，在此处使用具有较高的合理性；
5. *K-means* 算法收敛速度快，算法可解释性较强；
6. *Apriori* 算法适合稀疏、数据量小的样本空间，算法原理简单，容易得到直观的数据。

模型缺点：

1. *SVC* 对大规模的样本数据难以实施训练，并且对核函数的选择较为敏感，本题采用的是线性核函数，但当样本数据线性不可分时，该模型会失效；
2. *Critic* 权重法只能使用标准化前的数据；
3. *Apriori* 算法对少量的数据就可能产生庞大的频繁集，时空复杂度较高。

8.2 模型推广

1. 可以进一步探究隐性风化的发生机制；

2. 本题中所选择的训练模型均适用于数据量偏小的样本空间，当数据量增加时，可以将获取化学成分权重的 *critic* 算法改为熵权法；
3. *Apriori* 算法可以用于获取 *lift* 值，其中：

$$lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{support(A)}$$

表示当 A 的某些属性提升时，B 会提升$lift(A \rightarrow B)$倍
（可以以此来探究当某一化学成分含量发生变化，对其他化学成分含量的影响）为风化程度的判断提供了有力证据。

高钾玻璃：

前情(A)	后果(B)	提升率(lift)
氧化钙	氧化钾	1.313
氧化钾	氧化钙	1.313
氧化铁	氧化钙	1.071
氧化铁	氧化镁	1.469
氧化镁	氧化铁	1.469
氧化锶	氧化镁	1.714
氧化镁	氧化锶	1.714
氧化铁	氧化铜	1.429
氧化铜	氧化铁	1.429

铅钡玻璃：

前情(A)	后果(B)	提升率(lift)
氧化锶	氧化镁	1.327
氧化镁	氧化锶	1.327
氧化铅	氧化锶	1.548
氧化锶	氧化铅	1.548

九、参考文献

[1]王婕,李沫,马清林,张治国,章梅芳,王菊琳.一件战国时期八棱柱状铅钡玻璃器的风化研究[J].玻璃与搪瓷,2014,42(02):6-13.DOI:10.13588/j.cnki.g.e.1000-2871.2014.02.002.

[2]王承遇,陶瑛.硅酸盐玻璃的风化[J].硅酸盐学报,2003(01):78-85.

[3]陈佳丽. 基于最小边际间隔的支持向量聚类方法研究[D].吉林大学,2022.

[4]浦慧忠.基于 Apriori 算法的关联规则挖掘在人工智能+个性化学习系统中的实践与研究[J].信息记录材料,2022,23(06):185-188.DOI:10.16009/j.cnki.cn13-1295/tq.2022.06.017.

十、附录

附录 A:问题一绘图程序

draw.py

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
import math

# 绘制10 号样本氧化钾含量估计值函数图像
sns.set()

a = np.arange(0, 2000, 0.01)
t = a
y = 9.851 - 0.1997 * np.sqrt(t)

plt.xlabel('时间 t/年')
plt.ylabel('百分比 ϕ/%')
plt.title("10 号样本氧化钾含量估计值", fontsize=15)
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
plt.plot(t, y)
plt.annotate("(0,9.851)", xy=(0, 9.851), xytext=(0, 9.851))
plt.annotate("(2000,0.92)", xy=(1980, 0.920), xytext=(1900, 1.2))
plt.annotate("(1000,3.5359)", xy=(1000, 3.536), xytext=(900, 3.8))
dx = [0, 1000, 2000]
dy = [9.851, 3.536, 0.920]
plt.scatter(dx, dy, c='r')

plt.show()

# 绘制纹饰-类型分类柱状图
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

sns.set()
# 构造需要显示的值
X = np.arange(0, 3, step=1)  # X 轴的坐标
Y = np.arange(0, 2, step=1)  # Y 轴的坐标
```

```
# 设置作图点的坐标
xpos, ypos = np.meshgrid(X[:-1] - 2.5, Y[:-1] - 2.5)
xpos = xpos.flatten('F')
ypos = ypos.flatten('F')
zpos = np.zeros_like(xpos)

