SVD

目录

摘要

这是一个关于，菜鸟小白的故事。小白作为 In+lab 的一个新人，初来乍到，刚刚结束大学本科浑浑噩噩的四年，正准备在研究生生涯励精图治，却被实验室各种机器学习方法折磨的死去活来，唯有长叹：壮志未酬身先死！

这日，小白又徜徉在机器学习的海洋里度日如年。忽然听见，旁边的师兄弟们正在讨论一个叫做 SVD 的机器学习方法。小白隐隐约约的听到一些，感觉非常感兴趣。但是，小白毕竟是小白，小白的世界就是白茫茫的一片。于是，遵循有事问百度的原则，小白毫不犹豫地问了度娘。度娘百科里面给出的答案，顿时让小白精神抖擞，SVD 那不正是穿越火线里面大名鼎鼎的 SVD 狙击步枪么。小白默默点开了 CF（话说此处 CF 不是协同过滤，而是 cross fire）……

言归正传，各位看客老爷们毕竟不是小白，大家伙可都是五颗星红星下的所谓的栋梁之才。稍稍用点手段，越过防火墙，google 一下 SVD，你就会发现 google 和百度的答案是完全迥异的两个世界。跟随 google 的步伐，让我们一起走进 SVD 的世界。这里我将为大家介绍两种 SVD：在矩阵分解的基础上演化的推荐 SVD 和线性代数 SVD 及其应用。

一、推荐 SVD

1)矩阵分解

总得来说推荐 SVD 就是以矩阵分解为基础的，将一个矩阵分解为

2 个矩阵。为了使讲解不枯燥，这里我直接从实例入手，如果有看不懂的可以从后面查阅。

场景如下图，我们已知一些用户的对电影的评分，然后是否能预测 U1 对《回魂夜》这部电影的评分（评分划定 5 个等级）？

用户-电影	东成西就	死神归来	回魂夜
U1	5	1	？
U2	1	4	4
U3	5	2	2

用户-电影评分表(1)

答案是肯定的。SVD(Singular Value Decomposition)的想法是根据已有的评分情况，分析出评分者对各个因子的喜好程度以及电影包含各个因子的程度，最后再反过来根据分析结果预测评分。电影中的因子可以理解成这些东西：电影的搞笑程度，电影的爱情爱得死去活来的程度，电影的恐怖程度。。。。。。如下图（这里列出俩个电影影响因子，实际可能有非常多）：

电影-电影因子	搞笑	恐怖
东成西就	5	0
死神归来	0	4
回魂夜	4	3

电影-电影因子评分表(2)

同理，用户对那些影响因子也都有自己的偏好，有人就喜欢搞笑片，有人就喜欢恐怖片点，如图（根据已知的用户-电影评分表，我

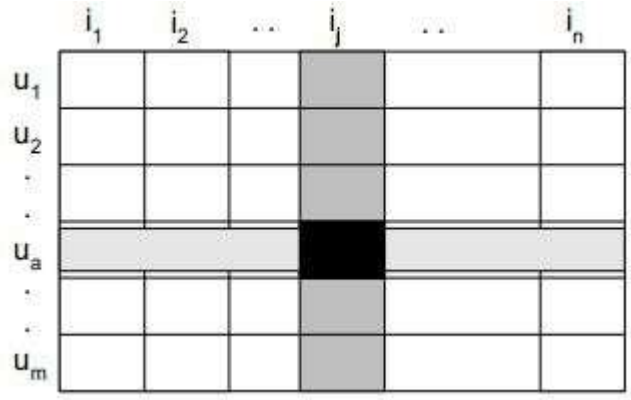
很明显的知道 U1 和 U3 可能更加喜欢喜剧，U2 偏好恐怖片点）：

用户-电影因子	搞笑	恐怖
U1	4	1
U2	1	4
U3	5	2

用户-用户因子评分表(3)

