# Predictive Learning with Sparse Heterogeneous Data

Vladimir Cherkassky, Fellow, IEEE, Feng Cai and Lichen Liang

***Abstract* — many applications of machine learning involve sparse and heterogeneous data. For example, estimation of predictive (diagnostic) models using patients' data from clinical studies requires effective integration of genetic, clinical and demographic data. Typically all heterogeneous inputs are properly encoded and mapped onto a single feature vector, used for estimating (training) a predictive model. This approach, known as standard inductive learning, is used in most application studies. More recently, several new learning methodologies have emerged. In particular, when training data can be naturally separated into several groups (or structured), we can view learning (estimation) for each group as a separate task, leading to Multi-Task Learning framework. Similarly, a setting where training data is structured, but the objective is to estimate a single predictive model (for all groups), leads to Learning with Structured Data and SVM+ methodology recently proposed by Vapnik. This paper demonstrates advantages and limitations of these new data modeling approaches for modeling heterogeneous data (relative to standard inductive SVM) via empirical comparisons using several publicly available medical data sets.**

## I. INTRODUCTION and MOTIVATION

Statistical data-driven computer aided diagnostics have been of growing interest in biomedical applications. Such approaches usually estimate diagnostic models from available (historical) data. Whereas machine learning and statistical approaches often pursue similar goals and use similar techniques, there is a key difference in perspective [3]. Under predictive learning, the main goal of modeling is good prediction (generalization) for future data. In contrast, statisticians view the probability model as the core of the analysis, with the idea that optimal predictions will arise from this probability model accurately estimated from data. Sometimes machine learning algorithms correspond to statistical models (e.g., mixture models), but other times the predictions feel more like they are coming from 'black boxes' with less statistical interpretation. This distinction is often known as generative (~statistical) vs discriminative (~ predictive) modeling. For multivariate sparse data sets common in biomedical applications, the predictive approach is more practical because

(a) there is simply not enough available data samples to estimate multivariate distributions (this is known as the curse of dimensionality).

(b) it may be possible to estimate accurate predictive models that reflect *certain properties* of unknown distributions [3,8,9]. For example, for classification problems, the goal of estimating decision boundary (for future predictions) does not require accurate estimation of class distributions. Moreover, Statistical Learning Theory aka VC theory [7-9] gives mathematical conditions under which good prediction (generalization) is possible with finite samples, *regardless of dimensionality* (the number of input variables).

The price paid for adopting the predictive approach is that the estimated models may *accurately predict*, but only in a specific well-defined sense (known as 'generalization'). This places an additional burden on a data modeler who needs to come up with a *meaningful formalization* of an application domain at hand. In particular, this approach requires *close collaboration* between data modelers and clinicians (who provide the data and will use data-driven predictive models). It also implies that medical researchers/clinicians should understand better conceptual aspects of predictive learning. Another important difference is that predictive models may not be easily interpretable, because they do not approximate 'true' distributions, but rather imitate certain properties of unknown distributions.

Future advances in the area of data-driven biomedical applications are limited by two fundamental factors: (a) high dimensionality of the input data (i.e., large number of input variables) and (b) heterogeneous nature of the input data. *High-dimensional, low sample size* (HDLSS) data is common in many biomedical applications, especially studies involving genetic data. For example, a 'typical' clinical study may result in a data set of a few hundred to a couple of thousands patients ('samples'), where each patient has a few hundred genetic predictors (for instance, ~ 400 genetic polymorphisms), in addition to a few dozen clinical and demographic inputs. All these heterogeneous inputs may be used as possible predictors for diagnosing a disease or predicting the outcome of a medical treatment procedure.

For such datasets, the dimensionality $d$ of the data vector may be larger than/ similar to the sample size $n$. Such sparse training data sets present new challenges to classification methods that estimate classification decision boundaries from HDLSS data. Note that commonly used discriminative methods (such as neural networks and support vector machines) require significant modifications and/or clever preprocessing in dealing with HDLSS data. *Heterogeneous*

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

Feng Cai is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455, USA. (e-mail: caixx043@umn.edu).

Lichen Liang is with Massachusetts General Hospital, Boston, MA 02114, USA. (e-mail: lian0064@umn.edu).

.

