Parallel Vector Field Regularized Non-Negative Matrix Factorization for Image Representation

Y. Peng[1], R. Tang[1], W. Kong[1], F. Qin[1] and F. Nie[2]

[1]School of Computer Science and Engineering, Hangzhou Dianzi University
[2]Center for OPTical IMagery Analysis & Learning, Northwestern Polytechnical University

yongpeng@hdu.edu.cn

April 18, 2018

Overview

1. Introduction
2. Related Works
   - Non-negative Matrix Factorization
   - Parallel Fields and Linear Functions
3. Parallel Vector Field Regularized NMF
   - Model Formulation
   - Model Optimization
   - Algorithm Workflow
4. Experimental Studies
   - Experimental Settings and Datasets
   - Experimental Results
5. Conclusion
6. References

Motivation

- Non-negative matrix factorization (NMF) aims to learn parts-based representation by seeking for two non-negative matrices whose product can best approximate the original matrix;
- However, the manifold structure is not considered by NMF and many of the existing work use the graph Laplacian to ensure the smoothness of the learned representation coefficients on the data manifold. Further, beyond smoothness, it is suggested by recent theoretical work that we should ensure second order smoothness for the NMF mapping, which measures the linearity of the NMF mapping along the data manifold.
- Therefore, based on the equivalence between the gradient field of a linear function and a parallel vector field, we propose to find the NMF mapping which minimizes the approximation error, and simultaneously requires its gradient field to be as parallel as possible.

NMF

Given a data matrix $X = [x_1, \cdots, x_n] \in \mathbb{R}^{m \times n}$, NMF aims at finding two non-negative matrices $U = [u_{ik}] \in \mathbb{R}^{m \times K}$ and $H = [h_{jk}] \in \mathbb{R}^{n \times K}$ by minimizing

$$\mathcal{O}_{\text{NMF}} = \|X - UH^T\|^2, \quad s.t.\ U \geq 0, H \geq 0. \tag{1}$$

The K columns of U are basis vectors, and each column of H is an encoding of a sample vector in X and is an one-to-one mapping.

- Property. The non-negative property enforced on both U and H allows only addictive combinations among different bases, which makes NMF learn parts-based representation.
- Optimization. $\mathcal{O}_{\text{NMF}}$ is convex when updating variable one by one.

Parallel Fields and Linear Functions

In geometry and vector calculus, a vector field is a mapping from a manifold $\mathcal{M}$ to tangent spaces. The parallel vector field has close relationship to a linear function on the manifold, which can be described below.

Definition

(Parallel Field). A vector field X on manifold $\mathcal{M}$ is a parallel field if $\nabla X \equiv 0$, where ∇ is the covariant derivative on $\mathcal{M}$.

Definition

(Linear Function). A continuous function $f : \mathcal{M} \to \mathbb{R}$ is said to be linear if $(f \circ \gamma)(t) = f(\gamma(0)) + ct$ for each geodesic γ.

- In this work, a function f is linear means that it varies linearly along the geodesics of the manifold.
- The following proposition reveals the relationship between a parallel vector field and a linear function on the data manifold.

Proposition

Let V be a parallel field on the manifold. If it is also a gradient field for function f, $V = \nabla f$, then f is a linear function on the manifold.

- We will not strictly distinguish between the concepts of covariant derivative and gradient field in this work.

PFNMF Model Formulation

The basic rules are

- learn vector fields on manifold from data samples to approximate the gradient field of the NMF mapping function, and encourage the vector fields to be as parallel as possible;
- learn NMF embedding while enforcing the encodes to be as close to the estimated parallel fields as possible.

Since the manifold $\mathcal{M}$ is unknown, the mapping function in NMF $f_k(x_j) = h_{jk} \triangleq f_j^{(k)}, j = 1, \ldots, n$ has no explicit form. First we need to estimate the tangent space of each data point, which will be used for discretizing the continuous objective function form when estimating parallel vector fields and learning NMF embedding. Therefore,

- Compute the local tangent spaces;
- Estimate the parallel vector field;
- Learn the NMF mapping;

Compute the local tangent spaces $\mathcal{T}_{x_i}\mathcal{M}$

- Let W be the corresponding affinity matrix of graph $\mathcal{G}$ and W is simply defined by 0-1 weight.
- Let $x_i \sim x_j$ denote x_i and x_j are neighbors. For each x_i, we can estimate its tangent space $T_{x_i}\mathcal{M}$ by performing PCA on its local neighborhood. We choose the eigenvectors corresponding to the d largest eigenvalues since $T_{x_i}\mathcal{M}$ is d-dimensional. Let $T_i \in \mathbb{R}^{m\times d}$ be the matrix whose columns constitute an orthogonal basis for $T_{x_i}\mathcal{M}$.
- $P_i = T_i T_i^T$ is the *unique* orthogonal projection from $\mathbb{R}^m$ onto the tangent space $T_{x_i}\mathcal{M}$. That is, for any vector $a \in \mathbb{R}^m$, we have $P_i a \in T_{x_i}\mathcal{M}$ and $(a - P_i a) \perp P_i a$.