事实上，以上三个表格，除了第一个用户-电影评分是现实已知的矩阵，其余两个是我们人为分析的结果。那么电脑是怎么得到那两个分析后的矩阵的呢？

电脑是通过矩阵分解得到的。如下：



图表 1 评分矩阵形式（引用）

评分矩阵 U（形式如上图）可被分解为两个矩阵相乘

$$U = \begin{bmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mk} \end{bmatrix} \begin{bmatrix} i_{11} & \cdots & i_{1n} \\ \vdots & \ddots & \vdots \\ i_{k1} & \cdots & i_{kn} \end{bmatrix}$$

将这种分解方式体现协同过滤中，即有：

$$\hat{r}_{ui} = q_i^T p_u \quad \text{(matrix factorization model)}$$

在这样的分解模型中,Pu 代表用户隐因子矩阵（表示用户 u 对因子 k 的喜好程度),Qi 表示电影隐因子矩阵（表示电影 i 在因子 k 上的程度）。其实到这一步为止是推荐系统比较有名的矩阵分解预测模型。

2）Baseline Predictors

实际上，我们给一部电影评分时，除了考虑电影是否合自己口味外，还会受到自己是否是一个严格的评分者和这部电影已有的评分状况影响。例如：一个严格评分者给的分大多数情况下都比一个宽松评分者的低。你看到这部电影的评分大部分较高时，可能也倾向于给较高的分。在 SVD 中，口味问题已经有因子来表示了，但是剩下两个还没有相关的式子表示。因此有必要加上相关的部分，提高模型的精准度。引入 Baseline Predictors 使用向量 bi 表示电影 i 的评分相对于平均评分的偏差，向量 bu 表示用户 u 做出的评分相对于平均评分的偏差，将平均评分记做 μ 。

$$\hat{r}_{ui} = \mu + bi + bu \quad \text{(Baseline Predictors)}$$

所以，SVD 就是一种加入 Baseline Predictors 优化的 matrix factorization model。

SVD 公式如下：

$$\hat{r}_{ui} = \mu + bi + bu + q_i^T p_u$$

加入防止过拟合的 λ 参数，可以得到下面的优化函数（主体就是真实减去预测值，就是控制这个值最小）：

$$\min \sum (r_{ui} - \mu + bi + bu + q_i{}^T p_u)^2 + \lambda(b_i{}^2 + b_u{}^2 + \|q_i\|^2 + \|p_u\|^2)$$

对上述公式求导，我们可以得到最终的求解函数:

$$\begin{aligned} e_{ui} &= r_{ui} - \hat{r}_{ui} \\ b_u &\leftarrow b_u + \gamma \cdot (e_{ui} - \lambda \cdot b_u) \\ b_i &\leftarrow b_i + \gamma \cdot (e_{ui} - \lambda \cdot b_i) \\ p_u &\leftarrow p_u + \gamma \cdot (e_{ui} - \lambda \cdot p_u) \\ q_i &\leftarrow q_i + \gamma \cdot (e_{ui} - \lambda \cdot q_i) \end{aligned}$$

然后，根据随机梯度下降法（随机梯度下降法可查阅实验室其他博文），需要将参数沿着最速下降方向向前推进，最终得到预测值。

二、线性代数 SVD 背景介绍

SVD（Singular Value Decomposition）奇异值分解[1]实际上是线性代数的内容，是一种数学常用的矩阵分解技术，但是这个方法已经应用的多个领域之中，本文主要介绍的是机器学习中的 SVD。这里主要先介绍一下数学背景知识，以及数学 SVD 的推导。