*data* in biomedical applications may include clinical, genomic and demographic data used as input variables for constructing a predictive (diagnostic) model. These inputs can be viewed as several feature sets, and the challenge is to integrate such input data from different modalities into learning with sparse high-dimensional data. There are two principal approaches for dealing with HDLSS and heterogeneous data [3]:

- *first approach* is to adopt *standard inductive learning* setting, and to reduce the problem dimensionality via clever preprocessing and feature extraction. That is, the problem of high-dimensional input space is addressed by dimensionality reduction (feature selection aka subset selection), and the problem of heterogeneous data is handled by encoding of all inputs into the same type. Then a standard inductive classifier (such as Support Vector Machine (SVM), or neural network, or logistic regression) is used to estimate a model. This approach has been successfully used in many biomedical and image processing applications [11]. Commonly used statistical approaches to modeling genetic data for diagnostic and prognostic classification follow feature selection strategy (aka subset selection) where a few strong informative inputs are selected from a large number of inputs, typically using greedy feature selection. Selection of inputs in the final model is performed via extensive use of resampling [12].
- *second approach* is to investigate new learning settings for dealing with HDLSS heterogeneous data. This approach is based on the fundamental principle (due to Vapnik) that for finite sample estimation problems one should always use the most appropriate *direct formulation* of the learning problem rather than a more general formulation. It can be argued that most recent advances in statistical learning (i.e., transduction, semi-supervised learning, single-class learning, multi-task learning) reflect an improved understanding of the learning problem setting.

In this paper, we investigate application of novel learning methodologies, such as SVM+, and Multi Task Learning (MTL), to classification problems using several medical data sets. The goal is to present several different ways to model heterogeneous data (as discussed in Section 2), and then investigate advantages and limitations of different learning approaches via empirical comparisons, presented in Section 3. Finally, conclusions and discussion are given in Section 4.

## II. APPROACHES for MODELING HETEROGENEOUS DATA

In this paper, we consider supervised learning applications where the training data includes additional (group) information about training samples. Examples include: (1) handwritten digit recognition where training examples are provided by several persons, (2) medical diagnosis where predictive (diagnostic) model, say for lung cancer, is estimated using a training data set of male and female patients, etc. Incorporating this additional information has lead to approaches known as Multi-Task Learning [1,2,6,10] and, more recently, to Learning with Structured Data (aka SVM+) [9], as briefly discussed next.

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

$$\{\{\mathbf{X}_r, \mathbf{Y}_r\}, r = 1,...,t\}, \{\mathbf{X}_r, \mathbf{Y}_r\} = \{\{\mathbf{x}_{r_1}, y_{r_1}\}, ..., \{\mathbf{x}_{r_{nr}}, y_{r_{nr}}\}\}$$

and can be thought as samples identically and independently generated from unknown distribution $P(\mathbf{x}, y) = \{P_r(\mathbf{x}, y), if \{\mathbf{x}, y\} \in \{\mathbf{X}_r, \mathbf{Y}_r\}\}$.

If the group labels of future test samples are not given, the problem is "**L**earning **W**ith **S**tructured **D**ata (LWSD)" formulation [9]. In this formulation, the goal is to find one best mapping function *f* such that the expected loss

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

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 it is known that *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 [1,2,6,8]. The goal in multi-task learning is to estimate *t* related classifiers $\{f_1, f_2, ..., f_t\}$ so that the sum of expected losses for each task

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

is minimized.

From the application point of view, different learning settings (standard inductive learning, multi-task learning and learning with structured data) handle training and test data in different ways. That is, standard inductive setting does not use (ignores) group information in the training data; MTL setting estimates *t* separate related predictive models; and LWSD estimates a single model that utilizes group information in the training data. Note that under LWSD test inputs do not have group information, whereas under MTL test inputs have (known) group labels.

Recently, Vapnik [9] proposed SVM-based optimization formulation called SVM+ for LWSD formulation. Liang and Cherkassky [5,6] showed empirical validation of SVM+ for *classification*, and showed its connection to Multi-Task Learning (MTL) classifiers in machine learning [1,2,6,10].

"Learning with structured data" formulation [9] and multi-task learning are similar in the sense that they both try to exploit the group information hidden in the data. Such 'group information' is common in many applications with *heterogeneous* data. For example, in medical diagnostic applications, certain inputs, for example patients' demographic features, such as Gender or Age, can be used to separate labeled training data into several groups. Proper selection of such a *group variable* is specific to each application at hand (see examples in Section 3).