# 设置每一个（X，Y）坐标所对应的Z 轴的值
Z = np.zeros(shape=(3, 2))
for i in range(3):
    Z[i, 0] = 6
    i += 1
Z[0, 1] = 0
Z[1, 1] = 16
Z[2, 1] = 24

xx, yy = np.meshgrid(X, Y)  # 网格化坐标
X, Y = xx.ravel(), yy.ravel()  # 矩阵扁平化
bottom = np.zeros_like(X)  # 设置柱状图的底端位值
Z = Z.ravel()  # 扁平化矩阵
width = height = 0.5  # 每一个柱子的长和宽

# 颜色设置
C = []
C.append('yellow')
C.append('purple')
C.append('limegreen')
C.append('skyblue')
C.append('red')
C.append('royalblue')
# 绘图设置
fig = plt.figure(figsize=(7, 7))
ax = fig.gca(projection='3d')  # 三维坐标轴
ax.bar3d(X, Y, bottom, width, height, Z, shade=True, color=C)
# 坐标轴设置
ax.set_zlabel('Z(数量)')

# 添加图例
ax.plot(0, 0, label='B/铅钡')
ax.plot(1, 0, label='B/高钾')
ax.plot(2, 0, label='C/高钾')
ax.plot(0, 1, label='A/高钾')
ax.plot(1, 1, label='A/铅钡')
ax.plot(2, 1, label='C/铅钡')
ax.legend(loc=1)
```

```
# 添加数据标签
ax.text(1.2, 1.2, 8, 6)
ax.text(0.2, 0.2, 8, 6)
ax.text(2.2, 0.2, 8, 6)
ax.text(1.2, 0.2, 2, 0)
ax.text(2.2, 1.2, 26, 24)
ax.text(0.2, 1.2, 18, 16)

ax.set_title('纹饰-类型分类柱状图', fontsize=15)
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

ax.set_xticks([])
ax.set_yticks([])

plt.show()
```

附录 B：支持向量机分类及其绘图

SVM.py

```
# 导入模块
import numpy as np
import pandas as pd
from pandas.plotting import parallel_coordinates
from sklearn import svm
from sklearn import preprocessing
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score as AUC
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
import seaborn as sns

# 设置后续的绘图风格
sns.set()
sns.despine()
sns.set_style("darkgrid")  # change the style to dark grid
sns.set_palette(["#39A7D0", "#36ADA4"])  # 自己选择颜色，可以输入 hex

# 加载中文字体
font = FontProperties(fname="./data/SimHei.ttf", size=14)
plt.rcParams["font.sans-serif"] = ["SimHei"]  # 设置字体
```

```
plt.rcParams["axes.unicode_minus"] = False  # 该语句解决图像中的“-”负号的乱码问题

# 导入数据
# 使用的数据是根据题目条件筛除后的数据
df = pd.read_excel("表单二.xlsx",
                   header=0,
                   index_col=0)

# 测试
# print(df)

# 进行数据的处理：X 表示化学变量，Y 表示类型变量
# 1 表示高钾，0 表示铅钡
X = df.iloc[:, 3:]
Y = df.iloc[:, 1]
# 测试
# print(X)
# print(Y)

# SVC 支持向量分类方法的模型及其参数
clf = svm.SVC(C=1,
              kernel='linear',
              max_iter=1000,
              degree=3,
              tol=0.001,
              gamma='scale',
              decision_function_shape='ovr',
              probability=True)
clf.fit(X, Y)

# 导入待预测的数据
# 表面风化:1
# 无风化:0
ans = pd.read_excel('附件（对拍）.xlsx',
                    header=0,
                    index_col=0,
                    sheet_name='表单 3')
ans.fillna(0)
pre = ans.iloc[:, :]
# 测试
# print(ans)
# print(pre)
```

```
# 数据的预测
# 1 表示高钾、0 表示铅钡
y_pre = clf.predict(pre)
y_score = clf.predict_proba(pre)
print('每个待预测样本在两个类别中的预测得分: ',
      y_score, sep='\n')
print('预测结果: ', y_pre)

