The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Efficient Test-Time Predictor Learning with Group-Based Budget

Li Wang
Department of Mathematics
University of Texas at Arlington
li.wang@uta.edu

Dajiang Zhu
Department of Computer Science
and Engineering
University of Texas at Arlington
dajiang.zhu@uta.edu

Yujie Chi
Physics Department
University of Texas at Arlington
yujie.chi@uta.edu

Abstract

Learning a test-time efficient predictor is becoming important for many real-world applications for which accessing the necessary features of a test data is costly. In this paper, we propose a novel approach to learn a linear predictor by introducing binary indicator variables for selecting feature groups and imposing an explicit budget constraint to up-bound the total cost of selected groups. We solve the convex relaxation of the resulting problem, with the optimal solution proved to be integers for most of the elements at the optima and independent of the specific forms of loss functions used. We propose a general and efficient algorithm to solve the relaxation problem by leveraging the existing SVM solvers with various loss functions. For certain loss functions, the proposed algorithm can further take the advantage of SVM solver in the primal to tackle large-scale and high-dimensional data. Experiments on various datasets demonstrate the effectiveness and efficiency of the proposed method by comparing with various baselines.

Machine learning algorithms have been widely used in many real-world applications, ranging from web-search engines (Chapelle and Chang 2011), medical diagnosis (Kononenko 2001) and computer vision (Vedaldi et al. 2009). A learning algorithm is able to achieve a good model in the training phase by leveraging all the admissible time and computational resources, but it can be costly to predict a new example whose features are difficult to access. In this paper, we are interested in the setting where the test-time cost consists of the time to extract features for learning models of various tasks (e.g., classification and regression), the extraction costs are associated to groups of features, and they vary highly across groups.

The test-time efficient prediction problem is becoming important in various domains (Reyzin 2011; Xu et al. 2013; Hu et al. 2016). In the Yahoo! Learning to Rank Challenge (Chapelle and Chang 2011), each query-document pair consists of 519 features. In the extraction phase, each feature has an acquisition cost as a discrete value in the set $[1, 5, 20, 50, 100, 150, 200]$. The cheapest features with cost 1 are those that can be acquired by looking up a table (such as the statistics of a given document), while the most expensive ones typically involve term proximity scoring (Xu, Weinberger, and Chapelle 2012). In medical diagnosis, a number of tests are usually requested to be performed and the test results are used by doctors to give an accurate prediction. The costs of tests can greatly vary in terms of money and valuable time. Thus, it is important to be able to diagnose as well as possible without performing unnecessary tests (Reyzin 2011). It is well known that one test usually generates a group of features for the purpose of diagnosis, which is associated to a single cost to perform the test. In computer vision, the groups of features are commonly extracted from a single image by taking a number of visual descriptors so as to describe elementary characteristics of images such as the distribution of edges, dense and sparse visual words (Vedaldi et al. 2009). Accessing these characteristics has varying cost, and each one of them is generally represented by a set of features (or group). The setting with a uniform cost on each group of features is widely studied in multiple kernel learning (MKL) (Sonnenburg et al. 2006).

To solve the test-time efficient prediction problems, we propose to learn a linear predictor by imposing a budget constraint over the costs of selected features. By introducing an indicator variable for each group of features, the total cost for test-time prediction can be easily modeled and the groups involved in the prediction can be selected simultaneously. In addition, the cost based on single feature is the special case of the cost for groups if each group only contains one feature. Instead of solving the integer programming, we solve a convex relaxation of the proposed problem. Experiments on various datasets demonstrate the effectiveness and efficiency of the proposed method by comparing with baselines. The contributions of this paper are as follows:

- We propose a novel formulation based on SVMs to learn a test-time efficient predictor with an upper bound on the total cost used to access groups of features in the test sample. The formulation directly handles groups of features, which is naturally applied to the costs of single features.
- We theoretically prove that the proposed relaxed problem has a solution pattern where most of variables are integers at optima. This makes the budget constraint close to the original constraint defined on the integer variables.
- The solution pattern of the indicator variables is independent of loss functions, so the same pattern is applied to different learning problems such as classification and regression. Our theoretical results hold for general losses.

- The resulting optimization problem is convex. We propose a general and efficient algorithm to solve it by leveraging the existing SVM solvers with various loss functions. For certain loss functions, the proposed algorithm can further take the advantage of SVM solver in the primal to tackle large-scale and high-dimensional data.

Related Work

Research on the problem of feature-efficient prediction has proceeded with various strategies. Greiner et al. (2002) considered the problem of feature efficient prediction, where a classifier must choose features to examine before predicting. Cesa-Bianchi et al. (2011) studied how to efficiently learn a linear predictor in the setting where the learner can access only a few features per example. In the similar setting, Schwing et al. (2011) trained a random forest to adaptively select experts at test-time via a tradeoff parameter. Pelossof et al. (2010) analyzed how to speed up margin-based learning algorithm by stopping criterion when the outcome is close to certain. Sun and Zhou (2013) considered how to order base learner evaluations so as to save prediction time.

The test-time budget for learning a predictor model is taken into account in this paper where costs are associated to features. Reyzin (2011) showed that sampling from a weight distribution of an ensemble yields a budgeted learner with similar properties to the original ensemble in boosting. He et al. (2012) trained an MDP for this task by jointly optimizing feature costs and errors. Xu et al. (2012) tackled a related feature efficient regression problem by training CART decision trees with feature costs incorporated as part of the impurity function. In a principled fashion, Xu et al. (2013) proposed cost-sensitive tree of classifiers by constructing a tree of classifiers to reduce the average test-time complexity and maximize the performance, which was solved more efficiently by using approximate submodularity (Kusner et al. 2014). Wang et al. (2015) model the classification system as a directed acyclic graph for reducing test-time acquisition costs. Grubb and Bagnell (2012) presented to sequentially choose weak learner and vote weights so as to greedily minimize a loss function per unit cost until a budget runs out. Huang et al. (2015) improved on Reyzin's approach by considering the feature budget during training, and the proposed greedy method was actually optimize the similar objective function as the work (Grubb and Bagnell 2012). Nan et al. (2016) proposed to prune a random forest for resource-constrained prediction. The features in groups that have costs have also been studied by Hu et al. (2016), where orthogonal matching pursuit and forward regression were extended to solve the group setting with costs. Different from the above methods, we learn an optimal linear predictor in the training stage to satisfy each budget constraint instead of using ensemble predictors. Moreover, our model can naturally represent the group-based budgets and is computationally much more efficient on large-scale and high-dimensional data. In addition, our algorithm converges theoretically to an ϵ-optimal solution with a fast convergence.

