Neural Networks 22 (2009) 766–773

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2009 Special Issue

Predictive learning with structured (grouped) data

Lichen Liang, Feng Cai [*], Vladimir Cherkassky

Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 7 May 2009
Received in revised form 11 June 2009
Accepted 25 June 2009

Keywords:
Heterogeneous data
Learning with structured data
Model selection
Multi-task learning
SVM
SVM-Plus

ABSTRACT

Many applications of machine learning involve sparse and heterogeneous data. For example, estimation of 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 a classifier. This approach, known as standard inductive learning, is used in most application studies. Recently, several new learning methodologies have emerged. For instance, when training data can be naturally separated into several groups (or structured), we can view model estimation for each group as a separate task, leading to a Multi-Task Learning framework. Similarly, a setting where the training data are structured, but the objective is to estimate a single predictive model (for all groups), leads to the Learning with Structured Data and SVM+ methodology recently proposed by Vapnik [(2006). *Empirical inference science afterword of* 2006. Springer]. This paper describes a biomedical application of these new data modeling approaches for modeling heterogeneous data using several medical data sets. The characteristics of group variables are analyzed. Our comparisons demonstrate the advantages and limitations of these new approaches, relative to standard inductive SVM classifiers.

1. 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 (Cherkassky & Mulier, 2007). 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) versus discriminative (∼ predictive) modeling. For multivariate sparse data sets common in biomedical applications, the predictive approach is more practical because

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

(b) it may be possible to estimate accurate predictive models that reflect *certain properties* of unknown distributions (Cherkassky & Mulier, 2007; Vapnik, 1998, 2006). For example, for classification problems, the goal of estimating a decision boundary (for future predictions) does not require accurate estimation of class distributions. Moreover, Statistical Learning Theory (also known as VC theory) (Vapnik, 1998, 2006, 1982) 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*

[*] Corresponding author. Tel.: +1 612 219 3562
E-mail address: caixx043@umn.edu (F. Cai).

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.06.030

sample size (HDLSS) data are 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 thousand 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 data sets, 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 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 (Cherkassky & Mulier, 2007).

The *first approach* is to adopt a *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, also known as 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 a neural network, or logistic regression) is used to estimate a model. This approach has been successfully used in many biomedical and image processing applications (Camps-Valls, Rojo-Alvarez, & Martinez-Ramon, 2007). Commonly used statistical approaches to modeling genetic data for diagnostic and prognostic classification follow feature selection strategy (also known as 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 (Simon, Radmacher, Dobbin, & McShane, 2003).

The *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.

Multi-Task Learning, also known as transfer learning, has had a relatively long history in machine learning. Learning multiple related tasks simultaneously has been empirically (Ando & Zhang, 2005; Bakker & Heskes, 2004; Evgeniou & Pontil, 2004) as well as theoretically (Ando & Zhang, 2005; Ben-David & Schuller, 2003) shown to often significantly improve predictive performance relative to learning each task independently. So MTL approaches can benefit applications using HDLSS heterogeneous data where relatively few data samples per task are available. Most Multi-Task Learning techniques can be broadly grouped into several categories, depending on how task relatedness is modeled:

- methods where multiple tasks share the same internal representation, such as hidden units in neural networks (Ando & Zhang, 2005; Bakker & Heskes, 2004; Caruana, 1997; Liao & Carin, 2005),
- estimating a common set of latent variables consisting of linear combinations of the original input features, as in Partial Least Squares (PLS) statistical approaches (Momma & Bennett, 2006),
- probabilistic methods where task relatedness is modeled by sharing priors (Lawrence & Platt, 2004; Raina, Ng, & Koller, 2006),
- modeling task relatedness via common (shared) features (Argyriou, Evgeniou, & Pontil, 2006; Obozinski, Taskar, & Jordan, 2006),
- kernel methods where different tasks share common part in their decision functions (Evgeniou & Pontil, 2004; Liang & Cherkassky, 2008).

The methods discussed in this paper are most closely related to the last category.

