Nmf 与图像聚类

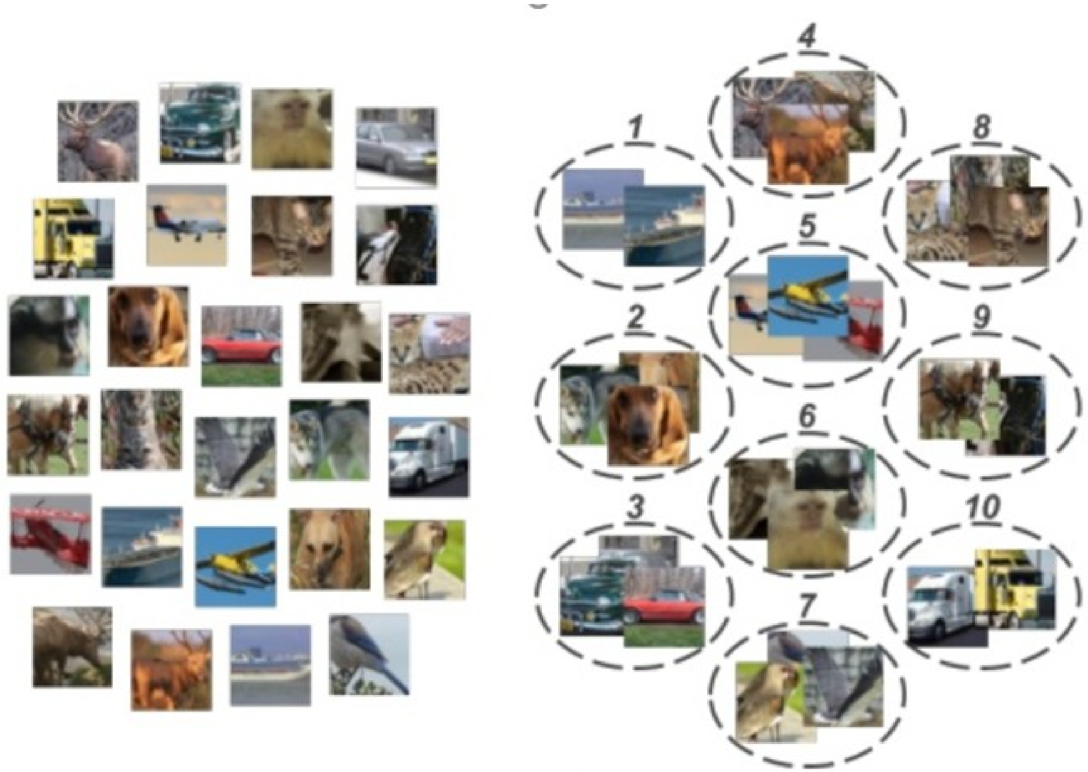
2020/04/20

Ma Li

mali_hp@pku.edu.cn

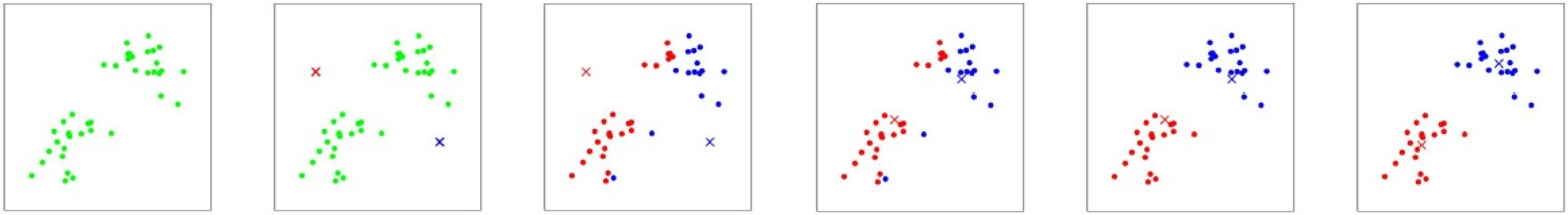
图像聚类

- 图像聚类
  - 特征提取
    - PCA
    - NMF
    - SIFT
  - 聚类
    - K-Means
    - 层次聚类
    - 谱聚类



