Non-Negative Matrix Factorization with Sinkhorn Distance

Wei Qian[†] **Bin Hong**[†] **Deng Cai**[†] **Xiaofei He**[†] **Xuelong Li**[‡]
[†]State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, China
{qwqjzju, hongbinzju, dengcai}@gmail.com xiaofeihe@cad.zju.edu.cn
[‡]Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, China
xuelong_li@opt.ac.cn

Abstract

Non-negative Matrix Factorization (NMF) has received considerable attentions in various areas for its psychological and physiological interpretation of naturally occurring data whose representation may be parts-based in the human brain. Despite its good practical performance, one shortcoming of original NMF is that it ignores intrinsic structure of data set. On one hand, samples might be on a manifold and thus one may hope that geometric information can be exploited to improve NMF's performance. On the other hand, features might correlate with each other, thus conventional L_2 distance can not well measure the distance between samples. Although some works have been proposed to solve these problems, rare connects them together. In this paper, we propose a novel method that exploits knowledge in both data manifold and features correlation. We adopt an approximation of Earth Mover's Distance (EMD) as metric and add a graph regularized term based on EMD to NMF. Furthermore, we propose an efficient multiplicative iteration algorithm to solve it. Our empirical study shows the encouraging results of the proposed algorithm comparing with other NMF methods.

1 Introduction

Data representation plays a fundamental role in various problems in the fields of pattern recognition, information retrieval and computer vision [Bengio *et al.*, 2013]. A good representation can significantly improve the performance of algorithms. Nowadays, it is easy to collect massive data with lots of features for the problems, which leads to large-scale high-dimensional data set. However, the intrinsic degrees of freedom could be far less. Naturally, one might hope to find the space where the data can be represented by a small number of semantic concepts. To achieve this goal, a large number of methods were proposed in the last decades, among which matrix factorization based methods [Wall *et al.*, 2003; Srebro *et al.*, 2004] have received considerable attentions.

Since its psychological and physiological interpretation of naturally occurring data, Non-negative Matrix Factorization (NMF) [Lee and Seung, 1999] is especially striking among these methods. Different from other methods, NMF uses non-negativity constraints and these constraints lead to a parts-based representation, which may be as same as the form that data is represented in human brains[Lee and Seung, 1999]. It has been shown to be superior to other matrix factorization based methods in various applications, such as face recognition [Guan *et al.*, 2012] and document clustering [Shahnaz *et al.*, 2006; Xu *et al.*, 2003]. One shortcoming of the original NMF is that it ignores the intrinsic structure of the data set. The problem is twofold. First, samples might be on a manifold and thus one may hope this information can be exploited to improve NMF's performance[Cai *et al.*, 2011]. Second, features might correlate with each other, thus conventional L_2 distance can not well measure the distance between samples [Sandler and Lindenbaum, 2011].

To address this limitation, researchers propose to improve NMF in different ways. [Cai *et al.*, 2011] uses a nearest neighbor graph to model the local geometry structure and aims at finding a matrix factorization that preserves the graph structure. Therefore, the proposed GNMF keeps the local structure on the data manifold. Instead of constructing a local connection graph, EMD NMF [Sandler and Lindenbaum, 2011] uses Earth Mover's Distance (EMD) [Rubner *et al.*, 2000] (also known as Wasserstein distance) to measure the difference between original data matrix and the product matrix. EMD is defined as the cost of the optimal transport plan for moving the mass between two histograms. It has been applied to a wide range of problems, including label propagation [Solomon *et al.*, 2014], and supervised learning [Frogner *et al.*, 2015]. EMD is sensitive to relationships between the different dimensions and agrees with perceptual dissimilarity better than other measures[Rubner *et al.*, 2000]. Thus EMD NMF achieves more robust results over traditional L_2 NMF for problems where the error mechanism follows complex local deformation[Sandler and Lindenbaum, 2011].

However, the computation of EMD NMF is time consuming, thus limits its scalability. Even though the wavelet EMD (WEMD)[Shirdhonkar and Jacobs, 2008], which is an accelerated approximation of EMD, is employed to accelerate optimization, EMD NMF is still slow. For a data matrix with 200 samples and 1024 features, one full iteration of WEMD based EMD NMF takes around 23 minutes and the algorithm converges after 4 hours by using Matlab on an Intel Core 2 Quad 2.5GHz processor[Shirdhonkar and Jacobs, 2008]. Moreover,

for the WEMD-based EMD NMF algorithm, the factorization results may not be non-negative since only soft penalty is used to encourage non-negativity.

