Applied Intelligence (2021) 51:3421–3434
https://doi.org/10.1007/s10489-020-01954-3

A new multi-task learning method with universum data

Yanshan Xiao[1] · Jing Wen[1] · Bo Liu[2]

Accepted: 16 September 2020 / Published online: 13 November 2020

Abstract

Multi-task learning (MTL) obtains a better classifier than single-task learning (STL) by sharing information between tasks within the multi-task models. Most existing multi-task learning models only focus on the data of the target tasks during training, and ignore the data of non-target tasks that may be contained in the target tasks. In this way, Universum data can be added to classifier training as prior knowledge, and these data do not belong to any indicated categories. In this paper, we address the problem of multi-task learning with Universum data, which improves utilization of non-target task data. We introduce Universum learning to make non-target task data act as prior knowledge and propose a novel multi-task support vector machine with Universum data (U-MTLSVM). Based on the characteristics of MTL, each task have corresponding Universum data to provide prior knowledge. We then utilize the Lagrange method to solve the optimization problem so as to obtain the multi-task classifiers. Then, conduct experiments to compare the performance of the proposed method with several baslines on different data sets. The experimental results demonstrate the effectiveness of the proposed methods for multi-task classification.

Keywords Multi-task learning · Universum learning · SVM · Prior knowledge

1 Introduction

Traditional machine learning always focuses on learning a model from one task, which can be called as single-task learning [1–3]. However, we can meet the learning case in which the data come from several domains, and each domain data have similar distribution with each other. We then need to build a model catering for the multi-task data, in which each task can help other tasks to build its predictive model. This is always called multi-task learning [4, 5]. Compared with single-task learning, multi-task learning can make better use of the information contained in related tasks to help build the model. To date, multi-task learning has been successfully used in many applications, such as in speech recognition [6], natural language processing [7], images recognition [8]. For example, in images recognition, the work in [9] uses a multitasking learning framework to build models that take advantage of various types of tag information to represent clothing images in a more granular way.

Over the years, quite a number of MTL methods are proposed, and they can be broadly grouped into the categories of SVM-based methods [10, 11], neural work-based methods [12, 13], and Bayesian based methods [14, 15]. For the SVM-based methods, each task shares a common parameters for the classifier, and builds a classifier similarly. For example, the work in [16], combines $\ell_{2,1}$ norm-regularization and hinge loss function to deal with feature selection problem. For the neural network-based method, each task has shared hidden neurons, and features in shared neurons are allowed to be used by other tasks to promote joint learning. For example, in the work [17], the authors use modular knowledge representation instead of hidden neurons to promote neural network multi-task learning. From the perspective of Bayesian-based methods, authors apply Bayesian optimization to parameter tuning to achieve better multitasking learning. A normal example is [18], Pearce et al. combine Bayesian optimization with Gaussian process to take advantage of covariance in task and parameter space to obtain optimal parameters. Compared with single-task learning methods, multi-task learning can train multiple models at the same time, and these models

✉ Bo Liu
csboliu@163.com

1 Department of Computer Science, Guangdong University of Technology, Guangzhou, Guangdong, China

2 Department of Automation, Guangdong University of Technology, Guangzhou, Guangdong, China

influence each other to improve generalization performance. Even if there is only one goal to solve the problem, the auxiliary model in multi-task learning can still improve the performance of the target task model. Although multi-task learning is widely used, the training of the model requires a lot of data, the computational complexity is high, and useful information is easily lost in the training process.

In real life, the datasets of many problems are multiple tags, most of them used for multitasking learning model training contains only positive and negative labels. Those that do not belong to any positive or negative label are called Universum data [19, 20]. Though Universum data does not belong to the indicated classes, they belong to the same domain as the problem of interest and are helpful for the target problem. Unlike semi-supervised unlabeled data, Universum data does not belong to indicated labels. As shown in Fig. 1, when the model do not use Universum data, some samples are misclassified by the dotted hyperplane. However, with the help of Universum data, the classifier hyperplane is refined and can correctly classify the misclassified samples [21]. The main advantage of Universum learning is that Universum data is easy to obtain, and the prior knowledge provided can better improve the performance of the model, and has good performance in some classification problems. For example, in text classification [22], Liu et al. use confidence as an aid to incorporate Universum data into the learning process and design a Universum logistic regression method to solve the problem of text classification. In graph classification, Pan et al. [23] propose a mathematical programming algorithm called ugBoost, which integrates discriminative subgraph selection and margin maximization into a unified framework to fully exploit the Universum data. Then, Richhariya et al. [24] use Universum data generated based on data information entropy to improve the effect of the human face recognition model.

In this paper, we use SVM-based methods to reduce the computational complexity of multi-task learning, and use Universum learning to discover hidden information in the training set. Considering the characteristics of multi-task learning, we assign a priori knowledge encoded by Universum to each task to improve data utilization. In addition, the existing methods combined with Universum are oriented to single-task learning. Considering the wide study and applications of multi-task learning, it is necessary to study the problem of multi-task learning with Universum data, and Universum learning can improve the data utilization rate of multi-task learning. In all, the main contribution of our work can be summarized as follows:

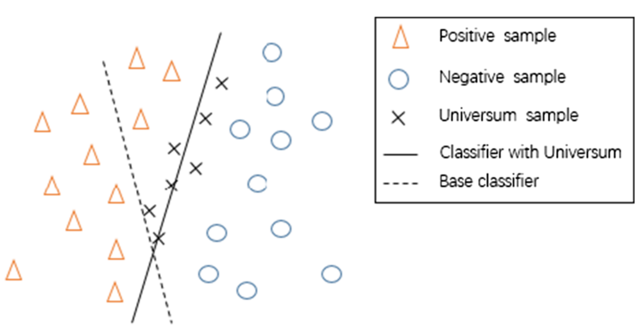


Fig. 1 Description of the influence of Universum on classification tasks

- In order to construct prior knowledge about data distribution in multi-task classification, we incorporate Universum learning into multi-task learning and propose U-MTLSVM method. The proposed model uses a hard parameter μ to associate each task together. In order to make full use of Universum data, which is applied to encode prior knowledge information of the training set, each task has corresponding Universum data. Then, we construct a hyperplane based on the original data and Univerusm data, and make the Universum data located near the hyperplane to obtain an accurate classifier.
- In order to optimize the Universum multi-task classifier with Universum data, we use the Lagrangian multiplication to convert the source objective model into its dual problem, and then optimize the model to obtain the classifier with help of the Universum data. U-MTLSVM not only integrates the information in the original sample, but also integrates the information implicit in the Universum sample into the model, which improves the data utilization rate and the generalization ability of the model. In this way, we can handle the multi-task classification problem when the Universum data is available in practice.
- We conduct extensive experiments to evaluate the performance of our proposed U-MTLSVM framework. The experimental results show that our method is better than the state-of-the-art MTL methods in terms of performance and sensitivity to noise.

The rest of this paper will be arranged as follows: Section 2 introduces related works of Universum learning and MTL. The proposed algorithm will be discussed in Section 3. The Section 4 is the experiment and result analysis. The conclusion and future work are given in Section 5.

2 Related work

In this section, we will introduce the related work of Universum learning and multi-task learning.

2.1 Universum learning

Universum data, containing multiple class labels, is first proposed by Vapnik [19], which allows one to encode prior knowledge by representing meaningful concepts in the same domain as the problem at hand. They also design a new algorithm called support vector machine with Universum data to leverage Universum data by maximizing the number of observed contradictions, which is an alternative capacity concept to the large margin approach [20]. The first Universum learning method is proposed by J.Weston et al. [25]. In the training process of the binary classification problem, they take a set of labeled samples and a set of unlabeled samples that do not belong to any target category as input information. Unlabeled samples are called Univerusm samples, which should be collected to reflect information about the domain of target problem. The experiment proves that the Universum sample can improve the performance of the model. The work in [26] illustrates that Universum learning has a filtering effect in the model training process. For $< x, x^* >$ in the dual equation, the training sample x suppresses the features specified by Universum sample x^* under the action of Universum learning.

