2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# Domain Transfer Nonnegative Matrix Factorization

Jim Jing-Yan Wang, Yijun Sun and Halima Bensmail*

*Abstract*— **Domain transfer learning aims to learn an effective classifier for a target domain, where only a few labeled samples are available, with the help of many labeled samples from a source domain. The source and target domain samples usually share the same features and class label space, but have significantly different In these experiments error of the classifier distributions. Nonnegative Matrix Factorization (NMF) has been studied and applied widely as a powerful data representation method. However, NMF is limited to single domain learning problem. It can not be directly used in domain transfer learning problem due to the significant differences between the distributions of the source and target domains. In this paper, we extend the NMF method to domain transfer learning problem. The Maximum Mean Discrepancy (MMD) criteria is employed to reduce the mismatch of source and target domain distributions in the coding vector space. Moreover, we also learn a classifier in the coding vector space to directly utilize the class labels from both the two domains. We construct an unified objective function for the learning of both NMF parameters and classifier parameters, which is optimized alternately in an iterative algorithm. The proposed algorithm is evaluated on two challenging domain transfer tasks, and the encouraging experimental results show its advantage over state-of-the-art domain transfer learning algorithms.**

## I. Introduction

DOMAIN transfer learning has attracted a lot of attentions from both the research and engineering areas [1]. It has a lot of real-world applications such as wireless WiFi localization [2] and cross-domain text classification [3]. It is defined as a problem of learning an effective classifier from samples of two different domains, which share the same feature space and class label space. One domain is called source domain, and the other one the target domain. In the source domain, data samples are labeled with a class label, which could be used to learn a classifier easily. However, in the target domain, only a few samples are labeled, and the remaining ones are unlabeled. Thus it is difficult to learn the classifier in the target domain for the classification problem. To solve this limitation, domain transfer learning tries to borrow the labeled samples from the source domain for the learning of classifier of the target domain. Nevertheless, due to the significant difference of distributions of the source and target domains, the labeled samples can not be used to learn the target domain classifier directly. For example, in text classification problem where each data sample is a text, we can define the newspaper as the source domain, while the email as the target domain. The newspaper articles are usually labeled as a category of news when it is written, but the email texts are usually not labeled by the email users. Thus when we try to develop a classifier to classify the email texts, only very few labeled samples are available, but many labeled newspaper articles can be used. However, it is obvious that the distribution of newspaper articles and email texts are different, and due to this difference, the classifier learned by using the newspaper articles can not be used directly to classify email texts. We can not combine the newspaper articles and email texts directly to obtain an enlarged dataset, because the difference between different domains may be larger than the difference between different classes.

To overcome this problem, various domain transfer methods have been proposed to learn the classifier for the target domain with the help of source domain. For example, Daume III [4] proposed the Feature Replication (FR) algorithm to augment features for domain transfer learning. Yang et al. [5] proposed the Adaptive SVM (A-SVM) by adapting the new SVM classifier for target domain from an existing classifier learned from the source domain. Jiang et al. [6] proposed the Cross-domain SVM (CD-SVM) by weighting each source domain sample for the learning of the target domain. Bruzzone and Marconcini [7] proposed the Domain Adaptation Support Vector Machine (DASVM) by extending the Transductive SVM (T-SVM) to label unlabeled target samples step by step and also removing some source domain samples at the same time. Duan et al. [1] proposed the Domain Transfer Multiple Kernel Learning (DTMKL) by learning the kernel function and the classifier to minimize the distribution mismatch between the samples from the source and the target domains.

Nonnegative Matrix Factorization (NMF) [8] has been studied very well as a data representation method. Given a nonnegative matrix, where each column is a nonnegative feature vector for a data sample, it tries to factorize it as the product of a basic matrix and a coding matrix, such that each sample could be represented as a coding vector in the coding matrix. The nonnegative constraints are also imposed on the basic matrix and the coding matrix. NMF is a popular data representation method and has been used in various ap-

Jim Jing-Yan Wang is with the University at Buffalo, The State University of New York, Buffalo, NY 14203, USA, and the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, 215006, China (E-mail: jimjywang@gmail.com). Yijun Sun is with the University at Buffalo, The State University of New York, Buffalo, NY 14203, USA. Halima Bensmail is with the Qatar Computing Research Institute, Doha 5825, Qatar.

*Halima Bensmail is the corresponding author.

This work is supported by Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, China (Grant No. KJS1324), and US National Science Foundation (Grant No. DBI-1322212).

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

plications, such as bioinformatics [9], computer vision [10], and pattern recognition [11], [12]. However, surprisingly, it is limited to single domain learning problem, and up to now, no work has been done to extend it to domain transfer learning problem. The only work that makes a breakthrough is [13]. Unlike previous NMF algorithms which are developed either from optimization or probability perspective, this work proposed an innovative NMF algorithm to integrate these two perspectives together. Furthermore, the work also proved that actually NMF algorithms can be transferred from one perspective to the other one. To fill this gap, we investigate in this paper the use of NMF for the transfer learning problem, and propose the first Domain Transfer NMF algorithm (DomTrans-NMF). Our algorithm tries to learn effective and unified basic and coding matrices so that the data samples from both source and target domains can be mapped into a common space with a common distribution. Moreover, the class labels from both the source and target domain samples are used to improve the discriminant ability of the coding vectors by learning a classifier and using it to regularize the coding vectors. The contribution of this paper is listed as follows:

1) To map the source and target domain samples into a common coding vector space with a common distribution. We employ the Maximum Mean Discrepancy (MMD) criteria [14] to reduce the mismatch of the distributions of source and target domain samples' coding vectors.
2) To utilize the class labels of labeled samples from both source and target domains, we also learn a linear classifier for samples of both domains in the coding vector space. The learning of the classifier parameters and the NMF parameter matrices are modeled within one single objective function so that they can be optimized simultaneously. In this way, the classifier can also regularize the coding vectors to improve their discriminant ability.
3) An unified objective function for both NMF and classifier learning is constructed and optimized alternately. Therefore, an iterative DomTrans-NMF is developed for domain transfer learning algorithm.

