IEICE TRANS. INF. & SYST., VOL.E94–D, NO.10 OCTOBER 2011

PAPER Special Section on Information-Based Induction Sciences and Machine Learning

Dimensionality Reduction for Histogram Features Based on Supervised Non-negative Matrix Factorization

Mitsuru AMBAI[†a)], ***Member***, **Nugraha P. UTAMA**[†], ***and*** **Yuichi YOSHIDA**[†], ***Nonmembers***

SUMMARY Histogram-based image features such as HoG, SIFT and histogram of visual words are generally represented as high-dimensional, non-negative vectors. We propose a supervised method of reducing the dimensionality of histogram-based features by using non-negative matrix factorization (NMF). We define a cost function for supervised NMF that consists of two terms. The first term is the generalized divergence term between an input matrix and a product of factorized matrices. The second term is the penalty term that reflects prior knowledge on a training set by assigning predefined constants to cannot-links and must-links in pairs of training data. A multiplicative update rule for minimizing the newly-defined cost function is also proposed. We tested our method on a task of scene classification using histograms of visual words. The experimental results revealed that each of the low-dimensional basis vectors obtained from the proposed method only appeared in a single specific category in most cases. This interesting characteristic not only makes it easy to interpret the meaning of each basis but also improves the power of classification.

key words: *dimensionality reduction, non-negative matrix factorization, histogram-based features*

1. Introduction

Histogram-based features such as HoG [1], SIFT [2], and bag-of-features representations using visual words [3]–[5] have been used for a broad range of applications; for instance, human detection, image matching, scene classification, and many more. Histogram-based features are generally represented by a few hundred [2], [3] to one million [4], [5] dimensional vectors. Because of the high dimensionality of the features, techniques of reducing dimensionality have often been applied to extract fundamental information that has made data analysis feasible [6].

Here, it should be noted that the histogram features are non-negative. For the purpose of constructing a low-dimensional feature space for non-negative data, use of non-negative matrix factorization (NMF) [7] has been focused on, instead of principal component analysis (PCA). The main focus of this paper is to propose a supervised method of reducing dimensionality for histogram-based features based on NMF.

1.1 Previous Work

NMF decomposes an input matrix $\mathbf{X}$ into two non-negative matrices: a basis matrix $\mathbf{F}$ and a coefficient matrix $\mathbf{G}$, minimizing some types of similarity measures between $\mathbf{X}$ and $\mathbf{F}\mathbf{G}^\top$. Interestingly, it has been reported that the non-negativity of $\mathbf{F}$ and $\mathbf{G}$ results in sparse and part-based representation of the input data [7]. For example, each of non-negative basis vectors obtained by factorizing face images obviously corresponds to a local part of the face. Thanks to this characteristic, we can automatically find common parts of the faces. In another example, semantic topics shared with a large number of documents can be found by decomposing term-document matrix of the documents. Thanks to the non-negativity of $\mathbf{F}$, relationship between terms and topics can be easily interpreted by observing elements in the $\mathbf{F}$. Thus NMF enables us to easily interpret the meaning of each basis. This is an important advantage of NMF.

Various kinds of similarity measures have been used as cost functions of NMF, e.g., the Frobenius norm [7], generalized divergence [8], the earth mover's distance [9] and many more [10]. The performance of NMF strongly depends on the type of input data and the choice of similarity measures; therefore, it is important to select a good similarity measure. For stochastic vectors such as histogram-based features, it has been reported that the use of a similarity measure that is defined from the perspective of information theory is important [10].

The generalized divergence proposed by Lee et al. [8] is one of the simplest and most common similarity measures based on information theory. This similarity measure is defined between two distributions $\mathbf{A}$ and $\mathbf{B}$ as:

$$D(\mathbf{A}||\mathbf{B}) = \sum_{ij} \mathbf{A}_{ij} \log\left(\frac{\mathbf{A}_{ij}}{\mathbf{B}_{ij}}\right) - \mathbf{A}_{ij} + \mathbf{B}_{ij}. \quad (1)$$

Figure 1 visualizes the generalized divergence and the Frobenius norm with colored images to enable comparison. The divergence takes frequently occurring signals as unimportant information. For example, let us consider an image recognition task using a histogram of visual words. Because it is natural to think that frequently appearing visual words can be regarded as common visual features among images, such visual words are not important to construct a discriminative subspace. In this case, use of the generalized divergence as a similarity measure has an advantage in image recognition.

There is another approach to give advantageous characteristics to NMF. By adding a penalty term into the cost function, application-specific characteristics can be incorporated into NMF [11]–[14]. The addition of penalty terms that reflect prior knowledge on input data to cost functions

Manuscript received December 28, 2010.
Manuscript revised May 5, 2011.

[†]The authors are with Denso IT Laboratory, Inc., Tokyo, 150–0002 Japan.
a) E-mail: manbai@d-itlab.co.jp

DOI: 10.1587/transinf.E94.D.1870

has recently attracted increasing attention to construct more discriminative feature space [15]–[18]. For example, Wang et al. [15] proposed Fisher NMF that minimizes within-class scatter and maximizes between-class scatter of labeled input data in the constructed feature space. The within- and between-class scatters were represented by two penalty terms.

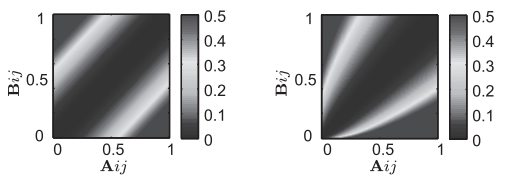
Constraint matrices have more recently been used to express prior knowledge on input data [16]–[18]. Yang et al. [16] expressed similarities and dissimilarities of inputs as a constraint matrix and embedded them into the cost function as a penalty term. Wang et al. [17] gave a computationally efficient algorithm to minimize Yang's cost function. A similar cost function using the constraint matrix has been proposed by Wang et al. [18] for the purpose of clustering semi-supervised data. The magnitude of each element in the constraint matrix expresses the reliability of prior knowledge. This expression is practical for a broad range of applications.

Previous methods [16]–[18] have used the constraint matrix together with the Frobenius norm. It is reasonable to expect that the use of a constraint matrix together with generalized divergence is effective for constructing a more discriminative feature space for histogram-based features.

1.2 Contribution

We propose a supervised method of reducing dimensionality for histogram-based features based on NMF using the divergence similarity measure with a constraint matrix in this paper. We describe our method as D-SNMF in the sections that follow. The two main contributions of this paper are:

1. NMF with the divergence similarity measure is reformulated within a supervised learning context. The prior knowledge on input data is expressed in the form of a constraint matrix, and incorporated into NMF as a penalty term.
2. A multiplicative update rule is proposed to optimize the new cost function. In exchange for mathematically guaranteed convergence, our update rule becomes numerically simple. Despite the compromise, we experimentally confirmed that the update rule works so robustly that convergence of the algorithm is not an issue in practice.