At present, the most research on Universum data is the traditional supervised learning based on SVM. On the basis of USVM, Qi et al. [27] introduce a more flexible U-NSVM to make better use of prior knowledge embedded in Universum data. A new twin support vector machine with Universum data (called U-TSVM) [21] is proposed. This method uses two non-parallel hyperplanes to construct the final classifier, and then the model assigns positive or negative class labels to the two hyperplanes based on their proximity to the samples. The Universum data in U-TSVM is determined by two hinge loss functions, located in non-parallel insensitive loss tubes. Afterwards, based on U-TSVM, $U_v - TSVM$ [28] and SUSVM [29] are proposed. Besides, the researchers also study the selection of Universum data to make the classifier perform better. Chen et al. [30] use new methods to pick useful Universum samples, which are defined as informative examples named in-between Universum examples, and apply them to semi-supervised learning. In the method proposed by Dhar et al. [31], authors add cost-sensitive learning to USVM and select suitable Universum data to reduce the misclassification cost of the classifier. The safe sample screening rule (SSSR) [32] is used in USVM to reduce computational cost.

In addition to SVM-based methods, Universum data is also used in other machine learning methods. In semi-supervised learning, Zhu et al. [33] introduce Universum learning to construct prior knowledge through the weights of views and features. This prior knowledge construction method makes full use of labeled and unlabeled sample. For multi-view learning, Chen et al. [34] combine CCA with Universum learning for multi-view learning and propose a method named UCCA. Wang et al. [35] exploit Universum data to gain a priori knowledge of the entire data distribution, which can improve the performance of MVL. In unsupervised learning, Deng et al. [36] design a novel unsupervised domain adaptation model to improve the performance of the systems evaluated in mismatched training and test conditions in deep learning. In multi-class learning, Songsiri et al. [37] present a method based on one-versus-one strategy, which aims at choosing the useful Universum data to build an effective binary classifier.

2.2 Multi-task learning

Since, the previous machine learning always focuses on single-task learning, it is easy to overlook other information that might help optimize the metrics. Specifically, the information may come from the training signals of the relevant tasks. By sharing the representations between related tasks, the model can be made to better summarize the original tasks. This method is called multi-task learning (MTL) [38].

There have been many MTL models that have been proposed, and their effects are much better than those without MTL. Fawzi et al. [39] uses multi-task learning to share transfer functions between models, reducing the complexity of the model and simplifying the exploration of the corpus. The work in [40], applies MTL to multiple robot with the goal of finding specific hazardous targets in an unknown area. Khosravan et al. [41] propose a multi-task CNN to deal with false positive (FP) nodule reduction and nodule segmentation. Wen et al. [42] use MTL to to train two-way recurrent neural networks for DNN-based speech synthesis. In the field of identity recognition, MTL also plays a considerable role [43–45]. Zhi et al. [46] propose lightweight volumetric multi-task learning model to reduce computing costs to solve real-time 3D image recognition problems.

The basic idea of the multi-task learning method is to share data between various tasks through hard or soft parameters. For example, the work in [47] applies multi-task learning to predict network-wide traffic speed. The model uses a set of hard parameters to share information between various links, and employ Bayesian optimization to tune the hard parameters of the MTL model. Huang et al. [48] uses two hierarchical Bayesian models as soft parameters to express the data correlation of each task, which improve the learning ability of the model. In addition, Zhang et al. [49] propose a multi-task multi-view clustering algorithm based on Locally Linear Embedding (LLE) and Laplacian Eigenmaps (LE). The model maps the information from multiple perspectives of each task to a common space, and then transforms it into a different task space, and finally uses k-means for clustering. A low-rank shared structure is used to model the sharing information across tasks [50]. A multi-stage multi-task feature Learning (MSMTFL) algorithm is designed by Gong et al. [51], and it aims at solving the non-convex optimization problem. Li et al. [52] design a new strategy to measure the relatedness that jointly learns shared parameters and shared feature representations. Yousefi et al. [53] propose a multi-task learning model using Gaussian processes, which can mine useful information in aggregated data.

In summary, the existing multi-task learning methods mostly optimize the multi-task learning problem by optimizing the task relationship parameters or feature sharing parameters. Under the premise that Universum data, which can provide prior knowledge, has a good effect in the field of machine learning. In this paper we target at starting from the task relationship parameters and using the Universum data to embed the prior knowledge to build a new multi-task learning model.

3 U-MTLSVM

In this section, we first present the definition of MTL with Universum data, and then propose our U-MTLSVM method. At the end of this section, we make an analysis of the proposed model.

3.1 Definition of MTL with universum data

For the multi-task problem, we have T training tasks, and assume the set $S_t = \{(x_{1t}, y_{1t}), ..., (x_{nt}, y_{nt})\}(t = 1, ..., T)$ store the labeled samples for the t^{th} task, and set $S_t^* = \{x_{1t}^*, ..., x_{mt}^*\}$ contain the Universum data for the t^{th} task. In S_t, x_t comes from the total data set, $y_t \in \{+1, -1\}$. For the task i, we assume that the function $f_t = w_t \cdot \phi(x)$ is a hyperplane, where $\phi(\cdot)$ means data x is mapped from input space to a feature space. For this task, we expect to build the classifier based on the labeled positive and negative examples as well as the Universum data of this task. However, there always exist relationship among the multi-tasks, which means the multi-task data can help to train a classifier for each individual task. In this kind, for every task $t \in \{1, 2, ..., T\}$, we define:

$$w_t = w_0 + v_t \tag{1}$$

In which w_t is the normal vector for the decision hyperplane and consists of two parameters. The first parameter is the common mean vector w_0 shared by all tasks. And the second parameter is the specific vector v_t for a specific task.

3.2 Multi-task SVM with universum data

In this section, we will detail the U-MTLSVM method. Supposed that we have an MTL problem with Universum data, we then introduce the Universum data into multi- task learning as follow:

$$\begin{aligned}
\min_{\omega_0, v_t, \psi_{ut}, \psi_{ut}^*, \xi_{it}} : \quad & \frac{1}{2}\|w_0\|^2 + \frac{1}{2}\mu \sum_{t=1}^{T} \|v_t\|^2 + C \sum_{i=1}^{m}\sum_{t=1}^{T} \xi_{it} \\
& + D \sum_{u=1}^{U}\sum_{t=1}^{T} (\psi_{ut} + \psi_{ut}^*) \\
s.t. \quad & \\
& y_{it}(w_0 + v_t) \cdot \phi(x_{it}) \geq 1 - \xi_{it} \\
& (w_0 + v_t) \cdot \phi(x_{ut}^*) \geq -\epsilon - \psi_{ut} \\
& (w_0 + v_t) \cdot \phi(x_{ut}^*) \leq \epsilon + \psi_{ut}^* \\
& \xi_{it} \geq 0, \psi_{ut} \geq 0, \psi_{ut}^* \geq 0.
\end{aligned} \tag{2}$$

- T represents the number of tasks, m represents the amount of data in the t^{th} task, and U represents the amount of Universum data in the t^{th} task. Parameter μ is the non-negative trade-off parameter, which controls preference of the tasks. Specially, If $\mu \to 0$, all tasks will be learned as the same task. C and D are penalty parameters for task data and Universum data. ξ_{it} is the corresponding slack variable. ψ_{ut} and ψ_{ut}^* denotes the slack variable for Universum data.
- The second and third constrains means the Universum data should be located in the insensitive region between hyperplanes. Parameter ϵ is defined by user.

The prior knowledge constructed by Universum learning is similar to Bayesian priors. Compared to prior distribution defined by parameters, it is much easier to use Universum learning to encode prior knowledge through a set of examples. Although, the prior knowledge provided by Universum learning is given by a series of data, they do not affect the distribution of training data. Therefore, we combine Universum learning with multi-task learning, and build prior knowledge through Universum samples to improve the generalization ability of the multi-task learning.

