Multitask Multiclass Privileged Information Support Vector Machines

You Ji, Shiliang Sun, Yue Lu
Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, China
jiyou09@gmail.com, slsun@cs.ecnu.edu.cn, ylu@cs.ecnu.edu.cn

Abstract

In this paper, we propose a new learning paradigm named multitask multiclass privileged information support vector machines. The starting point of our work is mainly based on the success of multitask multiclass support vector machines which cast multitask multiclass problems as a constrained optimization problem with a quadratic objective function. Learning using privileged information is an advanced learning paradigm integrated with the idea of human teaching in machine learning. This paper mainly extends multitask multiclass support vector machines to privileged information learning strategy. Our approach can take full advantages of the multitask learning and privileged information. Experimental results show that our approaches obtains very good results for multitask multiclass problems.

1. Introduction

The main goal of multitask learning (MTL) is to improve the generalization performance by learning multiple related tasks simultaneously while using a shared representation [1]. Past empirical work has shown that, it is beneficial to learn different but related tasks simultaneously instead of single task learning (STL) [2, 3]. The MTL strategy for support vector machines (SVMs) can be naturally extended to existing kernel-based on learning methods [4–6]. There are also a lot of researchers to theoretically study the MTL [4, 6]. The multitask multiclass SVMs (M^{2}SVMs) can solve multitask multiclass problems by a constrained optimization problem with a quadratic objective function [5]. With this strategy, M^{2}SVMs can learn multitask multiclass problems directly and effectively. Recently, Vapnik et al. [7] introduced a new learning model named learning using privileged information (LUPI). Besides learning with standard training data, the LUPI also supplies classifiers with additional information which can only be available for training instances [7]. In the optimistic case, the LUPI model can improve the probability bound of test errors from $O\left(\frac{1}{\sqrt{n}}\right)$ to $O\left(\frac{1}{n}\right)$ [7].

In this paper, we develop and discuss multitask multiclass privileged information SVMs (M^{2}PiSVMs) on the basis of M^{2}SVMs in detail. Experimental results demonstrate the effectiveness of the proposed method. The rest of this paper is organized as follows. Section 2 briefly reviews related work on multiclass SVMs (MSVMs). Section 3 thoroughly describes M^{2}SVMs. Section 4 presents multiclass privileged information SVMs (MPiSVMs). M^{2}PiSVMs are given in Section 5. Experiments and discussions are reported in Section 6. Conclusions and future work are provided in Section 7.

2. MSVMs

Multiclass SVMs (MSVMs) are generalizations of separating hyperplanes and margins to the scenario of multiclass problems [8]. Input (x_i, y_i) belongs to $X \times Y$, where $X = R^d$, $Y = \{1, 2, 3, \dots, K\}$ and $i \in \{1, \dots, m\}$. This framework uses classifiers of the form

$$H_M(x) = \arg\max_{k}^{K} \{M_k x\} \quad , \tag{1}$$

where M is a matrix of size $K \times d$, and M_k is the kth row of M. The quadratic optimization problem of MSVMs is defined as

$$\begin{aligned} &\min_{M} \tfrac{1}{2}\beta \left\|M\right\|_2^2 + \sum_{i=1}^{m} \varepsilon_i \\ &\text{s.t.}: \ \forall k, i, \\ &\qquad M_{y_i} x_i + \delta_{y_i,k} - M_k x_i \geq 1 - \varepsilon_i \ \ . \end{aligned} \tag{2}$$

3. M^{2}SVMs

In MTL settings, all data for T tasks come from the same distribution P on $X \times Y$ where $X = R^d, Y =$