# 模型评价
print('数据样本综合得分:', clf.score(X, Y))

# 绘制 ROC 曲线
FPR, recall, thresholds = roc_curve(Y, clf.decision_function(X), pos_
label=1)
# print(FPR, recall, thresholds)

# 我们这里的阈值之所以会超过 0-1 的范畴是因为以每个点到决策边界的距离为计算标
准的
area = AUC(Y, clf.decision_function(X))
# print(area)

recall_FPR_gap = (recall - FPR).tolist()
max_gap = max(recall_FPR_gap)
# print(max_gap)

max_index = recall_FPR_gap.index(max_gap)
# print(max_index)

# print(thresholds[max_index])

recall_FPR_gap = (recall - FPR).tolist()
max_gap = max(recall_FPR_gap)
# print(max_gap)  #0.914

max_index = recall_FPR_gap.index(max_gap)
# print(max_index)

# print(thresholds[max_index])

plt.figure()
plt.scatter(FPR[max_index], recall[max_index], c="black", s=30)
plt.plot(FPR, recall, color="red", label="ROC")
plt.plot([0, 1], [0, 1], color="black", linestyle="--")
plt.xlim([-0.05, 1.05])
```

```
plt.ylim([-0.05, 1.05])
plt.xlabel("False Positive Rate")
plt.ylabel("Recall")
plt.title("Receiver operating characteristic 曲线")
plt.legend(loc="lower right")
plt.savefig('SVC 模型的 ROC 曲线.png')
plt.show()

# 数据处理
X['类型'] = Y

# 测试
# print(X.columns)

# 进行数据的标准化，方便绘图
zscore = preprocessing.StandardScaler()
z_score = zscore.fit_transform(X)
X = pd.DataFrame(z_score, columns=X.columns, index=X.index)
# print(X)

# 多维可视化
plt.figure(figsize=(20, 10))
parallel_coordinates(X, alpha=0.8,
                     class_column='类型',
                     color=['r',
                            'limegreen'])
plt.xticks(size=20, rotation=30)
plt.yticks(size=20)
plt.legend(prop={"size": 20})
plt.title("类别与化学成分之间的关系",
          fontsize=35)
plt.legend(['高钾', '铅钡'],
           title='图例',
           loc=0,
           fancybox=True,
           shadow=True,
           fontsize=20,
           bbox_to_anchor=(1, 1))
# 设置紧凑型布局
plt.tight_layout()