3.3 Optimization

Problem (2) is a quadratic programming problem. We exploit the Lagrangian multiplication and KKT conditions to optimize problem 2. We first construct Lagrange equation, and each constraint should be connected with a Lagrange multiplier. So we set α_{it}, β_{it}, γ_{it}, θ_{it}, Ω_{it}, η_{it} as the Lagrange multipliers of (2). The Lagrange function can be shown as follow:

$$
\begin{aligned}
L(\varepsilon) = & \tfrac{1}{2}\|\omega_0\| + \tfrac{1}{2}\mu \sum_{t=1}^{T} \|v_t\| \\
& + C \sum_{i=1}^{m}\sum_{t=1}^{T} \xi_{it} + D \sum_{u=1}^{U}\sum_{t=1}^{T} (\psi_{ut} + \psi_{ut}^*) \\
& - \sum_{i=1}^{m}\sum_{t=1}^{T} \alpha_{it}[y_{it}(\omega_0 + v_t)\cdot\phi(x_{it}) - 1 + \xi_{it}] \\
& - \sum_{u=1}^{U}\sum_{t=1}^{T} \beta_{ut}[(\omega_0 + v_t)\cdot\phi(x_{ut}^*) + \epsilon + \psi_{ut}] \\
& + \sum_{u=1}^{U}\sum_{t=1}^{T} \gamma_{ut}[(\omega_0 + v_t)\cdot\phi(x_{ut}^*) - \epsilon - \psi_{ut}^*] \\
& - \sum_{i=1}^{m}\sum_{t=1}^{T} \theta_{it}\xi_{it} - \sum_{u=1}^{U}\sum_{t=1}^{T} \Omega_{ut}\psi_{ut} - \sum_{u=1}^{U}\sum_{t=1}^{T} \eta_{ut}\psi_{ut}^*
\end{aligned} \tag{3}
$$

According to KKT conditions, we solve the Lagrange equation by differentiating parameter ω_0, v_t, ξ_{it}, ψ_{ut}, ψ_{ut}^* and setting the differential equation into 0:

$$
\begin{aligned}
\frac{\partial Ł}{\partial \omega_0} = & \ \omega_0 - \sum_{i=1}^{m}\sum_{t=1}^{T}\alpha_{it}y_{it}\phi(x_{it}) - \sum_{u=1}^{U}\sum_{t=1}^{T}\beta_{ut}\phi(x_{ut}^*) \\
& + \sum_{u=1}^{U}\sum_{t=1}^{T}\gamma_{it}\phi(x_{ut}^*) = 0
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
\frac{\partial Ł}{\partial v_t} = & \ \mu v_t - \sum_{i=1}^{m}\sum_{t=1}^{T}\alpha_{it}y_{it}\phi(x_{it}) - \sum_{u=1}^{U}\sum_{t=1}^{T}\beta_{ut}\phi(x_{ut}^*) \\
& + \sum_{u=1}^{U}\sum_{t=1}^{T}\gamma_{it}\phi(x_{ut}^*) = 0
\end{aligned} \tag{5}
$$

$$\frac{\partial Ł}{\partial \xi_{it}} = C - \alpha_{it} - \theta_{it} = 0 \tag{6}$$

$$\frac{\partial Ł}{\partial \psi_{ut}} = D - \beta_{ut} - \eta_{ut} = 0 \tag{7}$$

$$\frac{\partial Ł}{\partial \psi_{ut}^*} = D - \gamma_{ut} - \Omega_{ut} = 0 \tag{8}$$

Then we can get the polynomial about each parameter and return them to the original Lagrangian equation. To simplify the representation of the dual form, we set a function $K_{st} < x, y >= (\delta_{st} + \frac{1}{\mu})x \cdot y$, where δ_{st}(s,t = 1,...,T) is the kronecker delta function. Finally, the dual form is shown as follow:

$$
\begin{aligned}
\max_{\alpha,\beta,\gamma} : \ & - \frac{1}{2}\sum_{i=1}^{m}\sum_{s=1}^{T}\sum_{j=1}^{m}\sum_{t=1}^{T}\alpha_{is}y_{is}\alpha_{jt}y_{jt}K_{st} < x_{is}, x_{jt} > \\
& - \frac{1}{2}\sum_{i=1}^{U}\sum_{t=1}^{T}\sum_{j=1}^{U}\sum_{s=1}^{T}\beta_{it}\beta_{js}K_{st} < x_{it}^*, x_{js}^* > \\
& - \frac{1}{2}\sum_{i=1}^{U}\sum_{t=1}^{T}\sum_{j=1}^{U}\sum_{s=1}^{T}\gamma_{it}\gamma_{js}K_{st} < x_{it}^*, x_{js}^* > \\
& - \sum_{i=1}^{m}\sum_{t=1}^{T}\sum_{j=1}^{U}\sum_{s=1}^{T}\alpha_{it}y_{it}\gamma_{js}K_{st} < x_{it}, x_{js}^* > \\
& + \sum_{i=1}^{m}\sum_{t=1}^{T}\sum_{j=1}^{U}\sum_{s=1}^{T}\alpha_{it}y_{it}\beta_{js}K_{st} < x_{it}, x_{js}^* > \\
& + \sum_{i=1}^{U}\sum_{t=1}^{T}\sum_{j=1}^{U}\sum_{s=1}^{T}\beta_{it}\gamma_{js}K_{st} < x_{it}^*, x_{js}^* > \\
& + \sum_{i=1}^{m}\sum_{t=1}^{T}\alpha_{it} - \epsilon\sum_{j=1}^{U}\sum_{t=1}^{T}\beta_{jt} - \epsilon\sum_{j=1}^{U}\sum_{t=1}^{T}\gamma_{jt} \\
s.t. \quad & 0 \le \alpha_{it} \le C, 0 \le \beta_{jt} \le D, 0 \le \gamma_{jt} \le D
\end{aligned} \tag{9}
$$

In all, the U-MTLSVM algorithm is shown in Algorithm 1. The first step of the procedure is that we are given multi-task data, and we set the target task and associated tasks. Then, the Universum data is divided into the same number of tasks, ensuring that each task has a corresponding Universum data to provide prior knowledge. The next step is to determine μ, C, D, and solve QP with dual form (5), obtain α_{it}, β_{jt}, γ_{jt} to calculate w_0 and v_t. Meanwhile, the label of a new sample x in the t^{th} task can be determined by:

$$f_t(x) = sgn[(w_0 + v_t)\cdot\phi(x)] \tag{10}$$

If $f_t(x) < 0$, then the label of sample x is predicted to be negative. If $f_t(x) > 0$, the label of sample x is predicted to be positive.

According to (9), despite the addition of Universum data, the dual form of U-MTLSVM is still dominated by the inner product of two feature matrices, so its time complexity is O($(s+m)^3$) (s is the number of training samples, m is the number of Universum samples). From (1), (4) and (5), the hyperplane coefficient is not only controlled by the hard parameter μ. The item $< x_{it}, x_{js}^* >$ makes the Universum data refine the multi-task data. Constraints 2 and 3 limit the Universum data to fall near the hyperplane of the corresponding task.

Algorithm 1 U-MTLSVM.

Input: Training data M and Universum data U;
Output: w_0 and v_t;

1: Determine the number of tasks in the data set, and set T = the number of tasks;
2: Select classification taskS_t(t=1,...,T) in training data M;
3: Divide Universum data U by task number T, get S_t^*(t=1,...,T);
4: Initial parameter μ, C, D;
5: Construct convex quadratic problem according to problem (3);
6: Solve convex quadratic problem and get parameter $\alpha_{it}, \beta_{jt}, \gamma_{jt}$;
7: Calculate w_0 by equation (6);
8: Calculate v_t by equation (7) and μ;
9: Return discriminant $f_t(x) = arg[(w_0 + v_t) \cdot \phi(x)]$;

4 Experiment

In this section, to assess the effectiveness of the proposed method, four baselines and four different data sets are used in the experiments. All the experiments are performed on a computer with a 2.8 GHz processor and 8GB D-RAM under the windows 10 system.

4.1 Baselines

In this section, we briefly introduce the baselines which are used into experiment.

- **IUTSVM** [54]**:** On the basis of UTSVM, this method makes the matrix in the optimization problem non-singular by adding a regularization term.
- **USVM** [25]**:** This method adds Universum data that can provide prior knowledge to the SVM and has achieved better performance in pattern recognition.
- **SUSVM** [29]**:** The author formulates SUSVM as a pair of linear programming problems instead of quadratic programming problems (QPPs) to reduce computing time.
- **MTLSTWSVM** [55]**:** This method combines multi-task learning and least squares learning to solve pattern recognition problem.
- **HGPMT** [56]**:** This method is a MTL model which jointly learns the latent shared information among tasks, and does not need to involve the cross covariance.

4.2 Data sets and settings

We used the following data set to compare the proposed method with other methods.