(a) $(\mathbf{A}_{ij} - \mathbf{B}_{ij})^2$ (b) $\mathbf{A}_{ij} \log\left(\frac{\mathbf{A}_{ij}}{\mathbf{B}_{ij}}\right) - \mathbf{A}_{ij} + \mathbf{B}_{ij}$

Fig. 1 Difference between Frobenius norm (left) and generalized divergence (right). When $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ are small, the generalized divergence becomes more sensitive to the difference between $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ than the Frobenius norm.

We tested our method on classification tasks using datasets for 13 Natural Scene Categories [19] that were encoded into the histograms of visual words. We examined various types of NMFs including our method for scene classification. The experimental results revealed that each basis vector obtained from our method was strongly associated with a single specific category in most cases. This interesting characteristic not only makes it easy to interpret what each basis means but also improves the power of classification.

Section 2 gives an overview of NMF with the Frobenius norm (L_2-NMF [7]) and with divergence (D-NMF [8]). Section 3 is the most important part of this paper, where we describe the modified cost function of D-NMF with prior knowledge (D-SNMF) in detail. The update rule for optimization is also proposed. In Sect. 4, we discuss the L_2-NMF with prior knowledge (L_2-SNMF) for the purpose of comparing it with our method. Readers are referred to [18] for the original discussion on L_2-SNMF. The experimental results are presented in Sect. 5.

2. L_2-NMF and D-NMF

The NMF decomposes a non-negative matrix $\mathbf{X}$ into two non-negative matrices $\mathbf{F}$ and $\mathbf{G}$ as:

$$\mathbf{X} \simeq \mathbf{F}\mathbf{G}^\top, \tag{2}$$

where $\mathbf{F} \in \mathbb{R}^{d\times k}$ is a basis matrix and $\mathbf{G} \in \mathbb{R}^{n\times k}$ is a coefficient matrix. In the convention we use, input data $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$ are column vectors in $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$. A low-dimensional representation of $\mathbf{x}_i$ is $\mathbf{g}_i \in \mathbb{R}^{k\times 1}$. They are stored as row vectors in $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n]^\top$ for $k < d$.

The cost functions of L_2-NMF and D-NMF are defined in Eqs. (3) and (4), respectively.

$$J_1(\mathbf{F}, \mathbf{G}) = \|\mathbf{X} - \mathbf{F}\mathbf{G}^\top\|_F^2 \quad s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0. \tag{3}$$

$$J_2(\mathbf{F}, \mathbf{G}) = D(\mathbf{X}\|\mathbf{F}\mathbf{G}^\top) \quad s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0. \tag{4}$$

The function $D(\cdot\|\cdot)$ is the generalized divergence defined in Eq. (1). The multiplicative update rules for minimizing these cost functions are summarized in Tables 1 and 2.

In Eqs. (3) and (4), the optimal solutions for $\mathbf{F}$ and $\mathbf{G}$ are not unique. For any non-negative and non-singular matrix $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{G}$ can be replaced with $\mathbf{F}\mathbf{U}$ and $\mathbf{G}\mathbf{U}^{-\top}$ without changing the value of the cost function. A diagonal matrix that normalizes the L_2 norm of the column vectors in $\mathbf{F}$ is often used as $\mathbf{U}$ to make the solution unique.

Table 1 Multiplicative update rule for L_2-NMF.

Algorithm 1 - L_2-NMF (Lee et al. [7], [8])
Step 1 Initialize $\mathbf{F}$ and $\mathbf{G}$ as a random dense matrix.
Step 2 Update $\mathbf{F}$ and $\mathbf{G}$ until convergence.
(a) $\mathbf{F}_{ij} \leftarrow \mathbf{F}_{ij}\frac{(\mathbf{X}\mathbf{G})_{ij}}{(\mathbf{F}\mathbf{G}^\top\mathbf{G})_{ij}}$
(b) Normalize $\mathbf{F}$ and $\mathbf{G}$. $\quad \mathbf{F} \leftarrow \mathbf{F}\mathbf{U},\ \mathbf{G} \leftarrow \mathbf{G}\mathbf{U}^{-\top}$
(c) $\mathbf{G}_{ij} \leftarrow \mathbf{G}_{ij}\frac{(\mathbf{X}^\top\mathbf{F})_{ij}}{(\mathbf{G}\mathbf{F}^\top\mathbf{F})_{ij}}$

Table 2 Multiplicative update rule for D-NMF.

Algorithm 2 - D-NMF (Lee et al. [7], [8])
Step 1 Initialize $\mathbf{F}$ and $\mathbf{G}$ as a random dense matrix.
Step 2 Update $\mathbf{F}$ and $\mathbf{G}$ until convergence.
(a) $\mathbf{F}_{ij} \leftarrow \mathbf{F}_{ij} \frac{\sum_k \mathbf{X}_{ik}\mathbf{G}_{kj}/(\mathbf{F}\mathbf{G}^\top)_{ik}}{\sum_k \mathbf{G}_{kj}}$
(b) Normalize $\mathbf{F}$ and $\mathbf{G}$. $\mathbf{F} \leftarrow \mathbf{F}\mathbf{U},\ \mathbf{G} \leftarrow \mathbf{G}\mathbf{U}^{-\top}$
(c) $\mathbf{G}_{ij} \leftarrow \mathbf{G}_{ij} \frac{\sum_k \mathbf{X}_{ki}\mathbf{F}_{kj}/(\mathbf{F}\mathbf{G}^\top)_{ki}}{\sum_k \mathbf{F}_{kj}}$

Table 3 Multiplicative update rule for D-SNMF.

Algorithm 3 - D-SNMF (our method)
Step 1 Initialize $\mathbf{F}$ and $\mathbf{G}$ as a random dense matrix.
Step 2 Update $\mathbf{F}$ and $\mathbf{G}$ until convergence.
(a) $\mathbf{F}_{ij} \leftarrow \mathbf{F}_{ij} \frac{\sum_k \mathbf{X}_{ik}\mathbf{G}_{kj}/(\mathbf{F}\mathbf{G}^\top)_{ik}}{\sum_k \mathbf{G}_{kj}}$
(b) Normalize $\mathbf{F}$. $\mathbf{F} \leftarrow \mathbf{F}\mathbf{U}$
(c) $\mathbf{G}_{ij} \leftarrow \mathbf{G}_{ij} \sqrt{\frac{\sum_k \frac{\mathbf{X}_{ki}\mathbf{F}_{kj}}{(\mathbf{F}\mathbf{G}^\top)_{ki}} + 2(\mathbf{C}^-\mathbf{G})_{ij}}{\sum_k \mathbf{F}_{kj} + 2(\mathbf{C}^+\mathbf{G})_{ij}}}$

3. D-NMF with Prior Knowledge (D-SNMF)

3.1 Penalty Term for Prior Knowledge

Imposing penalty terms on the cost function can control the characteristics of NMF [11]–[14].

$$J(\mathbf{F},\mathbf{G}) = D(\mathbf{X}\|\mathbf{F}\mathbf{G}^\top) + J_f(\mathbf{F}) + J_g(\mathbf{G}) \quad s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0. \tag{5}$$

$J_f(\mathbf{F})$ and $J_g(\mathbf{G})$ in Eq. (5) are the penalty terms of the coefficient and basis matrix, respectively. Various penalty terms have been proposed to yield application-specific characteristics. Prior knowledge of input data can also be represented as a penalty term. The distribution of $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n$ in a low-dimensional feature space should reflect prior knowledge; therefore, the penalty term should be a function of the coefficient matrix. Consequently, we only discuss the definition of $J_g(\mathbf{G})$ and let $J_f(\mathbf{F}) = 0$.