This paper describes 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, in Sections 3 and 4. Finally, conclusions and discussion are presented in Section 5.

2. Approaches for modeling heterogeneous data

In this paper, we consider supervised learning applications where the training data include additional (group) information about training samples. Examples include: (1) handwritten digit recognition where training examples are provided by several persons, (2) medical diagnosis where a 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 (Ando & Zhang, 2005; Ben-David, Gehrke, & Schuller, 2002; Evgeniou & Pontil, 2004; Liang & Cherkassky, 2008) and, more recently, to Learning with Structured Data (also known as SVM+) (Vapnik, 2006), as briefly discussed next.

Suppose that the training data can be represented as a union of t related groups, i.e. each group $r \in [1, 2, \ldots, t]$ contains n_r samples independently and identically generated from a distribution P_r on $\mathbf{x} \times y$. Therefore, the available data are a union of $t > 1$ groups: $\{\{\mathbf{X}_r, \mathbf{Y}_r\}, r = 1, \ldots, t\}, \{\mathbf{X}_r, \mathbf{Y}_r\} = \{\{\mathbf{x}_{r_1}, y_{r_1}\}, \ldots, \{\mathbf{x}_{r_{nr}}, y_{r_{nr}}\}\}$, and it can be thought of as samples identically and independently generated from an unknown distribution $P(\mathbf{x}, y) = \{P_r(\mathbf{x}, y), \mathit{if}\,\{\mathbf{x}, y\} \in \{\mathbf{X}_r, \mathbf{Y}_r\}\}$.

If the group labels of future test samples are not given, the appropriate formulation is known as "**L**earning **W**ith **S**tructured **D**ata (LWSD)" (Vapnik, 2006). In this formulation, the goal is to find the best mapping function f such that the expected loss

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

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 the 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 formalized as **M**ulti-**T**ask **L**earning (MTL) (Ando & Zhang, 2005; Ben-David et al., 2002; Liang & Cherkassky, 2008; Vapnik, 1998). The goal in multi-task learning is to estimate t related classifiers $\{f_1, f_2, \ldots, f_t\}$ so that the sum of expected losses for each task

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

is minimized.

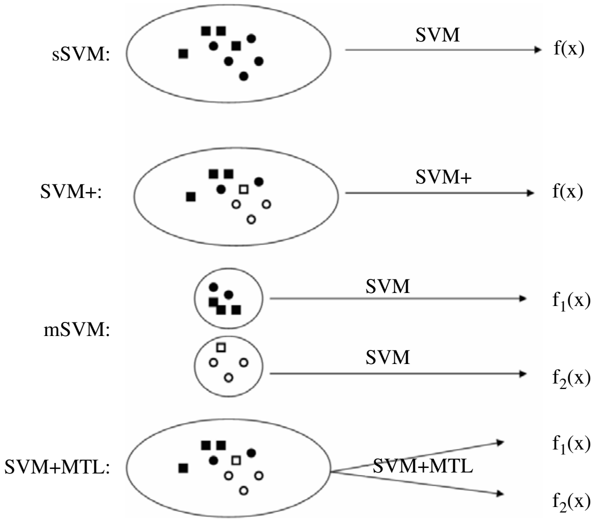

Fig. 1. Different ways of using group information in learning: (a) sSVM ∼ Single SVM classifier, (b) SVM+ classifier, (c) mSVM ∼ multiple independent SVMs, and (d) SVM+MTL ∼ SVM+ Multi-Task Learning (Liang & Cherkassky, 2008).

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, the standard inductive setting does not use group information in the training data; the 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 the test inputs do not have group information, whereas under MTL the test inputs have (known) group labels.

Recently, Vapnik (2006) proposed an SVM-based optimization formulation called SVM+ for LWSD formulation. Liang and Cherkassky (2008, 2007) showed empirical validation of SVM+ for *classification*, and showed its connection to Multi-Task Learning (MTL) classifiers in machine learning (Ando & Zhang, 2005; Ben-David et al., 2002; Evgeniou & Pontil, 2004; Liang & Cherkassky, 2008). "Learning with structured data" formulation (Vapnik, 2006) 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 the examples in Section 4).