Estimate the parallel vector field

Let V be a smooth vector field on $\mathcal{M}$. By definition, the covariant derivative of V should be zero. That is, $\nabla V \equiv 0$. For each point x_i

- let V_{x_i} denote the value of the vector field V at x_i
- $\nabla V|_{x_i}$ denote the value of ∇V at x_i

Then, V_{x_i} should be a vector in the tangent space $T_{x_i}\mathcal{M}$. and can be represented by the local coordinates of the tangent space, $V_{x_i} = T_i v_i$, where $v_i \in \mathbb{R}^d$.

The parallel field V can be obtained by solving

$$\min_{\mathbb{V}} \; E(\mathbb{V}) = \sum_{i,j=1}^{n} w_{ij} \| P_i T_j v_j - T_i v_i \|^2. \tag{2}$$

which has a compact form as $E(\mathbb{V}) \triangleq \mathbb{V}^T B \mathbb{V}$. By imposing constraint $\|\mathbb{V}\|^2 = 1$, the parallel vector field $\mathbb{V}$ can be estimated by solving the following eigenvalue decomposition problem

$$B\mathbb{V} = \lambda \mathbb{V}. \tag{3}$$

Learn the NMF embedding

If the dimensionality of learned NMF embedding is K; thus, we use $\tilde{\mathbb{V}} \in \mathbb{R}^{dn \times K}$ to denote the stack of K eigenvectors.
Once the parallel vector fields V_i are obtained, the embedding functions $f^{(k)} : \mathcal{M} \to \mathbb{R}$ can be constructed by requiring their gradient fields to be as close as V_i as possible, which can be achieved via minimizing

$$R(f^{(k)}) = \sum\nolimits_{i,j=1}^{n} w_{ij}((x_j - x_i)^2 T_i v_i - f_j^{(k)} + f_i^{(k)})^2. \tag{4}$$

Then we obtain the objective of PFNMF as

$$\mathcal{O}_{\text{PFNMF}} = \|X - UH^T\|^2 + \alpha \sum\nolimits_k R(f^{(k)}) \tag{5}$$

where

$$R(f^{(k)}) = 2f^{(k)T} L f^{(k)} + \mathbb{V}^T G \mathbb{V} - 2\mathbb{V}^T C f^{(k)}. \tag{6}$$

Using multiplicative updates, the updating rules are respectively

- Update U via

$$u_{ik} \leftarrow \frac{u_{ik}(XH)_{ik}}{(UH^T H)_{ik}}. \tag{7}$$

- Update H via

$$h_{jk} \leftarrow h_{jk} \frac{(X^T U + \alpha C^T \tilde{\mathbb{V}}^+ + 2\alpha WH)_{jk}}{(HU^T U + \alpha C^T \tilde{\mathbb{V}}^- + 2\alpha DH)_{jk}} \tag{8}$$

where $L = D - W$, $\tilde{\mathbb{V}} = \tilde{\mathbb{V}}^+ - \tilde{\mathbb{V}}^-$, $\tilde{\mathbb{V}}^+ = (|\tilde{\mathbb{V}}| + \tilde{\mathbb{V}})/2$, and $\tilde{\mathbb{V}}^- = (|\tilde{\mathbb{V}}| - \tilde{\mathbb{V}})/2$.

Algorithm Workflow

Algorithm 1 PFNMF

Input: Data samples $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{m \times n}$ and α;

Output: The basis matrix U and coefficient matrix H.

for $i = 1$ to n **do**

  Compute tangent spaces $T_{x_i}\mathcal{M}$ for each data sample by performing PCA on neighborhood of x_i;

end for

Construct matrix B according to (2);

Do eigen-decomposition on (3) to estimate $\tilde{\mathbb{V}}$;

// Iteratively optimize PFNMF model

while not converged **do**

  Update U according to (7) with H fixed;

  Update H according to (8) with U fixed;

end while

Experimental Settings

- We investigate the effectiveness of PFNMF on image clustering. We set the parameter K to be the number of clusters and use the obtained coefficient matrix H to determine the cluster label of each data point.
- Evaluation metrics are
  - ACCuracy (ACC)
  - Normalized Mutual Information (NMI)
- Date Sets are

Table 1: Properties of the used data sets.

dataset	#size	#dimensionality	#class
COIL20	1440	1024	20
ORL	400	4096	40
PIE	2856	1024	68