Assuming that available training data can be partitioned (in a meaningful way) into several groups, we can identify several learning approaches for utilizing this group information. These approaches are shown in Fig. 1 where, for simplicity, we show two groups, and use SVM classifier as a basic inductive learning method:

- *Single SVM* inductive model which estimates standard SVM classifier by pooling together training samples from different groups (i.e. group information is ignored);
- *multiple SVM* approach where a separate SVM classifier is estimated for each group (independently);
- *SVM+* approach where a single classifier model, utilizing available group information, is estimated from all data;
- *SVM+MTL* implementing multi-task learning, which estimates several related classification models following [5,6].

Various approaches for incorporating group data into learning process are presented in Fig. 1, showing for simplicity *t=2* groups.

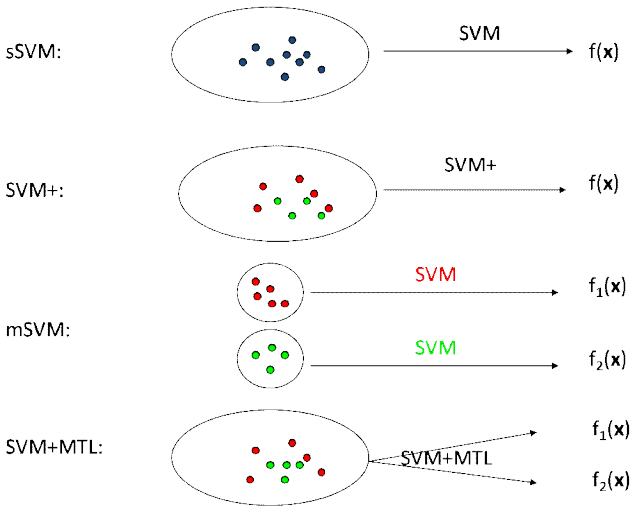


Figure 1: Different ways of using group information in learning: (a) sSVM ~ Single SVM classifier, (b) SVM+ classifier, (c) mSVM ~ multiple (independent) SVMs, and (d) SVM+MTL ~ SVM+ Multi-Task Learning [6].

In this paper, we use SVM as an underlying technology for implementing different approaches utilizing group information. However, one can use other learning techniques, for example, MLP networks, for implementing standard inductive learning and Multi-Task Learning. Theoretically, one can expect more sophisticated modeling approaches (utilizing the group information), i.e., SVM+ and SVM+MTL, to yield better generalization than single inductive SVM and multiple (independent) SVM's, respectively. In practice, the trade-off is not so clear, because more advanced approaches (SVM+ and SVM+MTL) have more tunable parameters (than standard SVM), and their potential advantages can be easily offset by more complex model selection.

Optimization formulation for SVM+ and SVM+MTL classification is given below. For detailed mathematical description of SVM+ and SVM+MTL, see [9] and [5, 6] respectively.

*A. Standard SVM classifier*

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 [8,9]. This optimal decision function $f(\mathbf{x}) = (\mathbf{w},\mathbf{x}) + b$ is estimated from training data by solving the following optimization problem:

$$\min_{\mathbf{w},b} \ \frac{1}{2}(\mathbf{w},\mathbf{w}) + C\sum_{i=1}^{n}\xi_i \qquad \text{(OP1)}$$

$$\text{subject to: } y_i((\mathbf{w},\mathbf{x}_i) + b) \ge 1 - \xi_i$$

$$\xi_i > 0$$

Where $\xi_i, i = 1,...n$ are slack variables, measuring the deviation from the margin borders. The term $(\mathbf{w},\mathbf{w})$ controls the size of margin, and coefficient $C$ controls the trade-off between complexity and proportion of nonseparable samples. (see Figure 2)

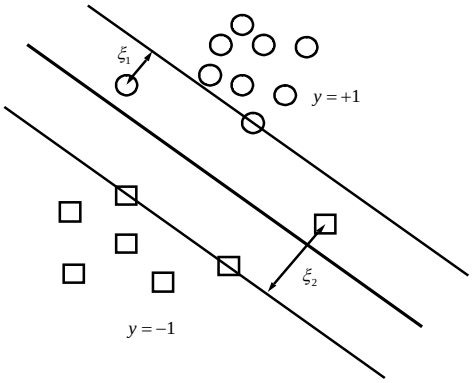


Figure 2. Binary classification for non-separable data involves two goals: (a) Minimizing the total error for data samples unexplained by the model, quantified as a sum of slack variables $\xi_i$ corresponding to deviation from margin borders; (b) Maximizing the size of margin.