- **20Newsgroups**[1] contains several top categories, such as “comp”, “rec”, “sci”, etc. Under the main categories, there are 20 sub-categories where each subcategory has 1,000 samples.
- **Reuters-21578**[2], is collected form Reuter news wire articles, and is organized into five top categories: “exchanges”, “orgs”, “people”, “place”, “topic”, and each category includes variable sub-categories.
- **Web-KB**[3] contains WWW-pages collected from computer science departments of various universities, which has 8,282 pages and are manually classified into 7 categories: “course”, “department”, “faculty”, “other”, “project”, “staff”, “student”.
- **Landmine**[4] consists of mine locations in 29 minefield areas, where the mine location in each area is represented by a 9-dimensional vector feature, and each area corresponds to a task. The first 15 regions are highly foliated, and the last 14 regions are bare earth or deserted.

For data sets with hierarchies structure, we use the following arrangements for data set process in order to achieve better and more fair experimental results. We select several subcategories from one top category as positive sub-data set (such as A(1), A(2), A(3) as the three classes from top category A) and the same number of positive sub-data set subcategories from another top category as negative sub-data set (such as B(1), B(2), B(3) as the three classes from top category B). Therefore, the multi-tasks are considered to be related since the positive classes and the negative classed belong to the same top categories. For example, in 20Newsgroups, to construct the data set (denoted as dataset1), we select 3 subcategories (“graphics”, “os.ms-windows”, “windows”) from top category “comp” as positive sub-data set, and select 3 subcategories (“med”, “crypt”, “space”) from “sci” as negative sub-data sets. We then can have a three-task data set, the task one regards the “graphics” and “med” for positive class and negative class, respectively; task two regards the “os.ms-windows” and “crypt” for positive class and negative class, respectively; task three regards “windows” and “space” for positive class and negative class. In addition, each data is represented as a binary vector of the 200 most characteristic words extracted by Alzaidy’s keyphrase extraction method [57].

Since, there is no hierarchy in the landmine data set, according to [51], we make the following arrangements for it. Since, negative labels in landmine are much more than positive labels, so we first remove some negative samples to

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

[2] http://www.daviddlewis.com/resources/testcollections/

[3] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz

[4] http://people.ee.duke.edu/ lcarin/LandmineData.zip

reduce the imbalance. As a result, the data from the same type of region can be considered as related, and different area types are irrelevant. We choose 4 areas from highly foliated region as positive sub-data set, and choose 4 area from bare earth or deserted region as negative sub-data set to build experimental data sets.

The text data we use generally does not contain Universum data, according to [25] and [21], we use U_{rest} Universum data construction method, with the appropriate number of subcategories other than positive and negative sub-data set as Universum data. For example, a data set contains 20 categories, 5 categories are used as classification tasks, 15 of the rest categories are used as Universum data (denoted $data_u$). For landmine data set, according to [51], we use the rest areas data as Universum data. In all, the dat sets are listed in Table 1.

4.3 Experiment settings

The experimental algorithms basic configuration will be arranged as follows. For the SVM-based methods, we use the linear kernel functon $k(x_i, x_j) = x_i \cdot x_j$, which performs well in text classification [58]. Since the data sets in this experiment are all text classification data sets, most of these classification problems are linearly separable. In addition, there are many features in these data sets, and there is no need to use other kernels to map the text features to a more higher-dimensional space. In addition, the processing speed of using linear kernels in text classification problems is faster. In our U-MTLSVM method, parameter μ is chosen from 10^{-3} to 10^3. The parameter which controls the tradeoff between Universum data in Universum-based methods is chosen from 10^{-3} to 10^3. The other regularization parameters in SVM-based methods are chosen from 10^{-3} to 10^3. In HGPMT, Squared Exponential kernel is used in this experiment.

For the data set, we use five-folder cross-validation to avoid sampling bias in the experiments, where four folds are selected as the training set, and the other one is considered as the testing set at each round.

4.4 Performance comparison

In this section, we compare the performance of U-MTLSVM and all the baseline methods. Table 2 shows the average accuracies, standard deviations and p-value on the 15 sub-data sets. P-value is calculated by performing a paired T-test of all other classifiers with U-MTLSVM

Table 1 Description of data sets

Sub-dataset	Data set	Task number	Positive sub-dataset	Negative sub-dataset	Universum
dataset1	20Newgroups	3	comp.{graphics, os.ms-windows,windows}	rec.{med,crypt, space}	$20Newgroups(1)_u$
dataset2	20Newgroups	3	rec.{autos, motorcycles,baseball}	sci.{crypt,electronics, med}	$20Newgroups(2)_u$
dataset3	20Newgroups	3	talk.{politics.gun, politics.mideast,politics.misc}	comp.{pc.hardware, mac.hardware,graphics}	$20Newgroups(3)_u$
dataset4	Web-KB	2	course(1)	department(1)	$Web-KB(1)_u$
dataset5	Web-KB	2	faculty(1)	project(1)	$Web-KB(2)_u$
dataset6	Web-KB	2	staff(1)	student(1)	$Web-KB(3)_u$
dataset7	Landmine	4	landmine.f(1)	landmine.b(1)	$Landmine(1)_u$
dataset8	Landmine	4	landmine.f(2)	landmine.b(2)	$Landmine(2)_u$
dataset9	Landmine	4	landmine.b(3)	landmine.f(3)	$Landmine(3)_u$
dataset10	Landmine	4	landmine.b(4)	landmine.f(4)	$Landmine(4)_u$
dataset11	Reuters-21578	5	exchanges.{amex,klce,mise, ipe,ose}	orgs(1).{aibd,bis,eib, ilo,imf}	$Reuters(1)_u$
dataset12	Reuters-21578	5	topics.{carcass,fishmeal,gold, jobs,rice}	places.{angola,benin,chile, fiji,laos}	$Reuters(2)_u$
dataset13	Reuters-21578	5	people.{amato,blix,eser, grosz,vancsa}	topics.{money-fx,lei,cpu, oilseed,sugar}	$Reuters(3)_u$
dataset14	Reuters-21578	5	places.{sumita,zak,walsh, ozal,keating}	exchanges.{fox,mnse,cboe, ase,mose}	$Reuters(4)_u$
dataset15	Reuters-21578	5	topices.{ship,silk,trade, lead,lumber}	orgs.{escap,icco,itc, un,worldbank}	$Reuters(5)_u$

Table 2 Accuracy obtained by USVM, IUTSVM, SUSVM, MTLSTWSVM, HGPMT and U-MTLSVM

Sub-dataset	USVM p-value	IUTSVM p-value p-value	SUSVM p-value p-value	MTLSTWSVM p-value	HGPMT p-value	U-MTLSVM
dataset1	70.4±1.2	74.2±1.6	72.1±2.1	75.4±1.7	77.9±1.8	**78.3**±0.9
	0.032	0.025	0.014	0.029	0.018	
dataset2	71.2±1.5	73.0±1.2	72.5±1.8	75.9±1.3	**78.2**±1.2	77.8±1.4
	0.015	0.038	0.026	0.041	0.024	
dataset3	72.8±1.9	76.5±2.1	75.1±2.0	78.4±1.8	79.6±1.5	**80.5**±1.7
	<0.001	0.028	<0.001	0.019	0.037	
dataset4	72.5±1.1	77.5±1.5	76.3±1.7	**80.1**±1.2	79.9±1.4	79.0±0.9
	0.041	0.021	0.029	0.039	0.018	
dataset5	77.5±1.6	81.9±1.3	80.8±2.0	81.7±1.8	83.1±1.6	**84.1**±1.5
	0.009	0.014	0.019	0.027	0.031	
dataset6	76.3±1.2	79.7±2.0	78.1±1.6	81.2±1.3	83.0±1.1	**83.8**±1.4
	0.022	0.39	0.028	0.004	0.017	
dataset7	79.4±1.4	82.4±1.3	80.9±1.4	84.3±1.5	85.3±1.2	**85.9**±1.6
	0.018	0.043	<0.001	0.020	0.036	
dataset8	78.4±2.0	82.5±0.8	81.6±1.0	83.1±1.4	86.0±1.3	**86.6**±0.7
	0.018	<0.001	0.021	0.042	0.028	
dataset9	79.5±1.6	83.7±1.6	82.2±1.2	86.8±1.7	**87.5**±1.4	86.1±1.3
	0.037	0.0032	0.026	0.042	0.039	
dataset10	78.6±1.2	82.4±0.6	80.7±2.0	83.1±1.1	86.2±1.5	**87.8**±1.2
	0.014	0.027	0.035	0.013	0.026	
dataset11	79.4±1.7	82.9±1.2	81.6±0.9	84.7±0.7	85.3±1.6	**86.7**±1.9
	0.015	0.041	0.040	0.024	0.018	
dataset12	78.3±2.1	81.8±1.4	82.3±1.8	84.4±1.3	**86.9**±1.7	86.3±1.5
	0.021	0.015	0.028	0.031	<0.001	
dataset13	80.0±2.3	82.5±0.8	81.4±1.5	85.1±1.6	**87.4**±1.2	87.2±1.1
	0.016	0.030	0.021	0.006	0.013	
dataset14	79.4±1.4	82.9±1.3	82.0±0.8	83.2±1.8	85.3±1.4	**86.5**±1.4
	0.009	0.025	0.037	0.035	0.043	
dataset15	82.5±1.7	84.1±1.2	85.0±1.2	87.3±1.4	**88.6**±0.7	87.0±1.7
	0.033	0.046	<0.001	0.029	0.010	

under the assumption that there is no difference between all classifications.