Clustering results on COIL20

K	Accuracy (mean±std-dev%)			
	Kmeans	NMF	GNMF	PFNMF
6	78.07±12.16	75.53±11.87	86.66±12.67	**92.82±8.20**
8	72.96±10.23	72.06±9.64	90.43±7.75	**95.29±6.48**
10	68.87±6.17	68.89±8.88	81.41±7.58	**87.44±6.51**
12	68.20±4.02	67.87±5.45	79.13±5.96	**86.17±5.32**
14	67.11±5.65	66.67±4.60	82.49±4.61	**84.67±4.09**
16	65.23±4.42	65.56±4.69	79.09±4.10	**80.90±3.72**
18	62.74±3.85	63.15±3.65	78.97±3.49	**80.41±4.29**
20	60.49	58.33	80.69	**85.14**
Avg.	69.75	68.65	83.51	**87.65**
K	Normalized Mutual Information (mean±std-dev%)			
	Kmeans	NMF	GNMF	PFNMF
6	74.83±12.79	71.99±11.91	87.91±8.84	**92.54±7.75**
8	74.23±7.65	72.02±7.57	91.29±5.72	**95.36±5.23**
10	72.68±5.95	71.72±7.49	86.80±5.07	**90.41±4.68**
12	73.22±3.25	72.36±3.95	86.83±3.43	**90.85±3.41**
14	74.19±3.93	72.92±3.75	89.29±2.80	**91.33±2.51**
16	73.88±2.71	72.61±3.38	88.36±2.10	**89.73±2.58**
18	73.25±2.54	72.28±2.41	88.36±1.57	**89.50±1.61**
20	73.86	71.51	89.12	**90.50**
Avg.	73.91	72.02	88.70	**91.64**

Clustering results on ORL

K	Accuracy (mean±std-dev%)			
	Kmeans	NMF	GNMF	PFNMF
8	66.68±8.74	67.06±7.18	70.56±7.28	**72.63±10.14**
12	62.58±6.58	63.50±6.81	68.42±7.26	**69.67±6.51**
16	59.94±5.91	59.53±4.67	64.63±6.11	**67.69±5.18**
20	57.30±4.92	54.75±4.99	62.47±4.05	**65.45±5.65**
25	56.18±4.04	53.98±4.29	61.38±3.25	**65.10±4.15**
30	55.63±3.33	52.23±3.29	59.33±2.91	**61.83±3.34**
40	53.50	49.75	59.75	**62.25**
Avg.	64.71	62.09	68.99	**71.46**
K	Normalized Mutual Information (mean±std-dev%)			
	Kmeans	NMF	GNMF	PFNMF
8	70.97±7.31	69.86±6.91	72.94±7.25	**75.37±7.78**
12	71.94±5.16	71.79±5.28	75.50±4.63	**76.99±5.13**
16	70.45±4.68	70.01±3.80	73.48±4.62	**76.04±3.93**
20	70.59±3.86	68.55±3.17	74.76±3.01	**76.99±3.54**
25	71.13±2.88	69.50±2.82	74.62±2.16	**77.50±2.79**
30	71.33±2.05	69.98±2.65	74.20±1.99	**76.01±2.11**
40	71.42	68.92	75.70	**78.23**
Avg.	71.65	68.09	74.91	**77.07**

Clustering results on PIE

K	Accuracy (mean±std-dev%)			
	Kmeans	NMF	GNMF	PFNMF
10	31.15±4.00	56.62±5.16	85.01±9.28	**91.39±7.38**
20	27.20±1.99	55.32±4.46	80.69±6.20	**83.72±5.64**
30	26.35±1.29	55.87±2.18	81.60±3.08	**84.19±3.51**
40	25.34±1.31	55.96±3.37	77.51±3.84	**79.81±3.18**
50	24.62±1.08	55.27±2.10	76.59±3.90	**78.83±4.41**
60	24.17±1.10	55.64±2.42	74.69±2.89	**76.34±3.29**
68	24.54	56.79	70.52	**73.01**
Avg.	26.20	55.92	78.09	**81.04**
K	Normalized Mutual Information (mean±std-dev%)			
	Kmeans	NMF	GNMF	PFNMF
10	37.75±6.40	66.22±2.88	89.53±5.28	**91.16±4.23**
20	44.24±2.30	72.77±2.88	88.85±2.86	**90.73±3.10**
30	48.19±2.05	76.47±1.04	89.85±1.25	**91.12±1.81**
40	50.08±1.68	78.20±1.34	88.86±1.53	**90.87±1.57**
50	51.33±1.40	78.75±0.85	88.67±1.46	**90.54±1.35**
60	52.67±1.21	80.36±0.99	88.36±0.92	**90.18±1.17**
68	53.77	80.18	87.21	**89.96**
Avg.	48.29	76.14	88.76	**90.65**

Basis Vectors

(a) Original images. (b) Basis by NMF. (c) Basis by PFNMF.

Figure 1: Basis vectors learned from the ORL data set. Large values are illustrated with white pixels.

Conclusion

- We have presented a novel method for matrix factorization algorithm to enforce the second order smoothness of data representation called PFNMF. Experiments shows that our proposed method performs better than other comparison methods in image clustering.
- Such regularizer is a general framework and can be incorporated into other models.

References

For comprehensively understanding the vector field theory, please refer to

- B. Lin, C. Zhang, X. He. Semi-supervised regression via parallel field regularization. NIPS 2011.
- B. Lin, X. He, C. Zhang, M. Ji. Parallel vector field embedding. JMLR 2013.

- Thanks.
- If you want to know more about the derivation, please send me an email yongpeng@hdu.edu.cn