Connection Between SVM+ and Multi-Task Learning

Lichen Liang and Vladimir Cherkassky

***Abstract*—Exploiting additional information to improve traditional inductive learning is an active research in machine learning. When data are naturally separated into groups, SVM+[7] can effectively utilize this structure information to improve generalization. Alternatively, we can view learning based on data from each group as an individual task, but all these tasks are somehow related; so the same problem can also be formulated as a multi-task learning problem. Following the SVM+ approach, we propose a new multi-task learning algorithm called svm+MTL, which can be thought as an adaptation of SVM+ for solving MTL problem. The connections between SVM+ and svm+MTL are discussed and their performance is compared using synthetic data sets.**

I. Introduction

Under inductive learning setting [3,4,5,6], the goal is to find a mapping function f which maps an input vector $\mathbf{x} \in \mathbf{X}$ to an output $y \in \mathbf{Y}$. This estimation (learning) is performed using a training set of i.i.d. samples generated from an unknown probability distribution $P(\mathbf{x}, y)$. The goal is to find the best mapping function f such that the expected loss

$$R(w) = \int L(f(\mathbf{x}, w), y) P(x, y) d\mathbf{x} dy$$

is minimized. Note that $L(f(\mathbf{x}, w), y)$ denotes a loss function such as classification error, or squared-loss.

Suppose that training data can be represented as a union of t related groups, i.e. each group $r \in [1,2,..,t]$ contains n_r i.i.d. samples from a distribution P_r on $\mathbf{X} \times \mathbf{Y}$. Therefore, available training data is a union of $t>1$ groups:

$\{\{X_r, Y_r\}, r = 1,...,t\}, \{X_r, Y_r\} = \{\{\mathbf{x}_{r_1}, y_{r_1}\},...,\{\mathbf{x}_{r_{nr}}, y_{r_{nr}}\}\}$

and can be though as samples identically and independently generated from the distribution $P = \cup_{r=1,..t} P_r$.

If the group labels of future test samples are not given, the problem is "**L**earning **W**ith **S**tructured **D**ata (LWSD)" formulation [7]. In this formulation, the goal is to find one best mapping function f such that the expected loss

$$R(w) = \int L(f(\mathbf{x}, w), y) P(\mathbf{x}, y) d\mathbf{x} dy$$

is minimized. Note that even though the expected loss is in the same form as in the supervised learning setting, the difference is that in supervised learning setting P is unknown, while in LWSD, P is a union of t sub-distributions.

On the other hand, if the group labels of future test samples are given, the problem is **M**ulti-**T**ask **L**earning (MTL) problem [1,2]. The goal in multi-task learning is to find t mapping functions $\{f_1, f_2,..., f_t\}$ such that the sum of expected losses for each task

$$R(w) = \sum_{r=1}^{t} (\int L(f_r(\mathbf{x}, w), y) P_r(\mathbf{x}, y) d\mathbf{x} dy)$$

is minimized. Figure 1 illustrates how standard supervised learning, multi-task learning and learning with structured data utilize training and test data in different ways.

"Learning with structured data" formulation and multi-task learning formulation are similar in the sense that they all try to exploit the group information. However, there are several important differences: (1) LWSD estimates a single model, while MTL estimates t models; (2) LWSD does not use the group membership of test data, whereas MTL does require it. Let's consider two application problems in order to illustrate the difference between the two formulations. One example is handwritten digit recognition, where the training data originates from t persons (each person provides labeled examples of all 10 digits). Then goal 1 (LWSD) is to find a classifier that can generalize well for other (previously unseen) samples written by these people (we don't know who generates test samples). In contrast, goal 2 (MTL) is improved generalization for each person who contributed to training data (i.e., the group membership for future test samples is known). Another application example is fMRI data analysis or more generally, medical diagnosis. Here the goal is to estimate a predictive model (predict/diagnose a disease) from the training samples from t patients. Then goal 1 (LWSD) is to find a predictive model that has good generalization for other (new) samples from these patients, whereas the goal 2 (MTL) is to built t specialized models (one for each patient). Note that estimating a classifier that can generalize well for data from a new group (previously unseen in the training data) does not fall into either LWSD or MTL formulation, and it represents a problem formulation beyond the scope of this paper. A recent empirical comparison study [9] shows that SVM+ provides an improved generalization accuracy for fMRI data. This paper attempts to extend the concepts developed for SVM+ algorithm, in order to solve multi-task learning problems.

