Journal of Information Security, 2012, 3, 245-250
http://dx.doi.org/10.4236/jis.2012.34031 Published Online October 2012 (http://www.SciRP.org/journal/jis)

Unsupervised Multi-Level Non-Negative Matrix Factorization Model: Binary Data Case

Qingquan Sun[1], Peng Wu[2], Yeqing Wu[1], Mengcheng Guo[1], Jiang Lu[1]
[1]Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, USA
[2]School of Information Engineering, Wuhan University of Technology, Wuhan, China
Email: quanqian123@hotmail.com

Received July 21, 2012; revised August 26, 2012; accepted September 7, 2012

ABSTRACT

Rank determination issue is one of the most significant issues in non-negative matrix factorization (NMF) research. However, rank determination problem has not received so much emphasis as sparseness regularization problem. Usually, the rank of base matrix needs to be assumed. In this paper, we propose an unsupervised multi-level non-negative matrix factorization model to extract the hidden data structure and seek the rank of base matrix. From machine learning point of view, the learning result depends on its prior knowledge. In our unsupervised multi-level model, we construct a three-level data structure for non-negative matrix factorization algorithm. Such a construction could apply more prior knowledge to the algorithm and obtain a better approximation of real data structure. The final bases selection is achieved through L_2-norm optimization. We implement our experiment via binary datasets. The results demonstrate that our approach is able to retrieve the hidden structure of data, thus determine the correct rank of base matrix.

Keywords: Non-Negative Matrix Factorization; Bayesian Model; Rank Determination; Probabilistic Model

1. Introduction

Non-negative matrix factorization (NMF) was proposed by Lee and Seung [1] in 1999. NMF has become a widely used technique over the past decade in machine learning and data mining fields. The most significant properties of NMF are non-negative, intuitive and part based representative. The specific applications of NMF algorithm include image recognition [2], audio and acoustic signal processing [3], semantic analysis and content surveillance [4]. In NMF, given a non-negative dataset $V \in R^{M\times N}$, the objective is to find two non-negative factor matrices $W \in R^{M\times K}$ and $H \in R^{K\times N}$. Here W is called base matrix and H is named feature matrix. In addition, W and H satisfy

$$V \approx WH \quad s.t. W \geq 0, H \geq 0 \tag{1}$$

K is the rank of base matrix and it satisfies the inequality $K \leq MN/(M+N)$.

For NMF research, the cost function and initialization problems of NMF are the main issues for researchers. Now the rank determination problem becomes popular. The rank of base matrix is indeed an important parameter to evaluate the accuracy of structure extraction. On the one hand, it reflects the real feature and property of data; on the other hand, more accurate learning could help us get better understanding and analyzing of data, thus improving the performance in applications: recognition [5,6] surveillance and tracking. The main challenge of rank determination problem is that it is pre-defined. Therefore, it is hard to know the correct rank of base matrix before the updating process of components. As the same as the cost function, there are no more priors added to the algorithm in previous methods. That is why the canonical NMF method and traditional probabilistic methods (*ML*, *MAP*) cannot handle the rank determination problem. Therefore in this paper, we propose an unsupervised multi-level model to automatically seek the correct rank of base matrix. Furthermore, we use L_2-norm to show the contribution of hyper-prior in correct bases learning procedure. Experimental results on two binary datasets demonstrate that our method is efficient and robust.

The rest of this paper is organized as follows: Section 2 provides a brief review of related works. In Section 3, we describe our unsupervised multi-level NMF model in details. The experimental results of two binary datasets are shown in Section 4. Section 5 concludes the paper.

2. Related Work

As we mentioned above, rank determination problem is a new popular issue in NMF research. Actually, there are few literatures discussing this issue. Although the author in [7] proposed a method based on sampler selection, it needs to pass through all the possible values of rank of base matrix to choose the best one. Obviously, this method is not impressive enough for unsupervised learning. In [8], the author proposed a rank determination method based on automatic relevance determination. In this method, a parameter is defined relevant to the columns of *W*. Then using *EM* algorithm to find a subset, however, this subset of bases is not accurate to represent true bases. Actually, the nature of this hyper-parameter is to affect the updating procedure of base matrix and feature matrix, thus affect the components' distributions.