The remaining of this paper is organized as the following: In section II, we propose the novel domain transfer NMF method; in section III, we evaluate the proposed algorithm by comparing it to several state-of-the-art domain transfer learning algorithm on two challenging domain transfer learning problems; in section IV, we conclude the paper .

## II. Domain Transfer Nonnegative Matrix Factorization

In this section, we introduce the proposed domain transfer nonnegative matrix factorization method.

### A. Objective Function

We suppose that we have a dataset of $N$ samples, and a $D$-dimensional nonnegative feature vector is extracted from each sample. The collection of the feature vectors of the $N$ samples are denoted as $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \in \mathbb{R}_+^D$, where $\mathbf{x}_i$ is the feature vector of the $i$-th sample. In the domain transfer problem, the samples belongs to two different domains — the source domain, and the target domain. The collection of the source domain samples' feature vectors are denoted as $\mathcal{S}$, while that of the target domain samples' feature vectors as $\mathcal{T}$. Thus, $\mathcal{X} = \mathcal{S} \bigcup \mathcal{T}$, and the number of source samples is denoted as $N_S = |\mathcal{S}|$, while the number of target samples as $N_T = |\mathcal{T}|$, so that $N = N_S + N_T$. Moreover, most samples in the source domain are labeled as positive or negative, while only a small number of samples in the target domain are labeled. We denote the collection of labeled samples' feature vectors as $\mathcal{L}$, while the unlabeled as $\mathcal{U}$, so that $\mathcal{X} = \mathcal{L} \bigcup \mathcal{U}$. Similarly, the number of labeled samples are denoted as $N_L = |\mathcal{L}|$, while unlabeled as $N_U = |\mathcal{U}|$, and $N = N_L + N_U$. We also define a class label vector as $\mathbf{y} = [y_1, \cdots, y_N] \in \{+1, -1, 0\}^N$ for all the samples. If the $i$-th sample is labeled as positive, $y_i = +1$; if it is labeled as negative, $y_i = -1$; and if it is unlabeled, $y_i = 0$.

The data samples are organized as a nonnegative matrix $X = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}_+^{D \times N}$, where the $i$-th column is the feature vector of the $i$-th sample. NMF seeks two low-rank nonnegative matrices $U \in \mathbb{R}_+^{D \times K}$ and $V \in \mathbb{R}_+^{K \times N}$, such that their product can approximate the original data matrix $X$, e.g., $X \approx UV$. Each column of $U$ can be referred as a basic vector, and each sample can be approximated as the linear combination of these $K$ basic vectors. The linear combination coefficients of the $i$-th sample are given in the $i$-th column in $V$, $\mathbf{v}_i \in \mathbb{R}_+^K$. In this way, the $i$-th sample, $\mathbf{x}_i$, can be represented by a $K$-dimensional coding vector, $\mathbf{v}_i$. To learn the effective factorization matrices $U$ and $V$ for samples of both source domain and target domains, we construct a unified objective function by combining the following three terms,

- **NMF Loss Term**: To measure the error of the approximation, NMF employs a loss function based on squared Euclidean distance between the original matrix $X$ and the approximated matrix $UV$, which is $\|X - UV\|_2^2$. This loss function should be minimized with respect to $U$ and $V$, so that the approximation could be as accurate as possible. This problem can be modeled as the following minimization problem,

$$\begin{aligned} &\min_{U,V} \|X - UV\|_2^2 \\ &s.t. U \geq 0, V \geq 0. \end{aligned} \tag{1}$$

- **Domain Transfer Term**: To reduce the difference between the coding vector distributions of the source and target domains samples, we employ the MMD criterion which compares the data distributions based on the squared Euclidean distance between the means of samples from two domains in the coding vector space. The mean of source domain sample coding vector is calculated as $\frac{1}{N_S} \sum_{i:\mathbf{x}_i \in \mathcal{S}} \mathbf{v}_i$, while that of the target domain sample coding vectors as $\frac{1}{N_T} \sum_{j:\mathbf{x}_j \in \mathcal{T}} \mathbf{v}_j$. To

reduce the mismatch of the distributions, the following minimization problem is considered with respect to the coding matrix $V$,

$$\begin{aligned} \min_{V} & \left\{ \left\| \frac{1}{N_S} \sum_{i:\mathbf{x}_i \in \mathcal{S}} \mathbf{v}_i - \frac{1}{N_T} \sum_{j:\mathbf{x}_j \in \mathcal{T}} \mathbf{v}_j \right\|_2^2 \right. \\ & \left. = \left\| \sum_{i:\mathbf{x}_i \in \mathcal{X}} \pi_i \mathbf{v}_i \right\|_2^2 = \|V\boldsymbol{\pi}\|_2^2 \right\} \\ & s.t. V \geq 0, \end{aligned} \tag{2}$$

where

$$\pi_i = \begin{cases} \frac{1}{N_S}, & if\ \mathbf{x}_i \in \mathcal{S} \\ -\frac{1}{N_T}, & if\ \mathbf{x}_j \in \mathcal{T} \end{cases} \tag{3}$$

is the domain indicator of the $i$-th sample, and $\boldsymbol{\pi} = [\pi_1, \cdots, \pi_N]^\top$ is the domain indicator vector.

- **Classifier Term**: To utilize the class labels of both the source and target domain samples directly, we also try to learn a classifier in the coding vector space with the labeled samples. Given a coding vector $\mathbf{v} \in \mathbb{R}_+^K$, we try to design a linear classifier, $h(\mathbf{v}) = \mathbf{w}^\top \mathbf{v} + b$, to predict its corresponding class label $y$, where $\mathbf{w} \in \mathbb{R}^K$ is the classifier coefficient vector, and $b \in \mathbb{R}$ is the bias variable. We unify the coding vectors and class labels of the labeled samples in $\mathcal{L}$ to learn the classifier parameter $(\mathbf{w}, b)$. To measure the classification error of the classifier for the labeled samples, we use the squared loss for the $i$-th sample as $\left\|(\mathbf{w}^\top \mathbf{v}_i + b) - y_i\right\|_2^2$, where $\mathbf{v}_i$ is its coding vector and $y_i$ is its true class label. The classifier parameter will be learned to minimize the classification errors of all the labeled samples. To reduce the complexity of the classifier, we further introduce the $l_2$ norm-based regularization $\|\mathbf{w}\|_2^2$ term for $\mathbf{w}$. The classifier learning problem is modeled as the following minimization problem,