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 :

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

*B. SVM+*

Suppose that training data are the union of $t$>1 groups. Let us denote the indices of samples from group r by $T_r = \{i_{n1},...,i_{n_r}\}, r = 1,...,t$ . Then the total training data set is a union of $t$ groups:

$$\{\{\mathbf{X}_r, \mathbf{Y}_r, r = 1,...,t\}, \{\mathbf{X}_r, \mathbf{Y}_r\} = \{\{\mathbf{x}_{r_1}, y_{r_1}\},...,\{\mathbf{x}_{r_{nr}}, y_{r_{nr}}\}\}$$

To account for the group information, Vapnik [9] proposed to define the slacks within each group by so-called 'correcting function'

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

To define the correcting function $\xi_r(\mathbf{x}_i) = \phi_r(\mathbf{x}_i, \mathbf{w}_r)$ for group $T_r$, Vapnik [9] 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 specified as: $\xi_r(\mathbf{x}_i) = (\mathbf{x}_i, \mathbf{w}_r) + d_r, r = \{1,...,t\}$ . Mapping of the training data onto two spaces, decision and correcting space, is shown in Fig. 3, for *t=2* groups.

Compared to standard SVM, in SVM+ 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 are 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.

Correcting functions represent a unique way that SVM+ handles group information, and these correcting functions have the following unique characteristics:

(1) All slack variables are non-negative, so $\xi_r(\mathbf{x}_i) = (\mathbf{x}_i, \mathbf{w}_r) + d_r \geq 0, r = \{1,...,t\}$ . 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* correspond to the size of margin.

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

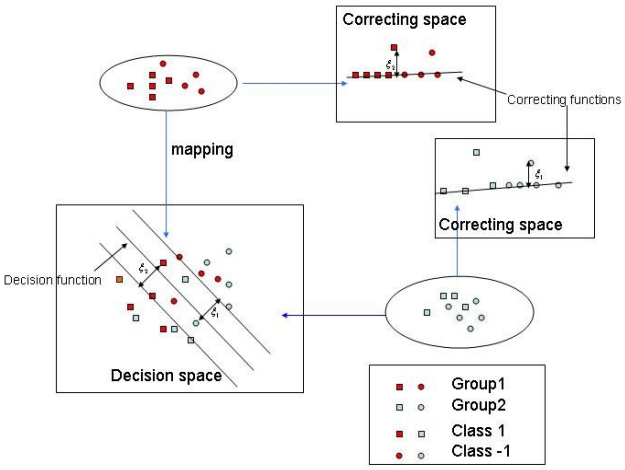


Figure 3: SVM+ maps data simultaneously into decision space and correcting spaces. Decision function is found in decision space. Slack variables are represented by correcting functions which are defined in correcting space.

Estimating SVM+ model from training data requires solving the following optimization problem [9]:

$$\min_{w,w_1,..,w_t,b,d_1,..d_t} \frac{1}{2}(\mathbf{w},\mathbf{w}) + \frac{\gamma}{2}\sum_{r=1}^{t}(\mathbf{w}_r,\mathbf{w}_r) + C\sum_{r=1}^{t}\sum_{i\in T_r}\xi_i^r \qquad \text{(OP2)}$$
subject to:
$y_i((\mathbf{w},\mathbf{z}_i)+b) \geq 1-\xi_i^r, i \in T_r, r = 1,...,t$
$\xi_i^r \geq 0, i \in T_r, r = 1,...,t$
$\xi_i^r = (\mathbf{z}_i^r, \mathbf{w}_r) + d_r, i \in T_r, r = 1,...,t$

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 function. $\gamma$ adjusts the relative weight of these two capacities. $C$ controls the trade-off between complexity and proportion 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 $\alpha_i$ maximize the functional:

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

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

subject to:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\sum_{i \in T_r} (\alpha_i + \mu_i) = |T_r| C, r = 1, ..., t$$

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

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

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

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 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 restricted $\alpha_i$'s .

### C. SVM+MTL

Now we discuss adaptation of SVM+ approach to multi-task learning (MTL). Application of SVM+ to MTL requires (1) specification (parameterization) of decision functions for different groups; (2) modeling the relatedness among the groups (tasks).