The only feasible solution is fully Bayesian models. Such kind of methods have been proposed in [9]. In this paper, the author addresses an *EM* based fully Bayesian algorithm to discover the rank of base matrix. *EM* based methods are an approximation solution. In comparison, a little more accurate solution is Gibbs sampling based methods. Such approach is utilized to find the correct rank in [10]. Although such kinds of methods are flexible, it requires successively calculation of the marginal likelihood for each possible value of each rank *K*. The drawback is too much computation cost involved. Additionally, when such methods are applied to real time application or some large scale dataset based applications, the high computation load is impractical. Motivated by the current condition, we propose a low computation, robust multi-level model for NMF to solve rank determination problem. Our unsupervised model with multi-lever priors only calculate once of the rank of base matrix and is able to successfully find the correct rank of base matrix given a large enough rank *K*. Therefore, our method involves less computation. This will be discussed in details in next section.

3. Unsupervised Multi-Level Non-Negative Matrix Factorization Model

In our unsupervised multi-level NMF model, we introduce a hyper-prior level. Hence, there are three levels in our model: data model, prior model, hyper-prior model. The model structure is shown in **Figure 1**. We will seek

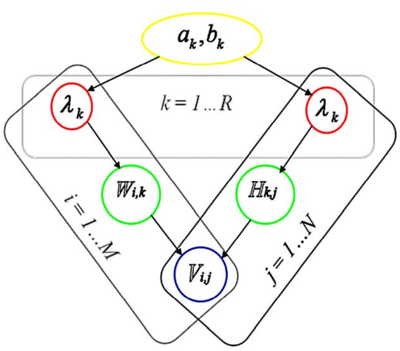

Figure 1. Unsupervised multi-level non-negative matrix factorization model.

the solutions through optimizing the maximum a posterior criterion. Our approach could be depicted by the following equation, here $=^c$ denotes equality up to a constant, $\boldsymbol{\lambda}$ is the prior of both $\boldsymbol{W}$ and $\boldsymbol{H}$.

$$MAP(\boldsymbol{W},\boldsymbol{H},\boldsymbol{\lambda}) =^c \log p(\boldsymbol{V}|\boldsymbol{WH}) + \log p(\boldsymbol{W}|\boldsymbol{\lambda}) + \log p(\boldsymbol{H}|\boldsymbol{\lambda}) + \log p(\boldsymbol{\lambda}) \quad (2)$$

The difference between our approach and the traditional *MAP* criterion is that in traditional one there is no hyper-prior added to the model. Moreover, in our model we attempt to update the hyper-priors recursively, but not just set it as a constant.

3.1. Model Construction

In NMF algorithm, the updating rules are based on the specific data model. Therefore, the first step is to set a data model for our problem. Here, in our experiment we assume that the data follows Poisson distribution. Consequently, the cost function of our model will be generalized *KL*-divergence. So given a variable *x*, which follows *Poisson* distribution with parameter λ, we have $p(x|\lambda) = e^{-\lambda} \times \lambda^x / \Gamma(x+1)$. Thus, in NMF algorithm, given dataset $\boldsymbol{V}$, we have the likelihood

$$p(\boldsymbol{V}|\boldsymbol{WH}) = e^{-\boldsymbol{WH}} (\boldsymbol{WH})^{\boldsymbol{V}} / \Gamma(\boldsymbol{V}+1) \quad (3)$$

The generalized *KL*-divergence is given by:

$$D_{KL}(\boldsymbol{V}|\boldsymbol{WH}) = \sum_{mn} \left(v_{mn} \log\left(\frac{v_{mn}}{[wh]_{mn}} \right) - v_{mn} + [wh]_{mn} \right) \quad (4)$$

Thus, the log-likelihood of the dataset $\boldsymbol{V}$ can be rewritten as:

$$\begin{aligned} &\log p(\boldsymbol{V}|\boldsymbol{WH}) \\ &= -D_{KL}(\boldsymbol{V}|\boldsymbol{WH}) \\ &\quad -\sum_m \sum_n \left[v_{mn}(1-\log v_{mn}) + \log\Gamma(v_{mn}+1) \right] \end{aligned} \quad (5)$$

From (2) and (5) we could conclude that maximizing a posterior is equivalent to maximizing the log-likelihood, and maximizing the log-likelihood is equivalent to minimizing the *KL*-divergence. Thus, maximizing a posterior is equivalent to minimizing the *KL*-divergence. Therefore, it is possible to find a base matrix $\boldsymbol{W}$ and a feature matrix $\boldsymbol{H}$ to approximate the dataset $\boldsymbol{V}$ via maximizing a posterior criterion.