图像聚类

- 聚类算法
  - K-Means
    1. 选取 K 个点做为初始聚集的簇心（也可选择非样本点）；
    2. 分别计算每个样本点到 K 个簇心的距离（这里的距离一般取欧氏距离或余弦距离），找到离该点最近的簇心，将它归属到对应的簇；
    3. 所有点都归属到簇之后， M 个点就分为了 K 个簇。之后重新计算每个簇的重心（平均距离中心），将其定为新的“簇心”；
    4. 反复迭代 2,3 步骤，直到达到某个中止条件（常用的有迭代次数、簇中心点变化率等）



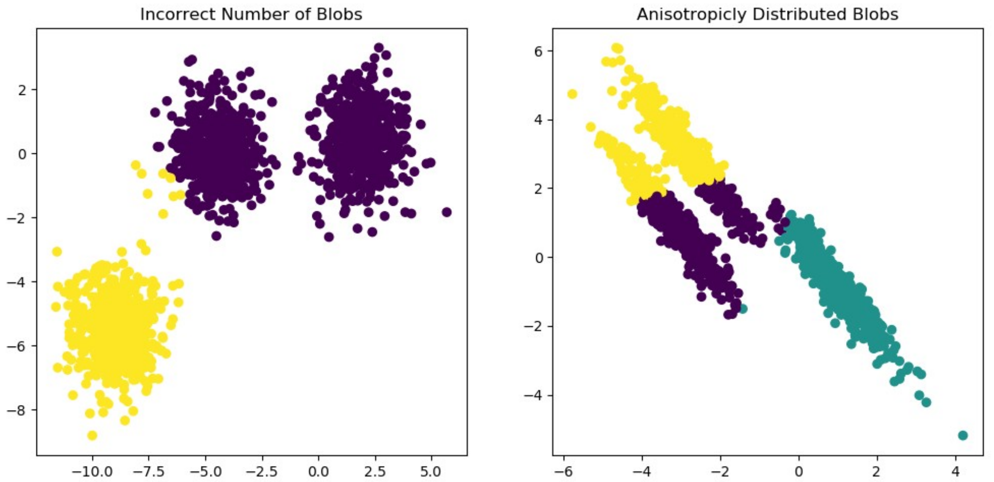
(a) (b) (c) (d) (e) (f)

图像聚类

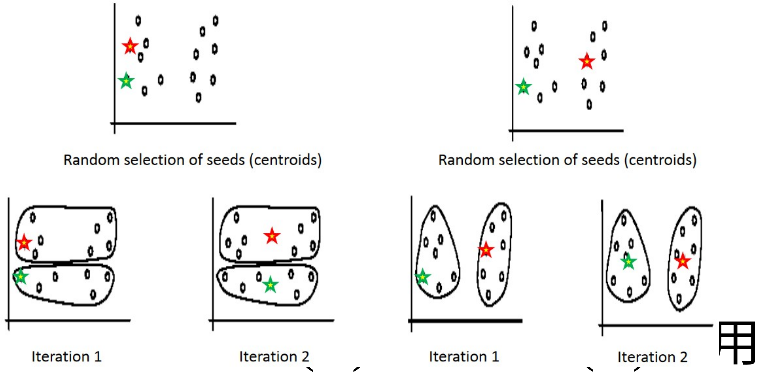
- 聚类算法
  - sklearn.cluster.Kmeans()

参数	说明
n-cluster	分类簇的数量
max_iter	最大的迭代次数
n_init	算法的运行次数
init	接收待定的string。kmeans++表示该初始化策略选择的初始均值向量之间都距离比较远，它的效果较好；random表示从数据中随机选择K个样本最为初始均值向量；或者提供一个数组，数组的形状为（n_cluster,n_features），该数组作为初始均值向量。
precompute_distance	接收Boolean或者auto。表示是否提前计算好样本之间的距离，auto表示如果nsamples*n>12 million，则不提前计算。
tol	接收float，表示算法收敛的阈值。
N_jobs	表示任务使用CPU数量
random_state	表示随机数生成器的种子。
verbose	0表示不输出日志信息；1表示每隔一段时间打印一次日志信息。如果大于1，打印次数频繁。

