Abstract—Multi-task learning has proven to be useful to boost the learning of multiple related but different tasks. Meanwhile, latent semantic models such as LSA and NMF are popular and effective methods to extract discriminative semantic features of high dimensional dyadic data. In this paper, we present a method to combine these two techniques together by introducing a new matrix tri-factorization based formulation for semi-supervised latent semantic learning, which can incorporate labeled information into traditional unsupervised learning of latent semantics. Our inspiration for multi-task semantic feature learning comes from two facts, i.e., (1) multiple tasks generally share a set of common latent semantics; and (2) a semantic usually has a stable indication of categories no matter which task it is from. Thus to make multiple tasks learn from each other we wish to share the associations between categories and those common semantics among tasks. Along this line, we propose a novel joint Nonnegative matrix tri-factorization framework with the aforesaid associations shared among tasks in the form of a semantic-category relation matrix. Our new formulation for multi-task learning can simultaneously learn (1) discriminative semantic features of each task; (2) predictive structure and categories of unlabeled data in each task; (3) common semantics shared among tasks and specific semantics exclusive to each task. We give alternating iterative algorithm to optimize our objective and theoretically show its convergence. Finally extensive experiments on text data along with the comparison with various baselines and three state-of-the-art multi-task learning algorithms demonstrate the effectiveness of our method.

Keywords—multi-task learning; semantic feature learning; semi-supervised learning; joint nonnegative matrix tri-factorization; text classification;

I. INTRODUCTION

Multi-task learning (MTL) refers to learning multiple related tasks together so as to improve the performance of each task relative to learning them separately. Over the past decade, MTL has attracted more and more attention in the community of machine learning and data mining, and has been applied to many important areas including computer vision [1], [2], natural language processing [3], and bioinformatics [4], [5]. As well known, what is crucial in MTL is the knowledge sharing scheme among tasks. Typical approaches in the literature can be roughly categorized into two kinds. One kind are the model based approaches which can be further divided into two forms, i.e., placing a common prior on the model parameters of each task in hierarchical Bayesian models [6], [7], [8], [9], [10] and explicitly sharing some model parameters or model structure among tasks [3]. So far, model based methods have mainly focused on discriminative models such as linear regression and logistic regression. The other kind are the feature based approaches which share a set of common features among tasks in original input space or the transformed feature space [11], [12], [13], [5], [1]. To learn new features, most previous work have used a common feature transformation for all tasks on the original input data. While having the convenience to be formulated, this may be not the best choice in some problems, e.g., in multi-task news classification, sports news about baseball may have rather different key words from the news about hockey since different sports have different terms. In such cases, simply applying the same feature transformation for all tasks will either not help much or lose much useful information, and one may most want to extract the semantic features in each task.

On the other hand, latent semantic models such as LSA [14], NMF [15] and LDA [16] have provided an effective way to learn the semantic features of dyadic data in unsupervised or supervised manner [17], [18]. While unsupervised latent semantic models are popular for dimensionality reduction, their supervised counterparts are similar in spirit to supervised dimensionality reduction [19], which aims at finding a low-dimensional representation of data such that the supervision information can be well fitted by some predictive model that will be used to predict unlabeled data. However, single task supervised feature learning may be not reliable when supervision information is not enough. To improve reliability, we may combine the ideas of supervised and unsupervised semantic models and yield a semi-supervised semantic learning model. Besides, by means of multi-task learning, we can learn multiple such problems jointly to further improve the performance of each problem.

Motivated by these aspects, in this paper, we combine multi-task learning, latent semantic learning and semi-supervised learning, yielding a novel multi-task semantic feature learning method with few labeled and abundant unlabeled data in each task. Our main inspirations come from multi-task text classification where despite of the different marginal distributions among different tasks in raw word feature space, there may be some common semantic topics
shared among tasks, and the associations between those common semantic topics and text categories may remain stable across different tasks. With these stable semantic-category associations being exploited for the knowledge sharing among tasks\(^1\), multiple tasks may learn from each other. To formulate our learning framework, first we provide a new matrix tri-factorization based formulation for semi-supervised latent semantic learning, which can incorporate labeled information into the traditional unsupervised learning of latent semantics and can combine the finding of discriminative semantic features, the learning of predictive structure and the predicting of unlabeled data in a unified optimization framework. Then we extend this single task formulation to the multi-task scenario, by sharing the associations between categories and those common semantics among tasks in the form of a semantic-category relation matrix. Our new formulation for multi-task semantic feature learning essentially is a joint Nonnegative matrix tri-factorization based optimization which can simultaneously learn (1) discriminative semantic features of each task; (2) predictive structure and categories of unlabeled data in each task; (3) common semantics shared among tasks and specific semantics exclusive to each task.

Comparison with typical model based approaches that either directly share same model parameters or place a common prior on the parameters, our method not only shares a part of model parameters but also automatically learns the correspondingly shared features that may have different forms in different tasks.