In data model $p(\boldsymbol{V}|\boldsymbol{WH})$ we regard $\boldsymbol{WH}$ as the parameter of data $\boldsymbol{V}$. With respect to the base matrix $\boldsymbol{W}$ and the feature matrix $\boldsymbol{H}$, we also introduce a parameter $\boldsymbol{\lambda}$ as a prior to them. Moreover, we define an independent *Exponent* distribution for each column of $\boldsymbol{W}$ and each row of $\boldsymbol{H}$ with prior λ_k because exponent distribution has sharper performance. It is no doubt that we can choose other exponential family distributions such as

Gaussian distribution, Gamma distribution, etc. Therefore, the columns of $\boldsymbol{W}$ and rows of $\boldsymbol{H}$ yield:

$$p\left(w_{mk}\middle|\lambda_k\right)=\lambda_k\times e^{-\lambda_k w_{mk}} \quad (6)$$

$$p\left(h_{kn}\middle|\lambda_k\right)=\lambda_k\times e^{-\lambda_k h_{kn}} \quad (7)$$

Then the log-likelihood of the priors could be rewritten as:

$$\log p\left(\boldsymbol{W}\middle|\boldsymbol{\lambda}\right)=\sum_m\sum_k\left(\log\lambda_k-\lambda_k\times w_{mk}\right) \quad (8)$$

$$\log p\left(\boldsymbol{H}\middle|\boldsymbol{\lambda}\right)=\sum_k\sum_n\left(\log\lambda_k-\lambda_k\times h_{kn}\right) \quad (9)$$

Compare to setting λ as a constant, the diversity of λ_k and recursively updating of λ_k enable the inference procedure to converge at the stationary point. Through calculating the L_2-morm of each column of base matrix W, we could discover that the data finally emerges to two clusters. One cluster contains the points of which the L_2-norm are much larger than 0, whereas in the other cluster the L_2-norm values are 0 or almost 0.

In order to find the best value for λ_k, here we introduce hyper-prior for λ_k. Since λ_k is the parameter of *Exponent* distribution, we define λ_k follows Gamma distribution which is the conjugate prior for Exponent distribution.

$$p\left(\lambda_k\middle|a_k,b_k\right)=\frac{1}{\Gamma(a)\times {b_k}^{a_k}}\lambda_k^{a-1}\exp\left(-\lambda_k/b_k\right) \quad (10)$$

Here a_k and b_k are the hyper-priors of λ_k. Thus, the log-likelihood of $\boldsymbol{\lambda}$ is given as:

$$\begin{aligned}&\log p(\boldsymbol{\lambda})\\&=\sum_k\left(-a_k\log b_k-\log\Gamma\left(a_k\right)+\left(a_k-1\right)\log\lambda_k-\lambda_k/b_k\right)\end{aligned} \quad (11)$$

3.2. Inference

After the establishment of data model and the deduction of log-likelihood of each prior, we can gain the maximum a posterior equation:

$$\begin{aligned}&MAP(\boldsymbol{W},\boldsymbol{H},\boldsymbol{\lambda})\\&=\sum_m\sum_n\left[v_{mn}\log\left(v_{mn}-1\right)+\log\Gamma\left(v_{mn}+1\right)\right]\\&-\Big[D_{KL}\left(\boldsymbol{V}\middle|\boldsymbol{WH}\right)+\sum_m\sum_k\left(\lambda_k w_{mk}-\log\lambda_k\right)\\&+\sum_k\sum_n\left(\lambda_k h_{kn}-\log\lambda_k\right)\\&+\sum_k\left(a_k\log b_k+\log\Gamma\left(a_k\right)+\lambda_k/b_k\right)-\left(a_k-1\right)\log b_k\Big]\end{aligned} \quad (12)$$

Since the first factor in (12) has nothing to do with the priors, and we have discussed the relationship between the posterior probability and *KL*-divergence, here we minimize the second factor to seek the solutions for this criterion. In our paper, we choose gradient decent updating method as our updating rule. Although multiplicative method is simpler, it has no detailed deduction about why the approach works. On the contrary, gradient decent updating will give us clear deduction about the whole updating procedure. We utilize this method to infer the priors $\boldsymbol{W}$ and $\boldsymbol{H}$, as well as the hyper-priors $\boldsymbol{\lambda}$ and $\boldsymbol{b}$. First we find the gradient of the parameters:

$$\frac{\partial f}{\partial W}=-\frac{\boldsymbol{V}}{\boldsymbol{WH}}\times\boldsymbol{H}^T+\boldsymbol{H}^T+\boldsymbol{\lambda} \quad (13)$$

$$\frac{\partial f}{\partial H}=-\boldsymbol{W}^T\times\frac{\boldsymbol{V}}{\boldsymbol{WH}}+\boldsymbol{W}^T+\boldsymbol{\lambda} \quad (14)$$

$$\frac{\partial f}{\partial\lambda_k}=\sum_m w_{mk}+\sum_n h_{kn}+1/b_k-\left(N+M+a_k-1\right) \quad (15)$$

$$\frac{\partial f}{\partial b_k}=-\frac{\lambda_k}{b_k^2}+\frac{a_k}{b_k} \quad (16)$$