图像聚类



- K-Means 不足
  - 初值敏感
  - K 值的选择
    - K 值是用户给定的，在进行数据处理前，K 值是未知的，给定合适的 k 值，需要先验知识，凭空估计很困难，或者可能导致效果很差
  - 异常点敏感
    - 特殊值(离群值或称为异常值)对模型的影响比较大
  - 不适合发现非凸形状的簇或者大小差别较大的簇



- K-Means++，K-Means||，Mini Batch K-Means
- K-Means++：对数据集中的每个点 x，计算 x 到所有已有聚类中心 用线性概率选择出下一个聚类中心点(距离较远的一个点成为新增的一个聚类中心点而不是最远的一个点是由于最远点可能为异常点，这里的选取规则是计算出 M 个距离团较远的点，然后随机选择出一点)

图像聚类

- 层次聚类
  - 凝聚型层次聚类的策略是先将每个对象作为一个簇，然后合并这些原子簇为越来越大的簇，直到所有对象都在一个簇中，或者某个终止条件被满足。绝大多数层次聚类属于凝聚型层次聚类，它们只是在簇间相似度的定义上有所不同。 这里给出采用最小距离的凝聚层次聚类算法流程：
    1. 将每个对象看作一类，计算两两之间的最小距离；
    2. 将距离最小的两个类合并成一个新类；
    3. 重新计算新类与所有类之间的距离；
    4. 重复 2， 3 两个步骤，直到所有类合并成一类或满足终止条件。
  - 优点：距离和规则的相似度容易定义，限制少；不需要预先制定聚类数；可以发现类的层次关系；可以聚类成其它形状。
  - 缺点：计算复杂度太高；奇异值也能产生很大影响；算法很可能聚类成链状

评价指标简介

$$E(RI)=E(\sum_{i,j}\binom{n_{ij}}{2})=[\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}]/\binom{n}{2}$$

- Rand index(兰德指数)(RI) 与 Adjusted Rand index(调整兰德指数)(ARI)

$$\text{RI}=\frac{a+b}{C_2^{n_{\text{samples}}}}\qquad \text{ARI}=\frac{\text{RI}-E[\text{RI}]}{\max(\text{RI})-E[\text{RI}]}$$

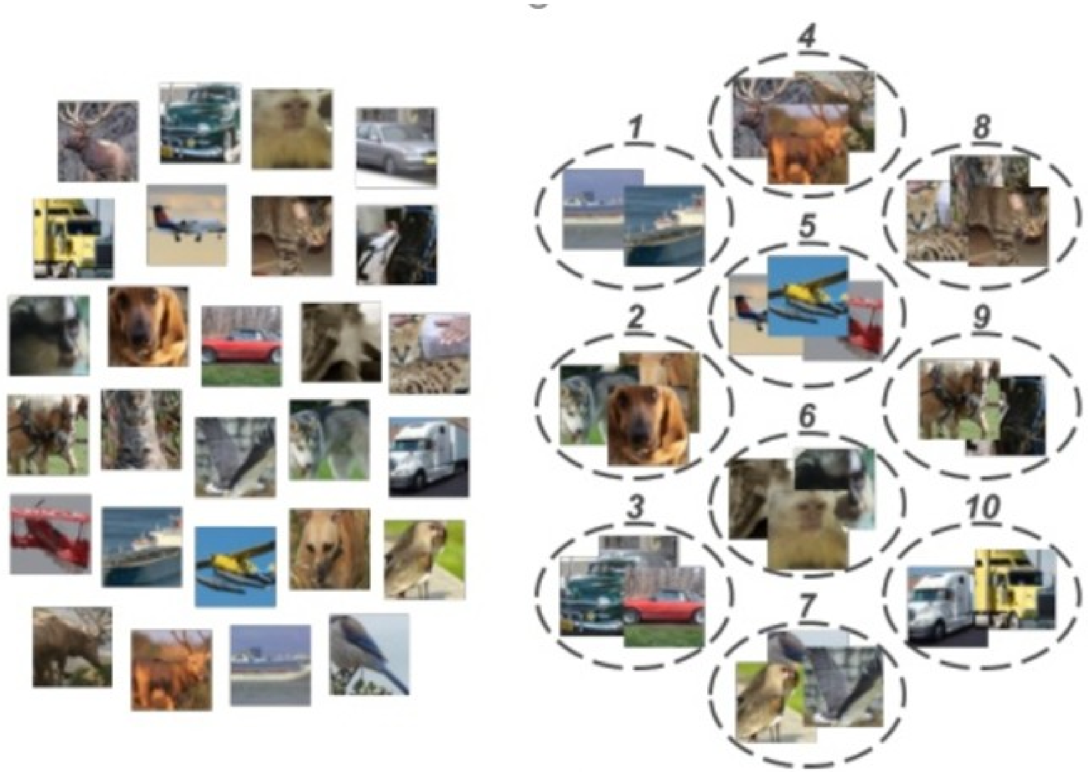
$$max(RI)=\frac{1}{2}[\sum_i\binom{n_{i.}}{2}+\sum_j\binom{n_{.j}}{2}]$$