In the method called SVM+MTL, similar to SVM+, we map the input vectors $\mathbf{x}_i, i \in T_r$ simultaneously into two different Hilbert spaces: into the decision space $\mathbf{z}_i = \Phi_z(\mathbf{x}_i) \in Z$ and into 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

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

Where 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 (unlike SVM+ that yields a function in the decision space only) .

In SVM+MTL, *t* tasks are related in the sense that decision functions for different tasks share a common decision function. Similar to SVM+, correcting functions of different groups may lie in the same correcting space or different correcting spaces.

SVM+MTL classifier is estimated from training data as a solution to the following optimization problem:

$$\min_{w,b} \frac{1}{2}(\mathbf{w}, \mathbf{w}) + \frac{\gamma}{2} \sum_{r=1}^{t} (\mathbf{w}_r, \mathbf{w}_r) + C \sum_{r=1}^{t} \sum_{i \in T_r} \xi_i^r \qquad \text{(OP3)}$$

subject to:

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

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

Here, the 2-norm of $\mathbf{w}, \mathbf{w}_r$ is used to control the capacity of the common decision function and of the correcting function, respectively. Parameter $\gamma$ adjusts the relative weight of these two capacities, and $C$ controls the trade-off between complexity and proportion of nonseparable samples. The slack variables $\xi_i^r$ measure the error that each of the final models (including common decision function and correcting function) makes on the data.

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:

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

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

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

Based on Karush-Kuhn-Tucker (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,

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

Based on optimization formulations for different learning settings (shown above), we can identify tunable parameters for modeling approach shown in Fig. 1:

- *single SVM* classifier: single parameter *C* (linear SVM is used), and 2 parameters *C*, $\sigma$ (RBF kernel is used);

- *multiple SVM*: $t$ parameters $C$ (linear SVM is used for each task) and $2t$ parameters $C$, $\sigma$ (RBF kernel is used for each task);

- *SVM+* classifier, where linear kernel is used for the decision space, and RBF kernel is used for correcting space, requires 3 parameters: $C$ (as in standard linear SVM), $\gamma$ and $\sigma$ (RBF witdh);

- *SVM+MTL* classifier requires 3 parameters $C$ and (as in standard linear SVM), $\gamma$ and $\sigma$ (RBF with parameter).

Note that the *same kernel parameter* $\sigma$ is used for all correcting functions in SVM+ and SVM+MTL methods. Note that standard linear SVM classifier has just one tunable parameter, whereas SVM+ and SVM+MTL each have 3 parameters. This crude analysis also suggests that relative performance of these methods may be strongly affected by sample size. For small sample size, standard SVM may still be the best method, simply because it has fewer tunable parameters.

Empirical comparisons of these learning methods (presented in Section 3) use double resampling procedure, i.e. one level of resampling for comparing prediction accuracy of learning methods, and the second level for tuning model parameters of each method. At each level, resampling was implemented using 5-fold cross-validation.

## III. EMPIRICAL COMPARISONS

This section describes empirical comparisons of various modeling approaches for classification with heterogeneous data, such as single SVM (sSVM), multiple SVM (mSVM), SVM+ and SVM+MTL. All comparisons use linear and rbf kernels for sSVM and mSVM, The common decision space for SVM+ and SVM+MTL use linear kernel while the unique correction space use RBF(Gaussian) kernel. All comparisons use the following experimental procedure:

(a) Select a group variable (from a list of input variables).

(b) Partition available data into several groups (tasks) corresponding to different values (or range of values) of group variable. Each group should be roughly of similar size.

(c) Within each group, order data samples by increasing value of the group variable.

(d) For estimating prediction error of a particular method, use 5-fold cross-validation, so that 80% of data samples are used for training and 20% of the data are used as test data. Note that conditions (b) and (c) ensure that each fold has approximately equal number of samples from all groups (tasks).

(e) For each training fold, parameter tuning (model selection) for different methods are via resampling within the training fold.

Comparison of methods' performance on several publicly available medical data sets is presented next. Comparison results show average test error (averaged over 5 folds) and its standard deviation.