The prior knowledge given to the low-dimensional coefficient matrix $\mathbf{G}$ can be incorporated into $J_g(\mathbf{G})$ by using an n-by-n constraint matrix $\mathbf{C} = \{\mathbf{C}_{ij}\}$. If input vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ stand close to each other in the feature space, $\mathbf{C}_{ij}$ is set to a negative value, representing a "*must-link*". However, if input vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ stand apart in the feature space, $\mathbf{C}_{ij}$ is set to a positive value, representing a "*cannot-link*". The magnitude of the element $|\mathbf{C}_{ij}|$ means the reliability of prior knowledge. If no prior knowledge is available between $\mathbf{x}_i$ and $\mathbf{x}_j$, we just set $\mathbf{C}_{ij} = 0$.

The penalty term $J_g(\mathbf{G})$ can be defined using the constraint matrix $\mathbf{C}_{ij}$ as:

$$J_g(\mathbf{G}) = \sum_{ij} \mathbf{C}_{ij}\mathbf{g}_i^\top\mathbf{g}_j = tr(\mathbf{G}^\top\mathbf{C}\mathbf{G}). \tag{6}$$

The $J_g(\mathbf{G})$ constrains the behavior of dot products between coefficient vectors in the optimization process. If $\mathbf{C}_{ij}$ is positive, $\mathbf{g}_i^\top\mathbf{g}_j$ becomes a small value. If $\mathbf{C}_{ij}$ is negative, $\mathbf{g}_i^\top\mathbf{g}_j$ becomes a large value. This enforces the feature space to be discriminative for the input data according to prior knowledge $\mathbf{C}_{ij}$. It should be noted, if the lengths of $\mathbf{g}_i$ and $\mathbf{g}_j$ are obviously different, it is still possible $\mathbf{g}_i$ and $\mathbf{g}_j$ stand apart from each other even if $\mathbf{C}_{ij}$ is negative. However the lengths of input vectors and basis vectors are normalized to 1 in our experiments. Therefore the lengths of all of the coefficient vectors are almost similar to each other. Thanks to this weak constraint on the lengths of the coefficient vectors, $\mathbf{g}_i$ and $\mathbf{g}_j$ stand close to each other in the reduced space by minimizing $J_g(\mathbf{G})$ when $\mathbf{C}_{ij}$ is set to a negative value.

3.2 Cost Function and Update Rule for D-SNMF

Based on the previous discussion, the cost function for D-SNMF is defined as:

$$J_3(\mathbf{F},\mathbf{G}) = D(\mathbf{X}\|\mathbf{F}\mathbf{G}^\top) + tr(\mathbf{G}^\top\mathbf{C}\mathbf{G}) \quad s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0. \tag{7}$$

This cost function can be minimized with the update rule in Table 3.

The update rule in Step 2 (a) in Table 3 is the same as that for D-NMF. The penalty term $tr(\mathbf{G}^\top\mathbf{C}\mathbf{G})$ in Eq. (7) stays constant during the updating process of $\mathbf{F}$. There are no differences in the updating process of $\mathbf{F}$ between D-NMF and D-SNMF.

The following normalization step in Table 3 is done differently between the two NMFs. For D-NMF, the column vectors in $\mathbf{F}$ can be normalized without changing the value of the cost function, because $J_2(\mathbf{F},\mathbf{G}) = J_2(\mathbf{F}\mathbf{U}, \mathbf{G}\mathbf{U}^{-\top})$ for any non-negative and non-singular matrix $\mathbf{U}$. However, this equation is not satisfied for D-SNMF, because Eq. (7) has a penalty term that only depends on $\mathbf{G}$.

$$J_3(\mathbf{F},\mathbf{G}) \neq J_3(\mathbf{F}\mathbf{U}, \mathbf{G}\mathbf{U}^{-\top}). \tag{8}$$

It is possible that normalizing $\mathbf{F}$ increases the value of the cost function. To avoid this problem, Yang et al. [16] introduced a more complicated penalty term that is independent of the length of the column vectors of $\mathbf{F}$. However, such a complicated penalty term makes the update rule for $\mathbf{F}$ and $\mathbf{G}$ more complex.

We instead avoid such complexity and allow the cost function to vary under normalization. The basis matrix $\mathbf{F}$ is only normalized during iteration in our proposed method. Although it is possible that normalization may increase the cost function, it was experimentally proved that the update rule for $\mathbf{F}$ and $\mathbf{G}$ in Step 2 (a) and (c) can decrease larger amounts of the value of the cost function in almost any situation. In exchange for compromising mathematically guaranteed convergence, our update rule becomes simpler than the previously proposed method [16]. This is a great advantage that enables computational efficiency. In the last of this section, we discuss about the order of the computation time in detail.

The update rule of $\mathbf{G}$ can be derived from the Karush-Kuhn-Tucker (KKT) condition in a similar fashion to that by Wang et al. [18]. If the cost function converges in the update

process, the final solution must satisfy the KKT condition. The KKT condition can be derived from the Lagrangian function for the cost function

$$L = J_3(\mathbf{F}, \mathbf{G}) - tr(\boldsymbol{\beta}\mathbf{G}^\top), \tag{9}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{n\times k}$ is a Lagrangian multiplier. Eq. (10) is a differentiation of L with respect to $\mathbf{G}$.

$$\frac{\partial L}{\partial \mathbf{G}_{ij}} = -\sum_k \frac{\mathbf{X}_{ki}\mathbf{F}_{kj}}{(\mathbf{F}\mathbf{G}^\top)_{ki}} + \sum_k \mathbf{F}_{kj} + 2(\mathbf{C}\mathbf{G})_{ij} - \boldsymbol{\beta}_{ij}. \tag{10}$$