Assuming that the 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 the SVM classifier as a basic inductive learning method.

The *Single SVM* inductive model, which estimates the standard SVM classifier by pooling together training samples from different groups (i.e. group information is ignored).

The *multiple SVM* approach, where a separate SVM classifier is estimated for each group (independently).

The *SVM+* approach, where a single classifier model, utilizing available group information, is estimated from all data.

The *SVM+MTL* implementing multi-task learning, which estimates several related classification models following Liang and Cherkassky (2008, 2007).

In Fig. 1, the class labels are represented by circles and squares, and the group membership is denoted by filled (dark) and non-filled symbols.

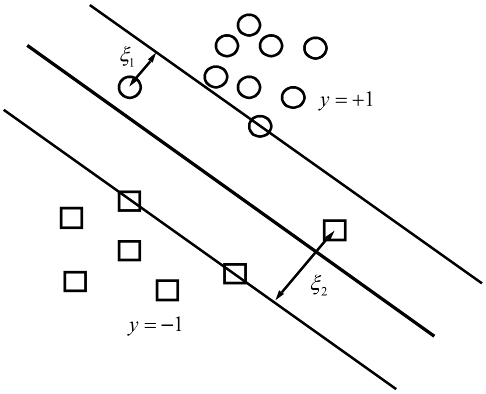

Fig. 2. Binary classification for non-separable data involves two goals: (a) minimizing the total error for data samples unexplained by the model, quantified as a sum of slack variables ξ_i corresponding to deviation from margin borders; and (b) maximizing the size of the margin.

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) SVMs, 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 descriptions of SVM+ and SVM+MTL, see Vapnik (2006), Liang and Cherkassky (2007), Liang (2008) and Liang and Cherkassky (2008), respectively.

2.1. 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 (Vapnik, 1998, 2006). This optimal decision function $f(\mathbf{x}) = (\mathbf{w}, \mathbf{x}) + b$ is estimated from the training data by solving the following optimization problem:

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

$$\text{subject to:} \quad y_i((\mathbf{w},\mathbf{x}_i)+b) \ge 1-\xi_i$$
$$\xi_i > 0$$

where ξ_i, $i = 1, \dots, n$ are slack variables, measuring the deviation from the margin borders. The term $(\mathbf{w}, \mathbf{w})$ controls the size of the margin, and coefficient C controls the trade-off between complexity and proportion of non-separable samples (see Fig. 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

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

$$\text{subject to:} \quad y_i((\mathbf{w},\mathbf{z}_i)+b) \ge 1-\xi_i$$
$$\xi_i > 0.$$

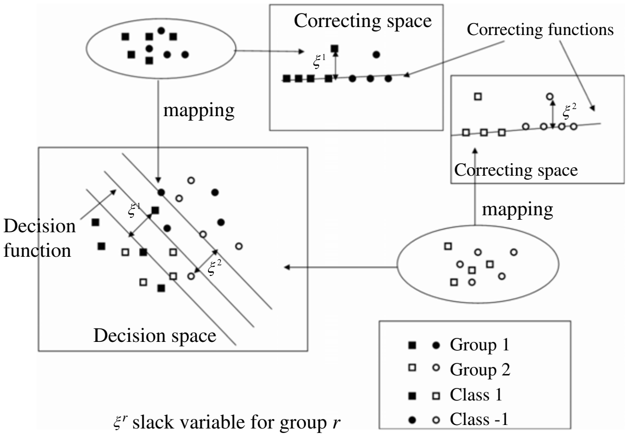

Fig. 3. SVM+ maps data simultaneously into the decision space and the correcting space. The decision function is found in the decision space. Slack variables are represented by correcting functions which are defined in the correcting space.