### *A. Statlog heart disease dataset*

This dataset is from UCI machine learning repository. There are 270 instances, each of which has 13 attributes. The goal is to predict absence or presence of heart disease using 13 input variables. We choose variable 'SEX' to separate the data into male & female groups: group1( $SEX = 0$, 87 instances)and group2( $SEX = 1$, 183 instances). The binary group variable was removed and modeling was performed with remaining 12 attributes. Possible choices of parameters for sSVM and mSVM are $C = [0.1\ 1\ 10\ 100]$ and $\sigma = [0.25\ 0.5\ 1\ 2\ 4]$. Possible choices of parameters for SVM+ and SVM+MTL are $C = [0.1\ 1\ 10\ 100]$, $\gamma = [10\ 1\ 0.1\ 0.01\ 0.001]$ and $\sigma = [0.25\ 0.5\ 1\ 2\ 4]$. Comparison results are shown in Table 1.

### *B. Ljubljana breast cancer dataset*

This dataset is available at UCI machine learning repository. It consists of 286 instances, each with 9 attributes. The dataset contains 9 instances with missing values and the remaining 277 instances are used. The goal is to predict the class (no-recurrence-events or recurrence-events) from 9 attributes. Variable *'age'* was selected to separate the data into 3 different groups: group 1( $age < 47$, 94 instances), group 2( $47 \le age < 55$, 93 instances) and group 3( $age \ge 55$, 90 instances). Since variable 'age' has different values within each group, this variable is still included in modeling. Possible choices of parameters for sSVM and mSVM are $C = [0.1\ 1\ 10\ 100]$ and $\sigma = [0.25\ 0.5\ 1\ 2\ 4]$. Possible choices of parameters for SVM+ and SVM+MTL are $C = [0.1\ 1\ 10\ 100]$, $\gamma = [10\ 1\ 0.1\ 0.01\ 0.001]$ and $\sigma = [0.25\ 0.5\ 1\ 2\ 4]$. Results are shown in Table 2.

### *C. Wisconsin breast cancer dataset*

This is another dataset from UCI machine learning repository. There are 699 instances, each of which has 9 continuous attributes. The measurements of attributes are assigned an integer value between 1 and 10. After removing 16 instances with missing values, we are left with 683 instances for

modeling. The goal is to predict the class(benign or malignant) using 9 input variables. We choose variable 'Clump Thickness' to separate the data into 3 groups: group1( $\text{Clump Thickness} < 4$ , 293 instances), group2( $4 \le \text{Clump Thickness} < 6$ , 207 instances)and group3 ( $\text{Clump Thickness} \ge 6$ , 183 instances). Since variable 'Clump Thickness' has different values within each group, this variable is still included in modeling. Possible choices of parameters for sSVM and mSVM are $C = [0.1\ 1\ 10\ 100]$ and $\sigma = [0.25\ 0.5\ 1\ 2\ 4]$. Possible choices of parameters for SVM+ and SVM+MTL are $C = [0.1\ 1\ 10\ 100]$ , $\gamma = [10\ 1\ 0.1\ 0.01\ 0.001]$ and $\sigma = [0.25\ 0.5\ 1\ 2\ 4]$. Results are shown in Table 3.

### *D. Hepatitis dataset*

This dataset can also be found at UCI machine learning repository. There are 155 instances, each of which has 19 attributes. After removing 75 instances with missing values, we are left with 80 instances for modeling. The goal is to predict the class (*dead / alive*) using 19 input variables. We separate data into 2 groups using binary variable 'HISTOLOGY': group 1( $HISTOLOGY = 1$, 47 instances) and group 2( $HISTOLOGY = 2$, 33 instances). Hence, all methods used only the remaining variables 18 attributes for prediction. Possible choices of parameters for sSVM and mSVM are $C = [0.1\ 1\ 10\ 100]$, $\sigma = [0.25\ 0.5\ 1\ 2\ 4]$. Possible choices of parameters for SVM+ and SVM+MTL are $C = [0.1\ 1\ 10\ 100]$, $\gamma = [10\ 1\ 0.1\ 0.01\ 0.001]$ and $\sigma = [0.25\ 0.5\ 1\ 2\ 4]$. Results are shown in Table 4.