$\{1, 2, 3, \ldots, K\}$. For each task we have m_t instances sampled from P_t ($t \in \{1, 2, 3, \ldots, T\}$). Assuming each instance belongs to one task, $\phi(i)$ stands for the ith instance's task index. For each task, we use a shared representation M_0 which stands for the common information between tasks while M_t stands for the tth task's classifier. Details of M^{2}SVMs can be found in [5]. By multiclass learning settings in [8], M^{2}SVMs have the following MTL model

$$\begin{aligned} &H_{M_t}(x) = \arg\max_k^K \{M_{t,k} x_t\} \\ &M_t = M_0 + V_t \quad , \end{aligned} \tag{3}$$

where M_t is a matrix of size $K \times d$, $M_{t,k}$ stands for the kth row of M_t, and $\arg\max_k^K \{M_{t,k} x_t\}$ finds the class having largest similarity score with the task's instance x_t. Then M^{2}SVMs will solve the following optimization problem

$$\begin{aligned} &\min_{V_t, M_0} \ \tfrac{\beta}{2} \sum_{k=1}^{K} \left[\sum_{t=1}^{T} \rho_t \|V_{t,k}\|^2 + \|M_{0,k}\|^2 \right] + \sum_{t=1}^{T} \sum_{i=1}^{m_t} \varepsilon_{t,i} \\ &\text{s.t.}: \ \forall t, k, i, \\ &(V_{t,y_i} + M_{0,y_i}) x_{t,i} + \delta_{y_i,k} - (V_{t,k} + M_{0,k}) x_{t,i} \\ &\quad \geq 1 - \varepsilon_{t,i} \quad , \end{aligned} \tag{4}$$

where $\delta_{p,q}$ is equal to 1 if $p = q$ and 0 otherwise, ρ_t is the weighted parameter between V_t and M_0 [5], and $\beta > 0$ and $\rho_t > 0$ are regularization constants [5]. Instead of solving Eq. 4 directly, the usually strategy is to solve Eq. 4's dual problem. The derivation in detail can be found in [5]. We may use the kernel trick to extend the M^{2}SVMs strategy to non-linear MTL. Then we can get the dual problem as

$$\begin{aligned} &\min \ Q(a) = \sum_{i,j,k} a_{i,k} \delta_{y_i,k} \\ &+ \tfrac{1}{2\beta} \sum_{i,j,k} \left[(a_{i,k} - \delta_{y_i,k}) \left(a_{j,k} - \delta_{y_j,k}\right) K_{\phi(i),\phi(j)} \langle x_i, x_j \rangle \right] \\ &\text{s.t.}: \ \forall i, j, k, \ \ a_{i,k} \geq 0, \ \ \sum_k a_{i,k} = 1 \quad , \end{aligned} \tag{5}$$

where K is given as

$$K_{\varphi(i)\varphi(j)}(x_i, x_j) = \left(1 + \frac{\delta_{\varphi(i),\varphi(j)}}{\rho_{\varphi(i)}}\right) \langle x_i, x_j \rangle \quad . \tag{6}$$

4. MPiSVMs

Firstly, we will extend MSVMs to MPiSVMs, then we extend MPiSVMs to the M^{2}PiSVMs. When we extend MSVMs Eq. 2 to the LUPI, some slack variables are also used in our MPiSVMs which result in the following optimization problem

$$\begin{aligned} &\min_{M, M^*} \ \tfrac{\beta}{2} \|M\|_2^2 + \tfrac{\gamma}{2} \|M^*\|_2^2 + \sum_i (\varepsilon_i + \varsigma_i) \\ &\qquad + \sum_{i,k} \left[M_{y_i}^* \cdot x_i^* + \delta_{y_i,k} - M_k^* x_i^* \right] \\ &\text{s.t.}: \ \forall k, i, \\ &M_{y_i} x_i + \delta_{y_i,k} - M_k x_i \\ &\qquad \geq 1 - \left(M_{y_i}^* \cdot x_i^* - M_k^* x_i^*\right) - \varsigma_i \\ &M_{y_i}^* \cdot x_i^* + \delta_{y_i,k} - M_k^* x_i^* \geq 1 - \varepsilon_i \quad . \end{aligned} \tag{7}$$