- RI 中 C 表示实际类别信息， K 表示聚类结果， a 表示在 C 与 K 中都是同类别的元素对数， b 表示在 C 与 K 中都是不同类别的元素对数，其中表示数据集中可以组成的对数， RI 取值范围为 [0,1]，值越大意味着聚类结果与真实情况越吻合。
- ARI 实现了 “在聚类结果随机产生的情况下，指标应该接近零” 的要求，它具有更高的区分度。 ARI 取值范围为 [−1,1]，值越大意味着聚类结果与真实情况越吻合。具体推导参见 https://link.springer.com/content/pdf/10.1007/BF01908075.pdf
- 轮廓系数
  - 簇内部相似度：计算样本 i 到同簇样本的平均距离 ， 越小，表示样本 i 越应该被聚类到改簇，簇 C 中的所有样本的的均值被称为簇 C 的簇不相似度。
  - 簇间不相似度：计算样本 i 到其他簇的所有平均距离， ；越大，表示样本 i 月不属于其他簇
  - 轮廓系数：值越接近 1 表示样本 i 聚类越合理，越接近 -1 ，表示样本 i 应该分类到其他簇中。近似为 0 表示赝本 i 应该在边界上；所有样本的均值被称为聚类结果的轮廓系数

$$s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$

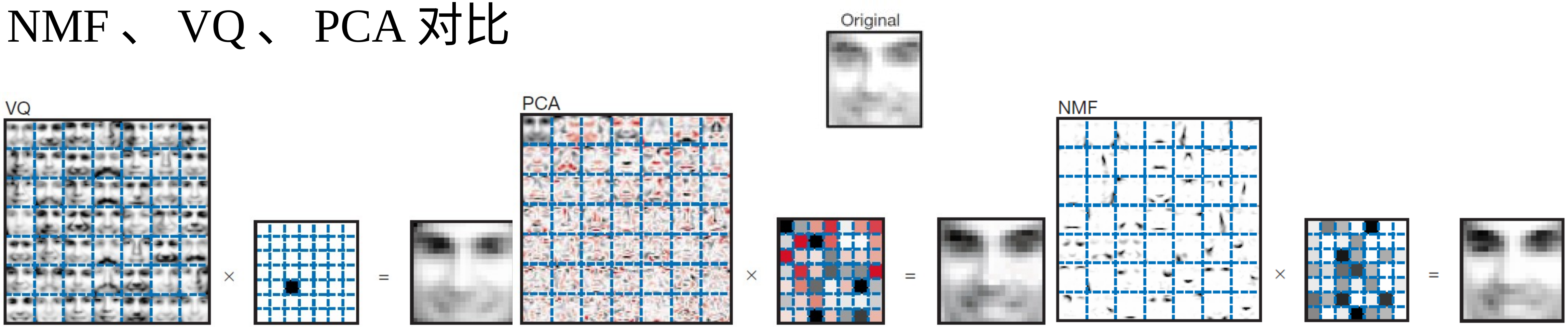
图像聚类

- 图像聚类
  - 特征提取
    - PCA
    - NMF
    - SIFT
  - 聚类
    - K-Means
    - 层次聚类
    - 谱聚类