1）线性代数基础

矩阵的秩[2]：是矩阵中线性无关的行或列的个数。

对角矩阵：对角矩阵是除对角线外所有元素都为零的方阵。

单位矩阵：如果对角矩阵中所有对角线上的元素都为零，该矩阵称为单位矩阵

2）特征值

在介绍奇异值分解之前，先介绍较为特殊的特征值分解[4]。A 是 n 阶方阵，如果存在 λ 和 n 维非零向量 x。如果说一个向量 x 是方阵 A 的特征向量，将一定可以表示成下面的形式：

$$Ax = \lambda x$$

这时候 λ 就被称为特征向量 x 对应的特征值，一个矩阵的一组特征向量是一组正交向量。特征值分解就是将一个矩阵分解成下面的形式：

设 A 有 n 个特征值及特征向量[5]，则：

$$A * x_1 = \lambda_1 * x_1$$
$$A * x_2 = \lambda_2 * x_2$$
$$\cdots$$
$$A * x_n = \lambda_n * x_n$$

将上面的写到一起成矩阵形式：

$$A * \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} * \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \cdots\cdots & & & \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

若(x1,x2,...,xn)可逆，则左右两边都求逆，则方阵 A 可直接通过特征值和特征向量进行唯一的表示，令

Q=(x1,x2,...,xn)

$\Sigma = \mathrm{diag}(\lambda 1, \lambda 2, ..., \lambda n)$

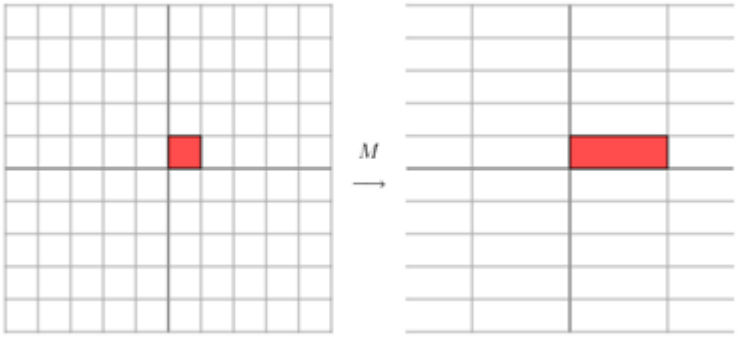
$A = Q\Sigma Q^{-1}$ 该表达式称为方阵的特征值分解，这样方阵 A 就被特征值和特征向量唯一表示。

其中 Q 是这个矩阵 A 的特征向量组成的矩阵，Σ是一个对角阵，

每一个对角线上的元素就是一个特征值。我这里引用了一些参考文献[3]中的内容来说明一下。首先，要明确的是，一个矩阵其实就是一个线性变换，因为一个矩阵乘以一个向量后得到的向量，其实就相当于将这个向量进行了线性变换。比如说下面的一个矩阵：

$$M = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$$

它其实对应的线性变换是下面的形式：



因为这个矩阵 M 乘以一个向量(x,y)的结果是：

$$\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3x \\ y \end{bmatrix}$$

上面的矩阵是对称的，所以这个变换是一个对 x，y 轴的方向一个拉伸变换（每一个对角线上的元素将会对一个维度进行拉伸变换，当值>1 时，是拉长，当值<1 时时缩短），当矩阵不是对称的时候，假如说矩阵是下面的样子：

$$M = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

它所描述的变换是下面的样子：

这其实是在平面上对一个轴进行的拉伸变换（如蓝色的箭头所示），在图中，蓝色的箭头是一个最主要的变化方向（变化方向可能

有不止一个），如果我们想要描述好一个变换，那我们就描述好这个变换主要的变化方向就好了。反过头来看看之前特征值分解的式子，分解得到的 Σ 矩阵是一个对角阵，里面的特征值是由大到小排列的，这些特征值所对应的特征向量就是描述这个矩阵变化方向（从主要的变化到次要的变化排列）

当矩阵是高维的情况下，那么这个矩阵就是高维空间下的一个线性变换，这个线性变化可能没法通过图片来表示，但是可以想象，这个变换也同样有很多的变换方向，我们通过特征值分解得到的前 N 个特征向量，那么就对应了这个矩阵最主要的 N 个变化方向。我们利用这前 N 个变化方向，就可以近似这个矩阵（变换）。也就是之前说的：提取这个矩阵最重要的特征。总结一下，特征值分解可以得到特征值与特征向量，特征值表示的是这个特征到底有多重要，而特征向量表示这个特征是什么，可以将每一个特征向量理解为一个线性的子空间，我们可以利用这些线性的子空间干很多的事情。不过，特征值分解也有很多的局限，比如说变换的矩阵必须是方阵。