Letting $\partial L/\partial \mathbf{G}_{ij} = 0$, we obtain

$$\boldsymbol{\beta}_{ij} = -\sum_k \frac{\mathbf{X}_{ki}\mathbf{F}_{kj}}{(\mathbf{F}\mathbf{G}^\top)_{ki}} + \sum_k \mathbf{F}_{kj} + 2(\mathbf{C}\mathbf{G})_{ij}. \tag{11}$$

Equation (12) is the KKT condition that must be satisfied at convergence.

$$\left(-\sum_k \frac{\mathbf{X}_{ki}\mathbf{F}_{kj}}{(\mathbf{F}\mathbf{G}^\top)_{ki}} + \sum_k \mathbf{F}_{kj} + 2(\mathbf{C}\mathbf{G})_{ij}\right)\mathbf{G}_{ij} = \boldsymbol{\beta}_{ij}\mathbf{G}_{ij} = 0. \tag{12}$$

Here, we consider the following update rule of $\mathbf{G}$.

$$\mathbf{G}_{ij} \leftarrow \mathbf{G}_{ij}\sqrt{\frac{\sum_k \mathbf{X}_{ki}\mathbf{F}_{kj}/(\mathbf{F}\mathbf{G}^\top)_{ki} + 2(\mathbf{C}^-\mathbf{G})_{ij}}{\sum_k \mathbf{F}_{kj} + 2(\mathbf{C}^+\mathbf{G})_{ij}}}, \tag{13}$$

where $\mathbf{C}^+$ and $\mathbf{C}^-$ are non-negative matrices that satisfy $\mathbf{C} = \mathbf{C}^+ - \mathbf{C}^-$. Because all the matrices that appeared in the update rule are non-negative, it is clear that $\mathbf{G}$ is always non-negative during iteration.

If the cost function converges in the update process, the left part of Eq. (13) becomes equivalent to the right part at the final solution as shown in Eq. (14).

$$\mathbf{G}_{ij} = \mathbf{G}_{ij}\sqrt{\frac{\sum_k \mathbf{X}_{ki}\mathbf{F}_{kj}/(\mathbf{F}\mathbf{G}^\top)_{ki} + 2(\mathbf{C}^-\mathbf{G})_{ij}}{\sum_k \mathbf{F}_{kj} + 2(\mathbf{C}^+\mathbf{G})_{ij}}}. \tag{14}$$

We rewrite Eq. (14) to obtain

$$\left(-\sum_k \frac{\mathbf{X}_{ki}\mathbf{F}_{kj}}{(\mathbf{F}\mathbf{G}^\top)_{ki}} + \sum_k \mathbf{F}_{kj} + 2(\mathbf{C}\mathbf{G})_{ij}\right)\mathbf{G}_{ij}^2 = 0. \tag{15}$$

This equation is equivalent to Eq. (12). Therefore, the update rule satisfies the KKT condition at its final solution.

It is still unknown whether the update rule monotonically decreases the cost function or not. Although an auxiliary function is often used for proving a monotonic decrease of NMF [8], it is difficult to define an auxiliary function for the cost function in Eq. (7). However, despite the lack of mathematical proof, the update rule did monotonically decrease the cost function for all input matrix and configuration parameters used in all of our experiments; thus, we believe that the proposed update rule is fairly robust in practical usage. The discussion about how the cost function decreased will be presented in the experiment section, 4.2.2.

It should be emphasized that the penalty term is not always bounded below. If a constraint matrix only has cannot-links, the penalty term is bounded below by zero because both $\mathbf{G}$ and $\mathbf{C}$ are non-negative. However, when a constraint matrix has strong must-links with large magnitudes, the penalty term occasionally goes to negative infinity. What happens in this case is that the strong must-links make the penalty term dominate over the divergence term. The magnitudes of the penalty term should be set so that the penalty term and the divergence term are well balanced for supervised learning. The magnitudes of must-links in the proposed update rule should be chosen very carefully. The discussion about how the must-links and cannot-links were set in our experiments will be presented in Sect. 4.1.3.

The previously proposed method [16] needs heavy computation of an n-by-n inverse matrix, where n is a number of input data. Since the computation time of the inverse matrix is proportional to the cube of n, the order of the computation for each updating steps of [16] is about $o(n^3)$. On the other hand, the bottle neck of our method is the computation of $\mathbf{C}^+\mathbf{G}$ and $\mathbf{C}^-\mathbf{G}$ if n is large. Since $\mathbf{C}^+\mathbf{G}$ and $\mathbf{C}^-\mathbf{G}$ are n-by-n matrices, the order of the computation for each updating steps is about $o(n^2)$. Besides, it has been reported in [17] that multiplicative update approach converges within a smaller number of iterations than such inverse-matrix-based update rule. Therefore our updating rule is significantly faster than [16].

3.3 L_2-NMF with Prior Knowledge (L_2-SNMF)

In this section, L_2-NMF with prior knowledge is discussed for the purpose of comparing it with D-SNMF. Similar to the discussion in the previous section, the L_2-NMF with prior knowledge can be defined as:

$$J_4(\mathbf{F}, \mathbf{G}) = \|\mathbf{X} - \mathbf{F}\mathbf{G}^\top\|_F^2 + tr(\mathbf{G}^\top\mathbf{C}\mathbf{G}) \quad s.t.\ \mathbf{F} \geq 0, \mathbf{G} \geq 0. \tag{16}$$

Similar optimization issues have already been tackled in previous work [16]–[18]. In particular, Eq. (16) is equivalent to the optimization problem discussed in [18] with additional constraint $\mathbf{F} \geq 0$. Taking this constraint into account, one can minimize the cost function with the update rule given in Table 4.

Note that the update process of $\mathbf{F}$ in Table 4 is exactly the same as that in Table 1, due to the fact that the penalty term, $tr(\mathbf{G}^\top\mathbf{C}\mathbf{G})$, stays constant during the updating process

Table 4 Multiplicative update rule for L_2-SNMF.

Algorithm 4 - L_2-SNMF (similar to Wang et al. [18])
Step 1 Initialize $\mathbf{F}$ and $\mathbf{G}$ as a random dense matrix.
Step 2 Update $\mathbf{F}$ and $\mathbf{G}$ until convergence.
(a) $\mathbf{F}_{ij} \leftarrow \mathbf{F}_{ij}\frac{(\mathbf{X}\mathbf{G})_{ij}}{(\mathbf{F}\mathbf{G}^\top\mathbf{G})_{ij}}$
(b) Normalize $\mathbf{F}$. $\mathbf{F} \leftarrow \mathbf{F}\mathbf{U}$
(c) $\mathbf{G}_{ij} \leftarrow \mathbf{G}_{ij}\sqrt{\frac{(\mathbf{X}^\top\mathbf{F})_{ij} + (\mathbf{C}^-\mathbf{G})_{ij}}{(\mathbf{G}\mathbf{F}^\top\mathbf{F})_{ij} + (\mathbf{C}^+\mathbf{G})_{ij}}}$

of **F**. In Step 2(b), the cost function is less affected by normalizing **F** as well as D-SNMF. Furthermore, it is clear from the discussion in Wang et al. [18] that the update rule of Step 2 (c) monotonically decreases the cost function.