Then we utilize gradient coefficient to get rid of the subtraction operation during the updating procedure for $\boldsymbol{W}$ and $\boldsymbol{H}$ to guarantee the non-negative constrain. The parameters λ_k and b_k are updated by zeroing.

The updating rules are listed as follows:

$$w_{mk}^*=w_{mk}\times\frac{\frac{v_{mn}}{[wh]_{mn}+\varepsilon}\times\sum_n h_{kn}}{\sum_n h_{kn}+\lambda_k+\varepsilon} \quad (17)$$

$$h_{kn}^*=h_{kn}\times\frac{\sum_m w_{mk}\times\frac{v_{mn}}{[wh]_{mn}+\varepsilon}}{\sum_m w_{mk}+\lambda_k+\varepsilon} \quad (18)$$

$$\lambda_k=\frac{M+N+a_k-1}{\sum_m w_{mk}+\sum_n h_{kn}+1/b_k+\varepsilon} \quad (19)$$

$$b_k=\frac{a_k}{\lambda_k+\varepsilon} \quad (20)$$

Then we find the correct bases and determine the order of the data model by:

$$R=\|B\|_1 \quad (21)$$

where B is defined as

$$B\triangleq\left\{\|w_k\|_2,\|w_k\|_2\gg 0\right\} \quad (22)$$

R is the rank of base matrix.

4. Experimental Results and Evaluation

In this section, we apply our unsupervised multi-level NMF algorithm on two binary datasets. One is fence dataset, and the other is famous swimmer dataset. Both of the experiment results demonstrate the efficacy of our method on the rank determination issue.

4.1. Fence Dataset

We first performed our experiments on fence dataset. Here I defined the data with four row bars (the size is 1 × 32) and four column bars (the size is 32 × 1). The size of each image is 32 × 32 with zero-value background, and the value of each pixel in eight bars is one. Each image is separated into five parts in both horizontal direction and vertical direction. Additionally, in each image the number of row bars and the number of column bars should be the same. For instance, there are two row bars in a sample image, then there should be two column bars in this image. Hence, the total number of the fence dataset is N = 69. The samples of Fence dataset are shown in **Figure 2**.

Here, we set the initial rank $K = 16$ (the initial value of rank K needs to be larger than the value of real rank of base matrix), the hyper-parameter $a = 2$, $b_k = [0.05 \cdots 0.05]_{1\times K}$. **Figure 3** shows the base matrix and feature matrix learned via our unsupervised multi-level NMF approach, we could see that the data is sparse, especially the base matrix. In both images, the color parts denote the effective bases or features, and the black parts denote irrelevant bases or features there. In addition, from image processing perspective, we can conclude that compared to the values of effective bases and features, the values of irrelevant bases and features are very small, since the color of such pixels are very dark. We could clearly find that there are eight color column vectors in the first image. Additionally, among the eight color vectors, four are composed of several separated color pixels, whereas the other four are composed of assembly pixels. Actually, the former four vectors are row bars, and the latter four vectors are column bars. We resize the dataset in columns during factorization procedure. Hence the row bars and column bars have different structures. Furthermore, there are also eight rows in the second image, which are the corresponding coefficients of the bases.

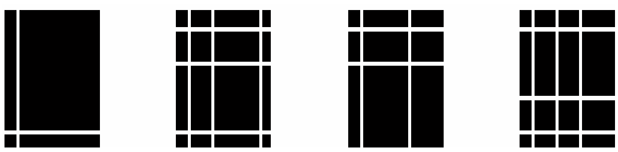

Figure 2. Sample images of fence dataset.

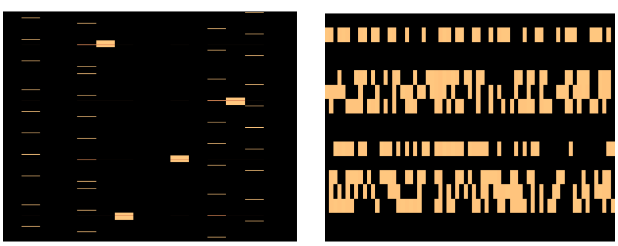

Figure 3. Base matrix *W* and feature matrix *H* learned via our algorithm.

