**注：**字典学习也是一种数据降维的方法，这里我用到SVD的知识，对SVD不太理解的地方，可以看看这篇博客：[《SVD（奇异值分解）小结 》](https://www.cnblogs.com/endlesscoding/p/10033527.html)。

**1、字典学习思想**

字典学习的思想应该源来实际生活中的字典的概念。字典是前辈们学习总结的精华，当我们需要学习新的知识的时候，不必与先辈们一样去学习先辈们所有学习过的知识，我们可以参考先辈们给我们总结的字典，通过查阅这些字典，我们可以大致学会到这些知识。

为了将上述过程用准确的数学语言描述出来，我们需要将“总结字典”、“查阅字典”做出一个更为准确的描述。就从我们的常识出发：

1. 我们通常会要求的我们的字典尽可能全面，也就是说总结出的字典不能漏下关键的知识点。
2. 查字典的时候，我们想要我们查字典的过程尽可能简洁，迅速，准确。即，查字典要快、准、狠。
3. 查到的结果，要尽可能地还原出原来知识。当然，如果要完全还原出来，那么这个字典和查字典的方法会变得非常复杂，所以我们只需要尽可能地还原出原知识点即可。

**注：** 以上内容，完全是自己的理解，如有不当之处，欢迎各位拍砖。

下面，我们要讨论的就是如何将上述问题抽象成一个数学问题，并解决这个问题。

**2、字典学习数学模型**

**2.1 数学描述**

我们将上面的所提到的关键点用几个数学符号表示一下：

* “以前的知识”，更专业一点，我们称之为**原始样本**，用矩阵**Y**

 表示；

 “字典”，我们称之为**字典矩阵**，用**D**表示，“字典”中的词条，我们称之为**原子（atom）**，用列向量**d***k*

 表示；

 “查字典的方法”，我们称为**稀疏矩阵**，用**X**

 ；

 “查字典的过程”，我们可以用矩阵的乘法来表示，即**DX**

* 。

用数学语言描述，字典学习的主要思想是，利用包含*K*

个原子**d***k*的字典矩阵**D**∈**R***m*×*K*，稀疏线性表示原始样本**Y**∈**R***m*×*n*（其中*m*表示样本数，*n*表示样本的属性），即有**Y=DX**（这只是我们理想的情况），其中**X**∈**R***K*×*n*

为**稀疏矩阵**，可以将上述问题用数学语言描述为如下优化问题

min**D, X**∥**Y**−**DX**∥2*F*,s.t. ∀*i*, ∥**x***i*∥0≤*T*0(2-1)

或者

min**D, X**∑*i*∥**x***i*∥0,s.t. min**D, X**∥**Y**−**DX**∥2*F*≤*ϵ*,(2-2)

上式中**X**

为稀疏编码的矩阵，**x***i* (*i*=1,2,⋯,*K*)

为该矩阵中的行向量，代表字典矩阵的系数。

**注：** ∥**x***i*∥0

表示零阶范数，它表示向量中不为0的数的个数。

**2.2 求解问题**

式（2-1）的目标函数表示，我们要最小化查完的字典与原始样本的误差，即要尽可能还原出原始样本；它的限的制条件∥**x***i*∥0≤*T*0

，表示查字典的方式要尽可能简单，即**X**

要尽可能稀疏。式（2-2）同理。

式（2-1）或式（2-2）是一个带有约束的优化问题，可以利用拉格朗日乘子法将其转化为无约束优化问题

min**D, X**∥**Y**−**DX**∥2*F*+*λ*∥**x***i*∥1(2-3)

**注：** 我们将∥**x***i*∥0

用∥**x***i*∥1代替，主要是∥**x***i*∥1

更加便于求解。

这里有两个优化变量**D, X**

，为解决这个优化问题，一般是固定其中一个优化变量，优化另一个变量，如此交替进行。式（2-3）中的稀疏矩阵**X**可以利用已有经典算法求解，如Lasso（Least Absolute Shrinkage and Selection Operator）、OMP（Orthogonal Matching Pursuit），这里我重点讲述如何更新字典**D**，对更新**X**

不多做讨论。

假设**X**

是已知的，我们逐列更新字典。下面我们仅更新字典的第*k*列，记**d***k*为字典**D**的第*k*列向量，记**x***kT*为稀疏矩阵**X**的第*k*

行向量，那么对式（2-1），我们有

∥**Y**−**DX**∥2*F*===∥∥∥∥**Y**−∑*j*=1*K***d***j***x***jT*∥∥∥∥2*F*∥∥∥∥(**Y**−∑*j*≠*k***d***j***x***jT*)−**d***k***x***kT*∥∥∥∥2*F*∥∥**E***k*−**d***k***x***kT*∥∥2*F*(2-4)

上式中残差**E***k*=**Y**−∑*j*≠*k***d***j***x***jT*

，

此时优化问题可描述为

min**d***k*, **x***kT*∥∥**E***k*−**d***k***x***kT*∥∥2*F*

因此我们需要求出最优的**d***k*, **x***kT*

，这是一个最小二乘问题，可以利用最小二乘的方法求解，或者可以利用[SVD进行](https://www.cnblogs.com/endlesscoding/p/10033527.html)求解，这里利用SVD的方式求解出两个优化变量。