$$\begin{aligned} \min_{V,w,b} & \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 + \gamma \sum_{i:\mathbf{x}_i \in \mathcal{L}} \left\|(\mathbf{w}^\top \mathbf{v}_i + b) - y_i\right\|_2^2 \right. \\ & = \frac{1}{2}\|\mathbf{w}\|_2^2 + \gamma \sum_{i:\mathbf{x}_i \in \mathcal{X}} \left\|\left[(\mathbf{w}^\top \mathbf{v}_i + b) - y_i\right] \iota_i\right\|_2^2 \\ & \left. = \frac{1}{2}\|\mathbf{w}\|_2^2 + \gamma \left\|\left[(\mathbf{w}^\top V + b\mathbf{1}_\mathbf{N}) - \mathbf{y}\right] \boldsymbol{\iota}\right\|_2^2 \right\} \\ & s.t. V \geq 0. \end{aligned} \tag{4}$$

where $\iota_i$ is the labeled sample indicator defined as

$$\iota_i = \begin{cases} 1, & if\ \mathbf{x}_i \in \mathcal{L} \\ 0, & if\ \mathbf{x}_i \in \mathcal{U} \end{cases} \tag{5}$$

, $\boldsymbol{\iota} = [\iota_1, \cdots, \iota_N]^\top$ is the labeled sample indicter vector, and $\mathbf{1_N} = \underbrace{[1, \cdots, 1]}_{\mathbf{N}}$ is a $N$-dimensional vector of all ones. The first term in (4) is to reduce the complexity of the classifier, the second term is to reduce the error of the classifier, and $\gamma$ is the trade-off variable to balance these two terms.

By combining the three terms above together, we have the final optimization problem for the domain transfer nonnegative matrix factorization problem as follows,

$$\begin{aligned} \min_{U,V,\mathbf{w},b} & \|X - UV\|_2^2 + \alpha \|V\boldsymbol{\pi}\|_2^2 \\ & + \beta \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 + \gamma \left\|\left[(\mathbf{w}^\top V + b\mathbf{1}_\mathbf{N}) - \mathbf{y}\right] \boldsymbol{\iota}\right\|_2^2 \right\} \\ & s.t. U \geq 0, V \geq 0. \end{aligned} \tag{6}$$

where $\alpha$ and $\beta$ are trade-off parameters. Unlike the usual NMF problem set up, our learning set up has a broader parameter space. We, not only assume that $U$ and $V$ are variables, but also the the classifier parameters $\mathbf{w}$ and $b$. The class labels from both source and target domains will be used to learn the discriminant classifier, and the classifier will further be used to regularize the coding vectors in $V$.

### *B. Optimization*

Direct optimization of (6) is difficult. We adopt the alternate optimization strategy to solve the optimization problem in (6) in an iterative algorithm where in each iteration, one of $U$, $V$ and $(\mathbf{w}, b)$ is optimized while others fixed, and then their roles are switched. The iteration is repeated until a maximum number of iterations is reached.

*1) Optimization of $U$:* By fixing $V$, $\mathbf{w}$ and $b$, and removing the terms irrelevant to $U$, the optimization problem in (6) is reduced to the following one with respect to only $U$,

$$\begin{aligned} \min_{U} & \left\{ \|X - UV\|_2^2 \right. \\ & \left. = Tr(X^\top X) - 2Tr(XV^\top U^\top) + Tr(UVV^\top U^\top) \right\} \\ & s.t.\ U \geq 0. \end{aligned} \tag{7}$$

where $Tr(\cdot)$ denotes the trace of a matrix. To solve this constrained minimization problem, we employ the Lagrangian multiplier method for optimization [15]. We first introduce the Lagrangian multipliers matrix $\Phi \in R^{D \times K}$ to enforce nonnegative constraint of $U \geq 0$. The Lagrangian function of this problem is introduced as

$$\begin{aligned} \mathcal{L} = & Tr(X^\top X) - 2Tr(XV^\top U^\top) \\ & + Tr(UVV^\top U^\top) + Tr(\Phi U^\top) \end{aligned} \tag{8}$$

By using the zero gradient condition, we have

$$\frac{\partial \mathcal{L}}{\partial U} = -2XV^\top + 2UVV^\top + \Phi = 0 \tag{9}$$

By using the KKT condition $[\Phi] \circ [U] = 0$, where $[\cdot] \circ [\cdot]$ denotes the element-wise product of two matrices, we have

$$\left[-XV^\top\right] \circ [U] + \left[UVV^\top\right] \circ [U] = 0 \tag{10}$$

which leads to the following updating rule for $U$,

$$U \leftarrow \frac{[XV^\top]}{[UVV^\top]} \circ [U], \tag{11}$$

where $\frac{[\cdot]}{[\cdot]}$ is the element-wise division of two matrices.

*2) Optimization of $V$:* By fixing $U$, $\mathbf{w}$ and $b$, and removing the terms irrelevant to $V$, we can reduce (6) to the following optimization problem with respect to $V$,