In order to show the bases clearly, we draw the bases in **Figure 4**. Since we set the initial rank of base matrix K = 16, however, only eight images have non-zero values. Moreover, the eight images show 4 row bars and 4 column bars appearing in different positions. The results are perfectly consistent to the design of Fence dataset. Therefore, we could get the conclusion that our algorithm is very powerful and efficient to find the real basic components and the correct rank.

4.2. Swimmer Dataset

The other dataset we used is the swimmer dataset. Swimmer dataset is a typical dataset for feature extraction. Due to the clearly definition and composition of 16 dynamic parts, it is quite appropriate to the unique characteristic of NMF algorithm, which is to learn part-based data. As we know, however, the swimmer dataset is a gray-level image dataset. In our experiment, we focus on binary dataset, so first we need to convert this gray-level dataset to binary dataset. Then apply our approach to perform inference. In this swimmer dataset, there are 256 images totally, each of which depicts a swimming gesture using one torso and four dynamic limbs. The size of each image is 32 × 32. Each dynamic part could appear at four different positions. **Figure 5** shows some sample images of the swimmer dataset.

In this experiment part, the initial rank is set to $K = 25$, the initial values of hyper-parameters are $a = 2$, $b_k = [0.05 \cdots 0.05]_{1\times K}$. **Figure 6** shows the experiment

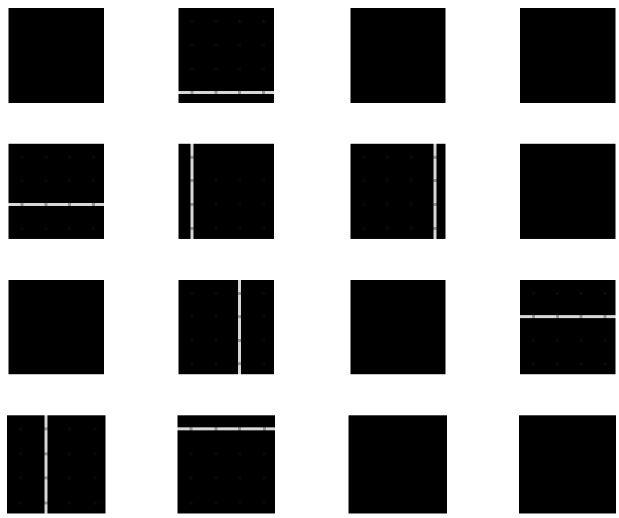

Figure 4. The bases obtained by our algorithm on fence dataset.

Figure 5. Sample images of the swimmer dataset.

results for the swimmer dataset. It could be observed that as for this dataset, we also could find out the correct bases via our algorithm. In this figure there are 25 base images. The black ones correspond to irrelevant bases, and the other 17 images depict the torso and the limbs at each possible position. We can see that the correct torso and limbs are discovered successfully.

The differences between the black images and the correct base images are shown in **Figure 7**. **Figure 7** depicts L_2-norm of each column of the base matrix. The total number of points in this figure is the same to the initial rank. Obviously, the points are classified into two clusters. One is zero-value cluster, and the other is larger-value cluster. Thus the rank of base matrix in swimmer dataset is $R = \|B\|_1 = 17$. The results of L_2-norm of base matrix not only tell us how we could find the correct bases, but also tell us how we could determine the correct rank of base matrix.

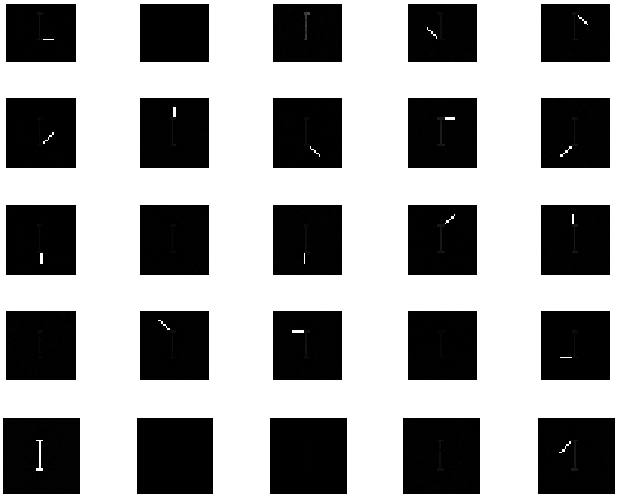

Figure 6. The bases of swimmer dataset learned by our algorithm.