但是，在这里我人需要注意的是，不能直接利用**E***k*

进行求解，否则求得的新的**x***Tk*不稀疏。因此我们需要将**E***k*中对应的**x***kT*不为0的位置提取出来，得到新的**E**′*k*

，这个过程如图2-1所示，这样描述更加清晰。

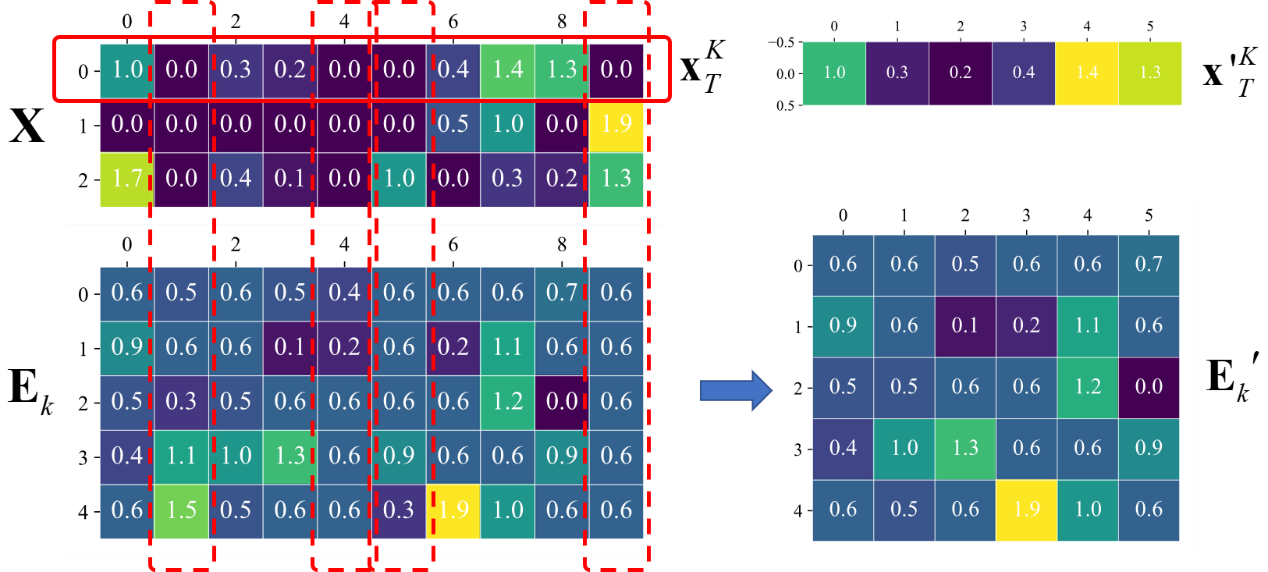


图2-1 提取部分残差

如上图，假设我们要更新第0列原子，我们将**x***kT*

中为零的位置找出来，然后把**E***k*对应的位置删除，得到**E**′*k*

，此时优化问题可描述为

min**d***k*, **x***kT*∥∥**E**′*k*−**d***k***x**′*kT*∥∥2*F*(2-5)

因此我们需要求出最优的**d***k*, **x**′*kT*

**E**′*k*=*U*Σ*VT*(2-6)

取左奇异矩阵*U*

的第1个列向量**u**1=*U*(⋅,1)作为**d***k*，即**d***k*=**u**1，取右奇异矩阵的第1个行向量与第1个奇异值的乘积作为**x**′*kT*，即**x**′*kT*=Σ(1,1)*VT*(1,⋅)。得到**x**′*kT*后，将其对应地更新到原**x***kT*

。

**注：** 式（2-6）所求得的奇异值矩阵Σ

中的奇异值应从大到小排列；同样也有**x**′*kT*=Σ(1,1)*V*(⋅,1)*T*，这与上面**x**′*kT*

的求法是相等的。

**2.3 字典学习算法实现**

据2.2小节，利用稀疏算法求解得到稀疏矩阵**X**

后，逐列更新字典，有如下算法1.1。

**算法1.1：字典学习（K-SVD）**

**输入：**原始样本，字典，稀疏矩阵  
**输出：**字典，稀疏矩阵

1. **初始化：** 从原始样本*Y*∈**R***m*×*n*

随机取*K*个列向量或者取它的左奇异矩阵的前*K*个列向量{**d**1,**d**2,⋯,**d***K*}作为初始字典的原子，得到字典**D**(0)∈**R***m*×*K*。令*j*=0

 ，重复下面**步骤2-3**，直到达到指定的迭代步数，或收敛到指定的误差：

 **稀疏编码：** 利用字典上一步得到的字典**D**(*j*)

，稀疏编码，得到**X**(*j*)∈**R***K*×*n*

 。

 **字典更新：** 逐列更新字典**D**(*j*)，字典的列**d***k*∈{**d**1,**d**2,⋯,**d***K*}

* 当更新**d***k*

时，计算误差矩阵**E***k*

**E***k*=**Y**−∑*j*≠*k***d***j***x***jT*.

  取出稀疏矩阵第*k*

个行向量**x***kT*不为0的索引的集合*ωk*={*i*|1≤*i*≤*n*, **x***kT*(*i*)≠0}，**x**′*kT*={**x***kT*(*i*)|1≤*i*≤*n*, **x***kT*(*i*)≠0}

  从**E***k*

取出对应*ωk*不为0的列，得到**E**′*k*

 .

 对**E**′*k*作奇异值分解**E***k*=*U*Σ*VT*，取*U*的第1列更新字典的第*k*列，即**d***k*=*U*(⋅,1)；令**x**′*kT*=Σ(1,1)*V*(⋅,1)*T*，得到**x**′*kT*后，将其对应地更新到原**x***kT*

 。

 *j*=*j*+1