2.2. SVM+

Suppose that the training data are the union of $t > 1$ groups. Let us denote the indices of samples from group r by $T_r = \{i_{n1}, \dots, i_{n_r}\}, r = 1, \dots, t$. Then the total training data set is a union of t groups: $\{\{\mathbf{X}_r, \mathbf{Y}_r\}, r = 1, \dots, t\}$, $\{\mathbf{X}_r, \mathbf{Y}_r\} = \{\{\mathbf{x}_{r_1}, y_{r_1}\}, \dots, \{\mathbf{x}_{r_{nr}}, y_{r_{nr}}\}\}$. To account for the group information, Vapnik (2006) proposed to define the slacks within each group by the so-called 'correcting function'

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

To define the correcting function $\xi_r(\mathbf{x}_i) = \phi_r(\mathbf{x}_i, \mathbf{w}_r)$ for group T_r, Vapnik (2006) 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 the 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 $\xi_r(\mathbf{x}_i) = (\Phi_{Z_r}(\mathbf{x}_i), \mathbf{w}_r) + d_r, r = \{1, \dots, t\}$. Mapping of the training data onto two spaces, the decision space and the 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 $\xi_r(\mathbf{x}_i) = (\Phi_{Z_r}(\mathbf{x}_i), \mathbf{w}_r) + d_r \geq 0, r = \{1, \dots, t\}$. Therefore mapping samples in the correcting space have to lie on one side of the corresponding correcting function. The correcting function also has to pass through some points with slack variables being zero.

(2) Like the decision function, the 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.

Estimating the SVM+ model from the training data requires solving the following optimization problem (Vapnik, 2006):

$$\min_{w,w_1,\dots,w_t,b,d_1,\dots,d_t} \frac{1}{2}(\mathbf{w},\mathbf{w}) + \frac{\gamma}{2}\sum_{r=1}^{t}(\mathbf{w}_r,\mathbf{w}_r) + C\sum_{r=1}^{t}\sum_{i\in T_r}\xi_i^r \quad \text{(OP2)}$$

subject to constraints:

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

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 the correcting function. γ adjusts the relative weight of these two capacities. C controls the trade-off between the complexity and proportion of non-separable samples. In this problem, the slack variables are represented as $(\mathbf{z}_i^r, \mathbf{w}_r) + d_r$, and they 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:

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

subject to constraints:

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

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 α_i's.

2.3. SVM+MTL (Liang, 2008)

Now we discuss the adaptation of the SVM+ approach to multi-task learning (MTL). The adaptation 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 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

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

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

The SVM+MTL classifier is estimated from the 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:

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

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 γ adjusts the relative weight of these two capacities, and C controls the trade-off between the complexity and proportion of non-separable samples. The slack variables ξ_i^r measure the error that each of the final models (including the common decision function and the 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:

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

Based on Karush–Kuhn–Tucker (KKT) conditions, we can express w, w_r in terms of the 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,

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

In practical applications, the complexity of advanced learning formulations, SVM+ and SVM+MTL, is further reduced by considering only linear decision space and introducing nonlinearity via correcting space, common for all groups/tasks. Based on optimization formulations for different learning settings (shown above), we can identify the tunable parameters for the modeling approach shown in Fig. 1:

– the *single SVM* classifier: the single parameter C (linear SVM is used), and two parameters C, σ (RBF kernel is used);

– the *multiple SVM*: t parameters C (linear SVM is used for each task) and $2t$ parameters C, σ (RBF kernel is used for each task);

– the *SVM+* classifier, where a linear kernel is used for the decision space, and the RBF kernel is used for the correcting space, requires three parameters: C (as in standard linear SVM), γ and σ (RBF width);

– the *SVM+MTL* classifier requires three parameters: C and (as in standard linear SVM) γ and σ (RBF with parameter).

Note that the standard linear SVM classifier has just one tunable parameter, whereas SVM+ and SVM+MTL each have three parameters. This crude analysis also suggests that the relative performance of these methods may be strongly affected by the sample size. For small sample size, standard SVM may still be the best method, simply because it has fewer tunable parameters. This effect of sample size on the relative performance of MTL approaches is illustrated in Section 3, showing empirical comparisons for a synthetic data set.