Lichen Liang is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455, USA. (e-mail: lian0064@ umn.edu).

Vladimir Cherkassky is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA. (e-mail: cherk001@ umn.edu) .

II. SVM+ APPROACH

SVM+ [7] is an algorithm for Learning with Structured Data developed as an extension of standard SVM. This section briefly reviews standard SVM classification (in order to introduce notation) and then describes SVM+.

A. Standard SVM

Given a training set $\{\{\mathbf{x}_i, y_i\}\}_{1\le i\le n}, \mathbf{x}_i \in R^d, y_i \in \{+1,-1\}$, SVM finds a maximum margin separating hyperplane $f(\mathbf{x}) = (\mathbf{w},\mathbf{x}) + b$ between two classes [5]. $f(\mathbf{x}) = (\mathbf{w},\mathbf{x}) + b$ is also called decision function. To this end, SVM solves the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}(\mathbf{w},\mathbf{w}) + C\sum_{i=1}^{n}\xi_i \qquad \text{(OP1)}$$
$$\text{subject to: } y_i((\mathbf{w},\mathbf{x}_i)+b) \ge 1-\xi_i$$
$$\xi_i > 0$$

ξ_i , $i = 1,...n$ are called slack variables, which indicates the deviation from the margin borders. $(\mathbf{w},\mathbf{w})$ indicates the size of margin, which represents the model complexity of SVM. The coefficient C controls the trade-off between complexity and proportion of nonseparable samples and must be selected by the user. (see Figure 2)

In the non-linear version of SVM, we first map the input training data into a feature space $\Phi(\mathbf{x}_i) = \mathbf{z}_i$, and then find the optimal decision function in that feature space. The non-linear form of SVM is similar to the optimization (OP1). The only difference is that $\mathbf{w}, \mathbf{z}_i$ (see OP1') are defined in the feature space. The non-linear SVM solves the optimization problem as:

$$\min_{w,b} \frac{1}{2}(\mathbf{w},\mathbf{w}) + C\sum_{i=1}^{n}\xi_i \qquad \text{(OP1')}$$
$$\text{subject to: } y_i((\mathbf{w},\mathbf{z}_i)+b) \ge 1-\xi_i$$
$$\xi_i > 0$$

sum of slack variables ξ_i corresponding to deviation from margin borders; (b) Maximizing the size of margin.

B. SVM+

Suppose that training data are the union of t>1 groups. Let us denote the indices from group r by $T_r = \{i_{n1},...,i_{n_r}\}, r = 1,...,t$. Then all training samples can be represented as:

$$\{\{X_r, Y_r\}, r = 1,...,t\}, \{X_r, Y_r\} = \{\{\mathbf{x}_{r_1}, y_{r_1}\},...,\{\mathbf{x}_{r_{nr}}, y_{r_{nr}}\}\}$$

To account for the group information, Vapnik [7] proposes to define the slacks inside one group by some *correcting function:*

$$\xi_i = \xi_r(\mathbf{x}_i) = \phi_r(\mathbf{x}_i, \mathbf{w}_r), \quad i \in T_r, r = 1,...,t.$$

To define the correcting function $\xi_r(\mathbf{x}_i) = \phi_r(\mathbf{x}_i, \mathbf{w}_r)$ for group T_r, Vapnik proposed to map the input vectors $\mathbf{x}_i, i \in T_r$ simultaneously into two different Hilbert spaces: into the decision space $\mathbf{z}_i = \Phi_z(\mathbf{x}_i) \in Z$ which defines the decision function and into correcting space $\mathbf{z}_i^r = \Phi_{z_r}(\mathbf{x}_i) \in Z_r$ which defines the set of correcting functions for a given group r. The correcting functions are represented by $\xi_r(\mathbf{x}_i) = (\mathbf{x}_i, \mathbf{w}_r) + d_r, r = \{1,...,t\}$

Compared to standard SVM, here the slack variables are restricted by the correcting functions, and the correcting functions represent additional information about the data. The goal is to find the decision function in decision space Z,

$$f(\mathbf{x}) = (\mathbf{w}, \Phi_Z(\mathbf{x})) + b$$

Note that data of different groups are mapped into the same decision space, and they all used to construct the decision function. However, there are different correcting functions for different groups. Correcting functions are defined in the correcting space. Different correcting functions can be defined either in the same correcting space or different correcting function spaces. Of course, if data of different groups are mapped to different correcting spaces, the correcting functions for different groups are different. If data of different groups are mapped to the same correcting space, we still can construct different correcting functions for different groups. The important point is that the correcting functions are different, not the correcting space.

