Algorithms For None-Negative Matrix Factorization

ShenHengheng

Institute of Big Data
Institute Of Computation Technology, Beijing

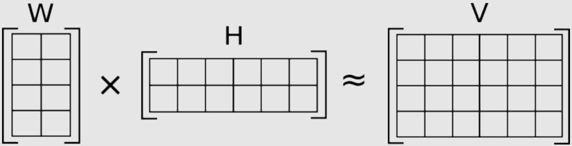
shenhengheng17g A.T ict.ac.cn

Overview

形式化的定义与应用

定义

非负矩阵分解（NMF），也被叫做非负矩阵近似，是多元/多变量数据和线性代数里面的一系列算法，对于任意给定的一个非负矩阵 V，NMF 算法能够寻找到一个非负矩阵 W 和一个非负矩阵 H，使得满足，从而将一个非负的矩阵分解为左右两个非负矩阵的乘积。然而这种非负矩阵分解并不是完全可解，所以往往利用数值分析中的数值逼近的思想及相关的算法近似求解.



应用

矩阵分解主要应用在机器人控制，图聚类，计算机视觉，音视频信号处理，推荐系统等.

非负矩阵的历史

著名的科学杂志《Nature》于 1999 年刊登了两位芬兰的科学家 Daniel D. Lee 和 H. Sebastian Seung 对数学中非负矩阵研究的突出成果。该文提出了一种新的矩阵分解思想——非负矩阵分解 (Non-negative Matrix Factorization，NMF) 算法，即 NMF 是在矩阵中所有元素均为非负数约束条件之下的矩阵分解方法。

分解 $\mathbf{V} = \mathbf{WH}$ 数学表示

定义问题

- 输入：V
- 输出：W and H
- 目标：$min||\mathbf{V} - \mathbf{WH}||$

- V -m x n 数据矩阵
  - m 非负矩阵的行数，no.rows
  - n 非负矩阵的列数，no.cols
- W -m x r 矩阵，每一向量为基向量
- H -r x n 系数矩阵

- Optimize accuracy of solution[Hoyer, 2004]:
  - $min||\mathbf{V} - \mathbf{W}\mathbf{H}||_F \;\; where \;\; \mathbf{W}, \mathbf{H} \geq 0$
  - We can drop nonnegative constraints
    - $min||\mathbf{V} - (\mathbf{W}.\mathbf{W})(\mathbf{H}.\mathbf{H})||$
- Many options for objective function
- Many options for algorithm
  - W,H will depend on initial choices
  - Convergence is not always guaranteed

为什么要使用 NMF 算法做矩阵分解？

NMF 算法提供了基于简单迭代的求解 $\mathbf{U}, \mathbf{V}$ 的方法，求解方法具有收敛速度快、左右非负矩阵存储空间小的特点，它能将高维的数据矩阵降维处理，适合处理大规模数据。利用 NMF 进行文本、图像大规模数据的分析方法，较传统的处理算法速度更快、更便捷。

Non-negative Matrix Factorization

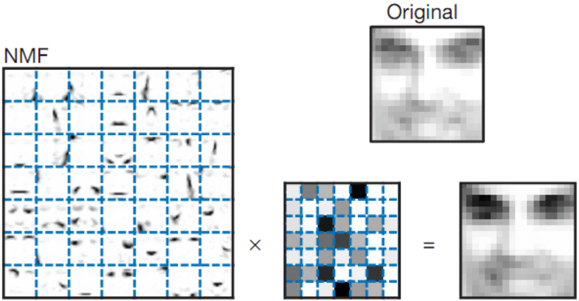


图 1: [Lee and Seung, 1999]NMF on facial images produces images containing facial parts.

PCA VS VQ

在机器学习算法中经常看到矩阵分解分析方法。比如：

1. 主成分分析算法 (PCA)
2. 矢量量化算法 (VQ)
3. 奇异值分解 (SVD)

它们广泛应用于信号（视频和图像）的压缩和处理，他们都是要解决非监督学习任务，主要是聚类任务，上面的三个算法可以看作将一个矩阵通过**设定不同的约束**，将一个大的矩阵来进行分解，从而实现数据的压缩的目的.

PCA VS VQ

- 主成分分析主要利用矩阵二次型的分解，首先要**限制**矩阵为非奇异矩阵，这样得到的**特征向量才可以满足单位正交**的特性.
- 矢量量化算法主要利用了 winner-take-all(赢家通吃) 的策略/限制，将数据进行聚类.