3）奇异值分解

特征值分解是一个提取矩阵特征很不错的方法，但是它只是对方阵而言的，在现实的世界中，我们看到的大部分矩阵都不是方阵，比如说有 N 个学生，每个学生有 M 科成绩，这样形成的一个 N * M 的矩阵就不可能是方阵，我们怎样才能描述这样普通的矩阵呢的重要特征呢？我们有必要先说说特征值和奇异值之间的关系。

对于特征值分解公式， ATA 是方阵，我们求 ATA 的特征值，即 $(A^T A)x = \lambda x$，此时求得的特征值就对应奇异值的平方，求得的特征向量 v 称为右奇异向量，另外还可以得到：

$$\sigma_i = \sqrt{\lambda_i}$$

$$u_i = \frac{Av_i}{\sigma_i}$$

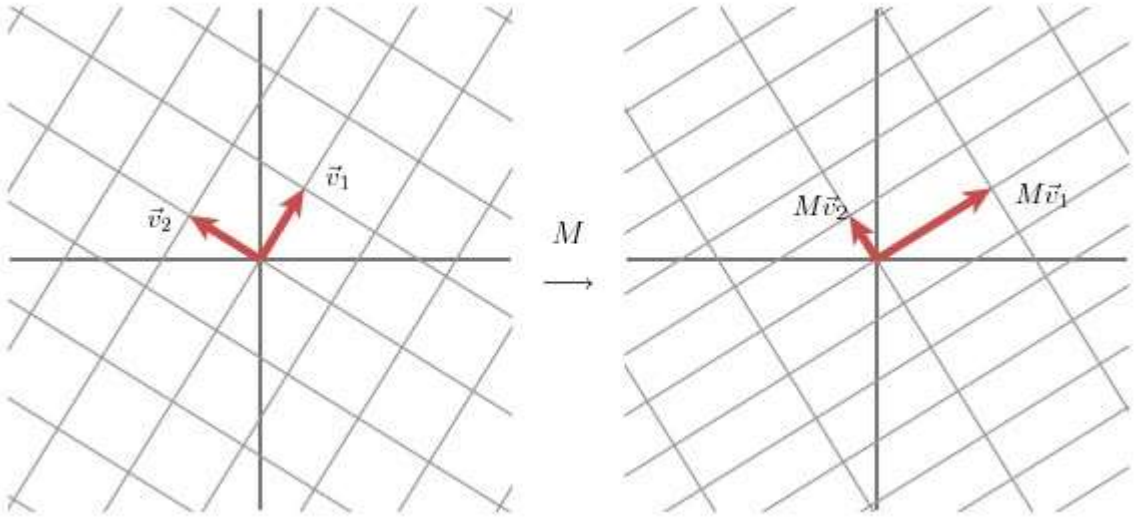
所求的 ui 就是左奇异向量， σi 就是奇异值。

奇异值分解[6]就是一个能适用于任意的矩阵的一种分解的方法：

$$A = U\Sigma V^T$$

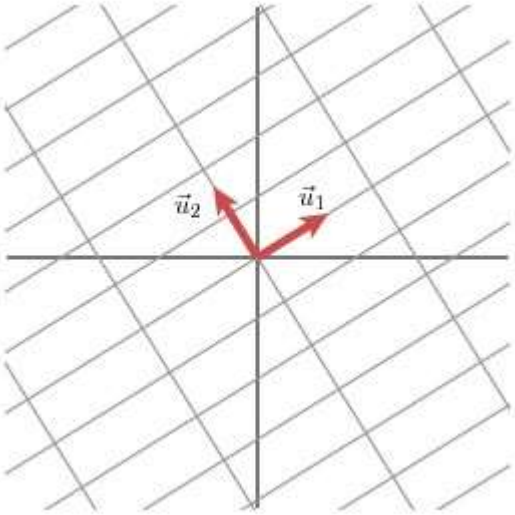
该部分是从几何层面上去理解二维的 SVD：对于任意的 2 x 2 矩阵，通过 SVD 可以将一个相互垂直的网格(orthogonal grid)变换到另外一个相互垂直的网格。