Empirical comparisons of various learning approaches for handling structured data are presented in Section 4 for several biomedical data sets. These comparisons use a double resampling procedure, i.e. one level of resampling for comparing prediction accuracy of learning methods, and the second level for tuning the model parameters of each method. At each level, resampling was implemented using five-fold cross-validation.

3. Empirical comparisons for synthetic data

This section describes empirical comparisons for various methods using a synthetic data set, generated as follows:

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

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

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

$$\beta_1 = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$\beta_2 = [1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$\beta_3 = [1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0].$$

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

For each task, 100 data samples are generated for training, 100 samples for validation, and 2000 samples for testing. The training data are used for model estimation, the validation data are used for model selection, and the testing data are used for estimating the prediction accuracy. Each experiment is repeated 10 times with different random realizations of (training, validation, test) data.

For the SVM+MTL and SVM+ methods, a linear kernel is used for the decision space, and a Gaussian kernel for the correcting space, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2)$. So these approaches have three tunable parameters, C, γ, σ, that need to be estimated using the validation data set. Possible choices for these parameters are $C = [0.1, 1, 10]$, $\gamma = [0.1, 1, 10]$, and $\sigma = [0.5, 1, 2]$. Table 1 shows the classification accuracy for each trial. Linear SVM with possible choices for $C = [0.1, 1, 10]$ is also used for comparison.

The average classification accuracy (in %) and standard deviation (shown in parentheses) for sSVM, SVM+, mSVM and SVM+MTL are 88.11 (0.65), 88.31 (0.84), 91.18 (1.26) and 91.47 (1.03), respectively. Both SVM+ and SVM+MTL outperform the standard linear SVM classifier. We note that SVM+MTL performs better than SVM+. This is not surprising, because SVM+MTL employs additional information about the group label of test data, which is not used in SVM+. Note that the multiple model methods (mSVM, SVM+MTL)

Table 1
Classification accuracy (%) for synthetic data (100 samples per task).

Trials	1	2	3	4	5	6	7	8	9	10
sSVM	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
mSVM	**92.28**	**91.01**	**92.58**	92.03	90.03	90.71	88.76	90.30	91.51	**92.63**
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 (%) for synthetic data (50 samples per task).

Trials	1	2	3	4	5	6	7	8	9	10
sSVM	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
mSVM	86.21	87.41	84.01	86.20	83.85	83.81	84.55	84.75	83.45	86.67
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 (%) for synthetic data (15 samples per task).

Trials	1	2	3	4	5	6	7	8	9	10
sSVM	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**
mSVM	72.78	75.23	65.85	70.65	70.95	67.90	68.56	72.51	71.65	71.25
SVM+MTL	**78.07**	79.38	79.43	78.82	80.92	73.90	78.58	85.15	77.85	80.25

outperform the single model methods (sSVM, SVM+). A possible reason for this is that, with 100 samples, the training data for each group have sufficient information for accurate model estimation.

To test the effect of the training sample size, in the next experiment the number of training samples is reduced to 50 per group. All other parameters in the experimental procedure remain the same. The comparison results are shown in Table 2. The average prediction accuracy and standard deviation for SVM, SVM+, mSVM and svm+MTL are 85.74 (1.36), 86.49 (1.69), 85.09 (1.40) and 87.39 (2.29). Similar to the previous experiment, both SVM+ and SVM+MTL outperform SVM. Note that mSVM's performance deteriorates dramatically when the training sample size reduces (from 100 to 50 to 15). In Table 3, mSVM's performance is the worst among all methods. The comparison results shown in Tables 1–3 indicate that the relative performance of different learning approaches depends strongly on the properties of the data. That is, the mSVM approach, which showed superior performance in Table 1, becomes inferior to other methods with a smaller sample size. When the sample size decreases more to only 15 per task, the single model methods sSVM and SVM+ improve, while the multiple model methods mSVM and SVM+MTL deteriorate. However, these results also suggest that the single model method SVM+ tends to be not worse than the standard SVM (except for two folds in Table 3), and that the multiple model method SVM+MTL tends to be consistently better than multiple SVM models independently estimated for each group (mSVM). Empirical comparisons presented later in Section 4 on several real-life data sets confirm this observation. Finally, we note that comparison results may be biased against SVM+ which does not use additional information about the group labels of test samples.