Our model for learning a linear predictor by imposing a test-time budget is closely related to feature selection using multiple kernel learning (MKL) (Sonnenburg et al. 2006; Rakotomamonjy et al. 2008). MKL has also been successfully used for feature selection (Xu et al. 2009), where a real variable in $[0, 1]$ is introduced to represent the selection of each feature and the total number of selected feature is bounded by a budget. Similar technique also was used in (Do, Kalousis, and Hilario 2009), but derived by optimizing the error bound in controlling both the margin and the radius of SVM. Feature generating machine (FGM) (Tan, Tsang, and Wang 2014) proposed to solve the similar binary classification problem by the cutting plane method (Kelley 1960). This budget constraint was also applied to optimize multivariate performance measurements (Mao and Tsang 2013). However, all above methods put the budget on the total number of selected features, so they ignore the varied costs associated to either features or groups of features. Our proposed method directly models these costs and put the budget on the total cost of the predictor. As a result, FGM is a special case of our model, i.e., these costs are all ones. Moreover, our method is natural to work for general loss functions.

Learning with a Test-time Budget

We first formulate the learning problem to learn a linear predictor with a test-time budget. After the theoretical analysis, we then propose a general optimization algorithm for various loss functions with guaranteed convergence.

Problem formulation with general loss function

Our training dataset consists of N input vectors $\{\mathbf{x}_i\}_{i=1}^N$ with corresponding labels $\{y_i\}_{i=1}^N$ drawn from an unknown distribution where $\mathbf{x}_i \in \mathbb{R}^D$ and either $y_i \in \{-1, 1\}$ for binary classification or $y_i \in \mathbb{R}$ for regression, $\forall i$, respectively. We assume that features are extracted from raw data in groups and the mth group of features has an acquisition cost $c_m > 0, \forall m$. Let $\mathbf{x}_i = [\phi_1(\mathbf{x}_i); \phi_2(\mathbf{x}_i); \ldots; \phi_M(\mathbf{x}_i)]$ where $\phi_m(\mathbf{x}_i) \in \mathbb{R}^{g_m}$ is a subset of the ith input vector corresponding to the mth group. Further, we assume that groups do not allow to share features, e.g., $\sum_{m=1}^M g_m = D$.

A budget B is defined as the upper bound of the total cost used for test-time prediction, By introducing indicator variable $d_m \in \{0, 1\}$ for the mth group, we can readily define the total cost of the selected groups as $\sum_{m=1}^M c_m d_m$ where $d_m = 1$ means the mth group is selected. The budget constrain is then formulated as $\sum_{m=1}^M c_m d_m \leq B$. Since the constraint is defined over integer variables, it is challenging to optimize for general objectives. Here, we seek to relax every integer constraint to a linear constraint $d_m \in [0, 1]$ and propose the following optimization problem

$$\min_{\mathbf{d}\in\mathcal{D}', \mathbf{w}, b, \mathbf{f}} \frac{1}{2}\sum_{m=1}^M \frac{||\mathbf{w}_m||^2}{d_m} + C\sum_{i=1}^N \ell(f_i, y_i) \quad (1)$$

$$\text{s.t. } f_i = \sum_{m=1}^M \langle \mathbf{w}_m, \phi_m(x_i)\rangle + b, \forall i,$$

$$\sum_{m=1}^M c_m d_m \leq B, 0 \leq d_m \leq 1, \forall m,$$

where $\mathbf{w} = [\mathbf{w}_1^T; \ldots, \mathbf{w}_M^T]$ is the coefficient vector of the linear predictor, $\mathbf{f} = [f_1, \ldots, f_N]$ with f_i as the decision

value of $\mathbf{x}_i$, and ℓ is a generic loss function defined on f_i and y_i. In the next section, we theoretically prove that the optimal solution of problem (1) keeps most d_ms as integer values at optima. Further, for $d_m = 0$, $\mathbf{w}_m = \mathbf{0}$ prevents the objective from infinity by letting $\frac{||\mathbf{w}_m||^2}{d_m} = 0$, but d_m cannot be zeros for all groups, otherwise $f_i = b, \forall i$. In other words, $d_m = 0$ means that the mth feature group is not used to calculate the decision value f. Moreover, minimizing $\frac{||\mathbf{w}_m||^2}{d_m}$ differs from minimizing $\sum_m d_m ||\mathbf{w}_m||^2$ where $d_m = 0$ always holds at the optima. Hence, problem (1) can automatically select a number of groups such that the sum of costs of these selected groups is less than B.

Theoretical analysis on optimality condition

Let $\mathbf{d} = [d_1, \ldots, d_m]$. To obtain the dual problem of (1), we introduce dual variables $\boldsymbol{\alpha}, \lambda \geq 0, \boldsymbol{\eta} \geq 0, \boldsymbol{\gamma} \geq 0$. The dual problem based on the Lagrangian $L(\mathbf{w}, \mathbf{d}, b, \boldsymbol{\alpha}, \lambda, \boldsymbol{\eta}, \boldsymbol{\gamma})$ of (1) is represented by a minimax problem as

$$\begin{aligned}\max_{\boldsymbol{\alpha},\lambda,\boldsymbol{\eta},\boldsymbol{\gamma}} \min_{\mathbf{w},\mathbf{d},b} \Bigg\{ & \frac{1}{2}\sum_{m=1}^{M} \frac{||\mathbf{w}_m||^2}{d_m} - \sum_{i=1}^{N} \alpha_i \Big(\sum_{m=1}^{M} \langle \mathbf{w}_m, \phi_m(x_i) \rangle + b \Big) \\ & + \lambda \Big(\sum_{m=1}^{M} c_m d_m - B \Big) - \boldsymbol{\eta}^T \mathbf{d} + \boldsymbol{\gamma}^T (\mathbf{d} - \mathbf{1}) \Bigg\} \\ & + \min_{\mathbf{f}} \left\{ C \sum_{i=1}^{N} \ell(f_i, y_i) + \sum_{i=1}^{N} \alpha_i f_i \right\}\end{aligned}$$