# 图片保存
plt.savefig('第二题（1）图.png')
plt.show()
```

```
print('Done!')
```

附录 C：k-means 聚类分析及其绘图

k-means.py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
import seaborn as sns

# 设置后续的绘图风格
sns.set()
sns.despine()
sns.set_style("darkgrid")  # change the style to dark grid
sns.set_palette(["#39A7D0", "#36ADA4"])  # 自己选择颜色，可以输入 hex

# 加载中文字体
font = FontProperties(fname="./data/SimHei.ttf", size=14)
plt.rcParams["font.sans-serif"] = ["SimHei"]  # 设置字体
plt.rcParams["axes.unicode_minus"] = False  # 该语句解决图像中的“-”负号的乱码问题

dfK = pd.read_csv('高钾数据集聚类标注.csv')
dfPb = pd.read_csv('铅钡数据集聚类标注.csv')
cenK = pd.read_csv('高钾聚类中心点坐标.csv')
cenPb = pd.read_csv('铅钡聚类中心点坐标.csv')

# 取出两次聚类中的各类数据
df1 = dfK[dfK['聚类种类'] == 1]
df2 = dfK[dfK['聚类种类'] == 2]
df3 = dfPb[dfPb['聚类种类'] == 1]
df4 = dfPb[dfPb['聚类种类'] == 2]

# 测试
print(dfK)
print(dfPb)
# print(df1)
# print(df2)
# print(df3)
# print(df4)
```

```
# 开始绘图
# 高钾聚类绘图
ax = plt.subplot(projection='3d')  # 创建一个三维的绘图工程
ax.set_title('高钾亚类区分图',
             fontsize=20)
x = np.array(df1['氧化钾(K2O)'])
y = np.array(df1['氧化钙(CaO)'])
z = np.array(df1['二氧化硅(SiO2)'])
ax.scatter(x, y, z, c='r')  # 绘制数据点 c: 'r'红色，'y'黄色，等颜色
x = np.array(df2['氧化钾(K2O)'])
y = np.array(df2['氧化钙(CaO)'])
z = np.array(df2['二氧化硅(SiO2)'])
ax.scatter(x, y, z, c='darkblue')  # 绘制数据点 c: 'r'红色，'y'黄色，等颜色
# x = np.array(cenK['中心值_氧化钾(K2O)'])
# y = np.array(cenK['中心值_氧化钙(CaO)'])
# z = np.array(cenK['中心值_二氧化硅(SiO2)'])
# ax.scatter(x, y, z, c='blue')
ax.legend(['无风化亚类',
           '似风化亚类'],
          title='图例',
          loc=0,
          fancybox=True,
          shadow=True,
          fontsize=10)
ax.set_xlabel('氧化钾(K2O)')  # 设置x坐标轴
ax.set_ylabel('氧化钙(CaO)')  # 设置y坐标轴
ax.set_zlabel('二氧化硅(SiO2)')  # 设置z坐标轴
ax.view_init(10, 70, 'z')
# 设置紧凑型布局
plt.tight_layout()
plt.savefig('高钾亚类区分图')
plt.show()

print(df3)

# 铅钡聚类绘图
plt.title('铅钡亚类区分图',
          fontsize=20)
x = np.array(df3['氧化铅(PbO)'])
y = np.array(df3['二氧化硅(SiO2)'])
plt.scatter(x, y, c='r')
x = np.array(df4['氧化铅(PbO)'])
```

```
y = np.array(df4['二氧化硅(SiO2)'])
plt.scatter(x, y, c='darkblue')
# x = np.array(cenPb['中心值_氧化铅(PbO)'])
# y = np.array(cenPb['中心值_二氧化硅(SiO2)'])
# plt.scatter(x, y, c='blue')
plt.legend(['无风化亚类',
            '似风化亚类'],
           title='图例',
           loc=0,
           fancybox=True,
           shadow=True,
           fontsize=10)
plt.xlabel('氧化铅(PbO)')
plt.ylabel('二氧化硅(SiO2)')
# 设置紧凑型布局
plt.tight_layout()
plt.savefig('铅钡亚类区分图')
plt.show()

print('Done!')
```