For $k = y_i$, the inequality constraints become $\varsigma_i \geq 0$ and $\varepsilon_i \geq 0$. After adding a dual set of variables and the derivation of the Lagrangian of the optimization problem, we can get Eq. 7's dual problem as

$$\begin{aligned} &\max \ Q(a, b) = \sum_{i,k} \delta_{y_i,k} (1 - a_{i,k} - b_{i,k}) \\ &- \tfrac{1}{2\beta} \sum_{i,j,k} \left[(a_{i,k} - \delta_{y_i,k}) \left(a_{j,k} - \delta_{y_j,k}\right) \langle x_i, x_j \rangle \right] \\ &- \tfrac{1}{2\gamma} \sum_{i,j,k} u_{i,j,k} \times \left\langle x_i^*, x_j^* \right\rangle \\ &s.t.: \quad \forall i, k, \\ &\sum_k a_{i,k} = 1, \quad \sum_k b_{i,k} = 1, \quad a_{i,k} \geq 0, \quad b_{i,k} \geq 0 \quad , \end{aligned} \tag{8}$$

where $u_{i,j,k}$ is defined as

$$\begin{aligned} u_{i,j,k} = &\left[(1 - a_{i,k} - b_{i,k}) - \delta_{y_i,k} (K - 2)\right] \\ &\times \left[(1 - a_{j,k} - b_{j,k}) - \delta_{y_j,k} (K - 2)\right] \quad . \end{aligned} \tag{9}$$

5. M^{2}PiSVMs

In MTL settings, we just consider all tasks' privileged information are uncoupled from each other. This means each task's privileged information has no relationship with others'. Although this paper just consider this situation, it is easy to extend our model to the situation that all tasks's privileged information come from the same distribution. After merging Eq. 4 and Eq. 7 together, we may get M^{2}PiSVMs as

$$\begin{aligned} &\min_{V_t, M_0, M_t^*} \ \tfrac{\beta}{2} \left(\|M_0\|_2^2 + \sum_t \rho_t \|V_t\|_2^2 \right) \\ &\quad + \tfrac{\gamma}{2} \sum_t \rho_t \|M_t^*\|_2^2 + \sum_{t,i} (\varsigma_{t,i} + \varepsilon_{t,i}) \\ &\quad + \sum_{t,i,k} \left[M_{t,y_i}^* \cdot x_{t,i}^* + \delta_{y_i,k} - M_{t,k}^* x_{t,i}^* \right] \\ &\text{s.t.}: \ \forall t, k, i \in \{1, 2, \ldots, m_t\}, \\ &(V_{t,y_i} + M_{0,y_i}) x_{t,i} + \delta_{y_i,k} - (V_{t,k} + M_{0,k}) x_{t,i} \\ &\qquad \geq 1 - \left(M_{t,y_i}^* \cdot x_{t,i}^* - M_{t,k}^* x_{t,i}^* \right) - \varsigma_{t,i} \\ &M_{t,y_i}^* \cdot x_{t,i}^* + \delta_{y_i,k} - M_{t,k}^* x_{t,i}^* \geq 1 - \varepsilon_{t,i} \quad , \end{aligned} \tag{10}$$

where m_t stands for the number of instances of the tth task. Similar to the derivation in [5], we can get Eq. 10's dual problem as

$$
\begin{array}{l}
\max\ Q\left(a,b\right)=\sum\limits_{i,k}\delta_{y_i,k}\left(1-a_{i,k}-b_{i,k}\right)\\
-\frac{1}{2\beta}\sum\limits_{i,j,k}\left[\begin{array}{l}\left(a_{i,k}-\delta_{y_i,k}\right)\left(\delta_{y_j,k}-a_{j,k}\right)\\ \times\left(1+\frac{\delta_{\varphi(i),\varphi(j)}}{\rho_{\varphi(i)}}\right)\left\langle x_i,x_j\right\rangle\end{array}\right]\\
-\frac{1}{2\gamma}\sum\limits_{i,j,k}u_{i,j,k}\times\left(\frac{\delta_{\varphi(i),\varphi(j)}}{\rho_{\varphi(i)}}\right)\left\langle x_i^*,x_j^*\right\rangle\\
\text{s.t.}:\ \forall i,j,k,\quad i,j\in\left\{1,2,\ldots,\sum\limits_{t=1}^{T}m_t\right\}\\
\sum\limits_{k}a_{i,k}=1,\ \sum\limits_{k}b_{i,k}=1,\ \ a_{i,k}\geq 0,\ \ b_{i,k}\geq 0\ \ .
\end{array}
\tag{11}
$$