4. Experiments on Image Dataset

We demonstrate that the proposed method, D-SNMF, can construct an appropriate low-dimensional feature space for classifying histogram features of images in this section. To compare it with existing methods, D-SNMF, D-NMF, L_2-SNMF, L_2-NMF, PCA, Linear Discriminant Analysis (LDA) [20] and Marginal Fisher Analysis (MFA) [21] were tested for a scene classification task using 13 Natural Scene Categories [19]. This database includes four indoor and nine outdoor categories. The sizes of images in the database are varied. The average size of them is approximately 250×300 pixels. Before testing LDA and MFA, we applied PCA to make a matrix non-singular in the same manner as those in [20], [21]. 90% cumulative energies of original signals were retained in the preprocessing.

We will discuss three main points: differences in decomposed factors with and without prior knowledge, convergence performance, and comparison of classification accuracies.

4.1 Experimental Setup

4.1.1 Extracting Histogram Features

First, an image was divided into blocks of 8×8 pixels, and then every 2×2 blocks were defined as image patches. SIFT descriptors [2] were computed from all the patches. Approximate k-means clustering was applied to random subsets of the descriptors obtained from the training set to define the visual dictionary. Vocabulary size k was set to 200. Training and testing images were encoded to visual words according to the visual dictionary. The images were spatially divided into 4×4 sub-regions to take into account the spatial layout of visual words as shown in Fig. 2. The pixel size of each sub-region depends on the size of the image. Every training/test image commonly has 16 sub-regions. We extracted 200 dimensional histogram features from each of the 16 sub-regions. Sixteen histogram features from the sub-regions were integrated into a single 3200 dimensional feature vector. The L_2 norm of the vector was normalized to one.

Fig. 2 The images were spatially divided into 4×4 sub-regions to take into account the spatial layout of visual words. A red rectangle represents each sub-region. The pixel size of the sub-regions depends on the size of the image as shown in this figure.

4.1.2 Learning and Classification

The 13 Natural Scene Categories consisted of four indoor and nine outdoor categories. Each category included a few hundred images. In each category, 100 images were randomly chosen as training samples and another 100 images were randomly chosen as testing samples. This means that a number of input data, n, is $100 \times 13 = 1{,}300$ in the both cases of training and testing. The experiments were repeated ten times with different training and testing sets under the same parameter conditions. The mean and standard deviation of the classification accuracies were evaluated. In our experiment, the dimension of feature vector, d, was 3,200 and a number of basis vectors, k, were varied.

The learning and classification algorithm of the NMF family is summarized in Table 5. The feature vectors of training data were stored as column vectors in a training matrix $\mathbf{X}_{train}$. The training matrix $\mathbf{X}_{train}$ was decomposed into $\mathbf{F}_{train}$ and $\mathbf{G}_{train}$ by using D-SNMF, D-NMF L_2-SNMF, and L_2-NMF. The testing matrix $\mathbf{X}_{test}$ was decomposed into $\mathbf{G}_{test}$ and $\mathbf{F}_{train}$, keeping $\mathbf{F}_{train}$ fixed. This factorization can be done by using D-NMF or L_2-NMF without Step 2 (a) and (b) in Table 1 or 2. The k-nearest neighbour algorithm was applied to the testing data in the low-dimensional feature space. Ten nearest neighbours were used in our experiment.

4.1.3 Adjusting Parameters of Constraint Matrix

The constraint matrices for D-SNMF and L_2-SNMF were determined as follows. Negative constant values were given to must-links that were all the pairs of training data belonging to the same category. Similarly, positive constant values were given to cannot-links that were all the pairs of training data belonging to different categories. Figure 3 shows the relationship between classification accuracy and the parameters of the cannot-links and must-links for L_2-SNMF and D-SNMF. The number of dimensions in reduced feature space was set to 25 in both cases. As can be seen

Table 5 Learning and classification algorithm.

Learning:
Input: Training images, **C**
Step 1 Define visual dictionary using training images.
Step 2 Extract histogram features from training images.
Step 3 Normalize L_2 norm of features to one and pack them into training matrix $\mathbf{X}_{train}$.
Step 4 Apply NMF to $\mathbf{X}_{train}$.
$\mathbf{X}_{train} \rightarrow \mathbf{F}_{train}\mathbf{G}_{train}^{\top}$
Output: $\mathbf{F}_{train}$, $\mathbf{G}_{train}$
Classification:
Input: Testing images, $\mathbf{F}_{train}$,$\mathbf{G}_{train}$
Step 1 Extract histogram features from testing images
Step 2 Normalize L_2 norm of the features to 1 and pack them into testing matrix $\mathbf{X}_{test}$.
Step 3 Apply NMF to $\mathbf{X}_{test}$ (only updating $\mathbf{G}_{test}$)
$\mathbf{X}_{test} \rightarrow \mathbf{F}_{train}\mathbf{G}_{test}^{\top}$
Step 4 Apply k-NN method using $\mathbf{G}_{train}$ and $\mathbf{G}_{test}$.
Output: Classification results.

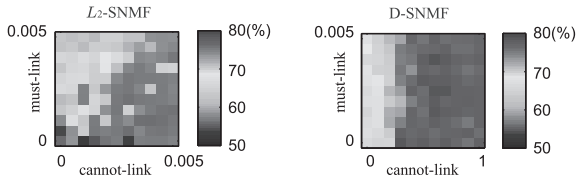
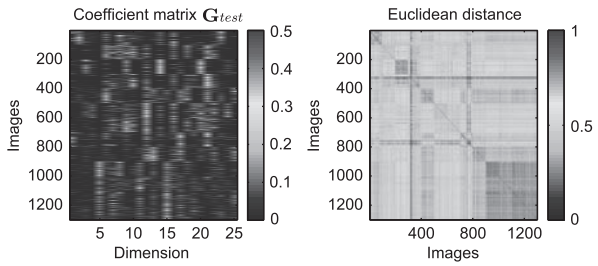
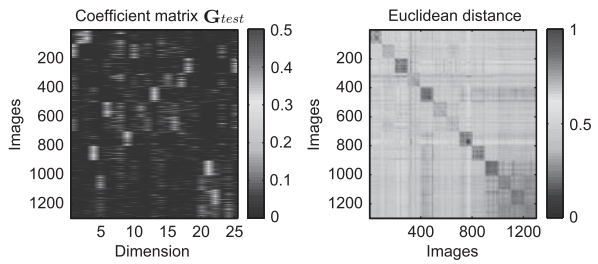


Fig. 3 Relationship between constraint matrix and classification accuracies of D-SNMF and L_2-SNMF. While cannot-links obviously improved accuracy, must-links had less effect on accuracy than cannot-links.