In this paper, we propose a novel NMF algorithm which exploits information in both data manifold and features correlation. Inspired by the recent progress on EMD [Cuturi, 2013; Frogner *et al.*, 2015], we propose to model the relationship between different feature dimensions by unnormalized Wasserstein distance with entropic regularization (also named as Sinkhorn distance in short), whose gradient can be solved efficiently. Meanwhile, graph regularization based on this distance is incorporated to preserve local geometry structure. Our model can capture both manifold structure and features correlation and is named as Non-negative Matrix Factorization with Sinkhorn distance (SDNMF). We further propose an efficient multiplicative update algorithm for the proposed NMF model. The non-negative factorization can then be efficiently computed. Experimental results on real-world data sets demonstrate the effectiveness of our model and the efficiency of our proposed multiplicative update algorithm.

The rest of the paper is organized as follows: section 2 provides a brief review of EMD and its entropic regularization. Our SDNMF model and multiplicative update rules are introduced in section 3. Experiments on two real-world data sets are presented in section 4. Related work is presented in section 5. Finally, we provide some concluding remarks in section 6.

2 Preliminary

Notation. Here we briefly introduce the notations used in this paper. We use italic uppercase letters to denote matrices, bold lowercase letters to denote vectors. Given an arbitrary $m\times n$ matrix A,we define $\mathbf{a}_j$ and a_{ij} as the j-th column vector and the (i,j)-th entry of matrix A respectively. $\odot$ represents element-wise multiplication and $\oslash$ represents element-wise division.

2.1 Earth Mover's Distance

Given two normalized histograms $\mathbf{x},\mathbf{y} \in \mathcal{R}^m(\sum_t x_t = \sum_t y_t = 1)$, and a distance metric matrix M, the Earth Mover's Distance $d_M(\mathbf{x},\mathbf{y})$ [Rubner *et al.*, 2000] is defined as:

$$\begin{aligned} d_M(\mathbf{x},\mathbf{y}) &= \min_{T_{pq}\geq 0} \sum_{p,q=1}^{m} M_{pq}T_{pq} \qquad (1)\\ s.t. &\quad \sum_{q=1}^{m} T_{pq} = x_p, \sum_{p=1}^{m} T_{pq} = y_q, \forall p,q \end{aligned}$$

Eq. (1) is the well-known transportation problem and the flow variable T_{pq} denotes the quantity transported from the p-th supply to the q-th demand. The parameter M_{pq} represents the ground distance between bins p and q. Usually M_{pq} is defined by L_1 or L_2 distance or is determined based on the priori knowledge of the features in the considered problem.

However, the computation of EMD is time consuming. [Cuturi, 2013] proposes a method to accelerate its computation. They smooth the classical optimal transportation problem with an entropic regularization term and show that the resulting optimum is also a distance which can be computed through Sinkhorn-Knopps matrix scaling algorithm at a speed that is several orders of magnitude faster than that of transportation solvers. They call the new distance as Sinkhorn distance and the the specific form is:

$$\begin{aligned} d_M^{\lambda}(\mathbf{x},\mathbf{y}) &= \min_{T_{pq}\geq 0} (\sum_{p,q} M\odot T + \frac{1}{\lambda}H(T)), \qquad (2)\\ s.t. &\quad T\mathbf{1} = \mathbf{x}, T^T\mathbf{1} = \mathbf{y} \end{aligned}$$

where $H(T) = -\sum_{p,q} T_{pq}\log T_{pq}$ is the entropy of T. It can approximate the exact EMD closely with λ large enough.

Moreover, [Frogner *et al.*, 2015] proposes a relaxation that extends the smoothed transport to unnormalized measures. They replace the equality constraints on the transport marginals in Eq. (2) with soft penalties with respect to KL divergence and obtain an unconstrained approximate transport problem. The distance becomes:

$$\begin{aligned} d_M^{\lambda,\gamma}(\mathbf{x},\mathbf{y}) &= \min_{T_{pq}\geq 0} \{\sum_{p,q} M\odot T + \frac{1}{\lambda}H(T) \qquad (3)\\ &+ \gamma(\widetilde{KL}(T\mathbf{1}||\mathbf{x}) + \widetilde{KL}(T^T\mathbf{1}||\mathbf{y}))\} \end{aligned}$$

where $\widetilde{KL}(\mathbf{w}||\mathbf{z}) = \mathbf{w}^T log(\mathbf{w}\oslash\mathbf{z}) - \mathbf{1}^T\mathbf{w} + \mathbf{1}^T\mathbf{z}$ is the generalized KL divergence between $\mathbf{w}$ and $\mathbf{z}$. It degenerates to Eq. (2) with γ large enough when $\sum_t x_t = \sum_t y_t = 1$.