4. Empirical comparisons for biomedical data sets

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. The comparisons use both linear and rbf kernels for sSVM and mSVM. The common decision space for SVM+ and SVM+MTL uses a linear kernel while the unique correction space uses an RBF (Gaussian) kernel. All comparisons follow the same experimental procedure:

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

Table 4
Prediction error for Statlog heart data set.

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

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

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

(d) For estimating the prediction error of a particular method, use five-fold cross-validation, so that every fifth sample in each group is used as test data, and the remaining samples constitute training 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, perform parameter tuning (model selection) via resampling within that fold.

Comparison of each method's performance on several publicly available medical data sets is presented next. The comparison results show the average test error (averaged over five folds) and its standard deviation. The model selection procedure tried (exhaustively) the following possible values of tuning parameters:

– for sSVM and mSVM methods, values of $C = [0.1110100]$ and $\sigma = [0.250.5124]$,

– for SVM+ and SVM+MTL methods, values of $C = [0.1110100]$, $\gamma = [1010.10.010.001]$ and $\sigma = [0.250.5124]$.

4.1. Statlog heart disease data set

This data set is from the UCI machine learning repository. There are 270 instances, each of which has 13 attributes. The goal is to predict the absence or presence of heart disease using 13 input variables. We choose variable 'SEX' to separate the data into male and female groups: group1 ($SEX = 0$, 87 instances) and group 2 ($SEX = 1$, 183 instances). The binary group variable was removed and modeling was performed with the remaining 12 attributes. The comparison results are shown in Table 4.

Table 5
Prediction error for Ljubljana breast cancer data set.

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 6
Prediction error for Wisconsin breast cancer data set.

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 7
Prediction error for the hepatitis data set.

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

4.2. Ljubljana breast cancer data set

This data set is available at the UCI machine learning repository. It consists of 286 instances, each with 9 attributes. The data set contains 9 instances with missing values, so the remaining 277 instances are used. The goal is to predict the class (no-recurrence events or recurrence events) from 9 attributes. The variable *'age'* was selected to separate the data into 3 different groups: group 1 ($age < 47$, 94 instances), group 2 ($47 \leq age < 55$, 93 instances) and group 3 ($age \geq 55$, 90 instances). Since the variable '*age*' has different values within each group, this variable is still included in the modeling. Results are shown in Table 5.

4.3. Wisconsin breast cancer data set

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 the variable 'Clump Thickness' to separate the data into 3 groups: group 1 ($\text{Clump Thickness} < 4$, 293 instances), group 2 ($4 \leq \text{Clump Thickness} < 6$, 207 instances)and group 3 ($\text{Clump Thickness} \geq 6$, 183 instances). Since the variable 'Clump Thickness' has different values within each group, this variable is still included in the modeling. Comparison results are shown in Table 6.

4.4. Hepatitis data set

This data set can also be found at the 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 the data into two groups using the binary variable 'HISTOLOGY': group 1 ($\text{HISTOLOGY} = 1$, 47 instances) and group 2 ($\text{HISTOLOGY} = 2$, 33 instances). This binary group variable was then removed, and all comparisons used the remaining 18 attributes for prediction. The results are shown in Table 7.

Table 9
Prediction error with group variable deg-malig.

Method	sSVM (linear)	SVM+	mSVM (linear)	SVM+MTL
Test error %	29.61 ± 6.6	27.79 ± 7.7	24.53 ± 5.7	24.90 ± 4.7

Table 10
Prediction error with group variable inv-nodes.