Table 1 Prediction error for statlog heart dataset

| method | sSVM(linear) | sSVM(rbf) | SVM+ |
|---|---|---|---|
| Test error % | 19.3 ± 7.5 | 18.2 ± 6.5 | 16.3 ± 6.1 |
| method | mSVM | mSVM(rbf) | SVM+MTL |
| Test error % | 16.6 ± 4.3 | 21.5 ± 5.3 | 15.2 ± 4.0 |

Table 2 Prediction error for Ljubljana breast cancer dataset

| method | sSVM(linear) | sSVM(rbf) | SVM+ |
|---|---|---|---|
| Test error % | 29.3 ± 6.2 | 25.7 ± 4.5 | 24.9 ± 4.8 |
| method | mSVM(linear) | mSVM(rbf) | SVM+MTL |
| Test error % | 29.6 ± 1.6 | 24.2 ± 2.5 | 23.5 ± 3.4 |

Table 3 Prediction error for Wisconsin breast cancer dataset

| method | sSVM(linear) | sSVM(rbf) | SVM+ |
|---|---|---|---|
| Test error % | 3.4 ± 1.3 | 3.8 ± 0.8 | 3.1 ± 1.0 |
| method | mSVM(linear) | mSVM(rbf) | SVM+MTL |
| Test error % | 3.4 ± 0.8 | 3.1 ± 1.0 | 2.9 ± 0.9 |

Table 4 Prediction error for hepatitis dataset

| method | sSVM(linear) | sSVM(rbf) | SVM+ |
|---|---|---|---|
| Test error % | 16.3 ± 8.4 | 17.5 ± 5.2 | 16.3 ± 8.4 |
| method | mSVM(linear) | mSVM(rbf) | SVM+MTL |
| Test error % | 16.3 ± 8.4 | 16.3 ± 8.4 | 15.0 ± 7.1 |

### *E. Comparison for Regression: Boston Housing Dataset*

Finally, we show comparison between different methods for *regression* problems. The same conceptual approaches, i.e. standard SVM, SVM+ and SVM+MTL can be developed for regression learning problem. We do not provide here detailed mathematical formulations for regression, due to space constraints. SVM+ regression was originally introduced by Vapnik [9] and description of SVM+MTL regression can be found in [13]. Comparisons between SVM regression, SVM+ regression, SVM+MTL regression, and multiple SVM regression are presented next using Boston Housing data set. It has 13 input variables (12 continuous and 1 Boolean) and 506 data samples. The goal is to estimate the median value of owner-occupied homes in $1000's from 13 attributes. We present two sets of comparisons for this dataset, using different group variables: 'RAD' ~ accessibility index to major highways) and 'DIS' ~ weighted distance to major employment centers in Boston area. *First*, variable 'RAD' is selected to separate data into 3 groups: group 1( $RAD < 5$, 192 instances), group 2( $5 \le RAD < 7.5$, 158 instances) and group 3( $RAD \ge 7.5$, 156 instances). *Second*, we separate data into 3 groups by another variable 'DIS': group 1( $DIS < 2.5$, 188 instances), group 2( $2.5 \le DIS < 4.5$, 163 instances) and group 3( $DIS \ge 4.5$, 155 instances). Therefore, all methods, sSVM, mSVM, SVM+ and SVM+MTL, used all 13 attributes for prediction.

Comparisons use different learning methods for regression, i.e. single SVM, multiple SVM, SVM+ and SVM+MTL, that implement different approaches for estimating regression models from heterogeneous training data. Distinction between these approaches is shown in Fig. 1 (where estimated models are real-valued functions). Comparisons use linear SVM regression for single and multiple SVMs, linear SVM for decision function in SVM+ and SVM+MTL, and RBF kernel in the correcting space. So each modeling approach has the following tunable parameters:

- *single SVM* regression: parameters $C$, epsilon and $\sigma$ (RBF width);
- *multiple SVM*: parameters $C$ ,epsilon and $\sigma$ (for each task);
- *SVM+* regression requires 5 parameters: $C$, epsilon and $\sigma_{common}$ (as in standard SVM), $\gamma$ and $\sigma_{correction}$ (RBF width);
- *SVM+MTL* regression requires 5 parameters: $C$, epsilon and $\sigma_{common}$ (as in standard SVM), $\gamma$ and $\sigma_{correction}$ (RBF width);

Parameters C and epsilon for SVM were tuned using analytic approach described in [3], whereas parameters $\gamma$ and $\sigma$ are tuned using resampling. Possible choices of parameters for SVM+ and SVM+MTL regression are:

$\gamma = [10\ 1\ 0.1\ 0.01\ 0.001]$ , $\sigma = [0.25\ 0.5\ 1\ 3\ 4]$ . Results are shown in Table 5 and Table 6. These results show prediction error (MSE) for each fold of 5-fold cross-validation procedure used to estimate test error, along with the mean (MSE) error and its standard deviation. Note that SVM+ regression is better than single SVM, and SVM+MTL is better than multiple SVM regression models. Overall, SVM+MTL achieves improvement, in terms of prediction MSE, over standard SVM regression.