Eq. (2), Eq. (3) and their gradients can be efficiently solved by Sinkhorn-like iterations [Frogner *et al.*, 2015].

3 SDNMF

Given a data matrix $X = [x_{ij}] = [\mathbf{x}_1, ..., \mathbf{x}_n] \in \mathbb{R}^{m\times n}$, where each column is a sample vector, NMF tries to find two non-negative matrices $U = [u_{ik}] = [\mathbf{u}_1, ..., \mathbf{u}_t] \in \mathbb{R}^{m\times t}$ and $V = [v_{jk}] = [\mathbf{v}_1, ..., \mathbf{v}_t] \in \mathbb{R}^{n\times t}$ such that $Y = [y_{ij}] = [\mathbf{y}_1, ..., \mathbf{y}_n]$ is an approximation of origin data matrix X under the new data representation $\mathbf{y}_j = \sum_{k=1}^{t} \mathbf{u}_k v_{jk}$. Previous works usually assume the error between X and Y is due to Gaussian error. However, this assumption neglects the relationship between different dimensions of features. Considering the case where we have two documents represented by Bag-of-Words which have the same topic "soccer", one is represented as $\{\text{"}soccer\text{"} : 100, \text{"}football\text{"} : 0, ...\}$ and the other is represented as $\{\text{"}soccer\text{"} : 0, \text{"}football\text{"} : 100, ...\}$. It does not affect their topics if we change the count of "soccer" and the count of "football" as long as we keep the count of "soccer" and "football". Noted that the topic vector "soccer" produced by NMF will probably have nonzero count on both of them. It causes that the new representation produced by NMF will have nonzero count on both of them and the error of NMF increases. Obviously, this kind of error can not be modeled by Gaussian error. Therefore, we adopt EMD as the metric. Optimizing with respect to the exact EMD is costly and it's difficult to ensure $\sum_i y_{ij} = \sum_i x_{ij}$ in the whole iterative solving process. Eq. (3) describes a regularized approximation which can be efficiently computed even with unnormalized data. So we change the objective function as:

$$\mathcal{O} = \sum_{j=1}^{n} d_M^{\lambda,\gamma}(\mathbf{x}_j, \mathbf{y}_j) \qquad (4)$$

In addition to the correlation of features, there exists a geometry structure in the sample space. We hope that the information of the geometric structure of samples can be exploited for better discovery of the basis $[\mathbf{u}_1, ..., \mathbf{u}_t]$. Inspired by [Cai *et al.*, 2011], we also use the manifold assumption on the data. This assumption can be explained as that if two data points $\mathbf{x}_j$, $\mathbf{x}_s$ are close in the intrinsic geometry of the data manifold, then $\mathbf{v}_j$ and $\mathbf{v}_s$, the representations of these two points in the new basis, are also close to each other. Formally, it can be written as:

$$\min_V \mathcal{R} = \sum_{j,s}^{n} W_{js} ||\mathbf{v}_j - \mathbf{v}_s||^2 \tag{5}$$

where W is a nearest neighbor graph on a scatter of data points. Different from [Cai *et al.*, 2011], our nearest neighbor graph is constructed based on Sinkhorn distance.

Combining both of them, we get our final objective function:

$$\mathcal{O} = \sum_{j=1}^{n} d_M^{\lambda,\gamma}(\mathbf{x}_j, \mathbf{y}_j) + \frac{\xi}{4}\mathcal{R} \tag{6}$$

where ξ is a regularization parameter that controls the trade-off between the features correlation and the data manifold.

3.1 Multiplicative Update Rules

The objective function $\mathcal{O}$ of SDNMF in Eq. (6) is not jointly convex in U and V. Therefore it is impractical to expect an algorithm to seek the global minimum of $\mathcal{O}$. Fortunately, the objective function $\mathcal{O}$ is convex in U and V separately. Similar to [Lee and Seung, 2001], we also adopt a two-stage multiplicative update rules which can keep non-negativity and find a local minimum:

$$u_{ik} \leftarrow u_{ik} \frac{\sum_s v_{sk} \frac{\sum_t T_s^{*it}}{y_{is}}}{\sum_s v_{sk}} \tag{7}$$

$$v_{jk} \leftarrow v_{jk} \frac{\sum_s u_{sk} \frac{\sum_t T_j^{*st}}{y_{sj}} + \xi \sum_{s \neq j} W_{js} v_{js}}{\sum_s u_{sk} + \xi v_{jk} \sum_{s \neq j} W_{js}} \tag{8}$$

where T_s^{*it} is the (i,t)-entry of the optimal transportation matrix between $\mathbf{x}_s$ and $\mathbf{y}_s$ and can be solved by a slight variation of algorithm 1 in [Frogner *et al.*, 2015] efficiently. What's more, the following theorem guarantees that the update rules of U and V in Eq. (7) and (8) converge and the final solution will be a local optimum. A detailed proof can be found in the appendix.