(a) D-NMF



(b) D-SNMF

Fig. 4 Coefficient matrix of testing data (left) and Euclidean distances between all pairs of factorized testing data (right).

from Fig. 3, must-links have less effect on the classification accuracy than cannot-links. We observed that the penalty term occasionally went to negative infinity, when the values of must-links were set below −0.005. These observations indicate weak must-links should be used. Throughout the experiment, the cannot-links were set to 1 for D-SNMF and the must-links were set to −0.005. For L_2-SNMF, the cannot-links were set to 0.005 and the must-links were set to −0.005.

4.2 Experimental Results and Discussion

4.2.1 Differences in Decomposed Factors

It can be observed from Figs. 4 and 5 that decomposed factors obtained with D-SNMF have significant differences compared with D-NMF. The left part of Fig. 4 shows the coefficient matrices $\mathbf{G}_{test}$ of the testing data. The right part of Fig. 4 shows the Euclidean distance between all pairs of testing data in the low-dimensional spaces. In the left and right parts, the testing data that belong to the same category are sequentially aligned along the axis. All the 13 scene categories and 25 dimensional feature space were used to obtain the results in Fig. 4.

Figure 5 visualizes the basis vectors for testing images with different colored markers. Only four scene categories (streets, offices, bedrooms, and living rooms) and four dimensional feature space were used to obtain the results in Fig. 5. The basis vectors were visualized by using the following procedure. A testing vector $\mathbf{x}$ consists of the four basis vectors $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4]$ weighted by $\mathbf{g} = [g_1, g_2, g_3, g_4]^\top$ as follows:

$$\mathbf{x} = g_1\mathbf{f}_1 + g_2\mathbf{f}_2 + g_3\mathbf{f}_3 + g_4\mathbf{f}_4, \tag{17}$$

where $\mathbf{f}_j = [f_{j,1}, \cdots, f_{j,3200}]^\top$ is a 3,200 dimensional basis vector. For each basis vector $\mathbf{f}_j$, we considered to extract specific visual words strongly relevant to the testing vector $\mathbf{x}$ by using the following thresholding method. If $g_j f_{j,v} > T$, we regarded v-th visual word as *relevant* to $\mathbf{x}$. Otherwise, we regarded v-th visual word as *irrelevant* to $\mathbf{x}$. All of the relevant visual words were drawn as colored markers. Since each visual word was extracted from an image patch as shown in Sect. 4.1.1, the colored marker was drawn at the center of the image patch which brought the relevant visual word. If v-th visual word was regarded as relevant, all of the image patches which brought the v-th visual word were marked by the colored markers.

As can be seen from the left part of Fig. 4, the coefficient matrix of D-SNMF became more sparse than D-NMF. Therefore, D-SNMF represented an input vector as a linear combination of fewer basis vectors than D-NMF, meaning that a basis vector of D-SNMF was mostly assigned to only a single category. The one-to-one correspondence between basis vectors and a category can also be observed from Fig. 5. We can clearly see that each basis vector obtained by D-SNMF is strongly associated with a particular category. However, for D-NMF, the basis vectors depicted by yellow markers appeared in all four categories. Such commonly observed basis vectors are not helpful for constructing a low-dimensional feature space.

Feature vectors in a low-dimensional space should have small mutual distances within a class compared to between classes to achieve good classification. One can observe in the right part of Fig. 4 that the difference between within-class distances (the block-diagonal elements) and between-class distances (the off-block-diagonal elements) of D-SNMF is clearer than that of D-NMF, for almost all category combinations. This indicates that supervised training by adding the penalty term to divergence results in superior performance in classification tasks, compared with unsupervised NMF.

4.2.2 Convergence Performance

Figure 6 shows how the cost function of D-SNMF, J_3, decreased during iterations. The first and the second term of J_3 in Eq. (7) are also shown in the figure. In all the experiments, the cost function, i.e., the sum of divergence and the penalty term, did converge after a few hundred iterations. Even though each term went up and down at the beginning of the iterations, both terms minimized toward their end. In

(a) D-NMF (a) D-SNMF

Fig. 5 Visualized basis vectors of (a) D-NMF and (b) D-SNMF: D-NMF and D-SNMF were applied to four scenes (streets, offices, bedrooms and living rooms). Testing data were factorized to four dimensional feature vectors. Each basis vector was depicted as colored makers on the images. Different basis vectors of D-NMF were mixed up within same category. For D-SNMF, only single basis vector was assigned to single category with few exceptions, e.g., "office-specific" vector or "living-room-specific" vector was singled out. (This figure is best viewed in color.)

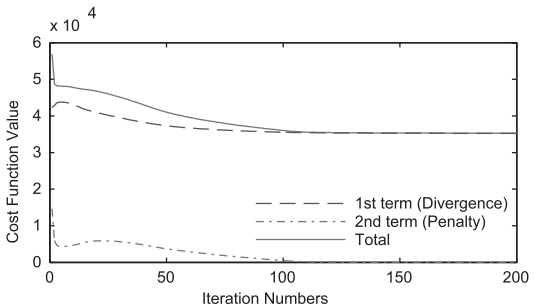


Fig. 6 Cost function of D-SNMF decreased by update rule in Table 4. First term (divergence) and second term (penalty) of cost function are also shown in figure.

the experiments, the update rule of D-SNMF always converged for any input matrix and number of dimensions. The update rule was demonstrated to be very robust for minimizing the cost function if the must-links were reasonably weak.

4.2.3 Comparison of Classification Accuracies

The classification accuracies of D-SNMF, D-NMF, L_2-SNMF, L_2-NMF, PCA, LDA, and MFA are summarized in Figs. 7, 8 and Table 6. To enable the effect of reduced dimensionality to be seen more clearly, classification accuracies without dimensionality reduction are also shown in Fig. 7 and Table 6. Figure 7 shows the relationship between mean accuracy and the number of dimensions. Table 6 lists the best mean accuracy and dimensions for each method. Figure 8 shows the confusion matrices for D-NMF and D-SNMF.