5.1. Non-linear M^{2}PiSVMs

The kernel trick can be used to avoid computing feature maps directly, and can also be applied to MPiSVMs. By the same strategy in [3], we will extend M^{2}PiSVMs to non-linear MTL as

$$
\begin{array}{l}
\max\ Q\left(a,b\right)=\sum\limits_{i,k}\delta_{y_i,k}\left(1-a_{i,k}-b_{i,k}\right)\\
-\frac{1}{2\beta}\sum\limits_{i,j,k}\left[\begin{array}{l}\left(a_{i,k}-\delta_{y_i,k}\right)\left(a_{j,k}-\delta_{y_j,k}\right)\\ \times K_{\delta_{\varphi(i),\varphi(j)}}\left(x_i,x_j\right)\end{array}\right]\\
-\frac{1}{2\gamma}\sum\limits_{i,j,k}u_{i,j,k}\times\left(\frac{\delta_{\varphi(i),\varphi(j)}}{\rho_{\varphi(i)}}\right)K^*\left(x_i^*,x_j^*\right)\\
\text{s.t.}:\ \forall i,j,k,\ \ \sum\limits_{k}a_{i,k}=1,\ \sum\limits_{k}b_{i,k}=1,\\
a_{i,k}\geq 0,\ \ b_{i,k}\geq 0\ \ ,
\end{array}
\tag{12}
$$

where kernel function K is given in Eq. 6. The kernel function K^* for privileged information is a normal kernel function. In the procedure of prediction, privileged information is not used, so the classifier is the same as [5]. The classifier of the tth task is

$$
\begin{array}{l}
H_t\left(x\right)=\arg\max\limits_{k=1}^{K}\left\{M_{t,k}x\right\}\\
=\arg\max\limits_{k=1}^{K}\left\{\frac{1}{2\beta}\sum\limits_{i=1}^{m}\left(\delta_{k,y_i}-a_{k,i}\right)K_{\varphi(i),t}\left(x_i,x\right)\right\}\ \ .
\end{array}
\tag{13}
$$

6. Experiments

We compare MSVMs, MPiSVMs, M^{2}SVMs and M^{2}PiSVMs with the LOQO[1] as the solver. The LOQO solver is based on interior point optimization algorithms which give the best results among the off-the-shelf optimizers [7].

We may find that there are two kernel functions in Eq. 12, one is a multitask kernel function while the other one is a normal kernel function. We use radial basis function (RBF) for both kernel functions (also in Eq. 7). Kernels can be written as

$$
\begin{array}{l}
K_{\varphi(i)\varphi(j)}\left(x_i,x_j\right)=\left(1+\frac{\delta_{\varphi(i),\varphi(j)}}{\rho_t}\right)k\left(x_i,x_j\right)\\
k\left(x_{i,}x_j\right)=\exp\left(-\sigma\|x_i-x_j\|^2\right),\ \sigma>0\\
K^*\left(x_i^*,x_j^*\right)=k\left(x_i^*,x_j^*\right)\ \ .
\end{array}
\tag{14}
$$

σ is RBF kernel parameter. There are another three parameters ρ_t,β,γ for our model. We use the same search strategy as [5]. We use a grid search strategy to select the best parameters from training sets which are also recommended in Libsvm.

6.1. Datasets

We test our algorithms on two datasets from UCI[2], the Isolet dataset and the spoken arabic digits (SAD) dataset. Although these datasets have no privileged information for the respective of LUPI, we extract the principal component information (98%)as privileged information by principal component analysis (PCA). Details about these two datasets are listed in Table 1 where x^* stands for the dimensionality of privileged information extracted by PCA.

The Isolet dataset with 7797 examples (three examples are historically missing) is collected from 150 subjects uttering all English alphabet twice. One task has 30 speakers. The representation of Isolet lends itself to the multitask multiclass learning [4] with $T = 5$ tasks and $K = 26$ labels. The SAD dataset with 8800 instances is collected from 88 speakers with 44 males (Task 1) and 44 females (Task 2) Arabic native speakers between ages 18 and 40 to represent ten spoken Arabic digits from 0 to 9. Each instance in the dataset is a matrix of size $row \times 13$ ($4 \leq row \leq 93$). For simplicity, we resize each matrix to 10×13. After the vectorization of these matrices, we get a vector of size 1×130 to represent one instance.