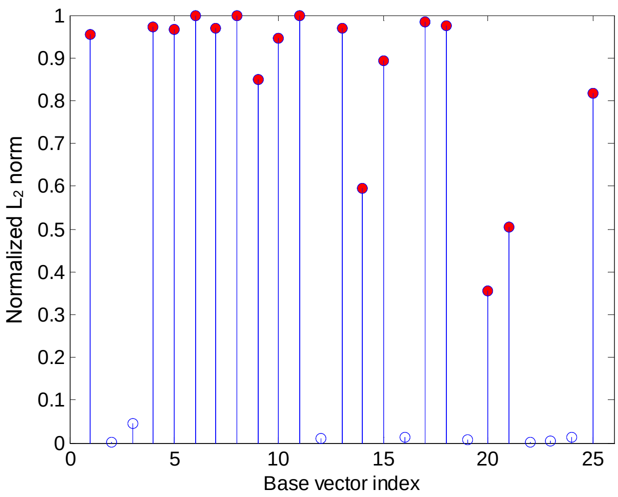

Figure 7. L_2-norm of base vectors.

5. Conclusion

We have presented an unsupervised multi-level non-negative matrix factorization algorithm which is powerful and efficient to seek the correct rank of a data model. This is achieved by introducing a multi-prior structure. The experiment results on binary datasets adequately demonstrate the efficacy of our algorithm. Compare to the fully Bayesian method, it is simpler and more convenient. The crucial points of this method are how to introduce the hyper-priors and what kind of prior is appropriate to a certain data model. This algorithm also could be extended to other data models and noise models. Although our experiment is based on binary dataset, this algorithm is suitable to other datasets such as gray-level dataset, colorful dataset, etc.

REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature*, Vol. 401, No. 6755, 1999, pp. 788-791. doi:10.1038/44565

[2] Z. Yuan and E. Oja, "Projective Non-Negative Matrix Factorization for Image Compression and Feature Extraction," Springer, Heidelberg, 2005.

[3] C. Fevotte, N. Bertin and J. L. Durrieu, "Non-Negative Matrix Factorization with the Itakura-Saito Divergence," *With Application to Music Analysis. Neural Computation*, Vol. 21, No. 3, 2009, pp. 793-830. doi:10.1162/neco.2008.04-08-771

[4] M. W. Berry and M. Browne, "Email Surveillance Using Non-Negative Matrix Factorization," *Computational and Mathematical Organization Theory*, Vol. 11, No. 3, 2005, pp. 249-264. doi:10.1007/s10588-005-5380-5

[5] Q. Sun, F. Hu and Q. Hao, "Context Awareness Emergence for Distributed Binary Pyroelectric Sensors," *Proceeding of* 2010 *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, Salt Lake City, 5-7 September 2010, pp. 162-167.

[6] F. Hu, Q. Sun and Q. Hao, "Mobile Targets Region-of-Interest via Distributed Pyroelectric Sensor Network: Towards a Robust, Real-Pyroelectric Sensor Network," *Proceeding of* 2010 *IEEE Conference on Sensors*, Waikoloa, 1-4 November 2010, pp. 1832-1836.

[7] Y. Xue, C. S. Tong and Y. C. W. Chen, "Clustering-Based Initialization for Non-Negative Matrix Factorization," *Applied Mathematics and Computation*, Vol. 205, No. 2, 2008, pp. 525-536. doi:10.1016/j.amc.2008.05.106

[8] Z. Yang, Z. Zhu and E. Oja, "Automatic Rank Determination in Projective Non-Negative Matrix Factorization," *Proceedings of* 9*th International Conference on LVA/ICA*, St. Malo, 27-30 September 2010, pp. 514-521.

[9] A. T. Cemgil, "Bayesian Inference for Non-Negative Matrix Factorization Models," *Computational Intelligence and Neuroscience*, Vol. 2009, 2009, Article ID: 785152. doi:10.1155/2009/785152

[10] M. Said, D. Brie, A. Mohammad-Djafari and C. Cedric,

"Separation of Nonnegative Mixture of Nonnegative Sources Using a Bayesian Approach and MCMC Sampling," *IEEE Transactions on Signal Processing*, Vol. 54, No. 11, 2006, pp. 4133-4145. doi:10.1109/TSP.2006.880310