附录 D：apriori 算法

correlation.py

```
# 导入库
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from sklearn import preprocessing

# 在这里，我们使用的是，apriori 算法进行关联度分析
# 我们定义：某一项数据在当前元组中出现等价于该项数据标准化后的值大于0
# 在这里，我们仍然使用z_score 进行数据的标准化

dfK = pd.read_excel('表单二.xlsx', sheet_name='高钾')
dfPb = pd.read_excel('表单二.xlsx', sheet_name='铅钡')
dfK = dfK[dfK['表面风化'] == 0]
dfPb = dfPb[dfPb['表面风化'] == 0]
dfK.drop('表面风化', axis=1, inplace=True)  # axis 参数默认为0
dfPb.drop('表面风化', axis=1, inplace=True)  # axis 参数默认为0

# 测试
```

```
# print(dfK)
# print(dfPb)

# 高钾
df = dfK.iloc[:, 3:]
# 测试
# print(df)
# 数据标准化
zscore = preprocessing.StandardScaler()
z_score = zscore.fit_transform(df)
df = pd.DataFrame(z_score, columns=df.columns, index=df.index)

# 根据定义将数据转换为bool 值
df.loc[df['二氧化硅(SiO2)'] >= 0, '二氧化硅(SiO2)'] = True
df.loc[df['二氧化硅(SiO2)'] < 0, '二氧化硅(SiO2)'] = False
df.loc[df['氧化钠(Na2O)'] >= 0, '氧化钠(Na2O)'] = True
df.loc[df['氧化钠(Na2O)'] < 0, '氧化钠(Na2O)'] = False
df.loc[df['氧化钾(K2O)'] >= 0, '氧化钾(K2O)'] = True
df.loc[df['氧化钾(K2O)'] < 0, '氧化钾(K2O)'] = False
df.loc[df['氧化钙(CaO)'] >= 0, '氧化钙(CaO)'] = True
df.loc[df['氧化钙(CaO)'] < 0, '氧化钙(CaO)'] = False
df.loc[df['氧化镁(MgO)'] >= 0, '氧化镁(MgO)'] = True
df.loc[df['氧化镁(MgO)'] < 0, '氧化镁(MgO)'] = False
df.loc[df['氧化铝(Al2O3)'] >= 0, '氧化铝(Al2O3)'] = True
df.loc[df['氧化铝(Al2O3)'] < 0, '氧化铝(Al2O3)'] = False
df.loc[df['氧化铁(Fe2O3)'] >= 0, '氧化铁(Fe2O3)'] = True
df.loc[df['氧化铁(Fe2O3)'] < 0, '氧化铁(Fe2O3)'] = False
df.loc[df['氧化铜(CuO)'] >= 0, '氧化铜(CuO)'] = True
df.loc[df['氧化铜(CuO)'] < 0, '氧化铜(CuO)'] = False
df.loc[df['氧化铅(PbO)'] >= 0, '氧化铅(PbO)'] = True
df.loc[df['氧化铅(PbO)'] < 0, '氧化铅(PbO)'] = False
df.loc[df['氧化钡(BaO)'] >= 0, '氧化钡(BaO)'] = True
df.loc[df['氧化钡(BaO)'] < 0, '氧化钡(BaO)'] = False
df.loc[df['五氧化二磷(P2O5)'] >= 0, '五氧化二磷(P2O5)'] = True
df.loc[df['五氧化二磷(P2O5)'] < 0, '五氧化二磷(P2O5)'] = False
df.loc[df['氧化锶(SrO)'] >= 0, '氧化锶(SrO)'] = True
df.loc[df['氧化锶(SrO)'] < 0, '氧化锶(SrO)'] = False
df.loc[df['氧化锡(SnO2)'] >= 0, '氧化锡(SnO2)'] = True
df.loc[df['氧化锡(SnO2)'] < 0, '氧化锡(SnO2)'] = False
df.loc[df['二氧化硫(SO2)'] >= 0, '二氧化硫(SO2)'] = True
df.loc[df['二氧化硫(SO2)'] < 0, '二氧化硫(SO2)'] = False
# 测试
# print(df)
```

```
freq = apriori(df, min_support=0.35, use_colnames=True)
# 保存数据
freq.to_excel('高钾化学元素频繁集.xlsx')
# print(freq)
# 由于数据原因，最小支持度设置成0.3

result = association_rules(freq, metric='confidence', min_threshold=0
.65)
# 保存数据
result.to_excel('高钾化学元素关联分析.xlsx')
# print(result)

# 铅钡
df = dfPb.iloc[:, 3:]
# 测试
# print(df)
# 数据标准化
zscore = preprocessing.StandardScaler()
z_score = zscore.fit_transform(df)
df = pd.DataFrame(z_score, columns=df.columns, index=df.index)

# 根据定义将数据转换为bool 值
df.loc[df['二氧化硅(SiO2)'] >= 0, '二氧化硅(SiO2)'] = True
df.loc[df['二氧化硅(SiO2)'] < 0, '二氧化硅(SiO2)'] = False
df.loc[df['氧化钠(Na2O)'] >= 0, '氧化钠(Na2O)'] = True
df.loc[df['氧化钠(Na2O)'] < 0, '氧化钠(Na2O)'] = False
df.loc[df['氧化钾(K2O)'] >= 0, '氧化钾(K2O)'] = True
df.loc[df['氧化钾(K2O)'] < 0, '氧化钾(K2O)'] = False
df.loc[df['氧化钙(CaO)'] >= 0, '氧化钙(CaO)'] = True
df.loc[df['氧化钙(CaO)'] < 0, '氧化钙(CaO)'] = False
df.loc[df['氧化镁(MgO)'] >= 0, '氧化镁(MgO)'] = True
df.loc[df['氧化镁(MgO)'] < 0, '氧化镁(MgO)'] = False
df.loc[df['氧化铝(Al2O3)'] >= 0, '氧化铝(Al2O3)'] = True
df.loc[df['氧化铝(Al2O3)'] < 0, '氧化铝(Al2O3)'] = False
df.loc[df['氧化铁(Fe2O3)'] >= 0, '氧化铁(Fe2O3)'] = True
df.loc[df['氧化铁(Fe2O3)'] < 0, '氧化铁(Fe2O3)'] = False
df.loc[df['氧化铜(CuO)'] >= 0, '氧化铜(CuO)'] = True
df.loc[df['氧化铜(CuO)'] < 0, '氧化铜(CuO)'] = False
df.loc[df['氧化铅(PbO)'] >= 0, '氧化铅(PbO)'] = True
df.loc[df['氧化铅(PbO)'] < 0, '氧化铅(PbO)'] = False
df.loc[df['氧化钡(BaO)'] >= 0, '氧化钡(BaO)'] = True
df.loc[df['氧化钡(BaO)'] < 0, '氧化钡(BaO)'] = False
df.loc[df['五氧化二磷(P2O5)'] >= 0, '五氧化二磷(P2O5)'] = True
df.loc[df['五氧化二磷(P2O5)'] < 0, '五氧化二磷(P2O5)'] = False
```

```
df.loc[df['氧化锶(SrO)'] >= 0, '氧化锶(SrO)'] = True
df.loc[df['氧化锶(SrO)'] < 0, '氧化锶(SrO)'] = False
df.loc[df['氧化锡(SnO2)'] >= 0, '氧化锡(SnO2)'] = True
df.loc[df['氧化锡(SnO2)'] < 0, '氧化锡(SnO2)'] = False
df.loc[df['二氧化硫(SO2)'] >= 0, '二氧化硫(SO2)'] = True
df.loc[df['二氧化硫(SO2)'] < 0, '二氧化硫(SO2)'] = False
# 测试
# print(df)

freq = apriori(df, min_support=0.35, use_colnames=True)
# 保存数据
freq.to_excel('铅钡化学元素频繁集.xlsx')
# print(freq)
# 由于数据原因，最小支持度设置成0.3

result = association_rules(freq, metric='confidence', min_threshold=0
.65)
# 保存数据
result.to_excel('铅钡化学元素关联分析.xlsx')
# print(result)

print('Done!')
```