The best mean accuracy of D-SNMF was 76.6%, which was observed when 18 dimensional feature space was used. Our D-SNMF achieved the highest classification rate for the lowest dimensions among the four kinds of NMFs: D-

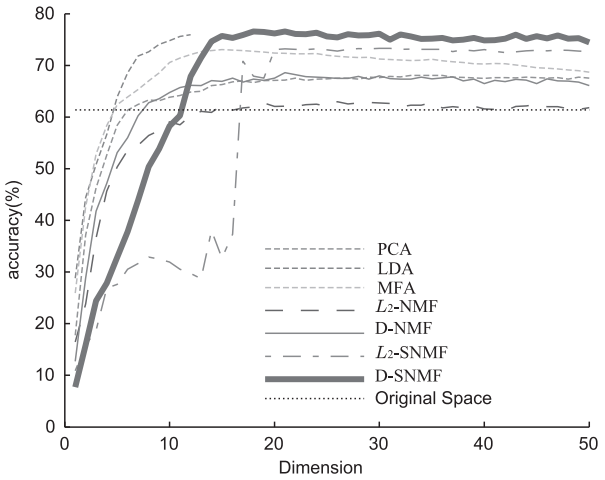


Fig. 7 Relationship between mean accuracy and dimensions.

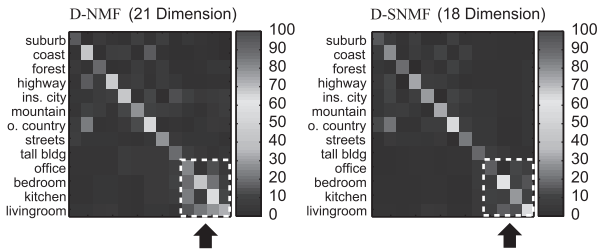


Fig. 8 Confusion matrices of D-NMF (left) and D-SNMF (right). D-SNMF distinguished four scene categories indicated by arrows more clearly than D-NMF.

Table 6 Comparison of best accuracies. Our D-SNMF achieved the highest classification rate for the lowest dimensions among the four kinds of NMFs: D-SNMF, L_2-SNMF, D-NMF and L_2-NMF.

Method	Dim.	Accuracy (%)
org.	3200	61.4±1.7
PCA	35	68.1±1.1
LDA	12	76.0±0.4
MFA	15	73.1±1.1
L_2-NMF	26	63.0±1.2
D-NMF	21	68.6±1.1
L_2-SNMF	31	73.3±0.9
D-SNMF	18	**76.6±0.6**

SNMF, L_2-SNMF, D-NMF and L_2-NMF. The best mean accuracy of D-NMF was at most 68.6%. It was proved that the constraint matrix had a significant effect on classification accuracy. The classification accuracy of D-SNMF was slightly superior to LDA and MFA, which were based on non-sparse expression. The effect of supervised learning can also be clearly observed from Fig. 8. For D-NMF, the classification results of offices, bedrooms, kitchens, and living rooms were confusing, because many image features of these categories resembled each other. D-SNMF ignores such common image features and only extracts features that are specific to each category. Figure 8 showed that D-SNMF clearly distinguished these four categories.

The advantage of divergence could be verified from the results of L_2- and D- (S)NMFs. Although Frobenius norms have been used as a standard metric for NMF, reduced dimensionality by using L_2-NMF had very little effect on histogram features. The best mean accuracy of L_2-NMF was 63.3% and that was almost equivalent to the results for original space. D-NMF performed better than L_2-NMF as is evident from Table 6. Although incorporating supervision into L_2-NMF increased classification accuracy, it was still inferior to D-SNMF. The best mean accuracy of L_2-SNMF was 73.3%, which needed 31 dimensional space. D-SNMF achieved even better accuracy with much smaller dimensions. Furthermore, the classification accuracy of L_2-SNMF tended to drop as dimensions decreased, such as from 1 to 15. The mean accuracy of D-SNMF was always superior to L_2-SNMF for any dimensions.

One reason why the divergence-based NMFs outperformed the L_2-based NMFs is that the generalized divergence regarded frequently occurring visual words as insignificant and less-frequently occurring visual words as significant. This characteristic is very important to improve classification accuracies because natural images include frequently-appearing texture patterns, (e.g., flat texture patterns) which are not informative and should be ignored in classification task. The divergence-based NMFs naturally ignore such insignificant visual words. This philosophy is similar to that of tf-idf weighting scheme, which gives large weights to meaningful visual words. Therefore it makes sense that the tf-idf scheme also contributes to improve object recognition accuracy as reported in [4].

It is interesting to discuss the reason why the classification performances of the supervised methods, D-SNMF and L_2-SNMF, were dropped compared to the other methods in the cases of lower dimensional spaces as shown in Fig. 7. We observed that the supervised NMFs tended to provide many-to-one correspondences from basis vectors to a single category, while the unsupervised NMFs gave many-to-many correspondences from basis vectors to different categories. In the case of low dimensions less than the number of categories, the supervised NMFs did not assign any basis vectors to some categories. However, when the number of dimensions was reached at 13 or more, the supervised NMFs gave good classification performances, because at least one basis vector was assigned to each category. This is an interesting observation.

5. Experiments on Document Dataset

In this section, we demonstrate that our method also works well for document classification. We show that D-SNMF can create an appropriate low-dimensional feature space even in the case where only a small number of training samples are available. The 20 Newsgroups dataset [22], which includes 18,774 documents and 61,188 words collected from 20 different newsgroups, were used in this experiments.

A method for learning and classification is almost the same as the procedure shown in Sect. 4.1.2. The major

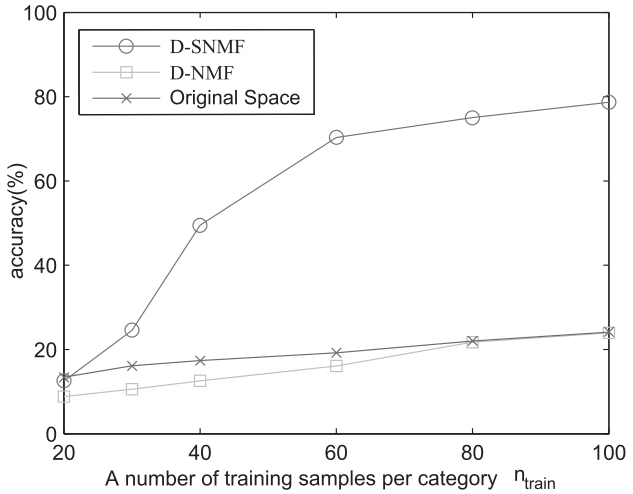


Fig. 9 Classification accuracies of the 20 Newsgroups dataset.

difference from the Sect. 4.1.2 is the way to choose training samples. In each category, n_{train} documents were randomly chosen as training samples and another 50 documents were randomly chosen as testing samples. This means that $20n_{train}$ training samples and 1,000 testing samples were used in total. We observed the classification accuracies of D-SNMF and D-NMF while increasing n_{train} from 20 to 100. A number of topics, k, was set to 20. Thus 61,188 dimensional feature vectors are mapped to 20 dimensional feature space by D-(S)NMFs.