非负矩阵分解 (non-negative matrix factorization)

- NMF 、 VQ 、 PCA 对比

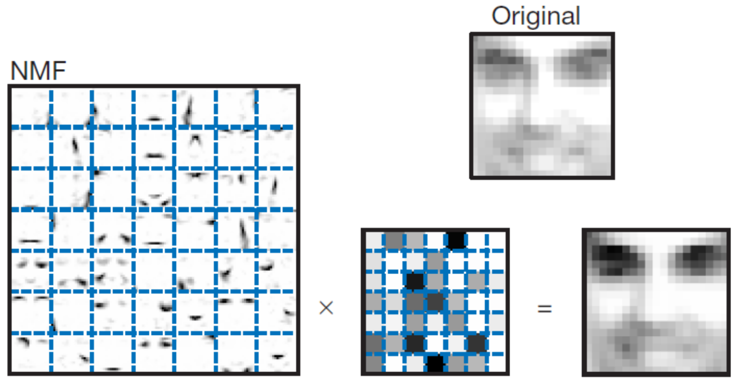


  - VQ: holistic representations
  - PCA: holistic representations
  - NMF: parts-based representation
- nmf 图像特征是稀疏的，因为它们不是全局性的（ non-global ），可以看到其中有几个不同版本的嘴和鼻子或者其他的面部部位，这些部位有不同的版本、位置或形式

非负矩阵分解 (non-negative matrix factorization)

- NMF： 给定矩阵$V \in R_+^{n\times m}$， 求$W \in R_+^{n\times r}, H \in R_+^{r\times m}$， 使得V=WH
- 在数学上，矩阵分解结果中存在负值是正确的，但负值元素在实际问题中往往是没有意义的。例如图像像素值均为非负，因此 NMF 具有良好的可解释性

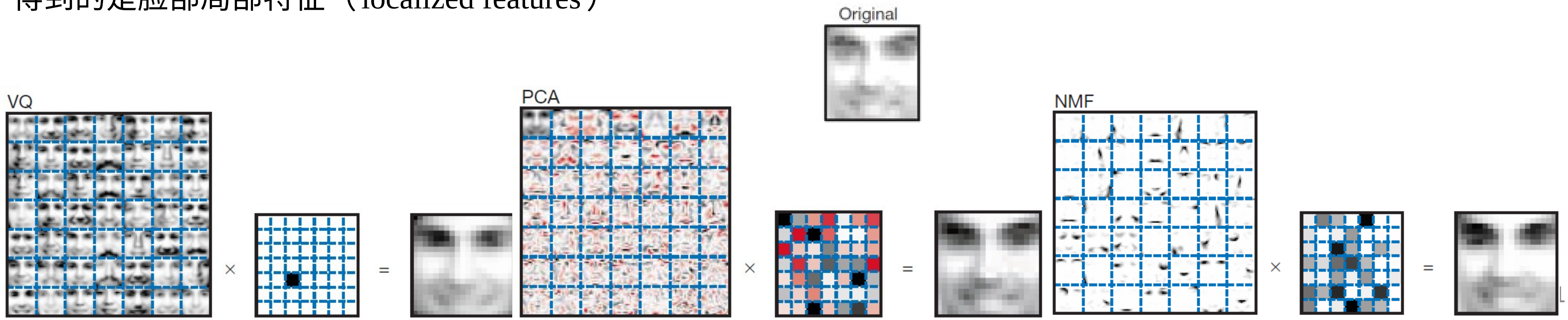
$$\underbrace{\mathrm{V}(:,j)}_{j\text{th facial image}} \approx \sum_{k=1}^{r} \underbrace{W(:,k)}_{\text{facial features}} \underbrace{H(k,j)}_{\substack{\text{importance of features}\\ \text{in } j\text{th image}}} = \underbrace{WH(:,j)}_{\substack{\text{approximation}\\ \text{of } j\text{th image}}} .$$



- 的每列代表特征（基图像 basis image），的列代表权重，此处取 49

非负矩阵分解 (non-negative matrix factorization)