From Table 2, we can observe that U-MTLSVM always performs better than baselines. For example, on dataset1, the accuracy of USVM, IUTSVM, SUSVM, MTLSTWSVM, HGPMT are "70.4", "74.2", "72.1", "75.4", "77.9", respectively. However, the proposed U-MTLSVM method can achieve the accuracy at "78.3", which performs better than other baselines. This occurs because U-MTLSVM adds Universum data to the process of model learning, helping to modify the classification decision boundaries. On the other hand, as the number of tasks increases, the multi-task approach is better than the single-task approach. In addition, the single-task with Universum data method USVM obtains less performance than the proposed MTLSVM method. Moreover, because of the nature of multi-task learning, the combination of data with Universum data can make full use of training data. Therefore, U-MTLSVM performs better than other multi-task baselines.

The standard deviation in this experiment is to describe the accuracy drift of the model in different data sets. As can be seen from Table 2, U-MTLSVM has a smaller standard deviation than most other baselines in most data sets, indicating that U-MTLSVM can deliver more stable performance.

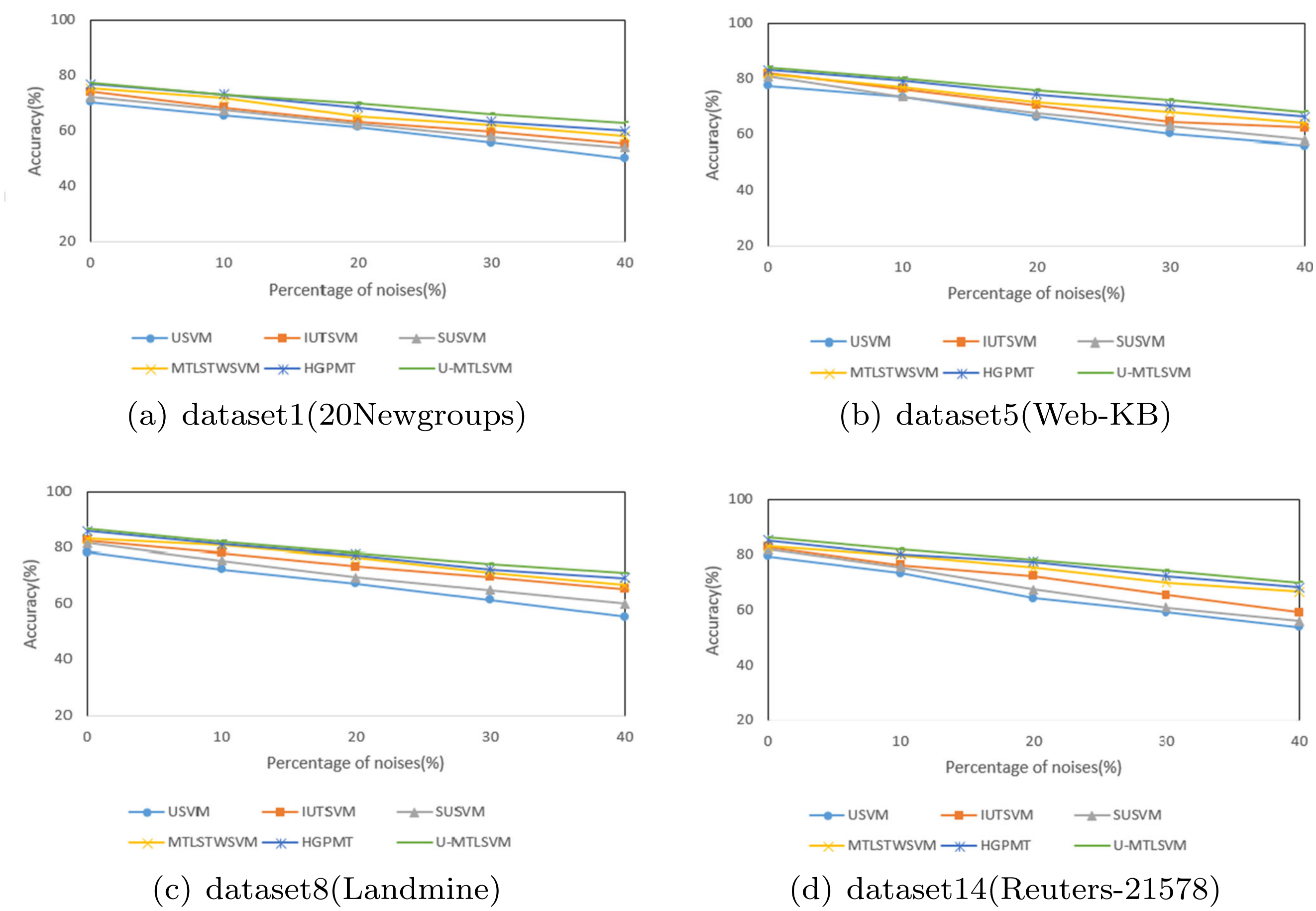
In addition, in order to demonstrate the difference between our proposed method and other baselines, we use the p-value calculated by t-test to reflect this difference. The t-test uses the t-distribution theory to infer the probability of a difference occurring, thereby comparing whether the difference between the two means is significant. If the p-value is below the confidence interval of 0.05, there is a significant difference between U-MTLSVM and other baselines. Otherwise there is no obvious difference between them. As can be seen from Table 2, we can observe the p-value of the proposed U-MTLSVM method over each baselines is almost less than the confidence level 0.05. For example, in dataset15, the p-values of the five baselines are all below 0.05. This shows that the U-MTLSVM method using Universum data can provide better performance than other baselines.

4.4.1 Performance on different noise levels

We investigate the sensitivity of USVM, IUTSVM, SUSVM, MTLSTWSVM, HGPMT and U-MTLSVM to data noise. Similar with the operation [59], we add the noise into the input data as follows. We randomly select data from the third major category as noise and add it to each classification task data in a certain proportion. Take dataset1 as a example: we choose noise data from top category "sci", then add them to the positive and negative sub-data sets in dataset1 according to a certain ratio. Take dataset1, dataset5, dataset8, dataset14 as examples, we add the noise and perform the methods on them. Figure 2 illustrates the variation of accuracy when the percentage of noises increases from 0% to 40% on 4 sub-data set. The x-axis stands for the percentage of noises added to the training data. The y-axis represents the average accuracy. It is easily discover that under the influence of noise, the accuracy of all models decreases at the same time. This occurs because as more noise is added into input data, the difference between the target category and the non-target category is less due to the influence of noise. But overall the accuracy of U-MTLSVM is still higher than other methods in all. This means that U-MTLSVM is less affected by noise compared with other methods.

4.4.2 Sensitivity of parameter μ

We test various values of the task-related parameter μ used in the proposed method to examine its effects. The test value we set for μ is 0.01, 0.1, 1, 2, 10, 1000. According to (2), when $\mu \rightarrow 0$, U-MTLSVM will be similar to USVM. We assume that all data come from the same task to correspond to the situation of $\mu \rightarrow 0$, let all tasks use one USVM for experiment. We set up individual USVM for each test task as a reference group for experiments. In this



(a) dataset1(20Newgroups) (b) dataset5(Web-KB)

(c) dataset8(Landmine) (d) dataset14(Reuters-21578)

Fig. 2 Sensitivity to labeling noise of different data set experiment, the training sets used in the experiment are all selected from Table 1, the number of tasks is 5. Dataset12 is chosen as the high similarity training set, and we select 5 data subsets from Reuters-21578 as low similarity training set. The penalty parameter C is set to 0.1, parameter D is set to 1, the number of Universum data is set to 200. The experiment results are shown in Fig. 3. The x-axis stands for the log(μ) of U-MTLSVM. The y-axis stands for the average accuracy. Figure 3 illustrates the importance of choosing μ correctly. If we choose the improper μ, which will make the model perform worse than solving 5 USVMs. For example, in Fig. 3a, when the task similarity is high, choosing a relatively large μ will seriously affect the model performance.