Theorem 1. The objective function $\mathcal{O}$ in Eq. (6) is non-increasing under the update rules in Eq. (7) and (8). The objective function is permanent under these updates if and only if U and V are at a stationary point.

4 Experiments

In this section, we test the performance of the proposed algorithm in the context of two challenging tasks. Previous studies show that NMF is very powerful on clustering, especially in the document clustering [Shahnaz *et al.*, 2006; Xu *et al.*, 2003] and image clustering tasks [Guan *et al.*, 2012]. For Image clustering, previous works usually align images first. But in the real world, images might be too complex to be aligned perfectly. There may still exist local deformation after alignment. In the first experiment, we use small random translation to simulate local deformation and show that our method is more robust than previous methods. In the second experiment, we use NMF to estimate texture descriptors for texture mosaic images and show that our method has a strong ability of description for images with high local variability.

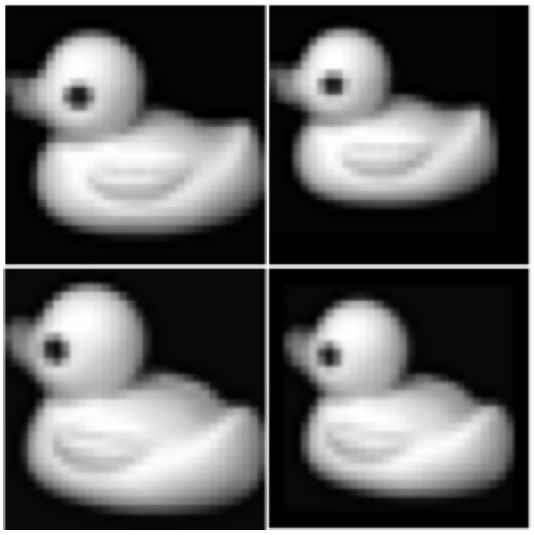

Figure 1: Examples of images with translation noise. Left images are original images and right images are generated with $\delta = 4$. The L_2 distance between right images is larger than that between left images.

4.1 Image Clustering with Translation Noise

Data Set. The data set used in this section is COIL20 image library [Nene *et al.*, 1996], which contains well aligned 32×32 gray scale images of 20 objects viewed from 72 varying angles. However images might be too complex to be aligned perfectly in the real world. There will still exist local deformation after alignment. We use small random translation to simulate local deformation and test our method on the new data set to show its robustness. Specifically, our data sets are generated by:

1. Resizing each image to $(32 - \delta) \times (32 - \delta)$ and placing it at the center of a 32×32 blank image, where δ is a small integer that controls the degree of random translation noise.
2. Generating a random integer vector (i, j), where i and j are sampled from a discrete uniform distribution on $\{-\delta, ..., \delta\}$ independently.
3. Translating the resized image with a vector (i, j).

Some examples are shown in Figure 1.

Compared Algorithms. We compare our method with the following four popular clustering algorithms on data sets generated with $\delta = 0, 1, 2, 3, 4$ respectively. For each compared

Table 1: Clustering performance on COIL20

δ	Accuracy(%)					Normalized Mutual Information(%)				
	Kmeans	NMF	GNMF	EMDNMF	SDNMF	Kmeans	NMF	GNMF	EMDNMF	SDNMF
0	59.98±3.89	58.85±2.82	**79.59±2.95**	58.89±3.43	78.16±4.35	71.37±1.23	69.72±1.53	88.32±1.31	69.06±1.58	**88.77±1.46**
1	50.16±3.43	48.77±2.37	51.12±2.99	45.64±4.35	**52.32±3.17**	64.20±2.00	59.96±1.70	69.24±2.18	57.95±2.98	**70.08±1.99**
2	33.78±0.82	31.67±1.74	47.53±1.97	30.49±1.73	**52.78±0.91**	48.78±1.03	44.17±2.13	62.60±1.90	41.57±1.89	**65.88±0.85**
3	24.77±1.11	22.98±1.28	33.27±1.63	22.74±1.33	**39.92±1.78**	35.79±0.78	32.29±1.09	46.60±0.91	31.42±0.93	**53.66±0.71**
4	19.55±0.80	19.52±1.06	21.31±0.77	18.70±0.88	**23.15±1.56**	27.88±1.24	26.48±1.08	29.79±1.12	24.75±0.51	**34.28±1.61**

method, several parameter configurations are tested and the best performance is reported.