There also exist key differences between our method and typical multi-task feature selection/learning (MTFS/L). First, typical MTFS only seeks to find a set of common features in the original input space\cite{13, 5}, while our method finds common features in the more meaningful semantic space; second, typical MTFL\cite{13, 1} enforces a common transformation for all tasks on the original input data, while our method allows the common semantic features learned among tasks to have different forms in terms of the multinomial distributions over words; third, most previous studies on MTFS/L mainly use the labeled data, while our method can naturally incorporate usually abundant unlabeled samples in each task for semantic feature learning.

We give alternating iterative algorithm for the optimization of our objective and theoretically show its convergence. Then we construct 144 4-task learning problems from the widely used 20 Newsgroups data set to evaluate our algorithm. Experimental results demonstrate our method can provide superior generalization performance compared with various baselines and several state-of-the-art multi-task learning algorithms.

The remainder is organized as follows. Section II states the notations and briefly covers the necessary preliminaries. Then in Section III we propose our multi-task semantic feature learning algorithm by first introducing a matrix tri-factorization based formulation for semi-supervised latent semantic learning. The experimental results are demonstrated in Section IV and related works are given in Section V. Finally we conclude the paper in Section VI.

II. PROBLEM SPECIFICATION AND PRELIMINARIES

A. Problem Specification and Notations

Assume we have \( T \) classification learning tasks indexed by \( i = 1, \cdots, T \), each of which with few labeled data \( \mathcal{D}_i^L = \{(x_{i,n}, y_{i,n})\}_{n=1}^{N_i} \) but sufficient unlabeled data \( \mathcal{D}_i^U = \{x_{i,n}\}_{n=1}^{N_{i,U}} \). Our objective is to learn these \( T \) tasks together with the hope that they will help each other in the learning process. The learning performance is measured by the averaged classification accuracy over all tasks, and a multi-task learning algorithm is evaluated by its performance improvements over single task learning (learning each task independently) and pooling (simply combine all labeled and unlabeled data in each task together).

We denote the set of real numbers and the set of nonnegative real numbers by \( \mathbb{R} \) and \( \mathbb{R}_+ \) respectively. \( x_i \in \mathbb{R}_{+}^{M_i \times N_i} \) is the word-document matrix\(^2\) in the \( i \)-th task, with \( N_i = N_{i,L} + N_{i,U} \) being the number of total documents in the \( i \)-th task and \( M \) being the number of vocabulary words. \( K \) is the number of total semantic topics in each task, while \( \mathcal{R} \) is the number of shared common topics among tasks. \( C \) denotes the number of data categories in each task. \( X_{ab} \) indicates the \( a \)-th row and the \( b \)-th column element of matrix \( X \), while \( \| \cdot \| \) and \( \text{Tr}(\cdot) \) denote the Frobenius norm and the trace of a matrix respectively.

B. Nonnegative Matrix Factorization (NMF)

As shown in\cite{15}, NMF can learn semantic features of text. It simply aims to factorize a data matrix \( X \) into two nonnegative factor matrices \( F \) and \( G \):

\[
X \approx FG, \quad F \geq 0, \quad G \geq 0,
\]

where \( X \in \mathbb{R}_{+}^{M \times N} \) contains \( N \) data samples in terms of \( M \) features, \( F \in \mathbb{R}_{+}^{M \times K} \) represents \( K \) latent semantics and \( G \in \mathbb{R}_{+}^{K \times N} \) contains the coordinates of each sample in the semantic space spanned by the columns of \( F \).

The nonnegativity constraints makes the learned semantics in NMF more interpretable than that in LSA which uses the Singular Value Decomposition (SVD) technique.

As two-factor NMF is restrictive sometimes, Ding et al. studied Nonnegative Matrix Tri-Factorization (NMTF)\cite{20}:

\[
X \approx FSG, \quad F \geq 0, \quad S \geq 0, \quad G \geq 0.
\]

NMTF can have many interpretations. In this paper, we will endow \( G \) with the label information of data, and the

\(^1\)These associations shared among tasks essentially are a part of model parameters in the latent semantic space of each task, but here the model is not discriminative, thus differs from typical model based MTL methods.

\(^2\)In the sequel, we only take text data as example though our method can address any dyadic data.
labels of training data will be fixed during the optimization. Accordingly, we endow $S$ with the semantic-category relations, which can be properly shared among tasks.

III. MULTI-TASK SEMI-SUPERVISED SEMANTIC FEATURE LEARNING FOR CLASSIFICATION

A. Semi-Supervised Semantic Feature Learning for Classification with NMTF

Let us begin with a semi-supervised classification setting of each learning task, i.e., there are a portion of labeled data and large amount of unlabeled data in each task. We formulate this problem for the $i$-th task as follows:

$$\min_{F_i, S_i, G_i} \| X_i - F_i S_i [G_i L, G_i U] \|^2$$

(1)

where

- $X_i = [X_{iL}, X_{iU}] \in \mathbb{R}_+^{M \times N_i}$ is the word-document matrix in the $i$-th task, with $X_{iL} \in \mathbb{R}_+^{M \times N_{iL}}$, being the labeled data and $X_{iU} \in \mathbb{R}_+^{M \times N_{iU}}$ the unlabeled data;
- $F_i \in \mathbb{R}_+^{P \times K}$ contains the latent semantic topics in the $i$-th task, which can be seen as semantic features;
- $S_i \in \mathbb{R}_+^{K \times C}$ is the semantic-category relation matrix in the $i$-th task;
- $G_i = [G_{iL}, G_{iU}] \in \mathbb{R}_+^{C \times N_i}$ contains the category information of data matrix $X_i$. Those supervision information is injected into $G_{iL}$3 and fixed during the optimization. $G_{iU}$ contains the predicted category information of unlabeled data $X_{iU}$4, which can be initialized by a supervised learner trained on those few labeled data;