Correcting functions represent a unique way that SVM+ handles group information (see Figure 3). Since correcting functions represent slack variables, they have some unique characteristics:

(1) All slack variables are non-negative, so $\xi_r(\mathbf{x}_i) = (\mathbf{x}_i, \mathbf{w}_r) + d_r \ge 0, r = \{1,...,t\}$. Therefore mapping samples in the correcting space have to lie on one side of the corresponding correcting function. Correcting function also has to pass through some points with slack variables being zero.

(2) Like decision function, correcting function is also chosen from a set of correcting functions, and $(\mathbf{w}_r, \mathbf{w}_r)$ reflects the capacity of the set of correcting functions; but this term does not have meaning of the size of margin.

(3) Correcting functions is not used to assign a sample a group membership.

Implementation of SVM+ is based on Vapnik's idea to control the capacity of a set of decision functions, and the capacity of a set of correcting functions. To this end, we need to solve the following optimization problem :

$$\min_{w,w_1,..,w_t,b,d_1,..d_t} \frac{1}{2}(\mathbf{w},\mathbf{w}) + \frac{\gamma}{2}\sum_{r=1}^{t}(\mathbf{w}_r,\mathbf{w}_r) + C\sum_{r=1}^{t}\sum_{i\in T_r}\xi_i^r$$

(OP2)

subject to:

$$y_i((\mathbf{w},\mathbf{z}_i)+b)\geq 1-\xi_i^r \quad ,i\in T_r, r=1,...,t$$
$$\xi_i^r \geq 0, i\in T_r, r=1,...,t$$
$$\xi_i^r = (\mathbf{z}_i^r,\mathbf{w}_r)+d_r, i\in T_r, r=1,...,t$$

The capacity of a set of decision functions is reflected by $(\mathbf{w},\mathbf{w})$ and the capacity of a set of correcting functions for group r is $(\mathbf{w}_r,\mathbf{w}_r)$. SVM+ directly controls the capacity of decision functions and correcting functions. Parameter γ adjusts the relative weight of these two capacities. C controls the trade-off between complexity and the number of nonseparable samples. In this problem, the slack variables are represented as $(\mathbf{z}_i^r,\mathbf{w}_r)+d_r$, and must be non-negative.

Using the dual optimization technique (similar to standard SVM) one can show that $\mathbf{w},\mathbf{w}_r$ can be expressed in terms of training samples:

$$\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i \mathbf{z}_i$$

$$\mathbf{w}_r=\frac{1}{\gamma}\sum_{i\in T_r}(\alpha_i+\mu_i-C)\mathbf{z}_i^r$$

where the coefficients α_i maximize the functional:

$$\max_{\alpha,\mu} W(\alpha,\mu)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{z}_i,\mathbf{z}_j)$$
$$-\frac{1}{2\gamma}\sum_{r=1}^{t}\sum_{i,j\in T_r}(\alpha_i+\mu_i-C)(\alpha_j+\mu_j-C)(\mathbf{z}_i^r,\mathbf{z}_j^r) \qquad (OP2')$$

subject to:

$$\sum_{i=1}^{n}\alpha_i y_i=0$$
$$\sum_{i\in T_r}(\alpha_i+\mu_i)=|T_r|C \quad ,r=1,...,t$$
$$\alpha_i\geq 0, \mu_i\geq 0, \quad i=1,...,n$$

Therefore, the optimal decision function in Z space has the form

$$f(\mathbf{x})=\sum_{i=1}^{n}\alpha_i y_i(\Phi_z(\mathbf{x}_i),\Phi_z(\mathbf{x}))+b,$$

Compared to SVM, SVM+ adds $\frac{\gamma}{2}\sum_{r=1}^{t}(\mathbf{w}_r,\mathbf{w}_r)$ in the objective function in the primal form, and also adds a new constraint $\xi_i^r=(\mathbf{z}_i^r,\mathbf{w}_r)+d_r$.

The dual form of SVM+ has an additional term $\frac{1}{2\gamma}\sum_{r=1}^{t}\sum_{i,j\in T_r}(\alpha_i+\mu_i-C)(\alpha_j+\mu_j-C)(\mathbf{z}_i^r,\mathbf{z}_j^r)$ in the objective function, and more constrained α_i's .