4.4.3 Universum data volume impact analysis

Through the above experiments, it can be seen that U-MTLSVM has a good performance in the field of multi-task learning. Next we will discuss the effect of the size of the Universum data on the accuracy of the classifier. In this part of the experiment, we chose dataset1, dataset2, dataset11, dataset12 as the experimental training set. All training samples were fixed at 100, and the Universum samples started from 200 and increased to 1400. The models participating in the experiment are USVM and U-MTLSVM, and the experimental results are shown in Fig. 4. According to Fig. 4, we can clearly see that the Universum data range from 200 to 800, and the accuracy of U-MTLSVM and USVM increase significantly. However, after 800 Universum data, the increase in accuracy slows down, and after the number reaches 1000, the accuracy changes tend to be stable. This shows Universum data can increase the performance as the number of them increase.

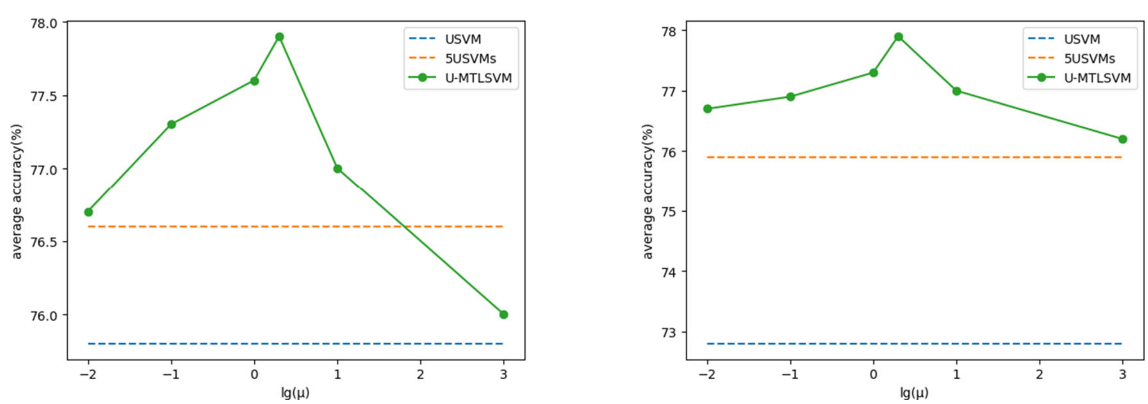
4.5 Running time analysis

In this part, we investigate the running time of U-MTLSVM and baselines on the data sets, and shown in Fig. 5. The shortest training time in the figure is IUTSVM, which is an improved method based on UTSVM, and the calculation speed of this model is faster than USVM. Although SUSVM and USVM are single-task learning methods, their calculation time are not much faster than MTLSTWSVM. This is because the USVM and SUSVM have added Universum data, and their processing time are correspondingly lengthened. Similarly, U-MTLSVM is also longer than SUSVM and MTLSTWSVM. The longest training time is HGPMT. The reason for the HGPMT training time is that the algorithm complexity is high. As the data set for training becomes larger, the training time required is longer.

4.6 Impact of different universum samples on the model

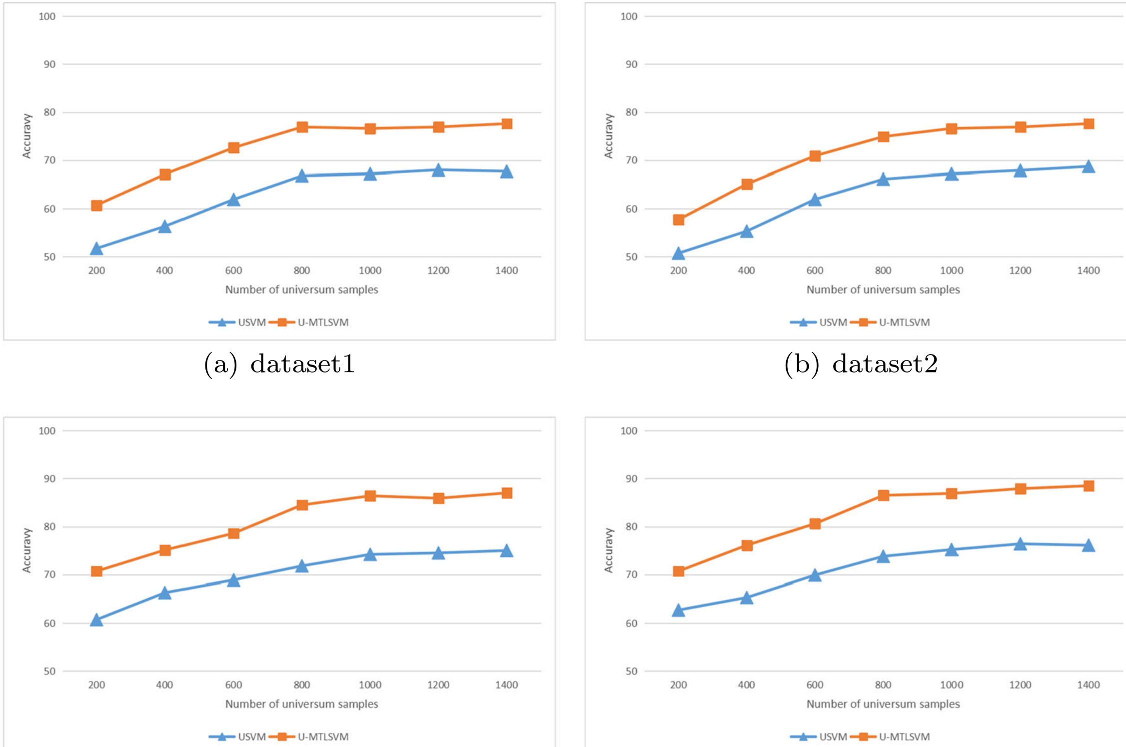
According to [25], we know that different Universum data has a certain influence on the model, and inappropriate Universum data will reduce the accuracy of the model. Therefore, we also perform experiments on different Universum data types, including U_{mean}, U_{gener}, U_{noise}, which are introduced as follows.

- U_{mean}:Select a word vector from each of the positive and negative classes of the target tasks, and then construct the mean of these two vectors as the new word vector.
- U_{gener}:Artificial text data is created by generating new word vectors based on the discrete empirical distribution of each word vector on the training set.