Method	sSVM (linear)	SVM+	mSVM (linear)	SVM+MTL
Test error %	29.63 ± 4.7	25.62 ± 5.5	26.70 ± 3.3	25.61 ± 4.9

Table 11
Prediction error with group variable *age*.

Method	sSVM (linear)	SVM+	mSVM (linear)	SVM+MTL
Test error %	29.3 ± 6.2	24.9 ± 4.8	29.6 ± 1.6	23.5 ± 3.4

Table 12
Prediction error with group variable breast.

Method	sSVM (linear)	SVM+	mSVM (linear)	SVM+MTL
Test error %	30.33 ± 4.7	27.07 ± 5.3	26.73 ± 5.8	26.03 ± 5.4

The empirical comparisons in Tables 3–6 illustrate the relative performance of the different approaches for predictive modeling of structured (grouped) data. These comparisons indicate that for the single model setting, SVM+ is better (or not worse) than the standard SVM classifier, and that for the multiple model setting, SVM+MTL is consistently better than several independent SVMs. Note that high values of standard deviations in the results are due to the variability of data in different folds.

However, all above modeling results assume some rational (effective) specification of the group variable. In order to understand better the selection of the group variable and its effect on the relative performance of different learning methods, the next set of the selection of different group variables for the Ljubljana breast cancer data set. The goal is to predict the class (non-recurrence or recurrence of breast cancer) from 9 attributes: *age, menopause, tumor-size, inv-nodes, node-caps, degree-of-malignancy, breast, breast-quad, irradiat*. The absolute values of the Pearson correlation coefficients between each attribute variable and the output class variable are shown in Table 8. Large coefficients indicate strong correlation with the output. The experimental results in Tables 9–12 show what happens when the selected group variable has strong correlation and weak correlation with the output. When a group variable is strongly correlated with the output (i.e. *deg-malig, inv-nodes*), the different groups tend to be quite independent. In this case, multiple model methods (mSVM, SVM+MTL) tend to outperform single model methods (sSVM, SVM+), as is evident from Tables 9 and 10. In particular, the simple mSVM approach that estimates several independent SVM models shows quite good performance. The more complex SVM+MTL method does not provide an improvement over mSVM, since there is little information shared among groups.

On the other hand, if a group variable (i.e. *age*) has a small correlation coefficient, there may be considerable overlap among different groups. More information can be shared among groups, and this leads to improved performance of the multi-task approaches SVM+ and SVM+MTL (see Table 11). However, a small correlation coefficient may be also due to irrelevant (noisy) input. For example, the input variable *breast* (with two values, left or right) has

Table 8
Absolute value of Pearson correlation coefficient between attributes and the output.

Attributes	Age	Menopause	Tumor-size	Inv-nodes	Node-caps	Deg-malig	Breast	Breast-quad	Irradiat
Coefficient	0.0675	0.0588	0.1784	0.3027	0.0289	0.2994	0.0586	0.0910	0.1939

a very small correlation coefficient, but it is clearly irrelevant for prediction. So this group variable (breast) cannot divide the data into meaningful groups. Hence advanced multi-task learning approaches are not likely to show much improvement over standard single SVM or mSVM (as shown in Table 12). Comparisons in Tables 9–12 show results only for linear sSVM and linear mSVM because nonlinear sSVM and mSVM don't show any improvement.

5. Conclusions and discussion

This paper presented comparison of different approaches for utilizing group information in learning problems. These include standard inductive SVM, multiple SVMs, SVM+ and SVM+MTL.

Our comparisons suggest the advantages of using advanced modeling approaches such as:

- SVM+ Vs standard SVM classifiers for single-model estimation,
- SVM+MTL for multiple model estimation.