For sake of brevity, we refer this formulation as Semi-Supervised semantic Feature Learning for Classification (S$^2$FLC) in the sequel. Note that S$^2$FLC can not only incorporate labeled information into the traditional unsupervised learning of latent semantics, but also combine the finding of discriminative latent semantics, the learning of predictive structure and the predicting of unlabeled data in a unified optimization framework. Zhuang et al. [21] achieved fairly good transfer learning results by using a special case of this formulation to model the fully labeled source domain data.

B. The Proposed Multi-Task Learning Formulation

As stated in the introduction, multiple related tasks generally shares a set of common semantics, and a semantic usually has a stable meaning of categories no matter which task it is from, so we may split the semantic-category relation matrix of each task into two parts: one part is for those common semantics and will be shared across tasks, while the other part corresponds to the specific topics of each task. Along this line of thoughts, we jointly factorize

3If the $n$-th sample in $X_{iL}$ belongs to category $c$, then $(G_{iL})_{c,n} = 1$, and for any $c' \in \{1, \ldots, C\}$, $c' \neq c$, $(G_{iL})_{c',n} = 0$.
4The $n$-th sample in $X_{iU}$ belongs to category $c = \arg \max_k (G_{iU})_{c,n}$.

$T$ word-document matrices $X_i$, $i = 1, \ldots, T$ as follows, with the hope that this will pass knowledge via the common semantic-category associations (contained in $S$) from some “confident” tasks to those unconfident ones:

$$\min_{F_i, S_i, R_i, G_{i}, \geq 0} \sum_{i=1}^{T} \| X_i - [F_i]_{S_i} [G_{iL}, G_{iU}] \|^2$$

(2)

where

- $X_i$, and $G_i = [G_{iL}, G_{iU}]$ have the same meanings as those in the single task S$^2$FLC introduced above;
- $F_i = [F_{is}, F_{ir}] \in \mathbb{R}_+^{M \times K}$ contains the semantic topics in the $i$-th task, with $F_{is} \in \mathbb{R}_+^{M \times K}$, $i = 1, \ldots, T$ containing the common semantic topics among tasks and $F_{ir} \in \mathbb{R}_+^{M \times (K-r)}$ containing the specific topics of the $i$-th task.
- $S_i = [S, R_i] \in \mathbb{R}_+^{K \times C}$ is the semantic-category relation matrix with $S \in \mathbb{R}_+^{K \times C}$ containing the shared semantic-category associations among tasks and $R_i \in \mathbb{R}_+^{(K-r) \times C}$ containing associations between the $i$-th task’s specific topics and text categories;

Note that, (a) common semantic topics among tasks mean that for any $1 \leq i, j \leq T$, $i \neq j$, $F_{is}$ and $F_{js}$ are semantically corresponding to each other but there is no need to satisfy $F_{is} = F_{js}$, which is to say, several corresponding topics have the same semantic relations to text categories, but they may have different forms in terms of the multinomial topic distributions over words; (b) the number $r$ of shared topics can be adjusted to accommodate to the relatedness of tasks. For closely related tasks we may use a large $r$ even equal to the total topics $K$, and for loosely related tasks we may use a small $r$ even equal to zero which is equivalent to no knowledge being shared; (c) we can employ a supervised learner i.e., Logistic Regression to pre-predict those unlabeled data and initialize $G_{iU}$ using these originally predicted probabilities. This pre-predicting process can be done either in the pooling manner or in single task learning manner; (d) since we have learned $F_i$ and $S_i$ during this transductive process, new data that not in the original factorization of $X_i$ can also be predicted by factorizing them according to (2) with $F_i$ and $S_i$ initialized to the original factorization results.

In what follows, we detail on how to solve the joint optimization problem in Eq. (2).

C. Optimization

The objective function $O$ in Eq. (2) is not convex in all its variables, thus, it is unrealistic to find the global minima for it. In the following, we provide an alternating iterative algorithm to obtain the local optima of (2).

Let $D_i = \| X_i - F_i S_i G_i \|^2$, $i = 1, \ldots, T$, then after some simple algebra we have

3Here we used the properties $\text{Tr}(A + B) = \text{Tr}(A) + \text{Tr}(B)$ and $\text{Tr}(AB) = \text{Tr}(BA)$. 