To simplify the above problem, the derivatives of the Lagrangian w.r.t. the primal variables, $\mathbf{d}, \mathbf{w}, b$, have to vanish, which leads to KKT conditions

$$\mathbf{w}_m = d_m \sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i), \forall m \tag{2}$$

$$\sum_{i=1}^{N} \alpha_i = 0, \tag{3}$$

$$-\frac{1}{2}\frac{||\mathbf{w}_m||^2}{d_m^2} + \lambda c_m - \eta_m + \gamma_m = 0, \forall m \tag{4}$$

$$\lambda \left(\sum_{m=1}^{M} c_m d_m - B \right) = 0, \tag{5}$$

$$\eta_m d_m = 0, \forall m \tag{6}$$

$$\gamma_m (d_m - 1) = 0, \forall m \tag{7}$$

$$\lambda \geq 0, \boldsymbol{\eta} \geq 0, \boldsymbol{\gamma} \geq 0. \tag{8}$$

In terms of (2) and (4), we have

$$-\frac{1}{2}||\sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i)||^2 + \lambda c_m - \eta_m + \gamma_m = 0, \forall m. \tag{9}$$

According to (2), (3), (8) and (9), we obtain dual problem

$$\max_{\boldsymbol{\alpha},\lambda \geq 0, \boldsymbol{\gamma} \geq 0} -\lambda B - \boldsymbol{\gamma}^T \mathbf{1} + \min_{\mathbf{f}} \left\{ C \sum_{i=1}^{N} \ell(f_i, y_i) + \sum_{i=1}^{N} \alpha_i f_i \right\} \tag{10}$$

$$\text{s.t. } \frac{1}{2}||\sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i)||^2 \leq \lambda c_m + \gamma_m, \forall m, \sum_{i=1}^{N} \alpha_i = 0.$$

As we see, the property of optimal values $\mathbf{d}$ is independent of the specific form of loss functions, which is illustrated in the following proposition.

Proposition 1. *An optimal solution of* $\mathbf{d}$ *in (10) is* $\forall m$,

$$d_m = \begin{cases} 0, & \eta_m > 0 \text{ or } \eta_m = \gamma_m = 0 \text{ and } \lambda = 0 \\ 1, & \gamma_m > 0 \\ \frac{B - \sum_{m \notin \mathcal{M}} c_m d_m}{|\mathcal{M}| c_m}, & \eta_m = \gamma_m = 0 \text{ and } \lambda > 0 \end{cases}$$

where $\mathcal{M} = \{m | \frac{1}{2c_m} ||\sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i)||^2 = \lambda > 0, \eta_m = \gamma_m = 0, \forall m\}$. *All the values* d_m *can be either* 0 *or* 1 *except those with* $m \in \mathcal{M}$.

Remark 1. In Proposition 1, any value d_m in $(0, 1)$ will satisfy the condition $\frac{1}{2c_m} ||\sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i)||^2 = \lambda > 0, \forall m \in \mathcal{M}$. Suppose that for $m, m' \in \mathcal{M}$, the given data must have the following equality:

$$\frac{c_m}{c'_m} = \frac{||\sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i)||^2}{||\sum_{i=1}^{N} \alpha_i \phi'_m(\mathbf{x}_i)||^2} = \frac{\boldsymbol{\alpha}^T K_m \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T K'_m \boldsymbol{\alpha}}, \tag{11}$$

where K_m is the gram matrix over $\phi_m(\mathbf{x}_i), \forall i$. In general, equality (11) holds for a small number of features which is observed in Table 2.

Remark 2. According to Proposition 1, the solution pattern of $\mathbf{d}$ is not directly related to the specific form of the loss function ℓ defined in the model. In other words, the same solution pattern of $\mathbf{d}$ is inherited by the model for various learning problems based on different loss functions.

Reformulation for efficient optimization

We propose to solve problem (10) with a general loss function. It is well-known that solving problem (10) is challenging due to the quadratic constraints. Thus, we resort to an equivalent reformulation by fixing $\mathbf{d}$ and obtain the duality problem with respect to $\mathbf{w}$ and b by following the above similar derivations.

Let $S(\boldsymbol{\alpha}) = \min_{\mathbf{f}} C \sum_{i=1}^{N} \ell(f_i, y_i) + \sum_{i=1}^{N} \alpha_i f_i$. As a result, problem (10) can be equivalently reformulated as a semi-infinite linear programming problem given by

$$\min_{\mathbf{d},\theta} \theta \ : \ \text{s.t.} \sum_{m=1}^{M} c_m d_m \leq B, 0 \leq d_m \leq 1, \forall m \tag{12}$$

$$S(\boldsymbol{\alpha}) - \frac{1}{2}\sum_{m=1}^{M} d_m ||\sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i)||^2 \leq \theta, \forall \boldsymbol{\alpha} : \mathbf{e}^T \boldsymbol{\alpha} = 0$$

where $\mathbf{e}$ is the vector with all ones. This can be solved by the exchange method (Hettich and Kortanek 1993). Let a set $\mathcal{K} = \emptyset$. Specifically, at the kth iteration, we solve a subproblem

$$\max_{\boldsymbol{\alpha}} S(\boldsymbol{\alpha}) - \frac{1}{2}\sum_{m=1}^{M} d_m ||\sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i)||^2 : \text{s.t. } \mathbf{e}^T \boldsymbol{\alpha} = 0. \tag{13}$$

Let $\boldsymbol{\alpha}^k$ be the optimal solution of (13) and $h(\boldsymbol{\alpha}, \mathbf{d}) = S(\boldsymbol{\alpha}) - \frac{1}{2}\sum_{m=1}^{M} d_m ||\sum_{i=1}^{N} \alpha_i \phi_m(\mathbf{x}_i)||^2$. If $h(\boldsymbol{\alpha}^k, \mathbf{d}) > \theta$, we add the $\boldsymbol{\alpha}^k$ to the set $\mathcal{K}$, otherwise the algorithm converges. We then solve the linear programming

$$\begin{aligned}\min_{\mathbf{d},\theta} \ \theta \ : \ \text{s.t.} & \sum_{m=1}^{M} c_m d_m \leq B, 0 \leq d_m \leq 1, \forall m \\ & h(\boldsymbol{\alpha}^k, \mathbf{d}) \leq \theta, \forall \boldsymbol{\alpha}^k \in \mathcal{K}.\end{aligned} \tag{14}$$

According to Proposition 1, we know that many d_m are zeros at the optima when B is small. Therefore, $\mathbf{d}$ can be very sparse. This is useful to tackle high-dimensional data.