III. SVM+ FOR MULTI-TASK LEARNING (SVM+MTL)

A. *svm+MTL*

This section describes an extension of SVM+ approach to multi task learning. In order to apply SVM+ to multi task learning, we need to specify: (1) how SVM+ can define decision functions for different groups; (2) how SVM+ can model task (group) relatedness.

The proposed method svm+MTL is described next. Similar to SVM+, we map the input vectors $\mathbf{x}_i, i\in T_r$ simultaneously into two different Hilbert spaces: the decision space $\mathbf{z}_i=\Phi_z(\mathbf{x}_i)\in Z$ and the correcting space $\mathbf{z}_i^r=\Phi_{z_r}(\mathbf{x}_i)\in Z_r$ for a given group r.

The goal is to find the *t* decision functions

$$f_r(\mathbf{x})=(\Phi_z(\mathbf{x}),\mathbf{w})+b+(\Phi_{z_r}(\mathbf{x}),\mathbf{w}_r)+d_r \quad ,r=1,...,t$$

Each decision function includes two parts: common decision function $(\Phi_z(\mathbf{x}),\mathbf{w})+b$ and unique correcting function $(\Phi_{z_r}(\mathbf{x}),\mathbf{w}_r)+d_r$. Common decision function is defined in the decision space Z and unique correcting function defined in the correcting space Z_r, so the final decision function actually involves two spaces: decision space and correcting space.

The *t* tasks are related in the sense that decision functions for different tasks share a common decision function.

Note that like SVM+, correcting functions of different groups may lie in the same correcting space or different correcting spaces.

Formally, the proposed svm+MTL method needs to solve the following optimization problem:

$$\min_{w,w_1,..,w_t,b,d_1,..d_r}\frac{1}{2}(\mathbf{w},\mathbf{w})+\frac{\gamma}{2}\sum_{r=1}^{t}(\mathbf{w}_r,\mathbf{w}_r)+C\sum_{r=1}^{t}\sum_{i\in T_r}\xi_i^r$$

(OP3)

subject to:

$$y_i^r((\mathbf{w},\mathbf{z}_i)+b+(\mathbf{w}_r,\mathbf{z}_i^r)+d_r)\geq 1-\xi_i^r, i\in T_r, r=1,...,t$$
$$\xi_i^r\geq 0 \quad ,i\in T, r=1,...,t$$

Following SVM+, we use 2-norm of $\mathbf{w},\mathbf{w}_r$ to control the common decision function capacity and correcting function capacity. Parameter γ adjusts the relative weight of these two capacities. C controls the trade-off between complexity and proportion of nonseparable samples. The slack variables ξ_i^r measure the error that each group models (including the common decision function and correcting function) makes on the training data.

This formulation (OP3) has some similarity to the regularized multi-task learning technique (rMTL) proposed by Evgeniou and Pontil [8] , which also assumes decision function has a common part and unique part. The relatedness of decision functions for different groups are characterized by the common decision function. rMTL solves the following optimization problem (OP4):

$$\min_{w,w_1,..,w_t} \frac{1}{2}(\mathbf{w},\mathbf{w}) + \frac{\gamma}{2}\sum_{r=1}^{t}(\mathbf{w}_r,\mathbf{w}_r) + C\sum_{r=1}^{t}\sum_{i\in T_r}\xi_i^r$$

(OP4)

subject to:

$$y_i^r((\mathbf{w},\mathbf{z}_i) + (\mathbf{w}_r,\mathbf{z}_i)) \geq 1-\xi_i^r, i\in T_r, r=1,...,t$$

$$\xi_i^r \geq 0 \quad , i\in T, r=1,...,t$$

However, there are two important differences between our svm+MTL and rMTL. First, under rMTL formulation the decision space and correcting space are the same, whereas the svm+MTL formulation these spaces may be different. Second, rMTL assumes that the decision function for each task is :

$f_r(x) = (\Phi_z(x), w) + (\Phi_z(x), w_r) \quad , r=1,...,t$, while svm+MTL considers a more general form with bias terms $(b,\ d_r)$.