(a) High similarity (b) Low similarity

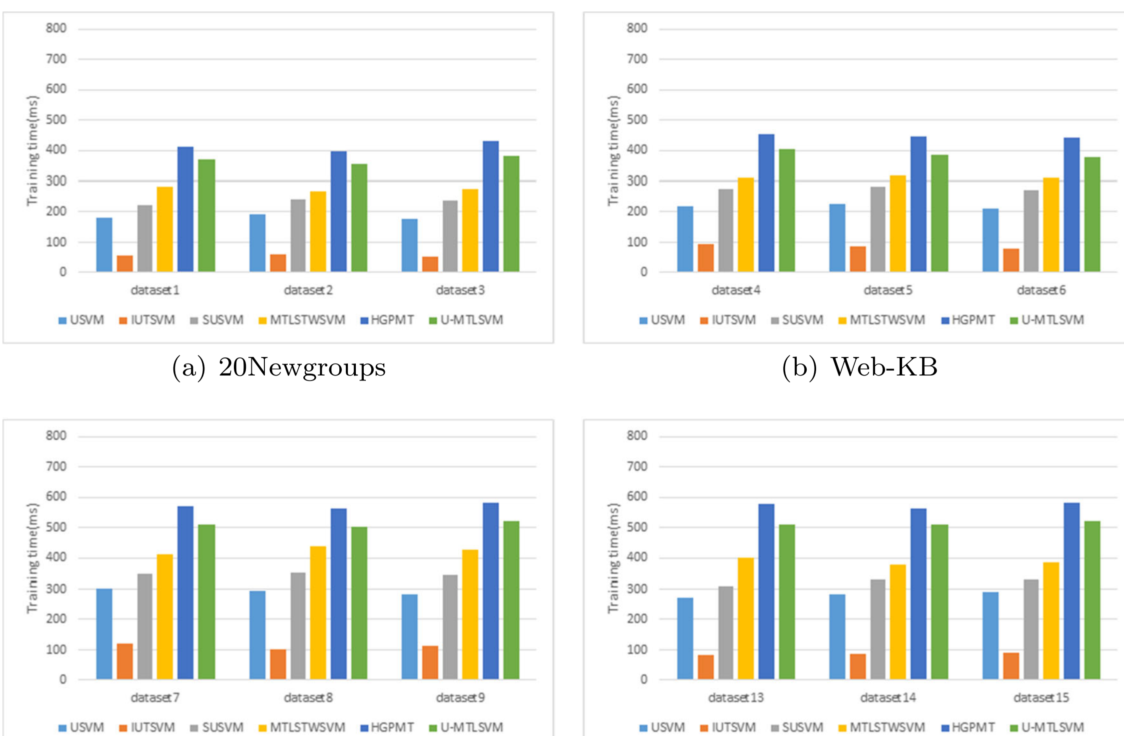
Fig. 3 Sensitivity of parameter μ



(a) dataset1 (b) dataset2

(c) dataset11 (d) dataset12

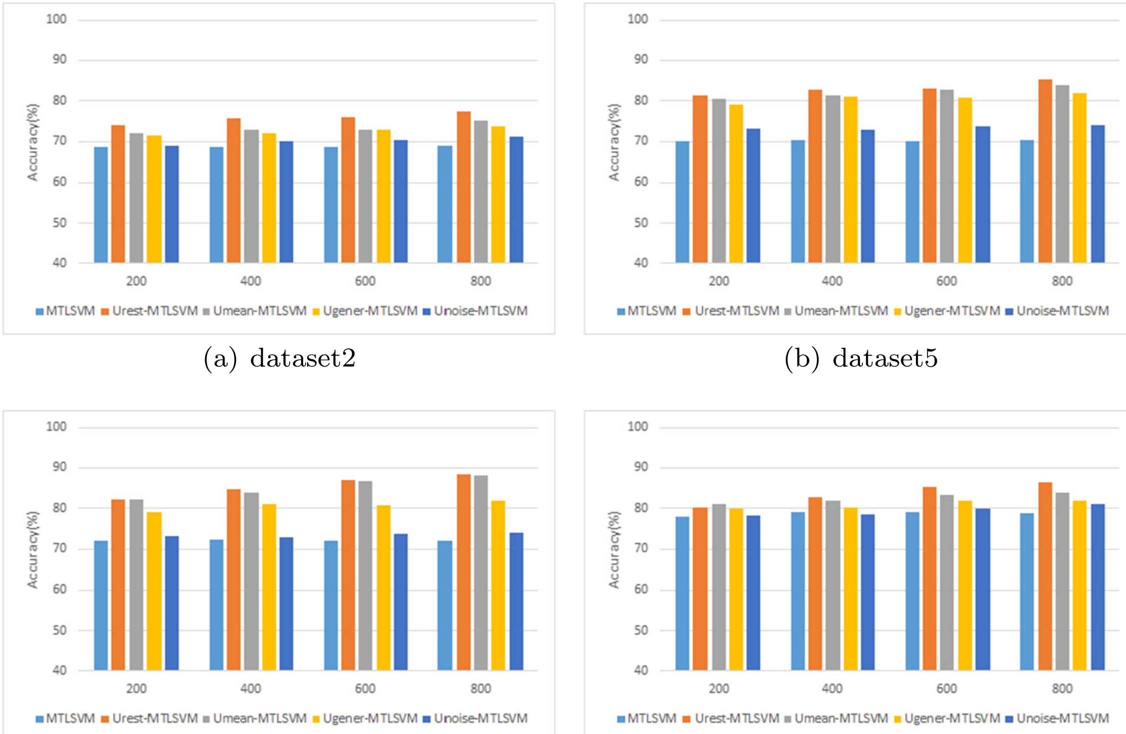
Fig. 4 The impact of the number of Universum data on the experimental results



(a) 20Newgroups (b) Web-KB

(c) Landmine (d) Reuters-21578

Fig. 5 Training time of USVM, IUTSVM, SUSVM, MTLSTWSVM, HGPMT, U-MTLSVM



(a) dataset2

(b) dataset5

(c) dataset9

(d) dataset13

Fig. 6 Influence of different Universum data strategies on experimental performance

- U_{noise}:Universum data composed of randomly generated text noise data.

We add MTLSVM [60] and Universum data constructed by the U_{rest} method as the control groups to the experiments to show the impact of different types of Universum data on the model. In this experiment, we choose dataset2, dataset5, dataset9 and dataset13 for the experiment, and all Universum samples are fixed at 600, and the Universum samples started from 200 and increased to 600. In MTLSVM, parameters λ_1 and λ_2 are chosen from 10^{-4} to 10^4 and make sure the ratio $\frac{\lambda_1}{\lambda_2}$ is less than 100 to prevent models from being the same. The experimental results are given in Fig. 6. The Universum data constructed by the U_{noise} method has almost no effect on improving the accuracy of the model. The Univerusm data constructed by U_{mean} and U_{gener} both have a good help on the accuracy of the model, while the univerusm data constructed by U_{rest} has the greatest improvement on the accuracy of the model. The possible reason that U_{noise} is not useful is that the prior knowledge provided by noise data has nothing to do with the target tasks, which causes the model to ignore this part of the data during training.

5 Conclusion

Most existing multitasking learning methods focus only on task-related data and build links between tasks with a few parameters. However, the use of data in this way will result in a lot of information loss. In this paper we proposed a novel method to solve multi-task learning problem with Universum data. First we use a parameter to share the information associated between each task and the target task. Then, the classifier is constructed on the premise that each task has corresponding Universum data, and the classifier is optimized and solved by iteration. Extensive experiments on the data sets have been conducted to investigate the performance of our proposed approach. The statistical results show that U-MTLSVM has a good performance on multi-task learning problems in the case of Universum data providing prior knowledge, and is superior to the classical multi-instance learning methods. In the future, we plan to study the multi-task learning with Universum data in the data stream environment.

Acknowledgments The authors would like to thank the reviewers for their very useful comments and suggestions. This work was supported in part by the Natural Science Foundation of China under Grant 61876044, 62076074 and Grant 61672169, in part by Guangdong Natural Science Foundation under Grant 2020A1515010670 and 2020A1515011501, in part by the Science and Technology Planning Project of Guangzhou under Grant 202002030141.