Loss functions for classification and regression

We consider loss functions for two different learning problems, i.e., classification and regression. By specifying a loss function ℓ, the key step is to solve the problem

$$\min_{\mathbf{f}} C\sum_{i=1}^{N}\ell(f_i,y_i)+\sum_{i=1}^{N}\alpha_i f_i. \tag{15}$$

Next, we take two specific loss functions as examples, and discuss other losses briefly.

Binary classification. The squared hinge loss $\ell_2(f_i,y_i)=\max(0,1-y_if_i)^2$ is widely used for classification. By introducing slack variables $\xi_i=\max(0,1-y_if_i)$, the Lagrangian function of (15) is written as $L_2(\mathbf{f},\boldsymbol{\xi})=C\sum_{i=1}^{N}\xi_i^2+\sum_{i=1}^{N}\alpha_i f_i+\sum_{i=1}^{N}\beta_i(1-y_if_i-\xi_i)-\sum_{i=1}^{N}\tau_i\xi_i$ where $\boldsymbol{\beta}\geq 0$ and $\boldsymbol{\tau}\geq 0$ are multipliers. We have the KKT conditions: $\partial_{\xi_i}L_2=2C\xi_i-\beta_i-\tau_i=0,\forall i$, $\partial_{f_i}L_2=\alpha_i-\beta_iy_i=0,\forall i$, $\beta_i\geq 0,\tau_i\geq 0,\forall i$. By substituting these conditions into (15), we obtain

$$S(\boldsymbol{\alpha})=\max_{\tau_i\geq 0}\sum_{i=1}^{N}\beta_i-\frac{1}{4C}\sum_{i=1}^{N}(\beta_i+\tau_i)^2:\text{s.t.}\boldsymbol{\beta}\geq 0,\alpha_i=\beta_iy_i\forall i.$$

Since $\beta_i\geq 0$ and $\tau_i\geq 0$, the optimal values of $\boldsymbol{\tau}$ should be zeros, i.e., $\tau_i=0,\forall i$. To solve (13), we prefer to solve $\boldsymbol{\beta}$ instead of $\boldsymbol{\alpha}$ as

$$\max_{\boldsymbol{\beta}}\sum_{i=1}^{N}\beta_i-\frac{1}{4C}\sum_{i=1}^{N}\beta_i^2-\frac{1}{2}\sum_{m=1}^{M}d_m||\sum_{i=1}^{N}\beta_iy_i\phi_m(\mathbf{x}_i)||^2 \tag{16}$$
$$\text{s.t. }\sum_{i=1}^{N}\beta_iy_i=0,\boldsymbol{\beta}\geq 0$$

which is equivalent to the dual problem of L2-regularized L2-loss SVC (Fan et al. 2008), and its primal problem is

$$\min_{\mathbf{w}}\frac{1}{2}||\mathbf{w}||^2+C\sum_{i=1}^{N}\max(0,1-y_i(\mathbf{w}^T\widehat{\mathbf{x}}_i+b))^2, \tag{17}$$

where $\widehat{\mathbf{x}}_i=[\sqrt{d_1}\phi_1(\mathbf{x}_i);\ldots;\sqrt{d_M}\phi_M(\mathbf{x}_i)]$. According to the above KKT conditions, we know that $\beta_i=2C\xi_i$. Hence, we can recover dual variables according to primal variables.

Regression. We consider the squared ϵ-insensitive loss function, $\ell_\epsilon(f_i)=\max(0,|y_i-f_i|-\epsilon)^2$ for regression problems. Let ξ and ξ^* be the slack variables to form the constraints $f_i-y_i\leq\epsilon+\xi_i,\xi_i\geq 0,\forall i, y_i-f_i\leq\epsilon+\xi_i^*,\xi_i^*\geq 0,\forall i$. The Lagrangian function of (15) using ℓ_ϵ can be written as $L_\epsilon(\mathbf{f},\boldsymbol{\xi},\boldsymbol{\xi}^*,\boldsymbol{\beta},\boldsymbol{\beta}^*,\boldsymbol{\tau},\boldsymbol{\tau}^*)=C\sum_{i=1}^{N}(\xi_i^2+(\xi_i^*)^2)+\sum_{i=1}^{N}\alpha_if_i+\sum_{i=1}^{N}\beta_i(f_i-y_i-\xi_i-\epsilon)-\sum_{i=1}^{N}\tau_i\xi_i+\sum_{i=1}^{N}\beta_i^*(y_i-f_i-\xi_i^*-\epsilon)-\sum_{i=1}^{N}\tau_i^*\xi_i^*$ where multipliers are $\boldsymbol{\beta}\geq 0,\boldsymbol{\beta}^*\geq 0,\boldsymbol{\tau}\geq 0,\boldsymbol{\tau}^*\geq 0$. We have the following KKT conditions: $\partial_{\xi_i}L_\epsilon=2C\xi_i-\beta_i-\tau_i=0,\forall i$, $\partial_{\xi_i^*}L_\epsilon=2C\xi_i^*-\beta_i^*-\tau_i^*=0,\forall i$, $\partial_{f_i}L_\epsilon=\alpha_i+\beta_i-\beta_i^*=0,\forall i$, $\beta_i\geq 0,\tau_i\geq 0,\forall i$. By substituting the above conditions into (15), we obtain the objective function given by

$$S(\boldsymbol{\alpha})=\max_{\boldsymbol{\tau}\geq 0,\boldsymbol{\tau}^*\geq 0}\sum_{i=1}^{N}(\beta_i^*-\beta_i)y_i-\epsilon\sum_{i=1}^{N}(\beta_i^*+\beta_i)$$
$$-\frac{1}{4C}\sum_{i=1}^{N}\left[(\beta_i+\tau_i)^2+(\beta_i^*+\tau_i^*)^2\right]$$
$$\text{s.t. }\boldsymbol{\beta}\geq 0,\boldsymbol{\beta}^*\geq 0,\alpha_i=\beta_i^*-\beta_i,\forall i.$$

Algorithm 1 Learning with test-time budget (LTB)

1: **Input:** Data $\mathbf{X}$, label $\mathbf{y}$, cost $\mathbf{c}$, budget B, groups $\{\phi_m\}_{m=1}^M$.
2: Initial $\mathbf{d}$, $\theta=-\infty$, $\mathcal{K}=\emptyset$
3: **repeat**
4: Obtain $\boldsymbol{\alpha}^k$ by solving SVM problem in the form of either primal or dual with the input $\widehat{\mathbf{x}}_i=\left[\sqrt{d_1^k}\phi_1(\mathbf{x}_i);\ldots;\sqrt{d_M^k}\phi_M(\mathbf{x}_i)\right]$
5: **if** $h(\boldsymbol{\alpha}^k,\mathbf{d}^k)>\theta^k$ **then**
6: $\mathcal{K}=\mathcal{K}\cup\{\boldsymbol{\alpha}^k\}$
7: **else**
8: Converge and exit
9: **end if**
10: Obtain $\mathbf{d}^{k+1}$ and θ^{k+1} by solving problem (14) with $\mathcal{K}$
11: **until** Convergence