Advantages of these new modeling approaches have been illustrated on several biomedical data sets. However, our comparison results can not be extrapolated to other data set. Relative performance of learning methods is always strongly affected by the properties of the application data at hand (Cai & Cherkassky, 2009; Liang & Cherkassky, 2008). New learning settings, such as SVM+ 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 the specification of 'good' group variable(s) that is likely to yield improved generalization. In most examples shown in this paper, the selected group variable typically had low correlation with the output (response) y, and also partitioned the data into meaningful groups. So partitioning of the training data into groups requires application domain knowledge, or common sense, and cannot be performed using statistical analysis alone. However, more research is needed on the proper selection of group variable(s).

In a broader context, the idea of using group information in the training data, underlying application of SVM+ and SVM+MTL methods is a special case of Vapnik's methodology called Learning Using Hidden Information (LUHI) or Learning With Teacher (Vapnik, 2006). Under LUHI, the training data contains some extra information, besides the usual labeled samples. This additional information can improve generalization of estimated models. The group information (used in this paper) is just one example of such as hidden information under LUHI. Strong theoretical justification for this new learning paradigm is given in (Vapnik, Akshay Vashist, & Pavlovitch, 2009), along with several application studies showing its advantages over standard SVM.

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

Ando, R., & Zhang, T. (2005). A Framework for Learning predictive structures from multiple tasks and unlabeled data *Journal of Machine Learning Research*,.

Argyriou, A., Evgeniou, T., & Pontil, M. (2006). Multi-task feature learning. In *NIPS*.

Bakker, B., & Heskes, T. (2004). Task clustering and gating for Bayesian multitask learning *Journal of Machine Learning Research*,.

Ben-David, S., Gehrke, J., & Schuller, R. (2002). A theoretical framework for learning from a pool of disparate data sources. In *ACM KDD*.

Ben-David, S., & Schuller, R. (2003). Exploiting task relatedness for multiple task learning. In *Proceedings of computational learning theory*.

Cai, F., & Cherkassky, V. (2009). SVM+ regression and multi-task learning. In *IJCNN*.

Camps-Valls, G., Rojo-Alvarez, J. L., & Martinez-Ramon, M. (Eds.). (2007). *Kernel methods in bioengineering, signal and image processing*. London: Idea Group Publishing.

Caruana, R. (1997). Multi-task learning. *Machine Learning*, *28*, 41–75.

Cherkassky, V., & Mulier, F. (2007). *Learning from data* (second edition). New York: John Wiley & Sons.

Evgeniou, T., & Pontil, M. (2004). Regularized multi–task learning. In *Proc. 17th SIGKDD conf. on knowledge discovery and data mining*.

Lawrence, N. D., & Platt, J. C. (2004). Learning to learn with the informative vector machine. In *ICML*.

Liang, L., & Cherkassky, V. (2008). Connection between SVM+ and multi-task learning. In *IJCNN*.

Liang, L., & Cherkassky, V. (2007). Learning using structured data: Application to fMRI data analysis. In *IJCNN*.

Liang, L. (2008). Application and development of new learning methodologies for fMRI data analysis. *Ph.D. Thesis*, Department of ECE, University of Minnesota.

Liao, X. J., & Carin, L. (2005). Radial basis function network for multi-task learning. In *NIPS*.

Momma, M., & Bennett, K. P. (2006). Constructing orthogonal latent features for arbitrary loss. In Nikravesh, Guyon, Gunn, & Zadeh (Eds.), *Feature extraction: Foundations and applications*. Springer.

Obozinski, G., Taskar, B., & Jordan, M. (2006). Multi-task feature selection. In *Technical report*.

Raina, R., Ng, Andrew Y., & Koller, D. (2006). Constructing informative priors using transfer learning. In *ICML*.

Simon, R., Radmacher, M. D., Dobbin, K., & McShane, L. M. (2003). Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *Journal of the National Cancer Institute*, *95*(1), 14–18.

Vapnik, V., Akshay Vashist, A., & Pavlovitch, N. (2009). Learning using hidden information (Learning with teacher). In *Proc. IJCNN*.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Vapnik, V. N. (2006). *Empirical inference science afterword of 2006*. Springer.

Vapnik, V. N. (1982). *Estimation of dependencies based on empirical data*. New York: Springer Verlag.