我们可以通过向量的方式来描述这个事实: 首先，选择两个相互正交的单位向量 v1 和 v2, 向量 Mv1 和 Mv2 正交。



u1 和 u2 分别表示 Mv1 和 Mv2 的单位向量，σ1 * u1 = Mv1 和 σ2 * u2 = Mv2。σ1 和 σ2 分别表示这不同方向向

量上的模，也称作为矩阵 M 的奇异值。



这样我们就有了如下关系式

$Mv1 = \sigma 1u1$

$Mv2 = \sigma 2u2$

我们现在可以简单描述下经过 M 线性变换后的向量 x 的表达形式。由于向量 v1 和 v2 是正交的单位向量，我们可以得到如下式子：

$x = (v1 \cdot x)\, v1 + (v2 \cdot x)\, v2$

这就意味着：

$Mx = (v1 \cdot x)\, Mv1 + (v2 \cdot x)\, Mv2$

$Mx = (v1 \cdot x)\, \sigma 1u1 + (v2 \cdot x)\, \sigma 2u2$

向量内积可以用向量的转置来表示，如下所示

$v \cdot x = vTx$

最终的式子为

$Mx = u1\sigma 1\ v1Tx + u2\sigma 2\ v2Tx$

$M = u1\sigma 1\ v1T + u2\sigma 2\ v2T$

上述的式子经常表示成

M = UΣVT

u 矩阵的列向量分别是 u1,u2 ，Σ 是一个对角矩阵，对角元素分别是对应的 σ1 和 σ2，V 矩阵的列向量分别是 v1,v2。上角标 T 表示矩阵 V 的转置。

这就表明任意的矩阵 M 是可以分解成三个矩阵。V 表示了原始域的标准正交基，u 表示经过 M 变换后的 co-domain 的标准正交基，Σ 表示了 V 中的向量与 u 中相对应向量之间的关系。(V describes an orthonormal basis in the domain, and U describes an orthonormal basis in the co-domain, and Σ describes how much the vectors in V are stretched to give the vectors in U.)

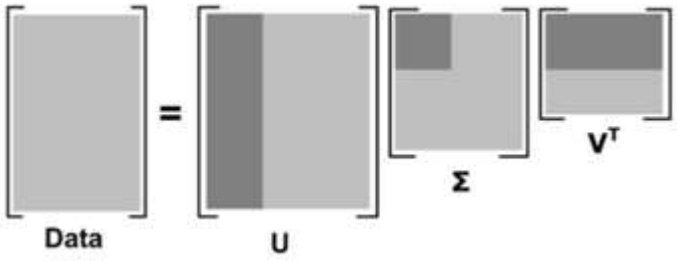
三、SVD 降维

1）降维原理

SVD 将矩阵分为三个矩阵的乘积，公式：

$$Data_{mxn} = U_{mxm}\Sigma_{mxn}V^{T}_{nxn}$$

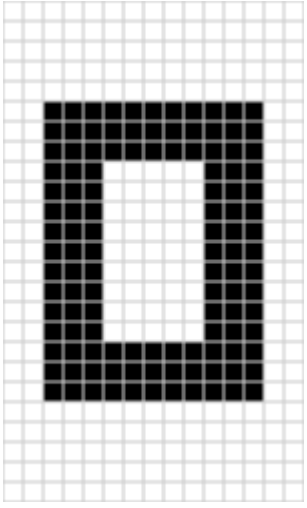
中间矩阵∑为对角阵，对角元素值为 Data 矩阵特征值 λi，且已经从大到小排序，即使去掉特征值小的那些特征，依然可以很好地重构出原始矩阵。如下图：其中阴影部分代表去掉小特征值，重构时的三个矩阵。