193
\[
D_i = \text{Tr}((X_i - F_i \tilde{S}_i G_i)\mathbf{T} (X_i - F_i \tilde{S}_i G_i)) = \text{Tr}(X_i^T X_i - 2X_i^T F_i \tilde{S}_i G_i + (F_i \tilde{S}_i G_i)^T F_i \tilde{S}_i G_i) \\
= \text{Tr}(X_i^T X_i - 2X_i^T F_i \tilde{S}_i G_i - 2X_i^T F_i \mathbf{r}_i G_i \mathbf{r}_i + (F_i \tilde{S}_i G_i)^T F_i \tilde{S}_i G_i + 2(F_i \tilde{S}_i G_i)^T F_i \mathbf{r}_i G_i i \\
+ (F_i \mathbf{r}_i G_i)^T F_i \mathbf{r}_i G_i)
\]

The partial derivatives of \(D_i\) are (we assume \(G_i = 1\) is fixed, and the matrix variables are \(F_i, G_{iU}, R_i\) and \(S\))

\[
\frac{\partial D_i}{\partial F_i} = 2F_i \tilde{S}_i G_i (\tilde{S}_i G_i)^T - 2X_i (\tilde{S}_i G_i)^T
\]

\[
\frac{\partial D_i}{\partial G_{iU}} = 2(F_i \tilde{S}_i)^T F_i \tilde{S}_i G_{iU} - 2(F_i \tilde{S}_i)^T X_i U
\]

\[
\frac{\partial D_i}{\partial R_i} = 2F_i^T F_i \tilde{S}_i G_i G_i^T + 2F_i^T F_i \mathbf{r}_i G_i G_i^T - 2F_i^T X_i G_i^T
\]

\[
\frac{\partial D_i}{\partial S} = 2F_i^T F_i \tilde{S}_i G_i G_i^T + 2F_i^T F_i \mathbf{r}_i G_i G_i^T - 2F_i^T X_i G_i^T
\]

Now we consider the updating of \(F_i\) when \(F_j, G_i, R_i\) and \(S\) are fixed, where \(i, j = 1, \cdots, T, j \neq i\). For this problem, Eq. (2) is equivalent to the following optimization:

\[
\min_{F_i \geq 0} \|X_i - F_i \tilde{S}_i G_i\|^2
\]

We can solve this constrained quadratic programming problem in many ways, e.g., the Reduced Gradient method and the Interior Point method. However, we find these methods either are complex to implement or converge very slow. Here we make use of the multiplicative updating method \([22, 20]\) which is free from selecting the step size that is needed in gradient based methods. By using the KKT complementarity condition of (8) and note (4), we have

\[
(2F_i \tilde{S}_i G_i (\tilde{S}_i G_i)^T - 2X_i (\tilde{S}_i G_i)^T)_{ab}(F_i)_{ab} = 0
\]

which is equivalent to

\[
(2F_i \tilde{S}_i G_i (\tilde{S}_i G_i)^T - 2X_i (\tilde{S}_i G_i)^T)_{ab} = 0
\]

Solving this equation leads to the following updating rule of \(F_i\):

\[
(F_i)_{ab} \leftarrow (F_i)_{ab} \cdot \frac{(\mathbf{D}_i (\tilde{S}_i G_i)^T)_{ab}}{\text{Tr}(\mathbf{D}_i (\tilde{S}_i G_i)^T)}
\]

Similarly we can get the following multiplicative update equations of other matrix variables in Eq. (2):

\[
(G_{iU})_{ab} \leftarrow (G_{iU})_{ab} \cdot \frac{(F_i \tilde{S}_i)^T X_i U_{ab}}{\text{Tr}(F_i \tilde{S}_i)^T X_i U}
\]

\[
(R_i)_{ab} \leftarrow (R_i)_{ab} \cdot \sqrt{\frac{\sum_{i=1}^{T} F_i^T F_i \tilde{S}_i G_i G_i^T}{\sum_{i=1}^{T} F_i^T F_i \tilde{S}_i G_i G_i^T}}
\]

\[
(S)_{ab} \leftarrow (S)_{ab} \cdot \sqrt{\frac{\sum_{i=1}^{T} F_i^T F_i \tilde{S}_i G_i G_i^T}{\sum_{i=1}^{T} F_i^T F_i \tilde{S}_i G_i G_i^T - 2F_i^T X_i G_i^T}}
\]

For these updating rules we have the following nonincreasing theorem, which is proved in a later subsection using the auxiliary function method.

**Theorem 1.** The objective function \(O\) in Eq. (2) is nonincreasing under the updates of \(F_i, G_{iU}, R_i\) and \(S\) by using Eqs. (11), (12), (13), (14) for each task \(i\).

Finally, Theorem 2 gives an alternating iterative solution to our optimization problem (2).

**Theorem 2.** Iteratively updating \(F_i, G_{iU}, R_i, i = 1, \cdots, T, \) and \(S\) will converge to a local minimum of (2).

**Proof:** Since \(O\) obviously has the lower bound zero and is nonincreasing in each round iteration, so this iterating process will converge to a local minimum of (2) finally.

**D. Learning Algorithm**

Based on Theorem 2, we develop our new multi-task learning algorithm which is summarized in Algorithm 1. To initialize the common topics \(F_{1s}, i = 1, \cdots, T\), we combine all the data in each task and conduct PLSA\(^6\) on them with topic number set to \(R_t\). All elements of \(F_{ir}, i = 1, \cdots, T, \) and \(S\) are initialized to constant \(1\), while the elements of \(R_i, i = 1, \cdots, T\) are randomly initialized as real numbers in \((0, 1)\). Then \(F_i, i = 1, \cdots, T\) are row-normalized while \(\tilde{S}_i, i = 1, \cdots, T\) are column-normalized to provide better initialization. We fix \(G_{iL}\) to the true label information of \(X_{iL}\), and initialize \(G_{iU}\), \(i = 1, \cdots, T\) using the originally predicted category probabilities by pooling Logistic Regression.