- Canonical K-means clustering method (Kmeans in short).
- Non-negative Matrix Factorization based clustering (NMF in short) [Lee and Seung, 2001].
- Graph Regularized Non-negative Matrix Factorization (GNMF in short) with Frobenius norm formulation. Following [Cai *et al.*, 2011], we use binary weighting scheme for constructing the 5-nearest neighbor graph and set the regularization parameter λ to 100.
- Non-negative Matrix Factorization with Earth Mover's Distance (EMD NMF in short). Since the WEMD algorithm in [Sandler and Lindenbaum, 2011] is too slow (the detailed performance will be discussed later) and can not ensure its result is non-negative, we use our efficient algorithm to solve it[1]. We set the λ and γ to 100 and 10 respectively and use the 2D distance of pixels' location of the image as the ground metric.
- Non-negative Matrix Factorization with Sinkhorn Distance (SDNMF in short). We use binary weighting scheme for constructing the 5-nearest neighbor graph for its simplicity. We set the λ , γ and ξ to 100,1 and 10 respectively and use the 2D distance of pixels' location of the image as the ground metric.

We evaluate the clustering performance by comparing the obtained label of each sample with the label provided by the data set. Two metrics, the accuracy (AC) and the normalized mutual information (NMI) are used to measure the clustering result. The detailed definitions of these two metrics can be found in [Cai *et al.*, 2005].

Clustering Results. Table 1 shows the clustering results on the COIL2 data sets with different levels of random translation noise. For each given random size δ, 10 test runs were conducted on different randomly chosen clusters. The table reports the mean and the standard error of the performance.

It can be seen from Table 1 that our method has the best performance in most cases. When the size of random noise is increasing, our method has the most robust performance.

Recall that for a data matrix with 200 samples and 1024 features, one full iteration of the algorithm in [Sandler and Lindenbaum, 2011] takes around 23 minutes and the algorithm converges after 4 hours by using Matlab on an Intel Core 2 Quad 2.5 GHz processor. Our algorithm needs only 2.9 seconds per full iteration and converges after 290.9 seconds by using Matlab on an Intel i7-4790K 4.0GHz processor on a data set with 1440 samples and 1024 features. It demonstrates the efficiency of our algorithm.

[1]Setting ξ to zero, our objective function Eq. (6) is another alternative approximation of EMD NMF

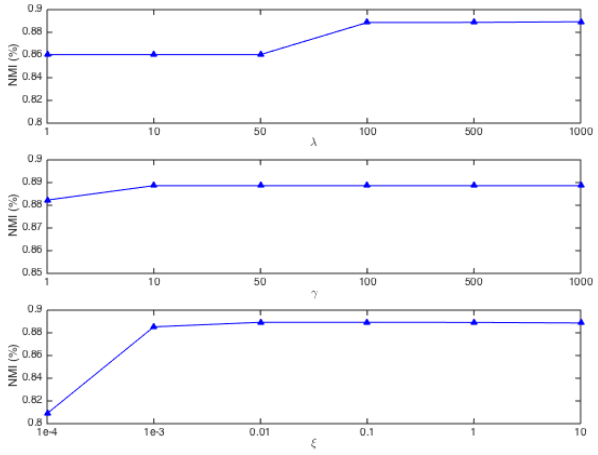

Figure 2: The performance of SDNMF vs. different parameters on original COIL20 data set.

Parameters Selection. Our SDNMF model has three essential parameters: the regularization parameter of Sinkhron distance λ, the relaxation parameter of unnormalized Sinkhorn distance γ and the trade-off parameter ξ. Figure 2 shows how the average performance of SDNMF varies with these parameters. As we can see, the performance of SDNMF is very stable with respect to these parameters. SDNMF achieves consistently good performance when these parameters varies in a large range.

4.2 Texture descriptor estimation

In this section we do the same experiment in section 6.2 of [Sandler and Lindenbaum, 2011]. Our task is estimating the texture descriptors associated with each texture class of the mosaic from [Haindl and Mikeš, 2008] (some examples are shown in Figure 3). Moreover, roughly classification of the textures in each mosaic location (e.g., for consecutive segmentation) is also expected. [Haindl and Mikeš, 2008] contains online generated 512×512 mosaics with different numbers of textures and the number can be chosen from 3 to 12. For each number, we choose 9 samples to make up our data set. For our task, we consider the texture in non-overlapping square image patches and assume the texture in each block is a positive mixture of the basic textures. Then NMF is used to analyse it.