Since multipliers are all non-negative, we have the optimal solution $\boldsymbol{\tau}=0$ and $\boldsymbol{\tau}^*=0$. As a result, we have the dual variables $\beta_i=2C\xi_i$ and $\beta_i^*=2C\xi_i^*,\forall i$. To solve (13), we prefer to solve $\boldsymbol{\beta}$ instead of $\boldsymbol{\alpha}$ as

$$\max_{\boldsymbol{\beta}}\mathbf{y}^T(\boldsymbol{\beta}^*-\boldsymbol{\beta})-\epsilon\mathbf{e}^T(\boldsymbol{\beta}^*+\boldsymbol{\beta})-\frac{1}{4C}(||\boldsymbol{\beta}^*||^2+||\boldsymbol{\beta}||^2)$$
$$-\frac{1}{2}\sum_{m=1}^{M}d_m||\sum_{i=1}^{N}(\beta_i^*-\beta_i)\phi_m(\mathbf{x}_i)||^2 \tag{18}$$
$$\text{s.t. }\mathbf{e}^T(\boldsymbol{\beta}^*-\boldsymbol{\beta})=0,\boldsymbol{\beta}\geq 0,\boldsymbol{\beta}^*\geq 0$$

which is equivalent to the dual problem of L2-regularized L2-loss SVR (Fan et al. 2008), and its primal problem is

$$\min_{\mathbf{w}}\frac{1}{2}||\mathbf{w}||^2+C\sum_{i=1}^{N}\max(0,|y_i-\mathbf{w}^T\widehat{\mathbf{x}}_i-b|-\epsilon)^2 \tag{19}$$

where $\widehat{\mathbf{x}}_i=[\sqrt{d_1}\phi_1(\mathbf{x}_i);\ldots;\sqrt{d_M}\phi_M(\mathbf{x}_i)]$. According to the above KKT conditions, we know that $\alpha_i=\beta_i^*-\beta_i=2C(\xi_i-\xi_i^*)$ where $\xi_i=\max(0,f_i-y_i-\epsilon)$ and $\xi_i^*=\max(0,y_i-f_i-\epsilon)$. Hence, dual variables are able to be recovered from primal solutions.

Discussion on generic loss functions According to the derivations of our proposed model, the differentiation of the given loss function ℓ with respect to f_i is not required. Thus, our derivation is more general than the derivations in MKL (Sonnenburg et al. 2006) where differentiable loss functions are demanded for the existence of inverse function with respect to the decision value f_i. As a result, non-differential loss functions such as hinge loss, ϵ-insensitive loss (Scholkopf and Smola 2001) and structured hinge loss (Tsochantaridis et al. 2005) can be applied with related derivations to get $S(\boldsymbol{\alpha})$ by solving (15). With a simple replacement of the loss function in (15) by a specified loss function, our method can be extended for various learning problems. This is also the key difference between our proposed method and FGM, where FGM only works for binary classification problems (Tan, Tsang, and Wang 2014).

Convergence and complexity analysis

Our proposed algorithm for learning with test-time budget is summarized in Algorithm 1, which solves SVM problems

and linear programming iteratively until convergence. For convenience, we name the proposed algorithm as LTB. To obtain $\boldsymbol{\alpha}$, the dual problem of SVMs can be solved by taking the relationship between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. For the loss functions discussed above, we can solve the primal problems in a more efficient way to obtain $\mathbf{w}$ and b, and then recover $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ from these dual variables.

Let $(\mathbf{d}^k, \theta^k)$ be the optimal solution of problem (14), and $(\mathbf{d}^*, \theta^*)$ be the optimal solution of problem (12). Let $\mathcal{A} = \{\boldsymbol{\alpha} | \sum_{i=1}^{N} \alpha_i = 0\}$, and $\mathcal{K}^k$ is the set $\mathcal{K}$ in kth iteration of Algorithm 1. Let $v(\mathbf{d}) := \sup_{\boldsymbol{\alpha} \in \mathcal{A}} h(\boldsymbol{\alpha}, \mathbf{d})$ be the value function. Let $\mathcal{D} = \{\mathbf{d} \in \mathbb{R}^M | \sum_{m=1}^{M} c_m d_m \leq B, 0 \leq d_m \leq 1, \forall m\}$. For convenience of discussing the convergence properties, in the following, we assume:

Assumption 1. *(1) For any loss function, both subproblems (13) and (14) are solved exactly. (2) For any loss function, $S(\boldsymbol{\alpha})$ is continuous for $\boldsymbol{\alpha} \in \mathcal{A}$. (3) Set $\mathcal{U} = \{(\boldsymbol{\alpha}^k, \mathbf{d}^k, \theta^k)\}_{k=1}^{\infty}$ generated by Algorithm 1 is bounded.*

Proposition 2. *For Algorithm 1, we always have*

$$\theta^k \leq \theta^{k+1} \leq \theta^*, \ \forall k. \tag{20}$$

Proof. Since the number of constraints in problem (14) is monotonically increasing and $\mathcal{K}^k \subset \mathcal{A}$. □

Theorem 1. *(1) If there is $k < \infty$, such that $v(\mathbf{d}^k) = h(\boldsymbol{\alpha}^k, \mathbf{d}^k) \leq \theta^k$, then $\theta^k = \theta^*$. Especially, if $\boldsymbol{\alpha}^k \in \mathcal{K}^{k-1}$, we have $\theta^k = \theta^*$. (2) Let $(\bar{\boldsymbol{\alpha}}, \bar{\mathbf{d}}, \bar{\theta})$ be the limit of any convergent subsequence $\{\boldsymbol{\alpha}^{k_j}, \mathbf{d}^{k_j}, \theta^{k_j}\}_{k_j=1}^{\infty}$ of $\mathcal{U}$. If the algorithm 1 does not stop in finite steps, we have $\lim_{k_j \to \infty} \theta^{k_j} = \theta^*$. (3) If the algorithm 1 does not stop in finite steps, for any $\epsilon > 0$, there exists $k < \infty$, such that $\theta^* - \epsilon \leq \theta^k \leq \theta^*$, i.e., Algorithm 1 always finds an ϵ-optimal solution in finite steps.*