**Algorithm 1** Multi-Task Semi-Supervised Semantic Feature Learning for Classification (MTS\(^3\)FLC)

**Input:** the word-document matrix \(X_i \in \mathbb{R}^{M \times N_i}\) in each task, the true label information of \(X_{iL} \in \mathbb{R}^{C \times N_i}\) in each task, the total topic number \(K\) in each task, the shared topic number \(R\), and the maximal iterating number \(maxIter\).

**Output:** classification results of unlabeled data.

1. Normalize \(X_i, i = 1, \cdots, T\) by columns, i.e., let \(\sum_{x_i} x_{i} = 1\).
2. Initialize the matrix variables as \(F_{i}^{(0)}, G_{iU}^{(0)}, S^{(0)}\) and \(R_i^{(0)}\).
3. for \(iter = 1\) to \(maxIter\) do
4. for \(t = 1\) to \(T\) do
5. Update \(R_i\) by Eq. (13).
6. end for
7. Update \(S\) by Eq. (14).
8. for \(t = 1\) to \(T\) do
9. Update \(F_i\) and \(G_{iU}\) by Eqs. (11), (12).
10. end for
11. end for
12. for any \(n = 1, \cdots, N_{t}, i = 1, \cdots, T,\) classify the \(n\)-th unlabeled sample in task \(i\), i.e., \((X_i)_{n}\), into category \(c = \arg\max_{c} (G_{iU})_{cb}\).

For each iteration of MTS\(^3\)FLC, the time complexity w.r.t. word number and document number is \(O(M \cdot \sum_{i=1}^{T} N_i)\).

**E. Convergence analysis**

In this subsection we give a proof of Theorem 1. The proof methods for each matrix variable are similar, so we only take \(F_i\) as example. Consider the updating of \(F_i\) when \(F_j, G_i, R_i\) and \(S\) are fixed, where \(i, j = 1, \cdots, T, \) \(j \neq i\)

\(^6\)http://www.kyb.tuebingen.mpg.de/bs/people/pgehler/code/
i. For this problem, Eq. (2) is equivalent to Eq. (8). To prove the nonincreasing property of the updating Eq. (11), we make use of the auxiliary function [22] method which can be described as follows.

**Definition 1.** A function $H(Y, \bar{Y})$ is called an auxiliary function of $T(Y)$ if it satisfies

$$H(Y, \bar{Y}) \geq T(Y), \quad H(Y, Y) = T(Y)$$

for any $Y, \bar{Y}$.

Then, according to Definition 1 we have

$$T(Y^{(t)}) = H(Y^{(t)}, Y^{(t)}) \geq H(Y^{(t+1)}, Y^{(t)}) \geq T(Y^{(t+1)})$$

which means that $T(Y)$ is non-increasing under the update rule of Equation (20).

Now ignoring unrelated terms in $D_t$ of Eq. (3), we write

$$D_t(F_i) = \text{Tr}(-2X_i^T F_i S_i G_i + (F_i S_i G_i)^T F_i S_i G_i),$$

and construct an auxiliary function of $D_t(F_i)$ as follow:

$$H(F_i, F'_i) = -2 \sum_{ab} (X_i G_i S_i^T)_{ab} (F'_i)_{ab} (1 + \log \frac{F_i}{F'_i})_{ab} + \sum_{ab} (F'_i S_i G_i S_i^T)_{ab} (\frac{F_i^2}{F'_i})_{ab} \geq \text{Tr}(S_i^T A S_i)$$

Obviously, when $F_i = F'_i$ the equality $H(F_i, F'_i) = D_t(F_i)$ holds. By comparing each term in $H(F_i, F'_i)$ and $D_t(F_i)$, we can also show the inequality $H(F_i, F'_i) \geq D_t(F_i)$ holds. E.g., the first term in $H(F_i, F'_i)$ is always smaller than the first term in $D_t(F_i)$, because of the inequality $z \geq 1 + \log(z), \forall z > 0$. The second term in $H(F_i, F'_i)$ is always bigger than the second term in $D_t(F_i)$, due to the following Proposition.

**Proposition 1.** For any matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{k \times k}$, $S \in \mathbb{R}^{n \times k}$, $S' \in \mathbb{R}^{n \times k}$, and $A$, $B$ are symmetric, the following inequality holds

$$\sum_{i=1}^{n} \sum_{p=1}^{k} (AS'B)_{ip} S^2_{ip} \geq \text{Tr}(S^T A S)$$

Then, we minimize $H(F_i, F'_i)$ while fixing $F'_i$. This can be achieved by letting

$$0 = \frac{\partial H(F_i, F'_i)}{\partial (F'_i)_{ab}} = -2(X_i (S'_i G_i)^T)_{ab} (F'_i)_{ab} + 2 (F'_i S'_i G_i (S'_i G_i)^T)_{ab} (F_i)_{ab} (F'_i)_{ab}$$

Solving for $(F_i)_{ab}$, we get

$$(F_i)_{ab} = (F'_i)_{ab} \cdot \sqrt{\frac{X_i (S'_i G_i)^T}_{ab} (F'_i S'_i G_i (S'_i G_i)^T)_{ab}}$$

Thus, we have proved the objective function is nonincreasing under the updating rule (11).