Figure 9 shows obvious difference between D-SNMF and D-NMF. This figure leads to the suggestion that D-NMF failed to discover category-specific topics from the document dataset. On the other hand, D-SNMF succeeded in finding the category-specific topics and gave good classification accuracies even in the case where only a small number of training samples, such as $n_{train} = 60$, were used. We believe that this characteristic is effective to classification problem with a much larger number of categories. Because even if we must cope with tens or hundreds of categories, D-SNMF holds promise for reducing a number of training samples to tractable size with small loss in classification accuracy.

Although LDA provided the second best results in the experiments on the image dataset as shown in Sect. 4, we couldn't tested the LDA on the document datasets because of the high dimensionality of the training/test samples. Since an original feature space is 61,188 dimensions, scatter matrices of size 61188×61188 have 3.7×10^9 elements which require approximately 14 GBytes in single precision. Therefore the computation of eigenvectors of the scatter matrices was infeasible. The memory usage of D-SNMF was so compact compared to LDA that the D-SNMF could cope with such high dimensional features.

We checked a basis matrix, $\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_{20}]$, to find relationship between terms and topics. The element of the basis vectors, $\mathbf{f}_j = (f_{j,1}, \cdots, f_{j,61188})^\top$, were thresholded by a constant value t. If $f_{j,v}$ is larger than t and $f_{i,v}$ is smaller than t for any $i \neq j$, we regarded v-th term as *category-specific term*. In our experiment, t was set to 0.05. Table 7 shows the category-specific terms obtained by D-SNMF. Although the 20 Newsgroups dataset includes similar categories such as *baseball* and *hockey*, D-SNMF assigned them to different basis vectors and succeeded in extracting category-specific words. On the other hand, D-NMF assigned *baseball* and *hockey* categories to the same basis vector. As shown in Table 7, it was easily possible to interpret the meaning of basis vectors obtained by D-SNMF at least 20 category classification. It is still not known if it is possible to interpret the meaning of basis vectors obtained from a dataset with much larger number of categories. We think such large-scale experiment is interesting for future work.

Table 7 Category-specific terms extracted by D-SNMF.

Category	Term
Baseball	san, ball, nl, red, al, jim, least, pretty, maybe, baseball, lost, duke, having, today, runs, bob, park, early, roger, average, hit, career, league, smith, bat, cubs, boggs, braves, pennant, pitching, pitcher, phillies, yankees, sox, hitter, giants, batting
Hockey	buffalo, ny, period, maine, goal, during, show, pre, far, wings, joseph, se, points, april, cup, chicago, andrew, cmu, flyers, night, finland, nj, la, stars, canada, patrick, pts, division, regular, score, sweden, italy, gary, boston, pittsburgh, beat, ericsson, pens, louis, winnipeg, blues, coverage, gerald, nyi, ulf, hockey, gld, jets, stanley, rangers, playoffs, detroit, coach, espn, playoff, leafs, bruins, messier, penguins, nhl, hawks

6. Conclusion and Future Work

We proposed a supervised method of reducing dimensionality for histogram-based features by using NMF framework in this paper. We reformulated NMF with a divergence similarity measure within a supervised learning context by using must-links and cannot-links between input data. A multiplicative update rule for minimizing the newly-defined cost function was also proposed. The experimental results revealed that supervised learning emphasized the sparsity of factorized matrices. All basis vectors obtained from D-SNMF represented a single specific category in most cases. The class-specific characteristics not only had positive effects on classification accuracy but also made it easier to interpret the bases.

It is naturally expected that D-SNMF can also construct appropriate low-dimensional feature spaces for HoG, SIFT, and other types of histogram-based features. Applying D-SNMF to other types of histogram-based features would be an interesting prospect for future work.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," CVPR, pp.886–893, 2005.
[2] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," IJCV, vol.60, no.2, pp.91–110, 2004.
[3] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," Workshop on Statistical

Learning in Computer Vision, ECCV, pp.1–22, 2004.
[4] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," CVPR, pp.2161–2168, 2006.
[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," CVPR, pp.1–8, 2007.
[6] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," CVPR, pp.506–513, 2004.
[7] D.D. Lee and H.S. Seung, "Learning the parts of objects by nonnegative matrix factorization," Nature, vol.401, pp.788–791, 1999.
[8] D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization," NIPS, pp.556–562, 2000.
[9] R. Sandler and M. Lindenbaum, "Nonnegative matrix factorization with earth mover's distance metric," CVPR, pp.1873–1880, 2009.
[10] A. Cichocki, R. Zdunek, A.H. Phan, and S. Amari, Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation, Wiley, 2009.
[11] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, and R.J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," Computational Statistics & Data Analysis, vol.52, no.1, pp.155–173, Sept. 2007.
[12] V.P. Pauca, J. Pipert, and R.J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," Linear Algebra and its Applications, vol.416, pp.29–47, July 2006.
[13] P.O. Hoyer, "Non-negative matrix factorization with sparseness constraints," J. Machine Learning Research, vol.5, pp.1457–1469, 2004.
[14] Z. Chen and A. Cichocki, "Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints," Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep, 2005.
[15] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Fisher non-negative matrix factorization for learning local features," ACCV, pp.27–30, 2004.
[16] J. Yang, S. Yan, Y. Fu, X. Li, and T.S. Huang, "Non-negative graph embedding," CVPR, pp.1–8, 2008.
[17] C. Wang, Z. Song, S. Yan, L. Zhang, and H.J. Zhang, "Multiplicative nonnegative graph embedding," CVPR, pp.389–396, 2009.
[18] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization," Proc. SIAM Int. Conf. on Data Mining, pp.1–12, 2008.
[19] F.F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," CVPR, pp.524–531, 2005.
[20] A.M. Martinez and A.C. Kak, "PCA versus LDA," IEEE Trans. Pattern Anal. Mach. Intell., vol.23, pp.228–233, 2001.
[21] S. Yan, D. Xu, B. Zhang, H.J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, pp.40–51, 2007.
[22] "The 20 Newsgroups data set." http://people.csail.mit.edu/jrennie/20Newsgroups/

Mitsuru Ambai received his B.E., M.S., and Ph.D. in information and computer science from Keio University in 2002, 2004 and 2007, respectively. Currently, he is Senior Engineer of research and development group at Denso IT Laboratory, Inc., Tokyo, Japan. His research interests include image processing and computer vision.

Nugraha P. Utama received B.Eng. in Engineering Physics from Bandung Institute of Technology in 2002, and M.A and Ph.D. in Computational Intelligence and Systems Science from Tokyo Institute of Technology in 2006 and 2009 respectively. Currently he is an assistant Professor at Bandung Institute of Technology. His research interests include Brain-Computer Interface and Image Processing.

Yuichi Yoshida received his B.E., M.S. in engineering science from Osaka University in 2001 and 2003, respectively. From 2003 to 2007, he was a researcher at NTT Corporation. Currently, he is Senior Engineer of research and development group at Denso IT Laboratory, Inc., Tokyo, Japan. His research interests include computer vision and human-computer interaction.