References

1. Smith FW (1968) Pattern classifier design by linear programming. IEEE Transactions on Computers C-17 4:367–372
2. Yuan R, Li Z, Guan X, Xu L (2010) An svm-based machine learning method for accurate internet traffic classification. Inf Syst Front 12(2):149–156
3. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232
4. Su C, Yang F, Zhang S, Tian Q, Davis LS, Gao W (2015) Multi-task learning with low rank attribute embedding for person re-identification pp 3739–3747
5. Xue Y, Liao X, Carin L, Krishnapuram B (2007) Multi-task learning for classification with dirichlet process priors. J Mach Learn Res 8:35–63
6. Tang Z, Li L, Wang D, Vipperla R (2017) Collaborative joint training with multitask recurrent model for speech and speaker recognition. IEEE Transactions on Audio Speech, and Language Processing 25(3):493–504
7. Do VH, Chen NF, Lim BP, Hasegawajohnson M (2018) Multitask learning for phone recognition of underresourced languages using mismatched transcription. IEEE Transactions on Audio Speech, and Language Processing 26(3):501–514
8. Xie X, Zhai X, Hou F, Hao A, Qin H (2019) Multitask learning on monocular water images: surface reconstruction and image synthesis. computer animation and virtual worlds
9. Yan C, Zhou L, Wan Y (2019) A multi-task learning model for better representation of clothing images. IEEE Access 7:34499–34507
10. Ji Y, Sun S (2013) Multitask multiclass support vector machines: Model and experiments. Pattern Recogn 46(3):914–924
11. Chen J, Tang L, Liu J, Ye J (2009) A convex formulation for learning shared structures from multiple tasks pp 137–144
12. Blackwood G, Ballesteros M, Ward T (2018) Multilingual neural machine translation with task-specific attention pp 3112–3122.
13. Das A, Dantcheva A, Bremond F (2018) Mitigating bias in gender, age and ethnicity classification: A multi-task convolution neural network approach pp 573–585
14. Salucci M, Poli L, Oliveri G (2019) Full-vectorial 3d microwave imaging of sparse scatterers through a multi-task bayesian compressive sensing approach
15. Huang Y, Beck JL, Li H (2019) Multitask sparse bayesian learning with applications in structural health monitoring. Computer-aided Civil and Infrastructure Engineering 34(9):732–754
16. Wang S, Chang X, Li X, Sheng QZ, Chen W (2016) Multi-task support vector machines for feature selection with shared knowledge discovery. Signal Process 120:746–753
17. Chandra R, Gupta A, Ong Y, Goh C (2018) Evolutionary multi-task learning for modular knowledge representation in neural networks. Neural Process Lett 47(3):993–1009
18. Pearce M, Branke J (2018) Continuous multi-task bayesian optimisation with correlation. European Journal of Operational Research p 270(3):1074–1085
19. Vapnik V (1998a) Statistical Learning Theory, dblp
20. Vapnik V (1998b) Estimation of Dependence Based on Empirical Data, springer
21. Qi Z, Tian Y, Shi Y (2012) Twin support vector machine with universum data. Neural Netw 36:112–119
22. Liu CL, Lee CH (2016) Enhancing text classification with the universum. In: 2016 12th International Conference on Natural Computation and 13th Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)
23. Pan S, Wu J, Zhu X, Long G, Zhang C (2017) Boosting for graph classification with universum. Knowledge & Information Systems 50(1):1–25
24. Richhariya B, Gupta D (2018) Facial expression recognition using iterative universum twin support vector machine, applied soft computing
25. Weston J, Collobert R, Sinz FH, Bottou L, Vapnik V (2006) Inference with the universum pp 1009–1016
26. Sinz F, Chapelle O, Agarwal A, Scholkopf B, Platt CJ, Koller D, Singer Y, Roweis S (2007) An analysis of inference with the universum. Advances in Neural Information Processing Systems 20(2008):1369–1376
27. Qi Z, Tian Y, Shi Y (2014) A nonparallel support vector machine for a classification problem with universum learning. J Comput Appl Math 263(1):288–298
28. Xu Y, Chen M, Yang Z, Li G (2016) V-twin support vector machine with universum data for classification. Applied Intelligence 44(4):956–968
29. Liu D, Tian Y, Bie R, Shi Y (2014) Self-universum support vector machine. Personal and ubiquitous computing 18(8):1813–1819
30. Chen S, Zhang C (2009) Selecting informative universum sample for semi-supervised learning. In: International Jont Conference on Artifical Intelligence
31. Dhar S, Cherkassky V (2015) Development and evaluation of cost-sensitive universum-svm. IEEE Transactions on Systems Man, and Cybernetics 45(4):806–818
32. Zhao J, Xu Y (2017) A safe sample screening rule for universum support vector machines. Knowledge Based Systems 138:46–57
33. Zhu C, Miao D, Zhou R, Wei L (2020) Weight-and-universum-based semi-supervised multi-view learning machine. Soft Comput 24(14):10657–10679
34. Chen X, Yin H, Jiang F, Wang L (2018) Multi-view dimensionality reduction based on universum learning. Neurocomputing 275:2279–2286
35. Wang Z, Zhu Y, Liu W, Chen Z, Gao D (2014) Multi-view learning with universum. Knowledge Based Systems 70:376–391
36. Deng J, Xu X, Zhang Z, Fruhholz S, Schuller B (2017) Universum autoencoder-based domain adaptation for speech emotion recognition. IEEE Signal Processing Letters 24(4):500–504
37. Songsiri P, Cherkassky V, Kijsirikul B (2018) Universum selection for boosting the performance of multiclass support vector machines based on one-versus-one strategy. Knowledge Based Systems 159:9–19
38. Ruder S (2017) An overview of multi-task learning in deep neural networks. arXiv: Learning
39. Fawzi A, Sinn M, Frossard P (2017) Multitask additive models with shared transfer functions based on dictionary learning. International conference on machine learning 65(5):1352–1365
40. Palmieri N, Yang X, De Rango F, Santamaria AF (2018) Self-adaptive decision-making mechanisms to balance the execution of multiple tasks for a multi-robots team. Neurocomputing 306:17–36
41. Khosravan N, Bagci U (2018) Semi-supervised multi-task learning for lung cancer diagnosis pp 710–713
42. Wen Z, Li K, Huang Z, Lee C, Tao J, Wen Z (2017) Improving deep neural network based speech synthesis through contextual feature parametrization and multi-task learning. Journal of Signal Processing Systems 90:0–1037

43. Chikontwe P, Lee HJ (2018) Deep multi-task network for learning person identity and attributes. IEEE Access 6:1–1
44. Su C, Yang F, Zhang S, Tian Q, Davis LS, Gao W (2017) Multi-task learning with low rank attribute embedding for multi-camera person re-identification. IEEE Transactions on Pattern Analysis & Machine Intelligence 40:1167–1181
45. Ma AJ, Yuen PC, Li J (2013) Domain transfer support vector ranking for person re-identification without target camera label information pp 3567–3574
46. Zhi S, Liu Y, Li X, Guo Y (2017) Toward real-time 3d object recognition: a lightweight volumetric cnn framework using multitask learning. Computers & Graphics 71(APR.):199–207
47. Zhang K, Zheng L, Liu Z, Jia N (2019) A deep learning based multitask model for network-wide traffic speed predication. Neurocomputing
48. Huang Y, Beck JL, Li H (2019) Multitask sparse bayesian learning with applications in structural health monitoring. Computer-Aided Civil and Infrastructure Engineering 34(9):732–754
49. Zhang Y, Yang Y, Li t, Fujita H (2018) A multitask multiview clustering algorithm in heterogeneous situations based on lle and le. knowledge based systems
50. Chen J, Liu J, Ye J (2012) Learning incoherent sparse and low-rank patterns from multiple tasks. Acm Transactions on Knowledge Discovery from Data 5(4):1–31
51. Gong P, Ye J, Zhang C (2013) Multi-stage multi-task feature learning pp 2979–3010
52. Li Y, Tian X, Liu T (2017) On better exploring and exploiting task relationships in multi-task learning: joint model and feature learning pp 1–11
53. Yousefi F, Smith M, Alvarez MA (2019) Multi-task learning for aggregated data using gaussian processes. arXiv: machine learning
54. Richhariya B, Sharma A, Tanveer M (2018) Improved universum twin support vector machine. In: IEEE Symposium Series on Computational Intelligence SSCI
55. Mei B, Xu Y (2019) Multi-task least squares twin support vector machine for classification. neurocomputing
56. Ping Li, Chen S (2018) Hierarchical gaussian processes model for multi-task learning, pattern recognition the journal of the pattern recognition society
57. Alzaidy RA, Caragea C, Giles CL (2019) Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents pp 2551–2557
58. Joachims T (1998) Text categorization with support vector machines: Learning with many relevant features, ecml
59. Liu B, Xiao Y, Hao Z (2018) A selective multiple instance transfer learning method for text categorization problems. Knowl-Based Syst 141(1):178–187
60. Evgeniou T, Pontil M (2004) Regularized multi–task learning. In: Tenth Acm Sigkdd International Conference on Knowledge Discovery & Data Mining

Yanshan Xiao received the PhD degree in computer science from the University of Technology, Sydney, in 2011. She is with the Faculty of Computer, Guangdong University of Technology. Her research interests include multiple-instance learning and support vector machine.

Jing Wen is a graduate student in the Faculty of Computer, Guangdong University of Technology, Guangzhou, China. His current research interests include multi-task learning, support vector machine, and data mining.

Bo Liu is a professor with the Faculty of Automation, Guangdong University of Technology. His research interests include machine learning and data mining. He has published papers on IEEE Transactions on Neural Networks, IEEE Transactions on Knowledge and Data Engineering, Knowledge and Information Systems, IEEE International Conference on Data Mining (ICDM), SIAM International Conference on Data Mining (SDM) and ACM International Conference on Information and Knowledge Management (CIKM).