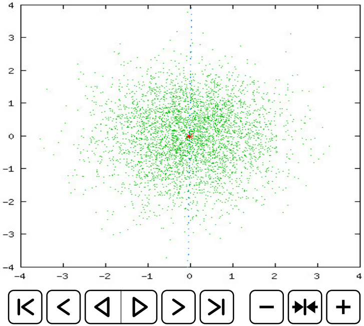


图 2: VQ 算法

具体比较

在 VQ 分解中，每一列的 H 被约束为一个一元矢量。其中只有一个元素为 1，其他元素为 0。若 H 的第一列中，第 r_1 个元素为 1，那么 V 中第一列的脸，就完全由基图像中的第 r_1 列数据表示。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）。

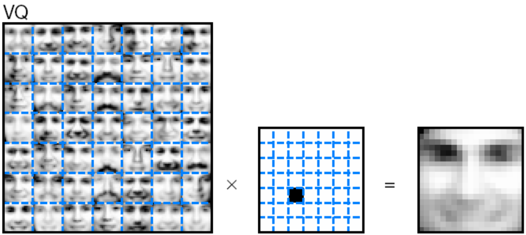


图 3: VQ 算法直接在原图像 V 中，选取了一个脸 (5,3) 代替整个图像 V

在 PCA 分解中，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”（distorted versions of whole faces）。

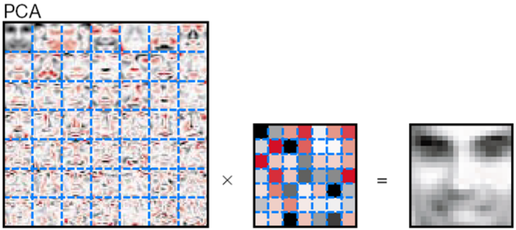


图 4: 通过基图像和权重图像向乘，会得到一个具有统计意义上的“特征脸”

在 NMF 中，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 得到的是脸部子特征（localized features）。

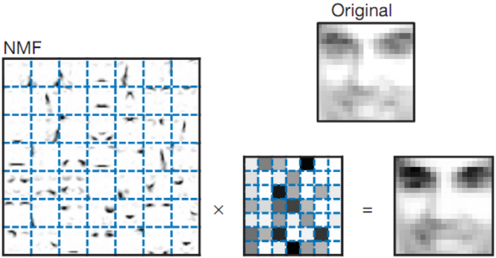


图 5: 由于严格意义上的“非负”约束，最后得到的是一个“子特征”，[1]

[1]权重矩阵 W 说明，红色表示负数，灰度像素表示证书，其中颜色越深数值越大.

求解 W 和 H 的算法

- Alternating Least Squares
  - Paatero 1994
- Multiplicative Update Rules
  - Lee-Seung 2000 Nature
  - Used by Hoyer
- Gradient Descent
  - Hoyer 2004
  - Berry-Plemmons 2004

Cost function

代价函数 1

- Cost function: $J = ||V - WH||^2,\ subjectto.\ W, H \geq 0$
- Object: $Minimize\ J$

代价函数 2

- Cost function:
  $J = D(V||WH) = \sum_{ij} V_{ij} log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij},\ s.t.\ W, H \geq 0$
- Object: $Minimize\ J$

学习算法-Gradient Descent[Roweis and Saul, 2000]

对第一个定义代价函数，它的更新规则为

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} + \eta_{\alpha\mu}[(W^T V)_{\alpha\mu} - (W^T WH)_{\alpha\mu})]$$

令 $\eta_{\alpha\mu} = \frac{H_{\alpha\mu}}{(W^T WH)_{\alpha\mu}}$ 那么

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{(W^T V)_{\alpha\mu}}{(W^T WH)_{\alpha\mu}}$$

同理

$$W_{i\alpha} \leftarrow W_{i\alpha} \frac{(VH^T)_{i\alpha}}{(WHH^T)_{i\alpha}}$$

对第 2 个定义代价函数同样应用上页的更新规则，可以得到.

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{\sum_i W_{i\alpha} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{k\alpha}}, W_{\alpha\mu} \leftarrow W_{\alpha\mu} \frac{\sum_\mu H_{\alpha\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_v H_{\alpha v}}$$

Matlab 内置函数[2]

```
[W,H] = nnmf(A,k)
[W,H] = nnmf(A,k,param1,val1,param2,val2,...)
[W,H,D] = nnmf(...)
```

给定矩阵 $A \in \mathbb{R}^{n\times m}$ 和整数 k，返回分解的非负矩阵 $W \in \mathbb{R}^{n\times k}, H \in \mathbb{R}^{k\times m}$.

[2] 官方文档：http://cn.mathworks.com/help/stats/nnmf.html

Reference I

Hoyer, P. O. (2004).
Non-negative matrix factorization with sparseness constraints.
Journal of machine learning research, 5(Nov):1457–1469.

Lee, D. D. and Seung, H. S. (1999).
Learning the parts of objects by non-negative matrix factorization.
Nature, 401(6755):788–791.

Roweis, S. T. and Saul, L. K. (2000).
Nonlinear dimensionality reduction by locally linear embedding.
science, 290(5500):2323–2326.