- VQ， H的每一列被约束为一个一元向量。其中只有一个元素为1，其他元素为0。此时得到的基图像称为原型基图像，这些原型图像表示一张原型脸（whole-face prototypes）
- PCA， W的各列之间相互正交， H各行之间相互正交。这个约束比VQ的松弛很多，也就是， H中的元素可为正也可为负。 V中每一张脸的每一个像素点都是W中各列对应的像素点的一个加权和。由于权重矩阵H中元素符号的任意性，所以基矩阵W表示出来并不像VQ中原型脸那样的直观可解释。此时将W的列数据画出来并不一定能直接看到一张“脸”。但是在统计上可以解释为最大方差方向，我们把这些“脸”称为“特征脸” （eigenfaces）
- NMF，由于加了非负约束。与VQ的单一元素不为0不同， NMF允许基图像H间的加权结合来表示脸部图像V；与PCA不同， NMF的加权系数H中的元素都为非负的。前两者得到的都是一个完整的脸部特征基图像，而NMF得到的是脸部局部特征（localized features）



非负矩阵分解 (non-negative matrix factorization)

- NMF 的缺点
  - 虽然 NMF 能够用于 facial parts 学习和语义主题分析，但是并不意味着它能从任何一个数据库中习得 parts，例如从不同角度拍照生成的人脸照片，或者是太复杂的事物。分析学习这些复杂模型的结构，可能需要多级的 hidden variables，也就是说多级的 NMF，而不是一级的 NMF。
  - 虽然非负的限制（non-negativity constraints）能够帮助进行 parts-based representations，但是，这并不表示 NMF 就足够了。 NMF 不能够学习到各个 part 之间的句法关系。 NMF 假定 H 是非负的，但是这并不进一步表示这些 parts 之间是统计独立的。

非负矩阵分解 (non-negative matrix factorization)

- Python 中的 NMF

```python
from sklearn.decomposition import NMF
```

```
n_components : int or None
    Number of components, if n_components is not set all features
    are kept.

init :  'random' | 'nndsvd' |  'nndsvda' | 'nndsvdar' | 'custom'
    Method used to initialize the procedure.
    Default: 'nndsvd' if n_components < n_features, otherwise random.
    Valid options:

    - 'random': non-negative random matrices, scaled with:
        sqrt(X.mean() / n_components)

    - 'nndsvd': Nonnegative Double Singular Value Decomposition (NNDSVD)
        initialization (better for sparseness)

    - 'nndsvda': NNDSVD with zeros filled with the average of X
        (better when sparsity is not desired)

    - 'nndsvdar': NNDSVD with zeros filled with small random values
        (generally faster, less accurate alternative to NNDSVDa
        for when sparsity is not desired)

    - 'custom': use custom matrices W and H
```

非负矩阵分解 (non-negative matrix factorization)

- 定义：
  - 给定矩阵，求，使得 V=WH
- 直观理解：
  - 原始矩阵的列向量是对左矩阵中所有列向量的加权和，而权重系数就是右矩阵对应列向量的元素，故称为基矩阵，为系数矩阵。一般情况下的选择要比小，，这时用系数矩阵代替原始矩阵，就可以实现对原始矩阵进行降维，得到数据特征的降维矩阵。
- 非负矩阵分解是一个 NP 问题，实际求解将其转化为最优化问题

$$min||V-WH||^2$$

$$s.t. W, H \geq 0$$

-
- 该问题在只考虑 W 或 H 之一时为凸，但在同时考虑 WH 两个变量时不为凸。寻找一种可以找到全局最小值的算法去解决以上两个最优化问题是不切实际的。但是，还有许多数值优化方法可以用于寻找局部最小值。

非负矩阵分解 (non-negative matrix factorization)