For each texture, we assume there exists a vector descrip-

Figure 3: Examples of texture mosaics. Mosaics involve several types of textures in random arrangements and textures have high local variability.

tor $\mathbf{u}_i^{true}$ connected with it. Therefore, a mosaic with C textures is connected with a texture descriptor matrix $U^{true} = [\mathbf{u}_1^{true}, ..., \mathbf{u}_{\mathrm{C}}^{true}]$. We expect that the texture descriptor in the j-th image patch should be $\mathbf{x}_j = U^{true}\mathbf{v}_j^{true}$, where $\mathbf{v}_j^{true}$ is the vector of true fractions of the j-th block area associated with each texture class. Then we can use NMF to roughly estimate texture descriptors and classify the textures in each mosaic location.

The specific process is:

1. Converting the image to a new representation which each location is represented by a vector of Gabor responses, since Gabor filters have been widely used in texture analysis [Ramakrishnan *et al.*, 2002].
2. Dividing the image into N non-overlapping patches and computing the mean feature vector $\mathbf{x}_j$ for each patches. Now, $X = [\mathbf{x}_1, ..., \mathbf{x}_{\mathrm{N}}]$ is a new representation of the image.
3. Using NMF to find the factorization $X \approx UV^T$.

The factorization results $U = [\mathbf{u}_1, ..., \mathbf{u}_{\mathrm{C}}]$ and $V = [\mathbf{v}_1, ...\mathbf{v}_{\mathrm{C}}]$ are the approximated representative texture descriptors and the approximated fraction of each texture in each patch respectively.

We conduct the tests with different number of patches N = 16, 64, 256, 1024 and these patches tessellate the image. Thus, N determines the patch size $128\times128, 64\times64, 32\times32$, and 16×16 pixels respectively. The U^{true} consists of the mean descriptors on the single texture segments of the mosaic and the V^{true} consists of the fractions of each class in the patch. We use average correlation between the true representation and the approximated one to evaluate the performance. This metric can be written as:

$$Q(W, W^{true}) = \frac{1}{\mathrm{C}} \sum_{i=1}^{\mathrm{C}} \frac{< \mathbf{w}_i, \mathbf{w}_i^{true} >}{||\mathbf{w}_i||||\mathbf{w}_i^{true}||}$$

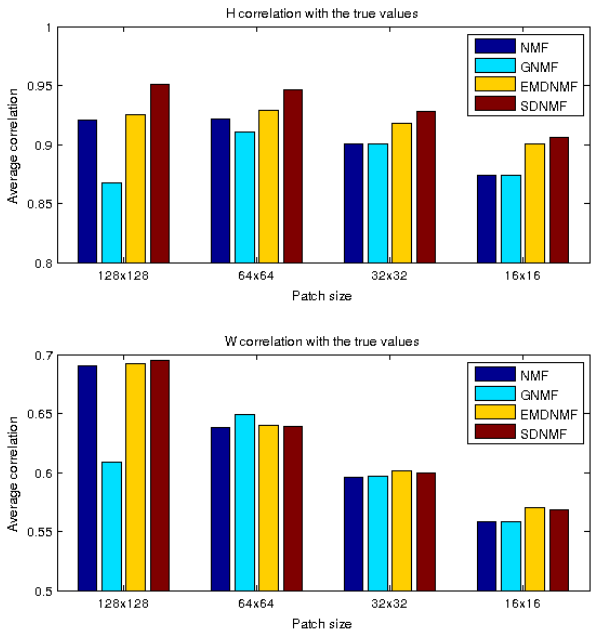

Figure 4: The performance of texture descriptor estimation

Figure 4 shows the performance of texture descriptor estimation. We still use NMF, GNMF and EMDNMF as compared algorithms. It can be seen from Figure 4 that our method has the best performance in most cases. Since textures exhibit lots of spatial variation, EMD might be preferable over L_2. Therefore, EMD based method will be more robust when the variation increasing (patch size decreasing). What's more, L_2 based graph regularizer will damage the performance.

5 Related Work

In the past decade, based on the standard NMF, many variants have been proposed to find effective data representation for various problems. The family of NMF algorithms have been applied in many areas, such as face recognition[Guan *et al.*, 2012], clustering[Xu *et al.*, 2003], hyperspectral unmixing[Jia and Qian, 2009], etc., and achieve remarkable performance.