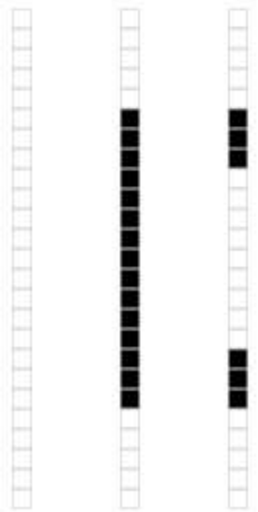
果 m 代表商品个数，n 代表用户个数，则 U 矩阵每行代表商品属性，现在通过降维 U 矩阵（取阴影部分）后，每个商品的属性可以用更低的维度表示（假设 k 维）。这样当新来一个用户的商品推荐向量 X，则可以根据公式 X*U1*inv(S1)得到一个 k 维的向量，然后在 V’中寻找最相似的的那个用户（相似度计算可用余弦公式），根据这个用户的评分来推荐（主要是推荐新用户未打分的那些商品）。

2）降维 SVD 实例

我们来看一个奇异值分解在数据表达上的应用。假设我们有如下的一张 15 x 25 的图像数据。



如图所示，该图像主要由下面三部分构成。



我们将图像表示成 15 x 25 的矩阵，矩阵的元素对应着图像的不同像素，如果像素是白色的话，就取 1，黑色的就取 0. 我们得到了一个具有 375 个元素的矩阵，如下图所示

$$
M=\begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{bmatrix}
$$

如果我们对矩阵 M 进行奇异值分解以后，得到奇异值分别是

σ1 = 14.72

σ2 = 5.22

σ3 = 3.31

矩阵 M 就可以表示成

M=u1σ1 v1T + u2σ2 v2T + u3σ3 v3T

vi 具有 15 个元素，ui 具有 25 个元素，σi 对应不同的奇异值。如上图所示，我们就可以用 123 个元素来表示具有 375 个元素的图像数据了。

四、结语

SVD 在一方面能有效的预测评分值，另一方面 SVD 是一种有效的降维工具。在数据维度庞大的时候，我们可以有效的利用 SVD 来提取重要的特种逼近真是矩阵。SVD 方法已经运用到多个领域，推荐 SVD 我们可以精确推荐，降维 SVD 可进行图像压缩、减噪、数据分析等等

但是 SVD 也存在一个问题，就是在大规模数据集上，SVD 计算困难。我们或许可以通过线下计算来一定程度上解决这个问题，可是对实时性有要求的问题上，这个问题特别突出。Google 已经对 SVD 进行了分布式计算，加快了计算，但仍然需要改进。期待小白和 Inplus+lab 共同进步，一起解决这些问题！

五、引用文献

[1]:https://zh.wikipedia.org/wiki/%E5%A5%87%E5%BC%82%E5%80%BC%E5%88%86%E8%A7%A3

[2]:http://baike.baidu.com/link?url=hG3S0FSpfe_ps9rLqG3CZTXaIxLNc_TVUYl89dUnUZM8zgJFnhdF3tit_eQLM1MfOxCAKbajh6rLa2QiB1UPC_

[3]:http://www.ams.org/samplings/feature-column/fcarc-svd

[4]:http://baike.baidu.com/link?url=a-eI6H0W463HPdu4s45Doak38bF2hXdlEljGkLvQyuERBDANlYXs35cDvLyrBfeDmJtvdJ3Da4Tl2gKlaCdCn_

[5]:http://blog.csdn.net/xiahouzuoxin/article/details/41118351

[6]:http://blog.sciencenet.cn/blog-696950-699432.html

[7]:http://m.blog.csdn.net/blog/DianaCody/40779505

[8]:机器学习实战。