The dual form of (OP3) is as follows:

$$\max_{\alpha,\mu} W(\alpha,\mu) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{z}_i,\mathbf{z}_j) - \frac{1}{2\gamma}\sum_{r=1}^{t}\sum_{i,j\in T_r}\alpha_i\alpha_j y_i y_j(\mathbf{z}_i^r,\mathbf{z}_j^r)$$

subject to:

$$\sum_{i\in T_r}\alpha_i y_i = 0 \quad , r=1,...,t$$

$$\alpha_i + \mu_i = C \quad , i=1,...,n$$

$$\alpha_i \geq 0, \mu_i \geq 0, \quad i=1,..,n$$

Based on KKT conditions, we can express w, w_r in terms of training samples:

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{z}_i$$

$$\mathbf{w}_r = \frac{1}{\gamma}\sum_{i\in T_r}\alpha_i y_i \mathbf{z}_i^r$$

Thus,

$$f_r(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i y_i(\mathbf{z}_i,\Phi_z(\mathbf{x})) + b + \frac{1}{\gamma}\sum_{i\in T_r}\alpha_i y_i(\mathbf{z}_i^r,\Phi_{z_r}(\mathbf{x})) + d_r, r=1,..,t$$

B. Connection between SVM+ and svm+MTL

OP3 can be though as an adaptation of SVM+ for solving MTL problems. Comparing OP2 and OP3, we can observe that both formulations utilize correcting functions. However, correcting functions play different roles in the two formulations. In SVM+, correcting functions are used to model slack variables, which have to be non-negative. In svm+MTL, correcting functions are used to fine-tune the decision function so that it fits the data from each group more appropriately; so it is not required for the correcting function to be non-negative.

The inherent connection between SVM+ and svm+MTL can be clarified by comparing their dual forms. We can see that the mappings $\mathbf{z}_i^r = \Phi_{z_r}(\mathbf{x}_i) \in Z_r$ in correcting spaces play different roles in these two formulations. In SVM+, the objective function contains the term

$$\frac{1}{2\gamma}\sum_{r=1}^{t}\sum_{i,j\in T_r}(\alpha_i+\mu_i-C)(\alpha_j+\mu_j-C)(\mathbf{z}_i^r,\mathbf{z}_j^r),$$

while the corresponding term in svm+MTL is:

$$\frac{1}{2\gamma}\sum_{r=1}^{t}\sum_{i,j\in T_r}\alpha_i\alpha_j y_i y_j(\mathbf{z}_i^r,\mathbf{z}_j^r).$$

Looking at constraints, we can observe that svm+MTL has more restricted constraints, because it requires $\sum_{i\in T_r}\alpha_i y_i = 0$ for data in each group (task). In contrast, SVM+ only places constraints $\sum_{i=1}^{l}\alpha_i y_i = 0$ for all data samples.

IV. EMPIRICAL COMPARISONS

This section describes empirical comparisons between SVM+ and svm+MTL, using synthetic data sets.

Synthetic data generation:

(1) Let number of input features be d=20, and number of tasks(groups) be t=3.

(2) Generate $\mathbf{x}\in R^{20}$ with each component $x_i \sim uniform(-1,1) \quad , i=1,...,20$.

(3) The coefficient vectors of three tasks are specified as

$\beta_1 = [1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0]$

$\beta_2 = [1,1,1,1,1,1,1,1,0,1,0,1,0,0,0,0,0,0,0,0]$

$\beta_3 = [1,1,1,1,1,1,1,1,0,1,0,0,0,0,0,1,0,0,0,0]$

(4) For each task and each data vector, $y = sign(\beta_i\mathbf{x} + 0.5)$

For each task, we generate 100 data samples for training, 100 samples for validation, and 2000 samples for testing, and repeat the process 10 times. Training data is used for training model, validation data used for model selection, and testing data for evaluating generalization performance of the final model. We use linear kernel for decision space, and Gaussian kernel for correcting space

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\|\mathbf{x}_i-\mathbf{x}_j\|^2/2\sigma^2).$$

Thus, in total, we have 3 tuning parameters for SVM+ and svm+MTL: C, γ, σ. The possible choices for parameters are C=[0.1, 1, 10], γ=[0.1,1,10], and $\sigma = [0.5,1,2]$. In Table 1, we show the classification accuracy for each trial. Additionally, linear SVM with C=[0.1, 1, 10] is also used for comparison. Namely, training samples from all tasks are pooled together and are all used to estimate a linear SVM classifier.

The average accuracy and standard deviation for SVM, SVM+ and svm+MTL are 88.11(0.65), 88.31(0.84) and 91.47(1.03), respectively. Both SVM+ and svm+MTL outperform SVM. We note that svm+MTL performs better than SVM+. It is not surprising because svm+MTL uses additional information about the group label of test data, which is not used in SVM+.