Some key properties of our proposed method are discussed in details. First, Algorithm 1 can be applied for general loss functions with the same solution pattern of $\mathbf{d}$ shown in Proposition 1. Second, solving problems (16) and (18) directly to obtain $\boldsymbol{\alpha}$ might not be efficient for large-scale and high-dimensional problems. However, their corresponding primal problems (17) and (19) can be solved efficiently, e.g., solvers in Liblinear (Fan et al. 2008), and their dual variables also can be recovered from primal solutions as discussed above. Third, the primal problems can be much more efficient for large-scale and high-dimensional datasets due to the sparsity of $\mathbf{d}$ since the input $\widehat{\mathbf{x}}_i$ only contains the groups with $d_m > 0$ to obtain $\boldsymbol{\alpha}$, so the primal solver only takes a small number of groups as the inputs. Similarly, the primal variable $\mathbf{w}$ can be reduced to the subset of groups that satisfies the test-time budget constraint, so the time complexity of solving the primal problems is reduced significantly.

Experiments

Several experimental settings are examined, including both binary classification and regression problems with a test-time budget on the total cost over features where each cost is associated to either a single feature or a group of features. Before that, we present a general setting of these experiments and the empirical analysis on the convergence and the parameter sensitivity of the proposed algorithm.

Table 1: Datasets used in the experiments.

Dataset	Train	Test	Dimension	Problem
blogData	52,392	7,624	280	Regression
slice_loc	42,800	10,700	385	Regression
Yahoo!	20,258	48,180	519	Classification
gisette	6,000	1,000	5,000	Classification
real-sim	35,582	8,894	20,958	Classification
news20.binary	15,998	3,998	1,355,191	Classification
E2006-log1p	16,087	3,308	4,272,227	Regression

Experimental setting

Seven real datasets in Table 1 are used in the experiments. The Yahoo! includes the computational cost of each features in $[1, 5, 20, 50, 100, 150, 200]$ (Xu, Weinberger, and Chapelle 2012). In total, 519 features are used. For the rest of datasets, they do not include the cost, so the synthetic costs are generated by drawing from the multinomial distribution with the probabilities of costs same as those in the Yahoo! data. Both slice_loc and blogData are freely available from UCI Machine Learning Repository[1] and the remaining datasets are from LIBSVM datasets[2]. To further examine the performance of the proposed algorithm on the costs of group-based features, we randomly partition the original features into groups with $1,000$ features and possibly one group with less then $1,000$ features by taking the same multinomial distribution as above. Datasets, news20.binary and E2006-log1p, with millions of features are used for studying cost budget constraints over groups for binary classification and regression problems, respectively.

We compare our proposed method **LTB** with several methods on the datasets listed in Table 1. The baseline methods include L2-regularized L2 loss SVC (in primal) with all features (**SVC**) for binary classification and L2-regularized L2 loss SVR (in primal) with all features (**SVR**) for regression (Fan et al. 2008), LTB with costs uniformly set as ones (**LTB-unif**), cost sensitive tree of classifiers [3] (**CSTC**) (Xu et al. 2013) for both binary classification and regression, and feature group sequencing[4] (**FS**) for regression problem (Hu et al. 2016) [5]. In the implementation, we take the primal solvers of both SVC and SVR as the SVM solvers in LTB for binary classification and regression, respectively.

The budgets are set to be a ratio ρ of the total cost in a fixed range, i.e., $\rho \in [0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$. Results of compared methods are reported by varying the budgets. The evaluation criteria are the accuracy for classification problems and the mean squared error (MSE) for regression problems. In the experiments, we tune the parameter C in the range $[0.01, 0.1, 1, 10, 100]$ and fix the parameter ϵ in SVR as 0.1. We also tune the parameters of baseline methods so as to reach the same level of budgets and report

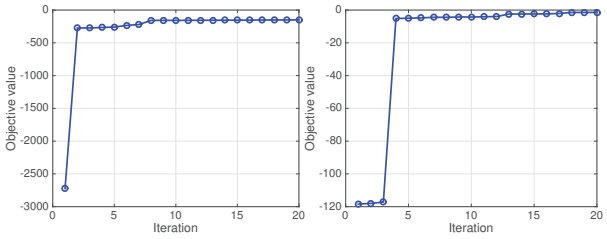
[1] https://archive.ics.uci.edu/ml/datasets.html

[2] https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

[3] http://www.cse.wustl.edu/∼xuzx/research/code/CSTC.zip

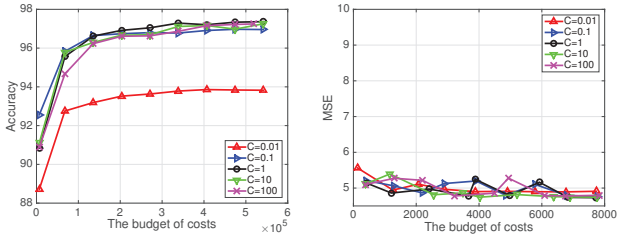
[4] https://github.com/agrubb/vowpal_wabbit/tree/omp-ext

[5] The OMP method is not reported in this paper because of the inconvenient adaptation of this method onto our datasets. As reported by authors, OMP has similar performance comparing to FS.



(a) real-sim (b) blogData

Figure 1: Convergence analysis about the objectives of LTB with respect to the number of iterations.



(a) real-sim (b) blogData

Figure 2: Parameter sensitivity analysis on the objectives of LTB with respect to the budgets by varying C.

the best results for fair comparisons.

Convergence and sensitivity analysis

We conduct experiments to study both convergence and parameter sensitivity of LTB. Figure 1 shows the convergence results of LTB on two datasets. We can observe that the objective value of problem (12) monotonically increases when the number of iterations increases, and Algorithm 1 usually converges in less than 20 iterations. We also report the experimental results of LTB on the above two datasets in terms of various budgets by varying the parameter C as shown in Figure 2. The best results among various Cs are quite robust with respect to the required budget. Meanwhile, the results of LTB over all budgets are consistently good in the most of parameter Cs. These experiments on both classification and regression problems show that LTB converges very fast to optimal solution, and is quite robust to C in a wide range. The empirical observations are consistent with the theoretical convergence analysis of the proposed method.