附录 E：关联矩阵热力图绘制

heatmap.py

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties

# 设置后续的绘图风格
sns.set()

# 加载中文字体
font = FontProperties(fname="./data/SimHei.ttf", size=14)
plt.rcParams["font.sans-serif"] = ["SimHei"]  # 设置字体
plt.rcParams["axes.unicode_minus"] = False  # 该语句解决图像中的“-”负号
的乱码问题

# 导入数据
conK = pd.read_excel('相关性.xlsx',
```

```
                        header=0,
                        index_col=0,
                        sheet_name='Sheet2')
conPb = pd.read_excel('相关性.xlsx',
                         header=0,
                         index_col=0,
                         sheet_name='Sheet3')

# print(conK)
# print(conPb)

# 绘制热力图
# 高钾
sns.heatmap(conK,
            linewidths=0.05,
            cmap="RdBu_r",
            annot=True,
            vmax=1, vmin=-1)
plt.title('高钾相关系数热力图',
          fontsize=15)
# 设置紧凑型布局
plt.tight_layout()
plt.savefig('高钾相关系数热力图.png')
plt.show()

# 铅钡
sns.heatmap(conPb,
            linewidths=0.05,
            cmap="RdBu_r",
            annot=True,
            vmax=1, vmin=-1)
plt.title('铅钡相关系数热力图',
          fontsize=15)
# 设置紧凑型布局
plt.tight_layout()
plt.savefig('铅钡相关系数热力图.png')
plt.show()

print('Done!')
```