To check the effect of the training sample size, we reduced the number of training and validation samples to 50 and 15 per group. The other parameters were same as used the last experiment. The result is shown in Table 2 and Table 3 respectively. For 50 training samples, the average

accuracy and standard deviation for SVM, SVM+ and svm+MTL are 85.74(1.36), 86.49(1.69), 87.39(2.29), respectively. Similar to the case of 100 training samples, both SVM+ and svm+MTL outperform SVM. In all trials, svm+MTL outperforms SVM+.

For 15 training samples, the average accuracy and standard deviation for SVM, SVM+ and svm+MTL are 80.10(3.42), 80.84(3.16) and 79.24(2.81) . In this case, SVM+ slightly outperforms better than SVM, and both SVM and SVM+ perform better than svm+MTL. This can be explained by observing that svm+MTL has to estimate 3 different model (vs a single model for SVM and SVM+), so it requires sufficient number of training samples. Clearly, 15 training samples for each group is not enough for the training svm+MTL.

V. CONCLUSION

In this paper, we investigated how to utilize available group information in order to improve prediction accuracy. We extended SVM+ (originally proposed for Learning with Structural Data) to the problem of multi-task learning. The proposed new technique called SVM+MTL can be applied to solving multi-task learning problem. Empirical comparisons illustrate the advantages of the proposed method (vs SVM+) for multi-task learning settings, when the number of training samples (per task) is sufficiently large. On the other hand, when the number of samples per task is small, standard SVM and SVM+ methods are shown to outperform svm+MTL.

REFERENCES

[1] Ando, R. and Zhang, T. A Framework for Learning predictive structures from multiple tasks and unlabeled data, Journal of Machine Learning Research, 2005.

[2] Ben-David, S., Gehrke, J. and Schuller, R. A theoretical framework for learning form a pool of disparate data sources. ACM KDD, 2002.

[3] Cherkassky, V. and Mulier, F. Learning from Data, John Wiley & Sons, New York, second edition, 2007.

[4] Hastie, T. , Tibshirani, R. and Friedman, J. The Elements of Statistical Learning. Data Mining, Inference and Prediction, Springer, 2001.

[5] Vapnik, V. Estimation of Dependences Based on Empirical Data, Springer Verlag, New York, 1982.

[6] Vapnik, V. Statistical Learning Theory, Wiley, New York, 1998.

[7] Vapnik, V. Empirical Inference Science Afterword of 2006, Springer, 2006.

[8] Evgeniou, T. and Pontil, M.. Regularized multi--task learning. In Proc. 17th SIGKDD Conf. on Knowledge Discovery and Data Mining, 2004.

[9] Liang, L. and Cherkassky, V. Learning Using Structured Data: Application to fMRI data analysis, IJCNN, 2007.

Table 1: Classification Accuracy (%) of SVM+ and svm+MTL for synthetic data (100 samples per task).

Trials	1	2	3	4	5	6	7	8	9	10
SVM	88.12	87.25	88.75.	88.50	88.10	89.15	87.27	87.82	88.60	87.52
SVM+	88.60	86.95	88.62	88.93	88.42	89.90	87.28	88.08	88.52	87.80
Svm+MTL	**91.55**	**89.82**	**91.93**	**92.82**	**91.28**	**91.57**	**89.60**	**92.33**	**92.17**	**91.62**

Table 2: Classification Accuracy (%) of SVM+ and svm+MTL for synthetic data (50 samples per task).

Trials	1	2	3	4	5	6	7	8	9	10
SVM	86.68	85.37	86.70	86.98	84.38	83.53	87.32	86.58	83.92	85.93
SVM+	87.73	86.35	87.43	87.45	**84.42**	**85.77**	**88.12**	88.07	82.97	86.55
Svm+MTL	**89.82**	**87.88**	**88.73**	**89.42**	83.68	84.20	87.42	**89.80**	**85.10**	**87.83**

Table 3: Classification Accuracy (%) of SVM+ and svm+MTL for synthetic data (15 samples per task).