$$\begin{aligned}
&\min_{V} \Big\{ \|X - UV\|_2^2 + \alpha \|V\boldsymbol{\pi}\|_2^2 + \beta\gamma \left\| \left[ (\mathbf{w}^\top V + b\mathbf{1_N}) - \mathbf{y} \right] \boldsymbol{\iota} \right\|_2^2 \\
&\quad = Tr(X^\top X) - 2Tr(U^\top X V^\top) + Tr(U^\top U V V^\top) \\
&\qquad + \alpha Tr(V\boldsymbol{\pi}\boldsymbol{\pi}^\top V^\top) \\
&\qquad + \beta\gamma Tr\left[ (\mathbf{w}^\top V + b\mathbf{1_N} - \mathbf{y})\boldsymbol{\iota}\boldsymbol{\iota}^\top (\mathbf{w}^\top \mathbf{V} + \mathbf{b1_N} - \mathbf{y})^\top \right] \\
&\quad = Tr(X^\top X) - 2Tr(U^\top X V^\top) + Tr(U^\top U V V^\top) \\
&\qquad + \alpha Tr(V\boldsymbol{\pi}\boldsymbol{\pi}^\top V^\top) \\
&\qquad + \beta\gamma Tr(\mathbf{w}\mathbf{w}^\top V\boldsymbol{\iota}\boldsymbol{\iota}^\top V^\top) + \beta\gamma Tr\left[ \mathbf{w}(b\mathbf{1_N} - \mathbf{y})\boldsymbol{\iota}\boldsymbol{\iota}^\top \mathbf{V}^\top \right] \\
&\qquad + \beta\gamma Tr\left[ (b\mathbf{1_N} - \mathbf{y})\boldsymbol{\iota}\boldsymbol{\iota}^\top (\mathbf{b1_N} - \mathbf{y})^\top \right] \Big\} \\
&s.t. V \geq 0.
\end{aligned} \tag{12}$$

We also adopt Lagrange multiplier error of the classifier method to optimize (12). The Lagrangian multiplier matrix $\Psi \in \mathbb{R}^{K\times N}$ is introduced to enforce nonnegative constraint $V \geq 0$. The Lagrange function is given as

$$\begin{aligned}
\mathcal{L} =& Tr(X^\top X) - 2Tr(U^\top X V^\top) + Tr(U^\top U V V^\top) \\
&+ \alpha Tr(V\boldsymbol{\pi}\boldsymbol{\pi}^\top V^\top) \\
&+ \beta\gamma Tr(\mathbf{w}\mathbf{w}^\top V\boldsymbol{\iota}\boldsymbol{\iota}^\top V^\top) + \beta\gamma Tr\left[ \mathbf{w}(b\mathbf{1_N} - \mathbf{y})\boldsymbol{\iota}\boldsymbol{\iota}^\top \mathbf{V}^\top \right] \\
&+ \beta\gamma Tr\left[ (b\mathbf{1_N} - \mathbf{y})\boldsymbol{\iota}\boldsymbol{\iota}^\top (\mathbf{b1_N} - \mathbf{y})^\top \right] \\
&+ Tr(\Psi V^\top)
\end{aligned} \tag{13}$$

The zero gradient condition gives

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial V} =& -2U^\top X + 2U^\top U V + \alpha V\boldsymbol{\pi}\boldsymbol{\pi}^\top + \beta\gamma \mathbf{w}\mathbf{w}^\top V\boldsymbol{\iota}\boldsymbol{\iota}^\top \\
&+ \beta\gamma \mathbf{w}(b\mathbf{1_N} - \mathbf{y})\boldsymbol{\iota}\boldsymbol{\iota}^\top + \boldsymbol{\Psi} \\
=& 0
\end{aligned} \tag{14}$$

We decompose the matrix $\boldsymbol{\pi}\boldsymbol{\pi}^\top$ into positive part $[\boldsymbol{\pi}\boldsymbol{\pi}^\top]^+ \in \mathbb{R}_+^{N\times N}$ and negative part and $[\boldsymbol{\pi}\boldsymbol{\pi}^\top]^- \in \mathbb{R}_+^{N\times N}$, so that $\boldsymbol{\pi}\boldsymbol{\pi}^\top = [\boldsymbol{\pi}\boldsymbol{\pi}^\top]^+ - [\boldsymbol{\pi}\boldsymbol{\pi}^\top]^-$, where $[X]^+$ or $[X]^-$ is defined as a matrix of the same size of $X$ with their elements as

$$\begin{aligned}
[X]_{ij}^+ &= \begin{cases} X_{ij}, & if\ X_{ij} \geq o, \\ 0, & otherwise. \end{cases} \\
[X]_{ij}^- &= \begin{cases} -X_{ij}, & if\ X_{ij} < o, \\ 0, & otherwise. \end{cases}
\end{aligned} \tag{15}$$

Similarly, $\mathbf{w}\mathbf{w}^\top = [\mathbf{w}\mathbf{w}^\top]^+ - [\mathbf{w}\mathbf{w}^\top]^-$, $\mathbf{w}(b\mathbf{1_N} - \mathbf{y}) = [\mathbf{w}(\mathbf{b1_N} - \mathbf{y})]^+ - [\mathbf{w}(\mathbf{b1_N} - \mathbf{y})]^-$. Equation (14) can be rewritten as

$$\begin{aligned}
&-2U^\top X + 2U^\top U V + \alpha V\left[\boldsymbol{\pi}\boldsymbol{\pi}^\top\right]^+ - \alpha V\left[\boldsymbol{\pi}\boldsymbol{\pi}^\top\right]^- \\
&+ \beta\gamma\left[\mathbf{w}\mathbf{w}^\top\right]^+ V\boldsymbol{\iota}\boldsymbol{\iota}^\top - \beta\gamma\left[\mathbf{w}\mathbf{w}^\top\right]^- V\boldsymbol{\iota}\boldsymbol{\iota}^\top \\
&+ \beta\gamma\left[\mathbf{w}(b\mathbf{1_N} - \mathbf{y})\right]^+ \boldsymbol{\iota}\boldsymbol{\iota}^\top - \beta\gamma\left[\mathbf{w}(b\mathbf{1_N} - \mathbf{y})\right]^- \boldsymbol{\iota}\boldsymbol{\iota}^\top + \Psi = 0
\end{aligned} \tag{16}$$