The works that are closely related to our method could be roughly categorized into two lines. First, various difference metrics are adopted to measure the dissimilarity between the original data matrix and the product. [Kompass, 2007] proposes a generalized divergence measure that interpolates between square loss and Kullback-Leibler divergence for non-negative matrix factorization. [Févotte *et al.*, 2009] uses Itakura-Saito divergence to incorporate Bayesian priors. However, none of these works model the features correlation explicitly and they may fail to capture the relationships of different dimensions. [Sandler and Lindenbaum, 2011] first proposes to use Earth Mover's Distance as difference metric, and employs wavelet EMD approximation to derive practical algorithm. Since EMD is sensitive to the feature correlation

and agrees with the perceptual dissimilarity better than other measures[Rubner *et al.*, 2000], EMD NMF achieves more robust results for problems where the error mechanism follows complex local deformation. However, the WEMD based EMD NMF is still computationally costly. In contrast, both the Sinkhorn Distance that is considered in our method and its gradient can be computed efficiently by Sinkhorn-like matrix scaling algorithm.

Second, considering the local geometry structure of the data manifold, [Cai *et al.*, 2011] constructs a nearest neighbor graph and finds a matrix factorization that preserves this graph structure. The proposed GNMF can keep the local structure on the data manifold, thus achieves superior results on clustering. GNMF is further extended to problems like co-clustering[Shang *et al.*, 2012], by adding more constraints. A semi-supervised NMF method is also proposed in [Liu *et al.*, 2012] to incorporate available labeled samples, which increases the discriminative ability of the obtained representation. However, the local graphs in these works are based on conventional L_2 distance or Kullback-Leibler divergence, which can not reflect the feature correlation. Meanwhile, currently we only consider the case without supervision.

6 Conclusions and Future Work

In this paper we propose a new NMF method, SDNMF, which exploits knowledge in both data manifold and features correlation. We use EMD to utilize information of feature correlation and use a graph regularizer to keep the local geometric structure of the data manifold. Although optimizing with respect to the exact EMD objective function is computationally costly, our approximate Sinkhorn distance objective function is efficiently computed. By this approximation, we propose a fast implementation of the proposed algorithm. SDNMF outperforms previous NMF based algorithms in the context of two challenging computer vision tasks.

Although our multiplicative update algorithm is efficient, SDNMF has to pay high computational cost in graph construction and subsequent matrix calculation, which limits its applicability to large-scale problems. Inspired by [Liu *et al.*, 2010] an interesting direction for future work may be to use landmark points to speed up our algorithm.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, National Natural Science Foundation of China (Grant nos. 61233011, 61125203), and National Youth Top-notch Talent Support Program.

A (Proofs of Theorem 1)

To prove Theorem 1, we will imitate the proof process from [Lee and Seung, 2001] and use its Definition 1 and Lemma 1. Please see [Lee and Seung, 2001] for details.

Definition 1. $G(h,h')$ is an auxiliary function for $F(h)$ if the conditions

$$G(h,h') \geq F(h), G(h,h) = F(h)$$

are satisfied.

Lemma 1. If G is an auxiliary function of F, then F is non-increasing under the update

$$h^{t+1} = arg\min_h G(h,h^t)$$

For simplicity, we only prove that $\mathcal{O}$ is non-increasing under the update step in Eq. (8) (the other half can be proved in similar way). Fixed U, we rewrite the objective function $\mathcal{O}$ as follows

$$F(V) = \sum_j d_M^{\lambda,\gamma}(\sum_k v_{jk}\mathbf{u}_k, \mathbf{x}_j) + \frac{\xi}{4}\sum_{j,s,k} W_{js}(v_{jk}-v_{sk})^2$$

According to Definition 1 and Lemma 1, we need to find an auxiliary function $G(V,V')$ for it.

Lemma 2. Setting $\alpha_{ik_i} = \frac{u_{ik}v_{jk}^{(q)}}{\sum_k u_{ik}v_{jk}^{(q)}}$ and $\boldsymbol{\alpha} = [\alpha_{1k_1},...,\alpha_{mk_m}]^T$. Then function

$$\begin{aligned} &G(V,V^{(q)}) \\ &= \sum_{j,k_1,...,k_m}\prod_i \alpha_{ik_i} d_M^{\lambda,\gamma}((\sum_k v_{jk}\mathbf{u}_k)\oslash\boldsymbol{\alpha}, \mathbf{x}_j) \\ &+\frac{\xi}{4}\sum j,s,kW_{js}(v_{jk}-v_{sk})^2 \end{aligned}$$

is an auxiliary function for $F(V)$.