### IV. Experiments

In this section, our algorithm is evaluated on the widely used 20 Newsgroups data set (Section V-A will detail on how this data set can be used to construct many 4-task learning problems). We compare our algorithm with various baselines and three state-of-the-art multi-task learning algorithms. We also visualize the learned semantic topics and investigate the parameter effect of our algorithm.

#### A. Data Preparation

20 Newsgroups\(^8\) is a widely used benchmark data set for text classification, which has approximately 20,000 newsgroup documents that are partitioned evenly into twenty different newsgroups. Since some of the newsgroups are very closely related, a part of these twenty newsgroups are further grouped into four top categories, e.g., the top category sci contains four subcategories sci.crypt, sci.electronics, sci.med and sci.space. All the four top categories and their subcategories are listed in Table I.

<table>
<thead>
<tr>
<th>Top Categories</th>
<th>Subcategories</th>
</tr>
</thead>
<tbody>
<tr>
<td>comp</td>
<td>comp.graphics, comp.os.ms-windows.misc, comp.sys.{ibm,pc,hardware, mac.hardware}</td>
</tr>
<tr>
<td>sci</td>
<td>sci.astronomy, sci.crypt, sci.electronics, sci.med, sci.space</td>
</tr>
<tr>
<td>talk</td>
<td>talk.politics.{guns, mideast, misc}, talk.religion.misc</td>
</tr>
</tbody>
</table>

Now we detail on how to re-organize this corpus to be used to evaluate our multi-task learning algorithm. Here we focus on 2-class classification although our algorithm can naturally deal with multi-class classification problem. It is easy to see that we can select any two of the four top categories sci, talk, comp and rec to construct 2-class classification data set. For the combination sci vs. talk, we construct a 2-class learning task by selecting one subcategory from the top category sci as the positive samples and one subcategory from talk as the negative samples. Since there are four subcategories in each top category, we can get an 4-task learning problem from the data set sci vs. talk by selecting subcategories in the order in Table I. The constructed 4-task classification problem is suitable for multi-task learning due to the facts that 1) the data in each of the four tasks are drawn from different distributions since they are from different subcategories; 2) the four tasks are related to each other since the positive (negative) samples in each task are from the same top categories. Similarly, we can also construct 4-task learning problems from the combinations comp vs. talk, rec vs. talk, sci vs. rec, sci vs. comp, and rec vs. comp. Therefore, we can totally get six orderly constructed 4-task learning problems from 20 Newsgroups. We can also randomly select subcategories

\(^8\)http://people.csail.mit.edu/jrennie/20Newsgroups/
when construct 4-task classification problems from any two top categories. There are totally $A_i^2 = 24$ such randomly constructed 4-task problems for each pair of top categories. We will first present results on the six orderly constructed problems then report the average results over all randomly constructed problems.

For all documents from the four top categories in 20 Newsgroups, we used $tf \cdot idf$ weighting scheme to represent them and all words with document frequency less than 25 were removed resulting in a vocabulary with 8198 terms.

B. Compared Algorithms and Evaluation Metric

We compare our multi-task learning algorithm MTS$^3$FLC with the following algorithms:

- Five baselines, i.e., the single task supervised classification algorithm Logistic Regression (STL-LG), the single task semi-supervised classification algorithm Transductive Support Vector Machine (STL-TSVM), the single task S$^3$FLC introduced in this paper (STL-S$^3$FLC), the supervised pooling (Pooling-LG) and the semi-supervised pooling (Pooling-TSVM);
- Multi-task feature learning (MTFL) [13]: shares a set of common features among tasks;
- Group multi-task feature learning (GMTFL) [1]: learns by the word clustering results of Five baselines, i.e., the single task supervised classification algorithm Transductive Support Vector Machine (STL-TSVM), the single task S$^3$FLC introduced in this paper (STL-S$^3$FLC), the supervised pooling (Pooling-LG) and the semi-supervised pooling (Pooling-TSVM);
- Bayesian multi-task learning with gaussian process and SVM such randomly initialized by single task Logistic Regression since we assume we are learning each task independently. Other initializations of variables are the same as that in MTS$^3$FLC.

The maximal iterating number is set to $maxIter = 20$ in all experiments of MTS$^3$FLC and S$^3$FLC as we empirically find they converge very quickly.

D. Experimental Results

1) Comparison on the 20 Newsgroups Data: In this subsection the proposed algorithm MTS$^3$FLC is compared with various baselines and three state-of-the-art multi-task learning algorithms. The shared topic number in MTS$^3$FLC is set to $R = 5$ for all experiments conducted here.