Using the KKT condition $[\Psi] \circ [V] = 0$, we have

$$\begin{aligned}
&\begin{bmatrix} 2U^\top U V + \alpha V\left[\boldsymbol{\pi}\boldsymbol{\pi}^\top\right]^+ \\ +\beta\gamma\left[\mathbf{w}\mathbf{w}^\top\right]^+ V\boldsymbol{\iota}\boldsymbol{\iota}^\top + \beta\gamma\left[\mathbf{w}(b\mathbf{1_N} - \mathbf{y})\right]^+ \boldsymbol{\iota}\boldsymbol{\iota}^\top \end{bmatrix} \circ [V] \\
&= \begin{bmatrix} 2U^\top X + \alpha V\left[\boldsymbol{\pi}\boldsymbol{\pi}^\top\right]^- \\ +\beta\gamma\left[\mathbf{w}\mathbf{w}^\top\right]^- V\boldsymbol{\iota}\boldsymbol{\iota}^\top + \beta\gamma\left[\mathbf{w}(b\mathbf{1_N} - \mathbf{y})\right]^- \boldsymbol{\iota}\boldsymbol{\iota}^\top \end{bmatrix} \circ [V]
\end{aligned} \tag{17}$$

which leads to the following update rule for $V$,

$$V \leftarrow \frac{\begin{bmatrix} 2U^\top X + \alpha V\left[\boldsymbol{\pi}\boldsymbol{\pi}^\top\right]^- \\ +\beta\gamma\left[\mathbf{w}\mathbf{w}^\top\right]^- V\boldsymbol{\iota}\boldsymbol{\iota}^\top + \beta\gamma\left[\mathbf{w}(b\mathbf{1_N} - \mathbf{y})\right]^- \boldsymbol{\iota}\boldsymbol{\iota}^\top \end{bmatrix}}{\begin{bmatrix} 2U^\top U V + \alpha V\left[\boldsymbol{\pi}\boldsymbol{\pi}^\top\right]^+ \\ +\beta\gamma\left[\mathbf{w}\mathbf{w}^\top\right]^+ V\boldsymbol{\iota}\boldsymbol{\iota}^\top + \beta\gamma\left[\mathbf{w}(b\mathbf{1_N} - \mathbf{y})\right]^+ \boldsymbol{\iota}\boldsymbol{\iota}^\top \end{bmatrix}} \circ [V]\,. \tag{18}$$

*3) Optimization of $\mathbf{w}$ and $b$:* By fixing $U$ and $V$, and removing the terms irrelevant to $\mathbf{w}$ and $b$, (6) is reduced to the following problem,

$$\min_{\mathbf{w},b} \beta\frac{1}{2}\|\mathbf{w}\|_2^2 + \beta\gamma \left\| \left[ (\mathbf{w}^\top V + b\mathbf{1_N}) - \mathbf{y} \right] \boldsymbol{\iota} \right\|_2^2 \tag{19}$$

By setting the derivative of the above objective function with respect to $b$ to zero, we have

$$\begin{aligned}
b\mathbf{1_N}\boldsymbol{\iota} &= (\mathbf{y} - \mathbf{w}^\top \mathbf{V})\boldsymbol{\iota} \\
\Rightarrow bN_L &= (\mathbf{y} - \mathbf{w}^\top V)\boldsymbol{\iota} \\
\Rightarrow b &= \frac{1}{N_L}(\mathbf{y} - \mathbf{w}^\top V)\boldsymbol{\iota}
\end{aligned} \tag{20}$$

where $N_L = \mathbf{1_N}\boldsymbol{\iota}$ is the number of labeled samples. By substituting (20) into (19), we have the following optimization problem with respect only to $\mathbf{w}$,

$$\begin{aligned}
&\min_{\mathbf{w}} \beta\frac{1}{2}\|\mathbf{w}\|_2^2 + \beta\gamma \left\| \left(\mathbf{w}^\top V - \mathbf{y}\right) \left[ I - \frac{1}{N_L}\boldsymbol{\iota}\mathbf{1_N} \right] \boldsymbol{\iota} \right\|_2^2 \\
&\quad = \beta\frac{1}{2}\|\mathbf{w}\|_2^2 + \beta\gamma \left\| \left(\mathbf{w}^\top V - \mathbf{y}\right) H \right\|_2^2 \\
&\quad = \beta\frac{1}{2} Tr(\mathbf{w}\mathbf{w}^\top) + \beta\gamma \left[ Tr(VHH^\top V^\top \mathbf{w}\mathbf{w}^\top) \right. \\
&\qquad \left. - 2Tr(VHH^\top \mathbf{y}^\top \mathbf{w}^\top) + Tr(\mathbf{y}HH^\top \mathbf{y}^\top) \right]
\end{aligned} \tag{21}$$

where $H = \boldsymbol{\iota} - \frac{1}{N_L}\boldsymbol{\iota}\mathbf{1_N}\boldsymbol{\iota}$. The closed form solution of $\mathbf{w}$ can be obtained by setting the derivative with respect to $\mathbf{w}$ to zero, as

$$\begin{aligned}
&\mathbf{w} + \gamma\left[2VHH^\top V^\top \mathbf{w} - 2VHH^\top \mathbf{y}^\top\right] = 0 \\
&\Rightarrow \mathbf{w} = \left[I + 2\gamma VHH^\top V^\top\right]^{-1} \left[2\gamma VHH^\top \mathbf{y}^\top\right].
\end{aligned} \tag{22}$$

### C. Algorithm

The iterative algorithm for the proposed domain transfer nonnegative matrix factorization is summarized in Algorithm 1. As we can see from Algorithm 1, the basic and coding matrices should be initialized. To this end, we perform a standard NMF algorithm to the original matrix $X$.

**Algorithm 1** DomTrans-NMF Learning Algorithm.

**INPUT**: Training set, $\mathcal{X}$, and corresponding class label vector $\mathbf{y}$.

Initialize the basic matrix, $U^0$, and the coding matrix, $V^0$.

**for** $t = 1, \cdots, T$ **do**

Update the classifier parameter $\mathbf{w}^t$ and $b^t$ by fixing the basic matrix $U^{t-1}$ and coding matrix $V^{t-1}$ as in (22) and (20) respectively.

Update the coding matrix $V^t$ by fixing the basic matrix $U^{t-1}$ and classifier parameter $\mathbf{w}^t$ and $b^t$ as in (17).

Update the basic matrix $U^t$ by fixing the coding matrix $V^t$ as in (11).

**end for**

**OUTPUT**: The basic matrix $U^T$, the coding matrix $V^T$ and the classifier parameter $\mathbf{w}^T$ and $b^T$.

## III. Experiments

In this section, we conducted two sets of experiments on two challenging domain transfer problems — the motor imagery classification in an EEG-based Brain-Computer Interface (BCI), and the glioblastoma tumor classification.

### A. Experiment I: Brain-Computer Interface

BCI is a technology which can translate human neuronal activities into user commands, by classifying Electroencephalography (EEG) singles according to the imagination of movements [16]. A direct communication pathway between the brain and a computer can be established via BCI. It should be noted that the EEG data acquired on a certain day for one subject may be very different to the EEG data acquired on another day. There are two possible reasons:

1) The brain of the subject may show different electrical activity on different days, due to the different state concerning motivation, fatigue, etc.
2) The EEG recording device may have some changes of electrode positions and impedances on different days.

It is worth noting that the design of a classifier for a BCI system is very challenging when a classifier trained on data acquired on a certain day should classify data recorded in other days without retraining. Thus we treat this problem as a domain transfer learning problem, and treat different days as different domains error of the classifier.

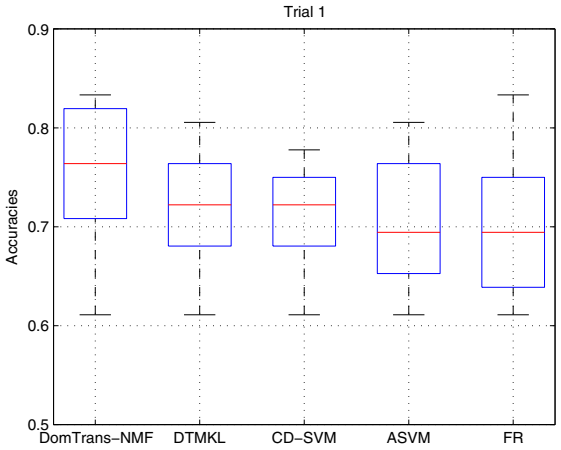
*1) Dataset and Setup:* In In these experiments error of the classifier, the BCI Competition 2008 — Graz data set A [17] was used to evaluate the proposed method. It is composed of EEG data of 9 individuals. The EEG data samples are classified into four different classes according to the motor imagery tasks, which are imagination of movement of the left hand, right hand, both feet, and tongue. The EEG data samples are recorded in two different days. Each day is treated as domain and thus there are two different domains. In each domain, there are in total 288 EEG samples, and 72 samples for each class respectively. We extract the feature vector for each EEG sample using the feature extraction method described in [18].

To conduct the experiment, we carried out two trials using the samples of two different days. In the first trial, we use the samples from the first day as source domain, and the samples from the second day as target domain. Moreover, in another trial of experiment, we switched the roles of these tow days. In each trial, all source domain samples are labeled. We apply the 8-fold cross-validation to the target domain to test the performance of the proposed algorithm. The EEG data sample set of the target domain was split into 8 folds randomly. In each fold, there are 36 samples, and 9 samples from each class of the four classes. One of the 8 folds was used as a test set in turns, while the remaining 7 folds were used as a training set. Such splits were repeated for 8 times. In the training set, only a small partition of the samples are labeled. We randomly selected 20% of them as labeled samples. All the feature vectors of the source domain and target domain training EEG samples were organized as a data matrix, which is further factorized by the proposed algorithm. After the NMF and classifier parameters were learned, we represented and classified the target domain EEG samples using these parameters. To handle the multiple class problem, we employ the one-against-all protocol. The classification results is compared against the true class labels to evaluate the classification performances.

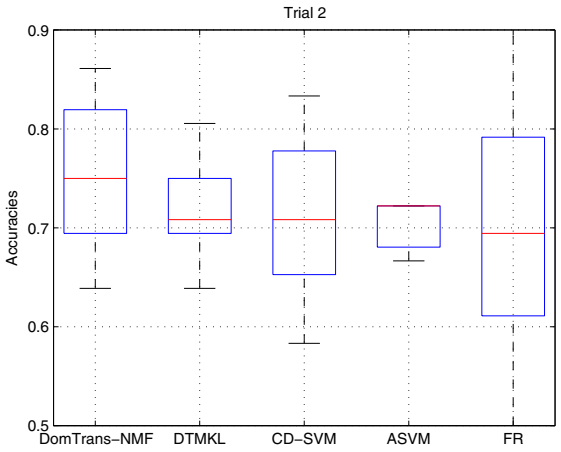
*2) Results:* In the experiment, we compared our algorithm to several popular domain transfer learning algorithms, including FR [4], A-SVM [5], CD-SVM [6] and DTMKL [1]. The boxplots of 8-fold cross-validation accuracy of different methods are given in Figure 1. The results in Figure 1 showed that DomTrans-NMF consistently outperformed other domain transfer learning methods across both two trials. The large improvement of our algorithm over the competing ones in this dataset is likely due to the large inter-domain variation and the part latent nature, which is better captured by both NMF representation and classification function provided by DomTrans-NMF. Interestingly, FR, which is a domain transfer representation algorithm, tends to do worse than the other three domain transfer classifiers, which are DTMKL, CD-SCM and ASVM. It seems that the difference of domain distributions in this dataset could be better modeled by domain transfer classifiers. However in general, we find that it is better to learn both representation and classification parameters, just as the proposed DomTrans-NMF does.

### B. Experiment II:Glioblastoma Grade Classification

In the second experiment, we evaluated the proposed algorithm on the problem of cross-domain glioblastoma grade classification based on the gene expression profile of the tumor samples.




(a) Trial 1




(b) Trial 2

Fig. 1. The boxplots of 8-fold cross-validation accuracy of two trials on the BCI datasst.

*1) Dataset and Setup:* In this experiment, we used two gene expression data sets of two different glioblastoma types — glioma III and IV. These dataset are generated by two different research groups [19], [20], using the same U133A platform, which share the same 12,287 genes. The first gene expression dataset contains 74 tumor samples, 50 of which belong to grade IV, while 24 of which belong to grade III. In the second dataset, there are in total 98 tumor samples, 75 of which belong to grade IV, and 23 of which belong to grade III. Generally speaking, since these two groups investigate the same glioblastoma cancer types using the same genes, the distributions of these two datasets should be similar. However, surprisingly, the distributions are significantly different from each other. The possible reasons are that the researchers of these two groups used different experiment protocol and data processing methods, or the conditions of the glioblastoma patients themselves are different. Thus these two datasets are treated as two different domains, and domain transfer learning is needed to learn from one dataset to another one.

To perform the domain transfer learning experiment, we also conducted two trials of experiments. In each trial, one dataset was used as the source domain while the other one as the target domain. The source domain samples are all labeled, and we perform the 5-fold cross-validation to the target domain samples. The target domain was split into 5 folds randomly, and each fold is used as the test set while the remaining four sets as the training set. In the training set, about 20 % samples were selected as labeled samples, while the remaining samples as unlabeled samples. Then we combined the source domain samples and the target domain training samples (labeled and unlabeled) and performed the proposed DomTrans-NMF algorithm to learn the basic matrix and the classifier parameter. Finally, the learned NMF and classifier parameter were used to represent and classify the target domain test samples.

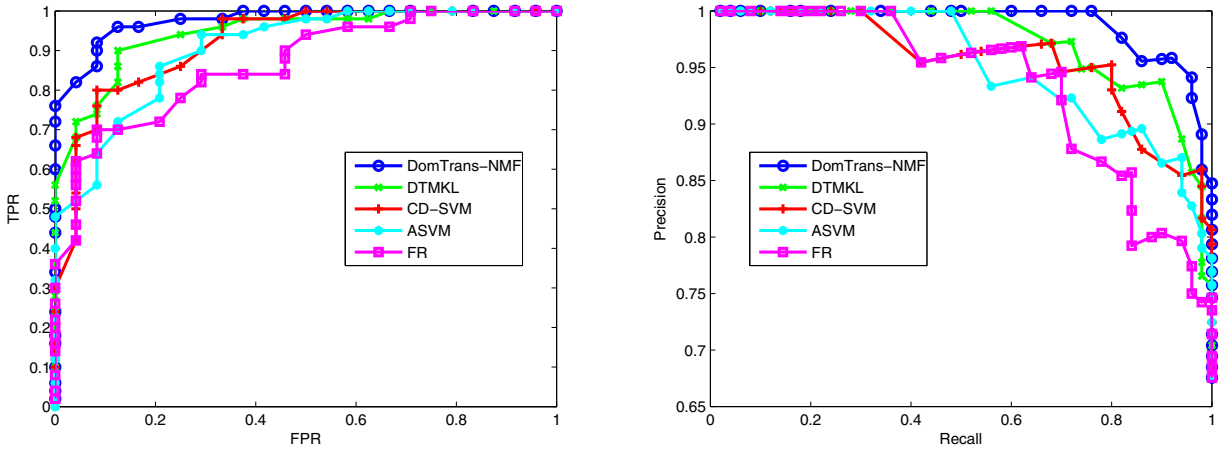
To evaluate the classification performance, we used the receiver operating characteristic (ROC) curve and the recall-precision curve as the performance metrics. The ROC curve is obtained by plotting the true positive rates (TPR) against the false positive rates (FPR) on various classification thresholds, while the recall-precession curve is obtained simply by plotting precision against recall values. A good classifier will have a ROC curve close to the left-top corner of the figure, and a recall-precision curve close to the right-top corner of the figure. We also used the area under ROC curve as a single performance metric.

*2) Results:* The ROC and recall-precision curves of two trials of experiments are given in Figure. 2. Our approach was performing better than the other methods in the first trial. In the second trial, the proposed method achieved the best performance in most cases. The DTMKL algorithm outperforms DomTrans-NMF in the seconde trial when the FPR is larger than 0.2. This is not surprising since it is based on kernel classifier while our algorithm learns a linear classifier, and in most cases, the kernel classifier outperforms the linear classifier. Additionally, our approach outperforms all the other domain transfer classification and representation methods when using the same percent of labeled samples.

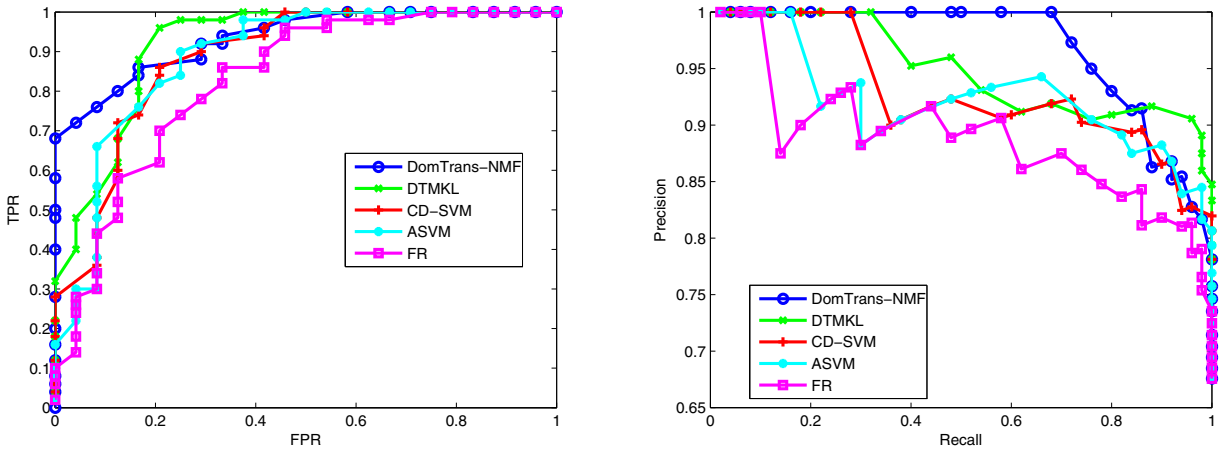
In addition, we provided the AUC values as shown in Figure. 3. In both trials our approach maintain a high AUC value (0.9758 and 0.9338) and outperform the other three competing approaches (CD-SVM, ASVM, and FR) significantly. The only exception is that in the second trial, where DTMKL almost catches up with DomTrans-NMF and the reason for that is it maps the samples from both domain into a kernel space.