Table 5 Prediction MSE for Boston housing dataset (group variable: RAD)

| Folds | 1 | 2 | 3 | 4 | 5 | Mean(st.dev) |
|---|---|---|---|---|---|---|
| sSVM | 8.9 | 26.1 | 8.5 | 5.9 | 10.9 | 12.1(8.0) |
| mSVM | 12.1 | 27.2 | 10.4 | 6.2 | 15.1 | 14.2(7.9) |
| SVM+ | 8.9 | 23.5 | 9.5 | 6.1 | 8.8 | 11.4(6.9) |
| SVM+MTL | **7.6** | **15.6** | **8.0** | **4.9** | **8.7** | **9.0(4.0)** |

Table 6 Prediction MSE for Boston housing dataset (group variable: DIS)

| Folds | 1 | 2 | 3 | 4 | 5 | Mean(st.dev) |
|---|---|---|---|---|---|---|
| sSVM | 8.9 | 8.3 | 11.1 | 9.0 | 18.4 | 11.1(4.2) |
| mSVM | 10.2 | 8.9 | 10.3 | 11.1 | 20.1 | 12.1(4.5) |
| SVM+ | 8.1 | 8.7 | 10.7 | **7.9** | **16.5** | 10.4**(3.6)** |
| SVM+MTL | **7.1** | **8.2** | **8.6** | 8.4 | 17.0 | **9.9**(4.0) |

## IV. CONCLUSIONS and DISCUSSION

This paper presents and compares different approaches for utilizing group information in learning problems. These include standard inductive SVM, multiple SVMs, SVM+ and SVM+MTL. Empirical comparisons presented using several medical data sets illustrate relative performance of these methods and various trade-offs. Our comparisons show that for the single-model setting, SVM+ is consistently better than standard SVM classifier, and that for multiple-model setting, SVM+MTL is consistently better than several independent SVMs.

Whereas our empirical comparisons suggest the advantages of SVM+MTL, we strongly warn against making such over-reaching conclusions. Relative performance of learning methods is always strongly affected by the properties of application data at hand [6, 13]. New learning settings, such as SVM+ regression and SVM+MTL, are more complex than standard SVM, and have more tuning parameters. So, effective model selection for these new methods is an open research area. Another important practical problem is specification of 'good' group variable(s) that is likely to yield improved generalization. In all examples shown in this paper, selected group variable typically has low correlation with the output (response) *y*. However, more research is needed in proper selection of group variable(s), in both classification and regression problems.

**Acknowledgements:** This work was supported, in part, by NSF grant ECCS-0802056, by the A. Richard Newton Breakthrough Research Award from Microsoft Research, and by the BICB grant from the University of Minnesota, Rochester.

## 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 from 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] Liang, L. and Cherkassky, V. Learning using Structured Data: Application to fMRI Data Analysis, IJCNN, 2007.
[6] Liang, L. and Cherkassky, V. Connection between SVM+ and Multi-Task Learning, IJCNN, 2008.
[7] Vapnik, V. Estimation of Dependencies Based on Empirical Data, Springer Verlag, New York, 1982.
[8] Vapnik, V. Statistical Learning Theory, Wiley, New York, 1998.
[9] Vapnik, V. Empirical Inference Science Afterword of 2006, Springer, 2006.
[10] Evgeniou, T. and Pontil, M.. Regularized multi--task learning. In Proc. 17th SIGKDD Conf. on Knowledge Discovery and Data Mining, 2004.
[11] Camps-Valls, G., Rojo -Alvarez, J. L., and M. Martinez-Ramon, Eds., **Kernel Methods in Bioengineering, Signal and Image Processing**, London: Idea Group Publishing, 2007.
[12] Simon, R., Radmacher, M.D., Dobbin, K. and L. M. McShane, Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification, Journal of the National Cancer Institute 2003, 95(1):14-18.
[13] Cai, F. and Cherkassky, V., SVM+ regression and multi-task learning, IJCNN 2009.