We first evaluate these algorithms on the six orderly constructed 4-task learning problems. For each of these six problems we vary the number of labeled samples in each task from 2 to 20, since as well known the merits of multi-task learning are more apparent when few data per task are available. For clarity, we present the comparison results between MTS$^3$FLC and the five baselines and the results between MTS$^3$FLC and other MTL algorithms separately. The former are shown in Figure 1 while the latter in Figure 2, and all these results are averaged over 30 independent trials (in each trial we randomly sample the labeled data and test on the remaining data). From Figure 1 we can see that owing to the exploration of unlabeled data and the elaborate knowledge sharing scheme, MTS$^3$FLC can provide very good predictive performance, and can substantially outperform the single task learning methods STL-LG, STL-TSVM and STL-S$^3$FLC as well as the pooling methods Pooling-LG and Pooling-TSVM on all six problems.

For the results in Figure 2, we directly applied MTFL$^1$ and BMTLGP$^12$ on the original 8198-dimensional bag-of-words data described in Section IV-A while PCA was used to reduce the dimension to 500 (as the author did in their paper) for GMTFL$^13$ on each of the six problems. The linear covariance function was used in BMTLGP (according to the author’s advice) since the high-dimensional text data makes the ARD covariance function infeasible. After careful investigation, the group number and the parameter $\gamma$ in GMTFL were set to 2 and 10 respectively, while the parameter $\gamma$ in MTFL was set to 0.01. From the results, we can clearly see that MTS$^3$FLC is consistently superior to all the three compared MTL algorithms on all six problems.

Then we conduct comparison experiments on all the 24 randomly constructed 4-task learning problems for each pair of top categories from 20 Newsgroups. There are totally 144 such problems since we have $C_4^2 = 6$ pairs of top categories. The experimental setting for each algorithm are the same as above, but here we only vary the number of labeled samples per task among $\{2, 10, 20\}$ because so many 4-task text classification problems are very time-consuming. We

---

9http://research.microsoft.com/~minka/papers/logreg/
10http://svmlight.joachims.org/
11http://tic.uchicago.edu/~argyriou/code/index.html/
12http://homepages.inf.ed.ac.uk/gsanguin/software.html/
13http://www-scf.usc.edu/~zkang/GroupMTLCode.zip

196
repeat the experiment 5 times (with labeled data randomly sampled every time) for each problem, and compute the mean accuracy. Finally the averaged results over all 24 problems for each pair of top categories are listed in Table II, from which we see obvious advantage of MTS$^3$FLC again.

2) Parameter Effect: Our preliminary experiments show that the performance of MTS$^3$FLC is not sensitive to the total topic number $K$ when $K$ chooses values in [20, 100], so here we mainly investigate the shared topic number $\bar{r}$ how affect the performance of MTS$^3$FLC. To this end, we fix $K = 20$ and vary $\bar{r}$ in the interval $[0, K]$. The averaged results over 30 repetitions on the six orderly constructed 4-task learning problems from 20 Newsgroups are shown in Table III (for comparison we also listed the results of LG and Pooling-LG), where we set the number of labeled samples per task to $\text{trainsize} = 6$ for the upper three problems and $\text{trainsize} = 14$ for the lower three problems to study whether the parameter effects of $\bar{r}$ are identical under different training sizes. From Table III we can see MTS$^3$FLC generally performs well in a wide range of $\bar{r}$, e.g., on the combination $\text{sci vs. rec}$ MTS$^3$FLC achieves the accuracy over 95% for arbitrary $0 < \bar{r} \leq K$, which means that we can choose $\bar{r}$ freely and boldly when we have no prior information of the task relatedness (if we have prior information, we may use a large $\bar{r}$ even equal to $K$ for closely related tasks, and a small $\bar{r}$ even equal to zero for loosely related tasks). We can also see that MTS$^3$FLC with medium large $\bar{r}$ performs better than both MTS$^3$FLC with too small $\bar{r}$ and MTS$^3$FLC with very large $\bar{r}$, e.g., MTS$^3$FLC with $\bar{r} = 5$ generally performs well than MTS$^3$FLC with $\bar{r} = 3$ and MTS$^3$FLC with $\bar{r} = 13$.

3) Visualization of The Learned Semantic Topics: To show the capability of MTS$^3$FLC to automatically find the common semantics shared among tasks and specific semantics exclusive to each task, we choose 20 most probable key words to express each topic according to the word clustering information $F_i$, $i = 1, \cdots, T$. Table IV lists one common topic among the four tasks and one specific topic of each task on the orderly constructed 4-task learning problem from $\text{comp vs. rec}$. Here, we set the number of shared topics to $\bar{r} = 5$ and the number of labeled samples per task to $\text{trainsize} = 10$. From these results, it can be seen that MTS$^3$FLC can effectively capture the common semantic topics among tasks using same or different key words about the topic, e.g., Topic 1 is a common topic among tasks discussing “computer” (The associations between such topic and the categories are shared among tasks for multi-task classification). Moreover, MTS$^3$FLC can also find the specific topics of each task which are generally expressed by different key words, e.g., Topic 2 in Table IV is a specific
Figure 2. Comparison with state-of-the-art MTL algorithms on the six orderly constructed 4-task classification problems from 20 Newsgroups.