## IV. Conclusion

We proposed the domain transfer NMF approach, for data representation in domain transfer learning problem. Our main contribution lies in explicitly reducing the mismatch of




(a) Trial 1




(b) Trial 2

Fig. 2. The ROC and recall-precision curves on the glioma tumor dataset.

coding vector distribution of different domain via the MMD criteria, and learning a linear classifier to utilize the class labels from both domains. The samples from the source domain and target domain can be mapped into a common space expanded In these experiments error of the classifier by the basic matrix. Additionally, our approach is able to learn the NMF and classifier parameters, simultaneously. The experimental results show that the proposed approach yields good classification results on two domain transfer learning datasets. Our approach outperforms recently proposed domain-transfer learning methods including CD-SVM, ASVM and FR. It is also comparable to a kernel based domain transfer classifier — DTMKL.

## REFERENCES

[1] L. Duan, I. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.

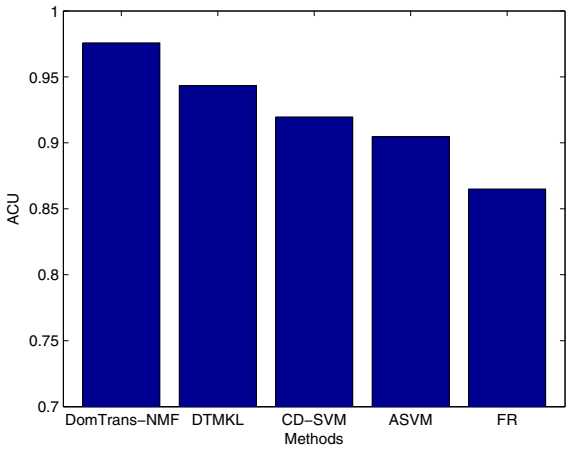
[2] J. Yang and Z. Fei, "Bipartite graph based dynamic spectrum allocation for wireless mesh networks," in *Proceedings - International Conference on Distributed Computing Systems*, 2008, pp. 96–101.

[3] F. Zhuang, Q. He, and Z. Shi, "Effectively constructing reliable data for cross-domain text classification," *IFIP Advances in Information and Communication Technology*, vol. 385 AICT, pp. 16–27, 2012.

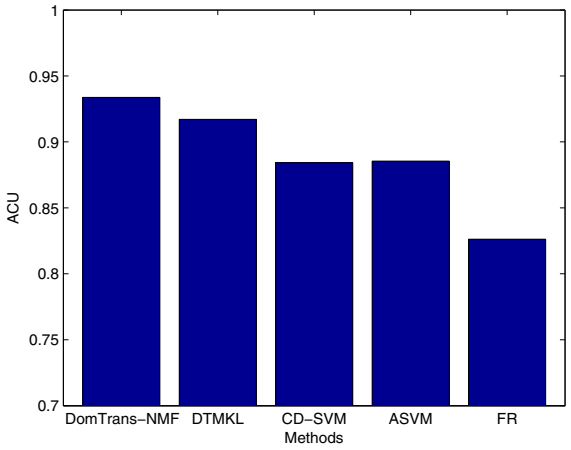
[4] H. Daume III, "Frustratingly easy domain adaptation," in *ACL 2007 - Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007, pp. 256–263.

[5] J. Yang, R. Yan, and A. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, 2007, pp. 188–197.

[6] W. Jiang, E. Zavesky, S.-F. Chang, and A. Loui, "Cross-domain learning methods for high-level visual concept classification," in *Proceedings - International Conference on Image Processing, ICIP*, 2008, pp. 161–164.




(a) Trial 1




(b) Trial 2

Fig. 3. The AUC valuses on the glioma tumor dataset.


[7] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 770–787, 2010.

[8] S. Nikitidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Subclass discriminant nonnegative matrix factorization for facial image analysis," *Pattern Recognition*, vol. 45, no. 12, pp. 4080–4091, 2012.

[9] J. J.-Y. Wang, X. Wang, and X. Gao, "Non-negative matrix factorization by maximizing correntropy for cancer clustering," *BMC Bioinformatics*, p. 107, 2013.

[10] H. Liu, Z. Wu, D. Cai, and T. S. Huang, "Constrained non-negative matrix factorization for image representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[11] Q. Sun, F. Hu, and Q. Hao, "Context awareness emergence for distributed binary pyroelectric sensors," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*. IEEE, 2010, pp. 162–167.

[12] Q. Sun, P. Wu, Y. Wu, M. Guo, and J. Lu, "Unsupervised multi-level non-negative matrix factorization model: Binary data case." *Journal of Information Security*, vol. 3, no. 4, 2012.

[13] Q. Sun, F. Hu, and Q. Hao, "Mobile target scenario recognition via low-cost pyroelectric sensing system: Toward a context-enhanced accurate identification," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 44, no. 3, pp. 375–384, March 2014.

[14] K. Borgwardt, A. Gretton, M. Rasch, H.-P. Kriegel, B. Scholkopf, and A. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[15] S. Tosserams, L. Etman, P. Papalambros, and J. Rooda, "An augmented lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers," *Structural and Multidisciplinary Optimization*, vol. 31, no. 3, pp. 176–189, 2006.

[16] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *Journal of Neural Engineering*, vol. 4, no. 2, pp. R1–R13, 2007.

[17] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "Bci competition 2008–graz data set a," *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, 2008.

[18] H.-I. Suk and S.-W. Lee, "A novel bayesian framework for discriminative feature extraction in brain-computer interfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 286–299, 2013.

[19] H. Phillips, S. Kharbanda, R. Chen, W. Forrest, R. Soriano, T. Wu, A. Misra, J. Nigro, H. Colman, L. Soroceanu, P. Williams, Z. Modrusan, B. Feuerstein, and K. Aldape, "Molecular subclasses of high-grade glioma predict prognosis, delineate a pattern of disease progression, and resemble stages in neurogenesis," *Cancer Cell*, vol. 9, no. 3, pp. 157–173, 2006.

[20] W. Freije, F. Castro-Vargas, Z. Fang, S. Horvath, T. Cloughesy, L. Liau, P. Mischel, and S. Nelson, "Gene expression profiling of gliomas strongly predicts survival," *Cancer Research*, vol. 64, no. 18, pp. 6503–6510, 2004.