Proof. It is straightforward to verify that $G(V,V) = F(V)$. Since $d_M^{\lambda,\gamma}$ is convex [Frogner *et al.*, 2015], we use the convexity for $i = 1,...,m$ one by one and obtain

$$\begin{aligned} &\sum_{k_1,...,k_m}\prod_i \alpha_{ik_i} d_M^{\lambda,\gamma}((\sum_k v_{jk}\mathbf{u}_k)\oslash\boldsymbol{\alpha}, \mathbf{x}_j) \\ &\geq d_M^{\lambda,\gamma}(\sum_k v_{jk}\mathbf{u}_k, \mathbf{x}_j) \\ &\Rightarrow G(V,V^{(q)}) \geq F(V) \end{aligned}$$

Thus we find a proper function $G(V,V^{(q)})$. □

Then we can prove Theorem 1:

Proof of Theorem 1. Note that the gradient of $d_M^{\lambda,\gamma}(\mathbf{y},\mathbf{x})$ with respect to $\mathbf{y}$ is: [Frogner *et al.*, 2015]

$$\nabla_{\mathbf{y}} d_M^{\lambda,\gamma}(\mathbf{y},\mathbf{x}) = \gamma(\mathbf{1} - T * \mathbf{1}\oslash\mathbf{y}).$$

By setting the gradient of $G(V,V^{(q)})$ with respect to V to zero, we can find $V^{(q+1)}$.

$$\begin{aligned} &\frac{\partial G(V,V^{(q)})}{v_{jk}} = 0 \\ \Rightarrow\quad & v_{jk}^{(q+1)} = v_{jk}^{(q)}\frac{\sum_s u_{sk}\frac{\sum_t T_j^{*st}}{y_{sj}} + \xi\sum_{s\neq j} W_{js}v_{js}^{(q)}}{\sum_s u_{sk} + \xi v_{jk}^{(q)}\sum_{s\neq j} W_{js}} \end{aligned}$$

According to Definition 1 and Lemma 1, Theorem 1 is proved □

References

[Bengio *et al.*, 2013] Yoshua Bengio, Aaron Courville, and Pierre Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

[Cai *et al.*, 2005] Deng Cai, Xiaofei He, and Jiawei Han. Document clustering using locality preserving indexing. *Knowledge and Data Engineering, IEEE Transactions on*, 17(12):1624–1637, 2005.

[Cai *et al.*, 2011] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1548–1560, 2011.

[Cuturi, 2013] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

[Févotte *et al.*, 2009] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.

[Frogner *et al.*, 2015] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, pages 2044–2052, 2015.

[Guan *et al.*, 2012] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. Online nonnegative matrix factorization with robust stochastic approximation. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(7):1087–1099, 2012.

[Haindl and Mikeš, 2008] Michal Haindl and Stanislav Mikeš. Texture segmentation benchmark. In *Pattern Recognition, 19th International Conference on*, pages 1–4. IEEE, 2008.

[Jia and Qian, 2009] Sen Jia and Yuntao Qian. Constrained nonnegative matrix factorization for hyperspectral unmixing. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(1):161–173, 2009.

[Kompass, 2007] Raul Kompass. A generalized divergence measure for nonnegative matrix factorization. *Neural computation*, 19(3):780–791, 2007.

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

[Lee and Seung, 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[Liu *et al.*, 2010] Wei Liu, Junfeng He, and Shih-Fu Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 679–686, 2010.

[Liu *et al.*, 2012] Haifeng Liu, Zhaohui Wu, Xuelong Li, Deng Cai, and Thomas S Huang. Constrained nonnegative matrix factorization for image representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1299–1311, 2012.

[Nene *et al.*, 1996] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). Technical report, Technical Report CUCS-005-96, 1996.

[Ramakrishnan *et al.*, 2002] AG Ramakrishnan, S Kumar Raja, and HV Raghu Ram. Neural network-based segmentation of textures using gabor features. In *Neural Networks for Signal Processing, Proceedings of the 12th IEEE Workshop on*, pages 365–374. IEEE, 2002.

[Rubner *et al.*, 2000] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[Sandler and Lindenbaum, 2011] Roman Sandler and Michael Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric for image analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1590–1602, 2011.

[Shahnaz *et al.*, 2006] Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.

[Shang *et al.*, 2012] Fanhua Shang, LC Jiao, and Fei Wang. Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognition*, 45(6):2237–2250, 2012.

[Shirdhonkar and Jacobs, 2008] Sameer Shirdhonkar and David W Jacobs. Approximate earth movers distance in linear time. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[Solomon *et al.*, 2014] Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *Proceedings of The 31st International Conference on Machine Learning*, pages 306–314, 2014.

[Srebro *et al.*, 2004] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2004.

[Wall *et al.*, 2003] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[Xu *et al.*, 2003] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.