Trials	1	2	3	4	5	6	7	8	9	10
SVM	74.97	**80.58**	**81.67**	79.80	85.80	77.65	79.48	85.32	77.52	78.25
SVM+	76.77	80.50	81.37	**79.82**	**86.25**	**77.77**	**79.97**	**86.32**	**79.28**	**80.38**
Svm+MTL	**78.07**	79.38	79.43	78.82	80.92	73.90	78.58	85.15	77.85	80.25

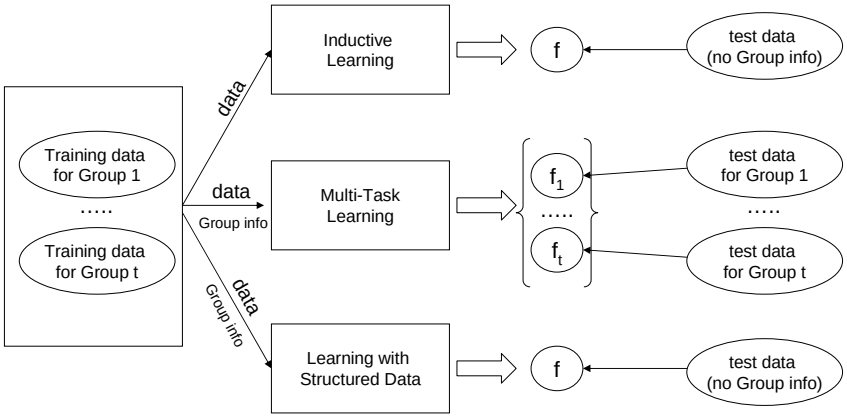

Figure 1: Inductive learning, Multi-task learning, and Learning with structured data use training and test data in different ways.

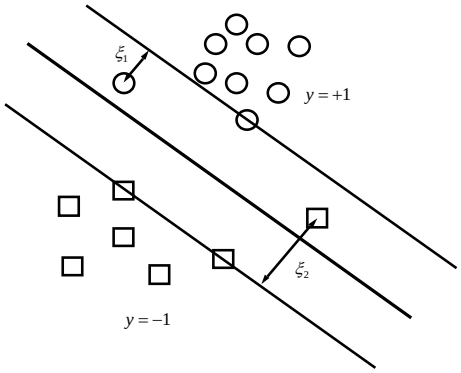

Figure 2. Binary classification for non-separable data involves two goals: (a) Minimizing the total error for data samples unexplained by the model, usually quantified as a sum of slack variables ξ_i corresponding to deviation from margin borders; (b) Maximizing the size of margin.

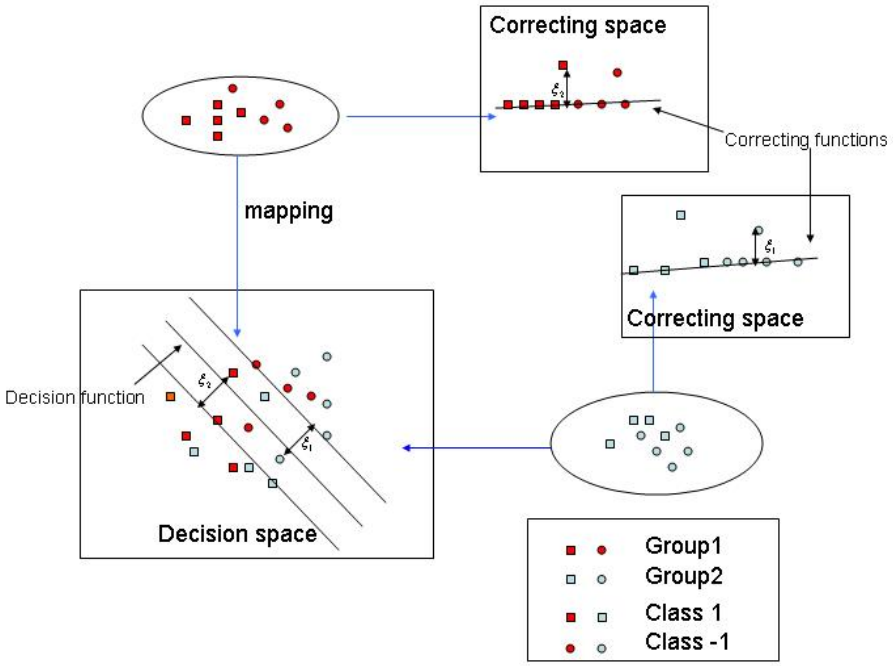

Figure 3: SVM+ maps data (from two groups) simultaneously into decision space and correcting spaces. Decision function is found in the decision space. Slack variables are represented by correcting functions defined in the correcting space.