Table 1. Details of datasets.

Name	Attributes	Instances	Classes	Tasks
Isolet	617(290*)	7797	26	5
SAD	130(24*)	8800	10	2

6.2. Results and Discussions

Error rates of our experiments are computed by averaged results of 5-fold cross-validation. We compare

[1] www.princeton.edu/~rvdb/loqo/LOQO.html

[2] http://archive.ics.uci.edu/ml/datasets.html

M^{2}PiSVMs with different baselines in Table 2 and Table 3. In these tables, ALL means merging all tasks' data together and taking these data with no task's differences. Results in Table 2 and Table 3 show that our model gets the best results on the two datasets. We may find that the LUPI can get higher accuracy.

Table 2. Error rates(%) on Isolet

STL	MSVMs	MPiSVMs
Task 1	8.66	**8.14**
Task 2	9.69	**9.17**
Task 3	10.71	**10.31**
Task 4	11.03	**10.56**
Task 5	11.23	**9.97**
ALL	7.67	**6.79**
–	M^{2}SVMs	M^{2}PiSVMs
MTL	5.39	**4.52**

Table 3. Error rates(%) on SAD

STL	MSVMs	MPiSVMs
Task 1	6.64	**6.28**
Task 2	11.66	**11.19**
ALL	9.34	**7.79**
–	M^{2}SVMs	M^{2}PiSVMs
MTL	4.85	**4.45**

We also compare these four methods with different training sizes. If the dataset have T tasks, with n instances for training, we randomly sample $\frac{n}{T}$ instances from each task. Then we randomly sample $\frac{n_*}{T}$ testing instances from the rest instances of each task. We set $n_* = 1500$ for Isolet dataset, $n_* = 1700$ for SAD dataset. Each result takes an average of 10 times the repetition. MSVMs and MPiSVMs just train all n instances together while ignore the information between tasks. From Fig. 1 (performance on SAD has a similar trend with Isolet, and thus its figure is omitted), we find that MTL learns better than STL. Fig. 1 also tells us learners with privileged information perform better than ones without this kind of information. Because of making full use of MTL and LUPI, M^{2}PiSVMs get the best results on two datasets.

7. Conclusion

In this paper, we present two new models MPiSVMs and M^{2}PiSVMs which are based on the LUPI. Because it can take full advantage of the multitask learning and privileged information, M^{2}PiSVMs get the best performance. As mentioned in [7], a SMO type algorithm for our model is valuable to study in the future. How to deal with privileged information which is related across different tasks is very interesting to study further. Extending M^{2}PiSVMs to multiple spaces' privileged information is also valuable for future investigation.

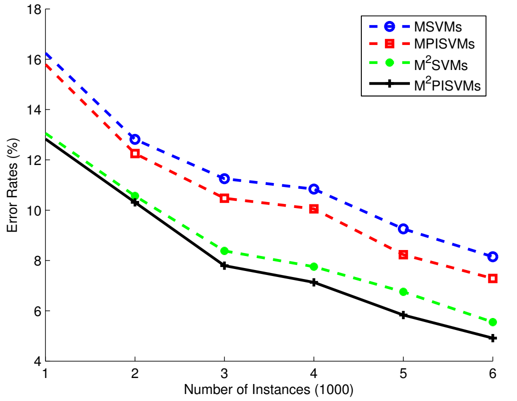

Figure 1. Performance with different training sizes on Isolet

References

[1] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[2] T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[3] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117, 2004.

[4] S. Parameswaran and K. Weinberger. Large margin multi-task metric learning. In *Advances in Neural Information Processing Systems*, 2010.

[5] Y. Ji and S. Sun. Multitask multiclass support vector machines. In *Proceedings of the International Conference on Data Mining Workshops*, pages 512–518, 2011.

[6] S. Sun. Multitask learning for EEG-based biometrics. In *Proceedings of the 19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[7] D. Pechyony, R. Izmailov, A. Vashist, and V. Vapnik. Smo-style algorithms for learning using privileged information. In *Proceedings of the International Conference on Data Mining*. Citeseer, 2010.

[8] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2002.