Analysis on the learned weights

We study the effect of the budget parameter on the number of selected features obtained by LTB in terms of some statistics on datasets, real-sim and blogData, including the number of features with learned weights equal to 1, equal to 0, and in between 0 and 1. Table 2 shows these statistics in terms of the varied cost ratio ρ. According to these results, we have the following observations. First, we can see that the increase of budget ratio ρ leads to the increase the number of selected features. The bigger the budget parameter is, the larger number of features are selected. This implies that the budget parameter indeed implicitly controls the number

Table 2: The statistics of the weights learned by LTB on real-sim and blogData. The cost ratio is the proportion of the total cost of all features. # one stands for the number of features with weights equal to one; # zero stands for the number of features with learned weights equal to zero; # (0,1) is the number of features with weights between 0 and 1.

cost ratio ρ	# one	# zero	# (0,1)	total cost	Accuracy
0.01	1374	19575	9	6958	94.25
0.1	5421	15526	11	68089	96.52
0.2	7829	13126	3	135602	96.78
0.3	9415	11541	2	203269	97.01
0.4	11391	9565	2	271211	97.23
0.5	13129	7827	2	338758	97.36
0.6	14573	6383	2	406507	97.28
0.7	16187	4770	1	474262	97.35
0.8	17605	3352	1	541997	97.37

(a) classification on real-sim

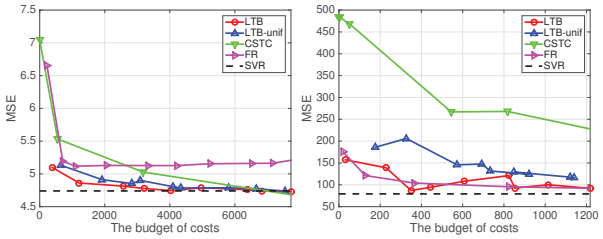
cost ratio ρ	# ones	# zeros	# (0,1)	total cost	MSE
0.01	14	264	2	127	5.56
0.1	54	213	13	1162	5.39
0.2	107	163	10	2203	5.22
0.3	102	176	2	2898	5.13
0.4	165	114	1	3879	5.25
0.5	174	102	4	4937	5.28
0.6	188	89	3	5923	5.16
0.7	217	62	1	6773	4.89
0.8	237	42	1	7735	4.91

(b) regression on blogData

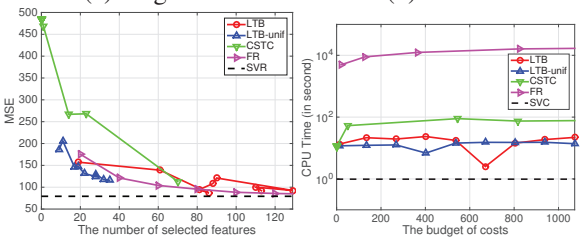
of selected features, which is consistent with the discussion based on Proposition 1. Second, the weights learned by LTB are most either 0 or 1, and few of them are values between 0 and 1. This agrees on the empirical discussion in Remark 1.

Regression

Experiments on regression problems are conducted on two datasets, where each feature is associated to one cost. Figure 3 shows the experimental results in terms of MSE. Figures 3(a) and 3(b) demonstrate that LTB can achieve better accuracy than others with the same test-time budget, and closely approximates to the best accuracy obtained by SVR using all features with the best tuned C (the dashed line). Moreover, the MSE in terms of the number of selected features is also reported on slice_loc as shown in Figure 3(c). We can see that LTB selects more features than LTB-unif for the same test-time budget, which means that LTB puts more emphasis on low-cost features than high-cost features. This is further verified by recording the features and their costs, which is summarized as a histogram shown in Figure 4(a), where the costs are reported as the original costs of features learned by LTB-unif. It is clear to see that LTB prefers to select features with low cost. Furthermore, we record the empirical time complexity in seconds shown in Figure 3(d), which shows that LTB is much more efficient than CSTC and FS. In summary, LTB performs better than others and much more efficient when a fixed test time budget is given.

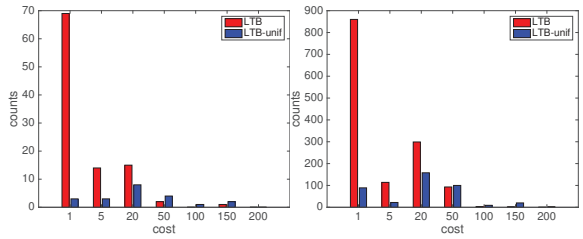


(a) blogData (b) slice_loc



(c) # features on slice_loc (d) CPU time on slice_loc

Figure 3: The MSE of compared methods on two regression datasets. (a)-(b) present the accuracy by varying the test-time budget on blogData and slice_loc, respectively. (c)-(d) record the number of selected features and the CPU time of compared methods on slice_loc, respectively.



(a) slice_loc (B=689) (b) gisette (B=12, 660)

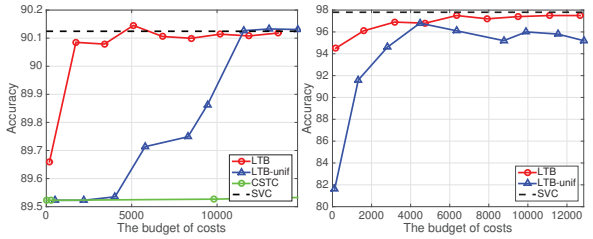
Figure 4: The histograms over the costs of the selected features by LTB and LTB-unif on two datasets with the test-time budget in the bracket that obtains the best performance.

Classification

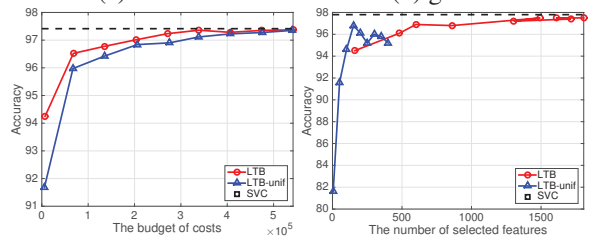
Experiments on binary classification problems are conducted on three datasets, where each feature is associated to one cost. Figure 5 shows the experimental results of compared methods on the three datasets. Note that CSTC cannot obtain results within 12 hours except Yahoo! data, so we will not report the results of CSTC on the other two datasets. Similar observations on regression problems can be found for binary classification problems. In other words, our proposed LTB for binary classification also prefers to select low-cost features in Figure 4(b) and is able to efficiently approximate the best performance of SVC.

Group-based cost budget

Different from the aforementioned experiments, we here verify our method in the setting of the cost associated to a group of features. Figure 6 shows the performance of compared methods on two high-dimensional datasets for both

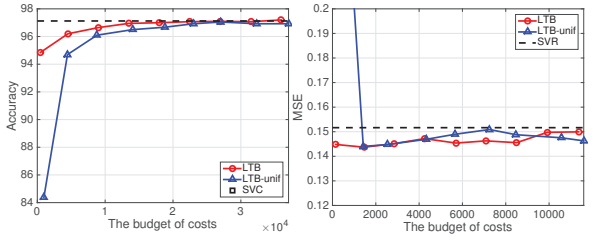


(a) Yahoo! (b) gisette



(c) real-sim (d) # features on gisette

Figure 5: Classification accuracy of compared methods on three binary classification datasets. (a)-(c) presents the accuracy by varying the test-time budget. (d) reports the accuracy with respect to the number of selected features on gisette.



(a) news20.binary (b) log1p_E2006

Figure 6: The performance of compared methods based on test-time budget over the costs of group-based features for both classification and regression.

classification and regression problems. Since the dimension of the two datasets is the size of millions, CSTC and FS have issues on either memory or time cost, we only report our methods and SVMs on all features. On news20.binary, the same result as the previous experiments can be observe. In addition, we find that the regression model based on test-time budget works better than the best result of SVR. These observations verify the effectiveness of the proposed LTB method in the setting of group-based budget.

Conclusions

In this paper, we propose a novel model for learning a linear predictor with a budget on the total cost of features where each cost can correspond to either one single feature or a group of features. Our model can incorporate any loss function without much added effort. An efficient optimization method is proposed to solve the problem with general loss functions. Two loss functions are specified as examples to demonstrate the flexibility of our model for binary classifi-

cation and regression. For these two losses, our algorithm can be much more efficient by solving SVMs in primal. Experiments on various sizes of real datasets demonstrate that our model for learning a predictor with a budget is more effective and efficient than baseline methods.

References

Cesa-Bianchi, N.; Shalev-Shwartz, S.; and Shamir, O. 2011. Efficient learning with partially observed attributes. *JMLR* 12(Oct):2857–2878.

Chapelle, O., and Chang, Y. 2011. Yahoo! learning to rank challenge overview. In *Yahoo! Learning to Rank Challenge*, 1–24.

Do, H.; Kalousis, A.; and Hilario, M. 2009. Feature weighting using margin and radius based error bound optimization in svms. *Machine Learning and Knowledge Discovery in Databases* 315–329.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *JMLR* 9(Aug):1871–1874.

Greiner, R.; Grove, A. J.; and Roth, D. 2002. Learning cost-sensitive active classifiers this extends the short conference paper [19]. *Artificial Intelligence* 139(2):137–174.

Grubb, A., and Bagnell, D. 2012. Speedboost: Anytime prediction with uniform near-optimality. In *AISTATS*, volume 15, 458–466.

He, H.; Eisner, J.; and Daume, H. 2012. Imitation learning by coaching. In *NIPS*, 3149–3157.

Hettich, R., and Kortanek, K. O. 1993. Semi-infinite programming: theory, methods, and applications. *SIAM review* 35(3):380–429.

Hu, H.; Grubb, A.; Bagnell, J. A.; and Hebert, M. 2016. Efficient feature group sequencing for anytime linear prediction. *UAI*.

Huang, Y.; Powers, B.; and Reyzin, L. 2015. Training-time optimization of a budgeted booster. In *IJCAI*, 3583–3589.

Kelley, Jr, J. E. 1960. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics* 8(4):703–712.

Kononenko, I. 2001. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine* 23(1):89–109.

Kusner, M. J.; Chen, W.; Zhou, Q.; Xu, Z. E.; Weinberger, K. Q.; and Chen, Y. 2014. Feature-cost sensitive learning with submodular trees of classifiers. In *AAAI*, 1939–1945.

Mao, Q., and Tsang, I. W.-H. 2013. A feature selection method for multivariate performance measures. *T-PAMI* 35(9):2051–2063.

Nan, F.; Wang, J.; and Saligrama, V. 2016. Pruning random forests for prediction on a budget. In *Advances in Neural Information Processing Systems*, 2334–2342.

Pelossof, R.; Jones, M.; and Ying, Z. 2010. Speeding-up margin based learning via stochastic curtailment. In *ICML/COLT Budgeted Learning Workshop*.

Rakotomamonjy, A.; Bach, F. R.; Canu, S.; and Grandvalet, Y. 2008. Simplemkl. *JMLR* 9(Nov):2491–2521.

Reyzin, L. 2011. Boosting on a budget: Sampling for feature-efficient prediction. In *ICML*, 529–536.

Scholkopf, B., and Smola, A. J. 2001. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schwing, A. G.; Zach, C.; Zheng, Y.; and Pollefeys, M. 2011. Adaptive random foresthow many experts to ask before making a decision? In *CVPR*, 1377–1384. IEEE.

Sonnenburg, S.; Rätsch, G.; Schäfer, C.; and Schölkopf, B. 2006. Large scale multiple kernel learning. *JMLR* 7(Jul):1531–1565.

Sun, P., and Zhou, J. 2013. Saving evaluation time for the decision function in boosting: Representation and reordering base learner. In *ICML*, 933–941.

Tan, M.; Tsang, I. W.; and Wang, L. 2014. Towards ultrahigh dimensional feature selection for big data. *JMLR* 15(1):1371–1429.

Tsochantaridis, I.; Joachims, T.; Hofmann, T.; and Altun, Y. 2005. Large margin methods for structured and interdependent output variables. *JMLR* 6(Sep):1453–1484.

Vedaldi, A.; Gulshan, V.; Varma, M.; and Zisserman, A. 2009. Multiple kernels for object detection. In *ICCV*, 606–613. IEEE.

Wang, J.; Trapeznikov, K.; and Saligrama, V. 2015. Efficient learning by directed acyclic graph for resource constrained prediction. In *Advances in Neural Information Processing Systems*, 2152–2160.

Xu, Z.; Jin, R.; Ye, J.; Lyu, M. R.; and King, I. 2009. Non-monotonic feature selection. In *ICML*, 1145–1152. ACM.

Xu, Z. E.; Kusner, M. J.; Weinberger, K. Q.; and Chen, M. 2013. Cost-sensitive tree of classifiers. In *ICML*, 133–141.

Xu, Z.; Weinberger, K.; and Chapelle, O. 2012. The greedy miser: Learning under test-time budgets. *ICML*.