- 理论推导：（以欧几里得距离作为代价函数）

$$\min D_E(V \| WH) = \frac{1}{2}\|V - WH\|_2^2$$

，

$$\mu_{ik} = \frac{w_{ik}}{\left[WHH^T\right]_{ik}}, \quad \eta_{kj} = \frac{h_{kj}}{\left[W^T WH\right]_{kj}}$$

$$w_{ik} \leftarrow w_{ik} - \mu_{ik} \frac{\partial D_E(V \| WH)}{\partial w_{ik}}$$

$$h_{kj} \leftarrow h_{kj} - \eta_{kj} \frac{\partial D_E(V \| WH)}{\partial h_{kj}}$$

导、

$$w_{ik} \leftarrow w_{ik} \frac{\left[VH^T\right]_{ik}}{\left[WHH^T\right]_{ik}}$$

$$h_{kj} \leftarrow h_{kj} \frac{\left[W^T V\right]_{kj}}{\left[W^T WH\right]_{kj}}$$

$$\frac{\partial D_E(V \| WH)}{\partial w_{ik}} = -\left[(V - WH) H^T\right]_{ik}$$

$$\frac{\partial D_E(V \| WH)}{\partial h_{kj}} = -\left[W^T (V - WH)\right]_{kj}$$

非负矩阵分解 (non-negative matrix factorization)

- 理论推导：（另一种代价函数）

$$\min D_{KL}(V\|WH)=\sum_{i=1}^{I}\sum_{j=1}^{J}\left(v_{ij}\ln\frac{v_{ij}}{[WH]_{ij}}-v_{ij}+[WH]_{ij}\right)\quad ,\qquad \mu_{ik}=\frac{1}{\sum_{j=1}^{J}h_{kj}},\quad \eta_{kj}=\frac{1}{\sum_{i=1}^{I}w_{ik}}$$

$$w_{ik}\leftarrow w_{ik}-\mu_{ik}\frac{\partial D_{KL}(V\|WH)}{\partial w_{ik}}$$

$$h_{kj}\leftarrow h_{kj}-\eta_{kj}\frac{\partial D_{KL}(V\|WH)}{\partial h_{kj}}$$

$$\frac{\partial D_{KL}(V\|WH)}{\partial w_{ik}}=-\sum_{j=1}^{J}\left(h_{kj}v_{ij}/[WH]_{ij}+h_{kj}\right)$$

$$\frac{\partial D_{KL}(V\|WH)}{\partial h_{kj}}=-\sum_{j=1}^{J}\left(w_{ik}v_{ij}/[WH]_{ij}+w_{ik}\right)$$

$$w_{ik}\leftarrow w_{ik}\frac{\sum_{j=1}^{J}h_{kj}v_{ik}/[WH]_{ik}}{\sum_{j=1}^{J}h_{kj}}$$

$$h_{kj}\leftarrow h_{kj}\frac{\sum_{i=1}^{I}w_{ik}v_{ik}/[WH]_{ik}}{\sum_{i=1}^{I}w_{ik}}$$

收敛性证明可以参考： Lee D D, Seung H S. Algorithms for Non-negative Matrix Factorization[C]// NIPS. 2000:556--562.

非负矩阵分解 (non-negative matrix factorization)

- 理论推导（nature1999 中的代价函数）

$$minD(V||WH) = \sum_{i=1}^{n}\sum_{\mu=1}^{m}[V_{i\mu}\log(WH)_{i\mu} - (WH)_{i\mu}]$$

$$W_{ia} \leftarrow W_{ia}\sum_{\mu}\frac{V_{i\mu}}{(WH)_{i\mu}}H_{a\mu}$$

$$W_{ia} \leftarrow \frac{W_{ia}}{\sum_{j}W_{ja}}$$

$$H_{a\mu} \leftarrow H_{a\mu}\sum_{i}W_{ia}\frac{V_{i\mu}}{(WH)_{i\mu}}$$

像素聚类

- 对像素进行聚类而不是对所有的图像进行聚类。这种将图像区域归并成“有意义的”组件称为图像分割。单纯在像素水平上应用 K-means 可以用于一些简单图像的图像分割。
- 对简单的图像利用 K-Means 进行像素聚类。
  - 用一个 100×100 的窗口在图像中滑动
  - 在 RGB 三通道上，分别求窗口所在位置中窗口包含像素值的平均值作为特征
  - 对这些特征利用 K-Means 进行聚类
  - 转至 ipynb

Q & A