V. RELATED WORK

Multi-task learning [23] has been the focus in the community of machine learning and data mining during the past ten years, and many algorithms on it have been proposed. In [13], Argyriou et al. proposed a multi-task feature learning method which learns a shared sparse representation for multiple related tasks. They show their $l_{1,2}$ norm based formulation is equivalent to a convex optimization problem and can be solved efficiently. Later, Zhang et al. [5] proposed a probabilistic interpretation of the general multi-task feature selection problem using $l_{1,2}$ or $l_{1,\infty}$ norm. Recently, a group multi-task feature learning method is proposed in [1], which assumes the tasks exist in groups and the tasks within each group share features. Our model is related to these works since we are also learning features for each task. However, our model is more flexible to capture the semantic features that exist in many dyadic data including text, image and gene expression. Besides, usually abundant unlabeled samples in each task is naturally incorporated in our model, which makes feature learning more reliable.

Since we have used unlabeled data during the learning phase, our algorithm should be categorized into semi-supervised multi-task learning [9], [4]. Though as two major directions to improve supervised learning in case of labeled samples are insufficient, semi-supervised learning and multi-task learning have been investigated extensively respectively, few works have combined these two lines together. While the manifold properties of unlabeled data are well exploited in [9], our method utilizes unlabeled data to learn not only the more robust predictive structures but also the common and specific semantic features of each task.

Our model is also related to a transfer learning model named MTrick [21], which shares the associations between categories and all semantics in source domain and target domain. Though multiple related tasks (domains) typically share semantics, treating all semantics as the shared ones may be not appropriate. In fact, our experiments on 20 Newsgroups data show the best number of shared semantics among tasks even less than a half of the total topic number.

Besides, the idea of distinguishing common semantic topics from specific ones is also considered in a recent work called Group Matrix Factorization (GMF) [24], which aims at scaling up topic modeling approaches and assumes there exist class-specific topics for each text class as well as shared topics across all classes. However, GMF only simply supposed that a shared topic has the same word distribution in each class, which differs from the semantic corresponding idea considered in this paper.
VI. Conclusions and Future Work

In this paper, we have proposed a novel multi-task semantic feature learning algorithm MTS^{3}FLC with few labeled and abundant unlabeled data in each task. The algorithm shares the associations between categories and common semantics among tasks and formulates its objective as a joint Nonnegative matrix tri-factorization based optimization which can simultaneously learn the topic correspondences among tasks and predict the categories of unlabeled data. We have developed alternating iterative algorithm to optimize our objective and theoretically showed its convergence. Extensive experiments on text data demonstrate our method can provide superior generalization performance compared with various baselines and several state-of-the-art multi-task learning algorithms.

Though we only studied text classification in this paper, MTS^{3}FLC is also applicable to many other learning problems, e.g., image classification in computer vision, where we have many "visual words" as the raw features, based on which we can also induce higher level semantic features.

For the future work, one challenging but important issue is to automatically resolve the appropriate number of shared semantics among tasks according to their relatedness. Luckily, the recent progress in nonparametric Bayesian has provided us a possible way to achieve this. Another issue that matters is to combine graph regularization with our framework to exploit the geometric structure of data. Since close samples tend to have the same label, predicting labels smoothly may further improve performance. We will report these results in a longer version of the paper.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61175052, 60975039, 61203297, 60933004, 61035003), National High-tech R&D Program of China (863 Program) (No. 2012AA011003), and National Program on Key Basic Research Project (973 Program) (No. 2013CB329502).
Table IV

<table>
<thead>
<tr>
<th>Topic 1 (Common topic)</th>
<th>Task 1</th>
<th>graphics.gif,thanks,format,files,tiff,fig,eb_image,program,bit,color,images,pct,advance,hi,va,visa,vga,xf,ftp,card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 2</td>
<td>windows,dos,files,os,drivers,driver,mouse,files,at,card,printer,thanks,program,window,version,ini,ns,win,memory,apps</td>
<td></td>
</tr>
<tr>
<td>Task 3</td>
<td>scsi,ide,drive,controller,card,mb,cpu,bus,drives,usa,pc,vib,disk,os,bios,mb,system,motherboard,flippy,port</td>
<td></td>
</tr>
<tr>
<td>Task 4</td>
<td>mac,apple,le,monitor,mb,drive,centris,simms,quadra,mb,scsi,card,lj,duo,thanks,test,video,ftp,lciii,ed</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic 2 (Specific topic)</th>
<th>Task 1</th>
<th>graphics,thanks,please,looking,where,format,imag,any,find,anyone,advance,can,mail,me,comp,hi,files,polygon,files,am</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 2</td>
<td>bike,dod,shaw,com,ride,my,motorcycle,bmw,you,drive,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bikes,riding,honda, rider,ca, dog, on, do, your, nec</td>
<td></td>
</tr>
<tr>
<td>Task 3</td>
<td>mattingly,he,career,nj,tesla,com,baseball,game,edu,year,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>games,team, his, cubs, runs, was, hit, pitcher, sox, last</td>
<td></td>
</tr>
<tr>
<td>Task 4</td>
<td>games, hockey, rhp, team, he, games go, mb, brains, was,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>wings, win, ca, season, detroit, play, pens, gm, players, they</td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES


