Veto-Consensus Multiple Kernel Learning

Yuxun Zhou
Department of EECS
UC Berkeley
yxzhou@berkeley.edu

Ninghang Hu
Amsterdam Machine Learning Lab
University of Amsterdam
huninghang@gmail.com

Costas J. Spanos
Department of EECS
UC Berkeley
spanos@berkeley.edu

Abstract

We propose Veto-Consensus Multiple Kernel Learning (VCMKL), a novel way of combining multiple kernels such that one class of samples is described by the logical intersection (consensus) of base kernelized decision rules, whereas the other classes by the union (veto) of their complements. The proposed configuration is a natural fit for domain description and learning with hidden subgroups. We first provide generalization risk bound in terms of the Rademacher complexity of the classifier, then a large margin multi-ν learning objective with tunable training error bound is formulated. Seeing that the corresponding optimization is non-convex and existing methods severely suffer from local minima, we establish a new algorithm, namely Parametric Dual Descent Procedure(PDDP), that can approach global optimum with guarantees. The bases of PDDP are two theorems that reveal the global convexity and local explicitness of the parameterized dual optimum, for which a series of new techniques for parametric program have been developed. The proposed method is evaluated on extensive set of experiments, and the results show significant improvement over the state-of-the-art approaches.

1 Introduction

In recent years, multiple kernel learning (MKL) has shown promising results in a variety of applications and has attracted much attention in machine learning community. Given a set of base kernels, MKL finds an optimal combination of them with which an appropriate hypothesis is determined on the training data. A large body of literature has been addressing the arising issues of MKL, mainly from three perspectives and their intersections, i.e. theoretical learning bound, related optimization algorithm, and alternative MKL settings. To list a few, the generalization bounds for learning linear combination of multiple kernels have been extensively studied in (Ying and Campbell 2009; Cortes, Mohri, and Rostamizadeh 2010; Hussain and Shawe-Taylor 2011) by analyzing various complexity metrics. Following the pioneer work (Lanckriet et al. 2004) that formulates linear MKL as a semi-definite program (SDP), a series of work is devoted to improve the efficiency with various optimization techniques, such as reduced gradient (Rakotomamonjy et al. 2008), Newtown's method (Kloft et al. 2011), and mirror descent (Jagarlapudi et al. 2009). Also data related issues such as sample adaptability and missing channels (Liu et al. 2014; 2015) have been addressed. Despite the substantial theoretical advancement and algorithmic progress made in linear MKL, few of the results could be directly applied to MKL that incorporates nonlinear combinations. Indeed non-linear MKL is usually studied on a case-by-case basis, such as polynomial combination (Cortes, Mohri, and Rostamizadeh 2009), hierarchical kernel (Bach 2009), hyperkernels (Ong, Williamson, and Smola 2005), etc.

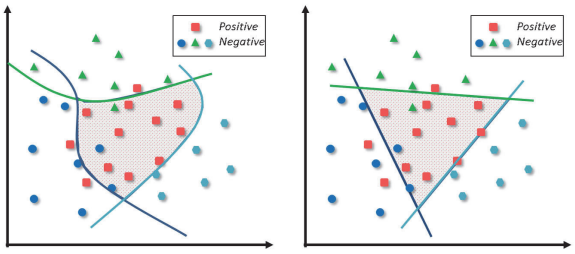

Figure 1: 2D VCMK with non-linear/all linear base kernels

In this work we propose a novel non-linear combination of multiple kernels. To motivate the configuration, Figure 1 illustrates a practical problem where part of the classes contains hidden structures. In this example, the positive class is labeled. In contrast, the negative class contains several subgroups but only a"single" label is provided. To compensate for this implicit information, we propose to describe the positive class by the intersection of the acceptance region of multiple base kernel decision rules, and the negative class by the union of their complements. Hence a sample is classified as negative as long as one or more rules "votes" negative (Veto), and a positive assignment is made for a sample if and only if all of the rules agree (Consensus). With this intuition, VCMKL is a natural solution for applications involving hidden structures. Moreover because the construction inherently emphasizes the sensitivity to negative class and the specificity to positive class, it is also a promising tool for domain description problems.

We discuss the proposed VCMKL from both theoretical and algorithmic perspectives. Firstly, we formalize the

the construction of the classifier and provide Rademacher complexity analysis. Then a large margin multi-ν learning formulation is proposed with training error controlled by hyperparameters. Another main contribution of this paper is a learning algorithm that can approach global optimum for the non-convex learning objective. We first show that the objective could be reformulated with hidden variables, and then we show that the dual solution is locally explicit and globally convex piece-wise quadratic in hidden variables. Based on this, a subgradient algorithm is proposed with guaranteed convergence to global optimum. We call the overall framework Parametric Dual Descent Procedure (PDDP). Although we develop it for the purpose of VCMKL, the method can be readily adapted to many other situations such as the training of latent structured SVM, ramp loss robust Regression/SVM and approximate L_0 regularized problems (Yu and Joachims 2009; Xu, Crammer, and Schuurmans 2006). Please refer to the supplementary material for details. To the best of our knowledge, PDDP is the first method that is able to find global optimum for this class of non-convex learning problems without resorting to combinatorial search.

2 Related Work

With only linear kernels, VCMKL is reduced to constructing convex polyhedrons in the original feature space for classification, as is shown in the right subplot of Figure 1. Recently several attempts were made for this purpose: Based on the AND-OR model, (Dundar et al. 2008) proposed an alternating algorithm, assuming some subclass labels are available. In (Manwani and Sastry 2010; 2011) the author proposed two methods, one with logistic link function and the other based on perception algorithm. In (Kantchelian et al. 2014) a Stochastic Gradient Descent (SGD) method is applied with a notion of "entropy" to alleviate collapsed solutions. Improved SGD with better initialization method is proposed in (Zhou, Li, and Spanos 2015; Zhou, Jin, and Spanos 2015). However, all of these methods only deal with the primal, which cannot be directly adopted in a general MKL setting. Moreover, even with searching heuristics these learning methods still suffer severely from local minimums.

As will be shown in section 3, with a slight reformulation, the VCMKL objective resembles a multi-kernel extension of SVMs with hidden/latent variables. Although the idea of including hidden variables in SVM variations has been suggested in literature (Yu and Joachims 2009; Felzenszwalb et al. 2010), the proposed VCMKL is novel in that multiple base kernels are combined and each hidden subgroup is dealt with by an optimal one. Regarding the learning algorithm, the aforementioned work uses either group alternating optimization or Concave-Convex Procedure (CCCP). Again these methods only converge to local minimas and the solution could be very deteriorated with large number of latent variables (Stephen Boyd 2004; Floudas 1995).

3 Problem Formulation

3.1 The Classifier and Generalization Bound

Let $\mathcal{S} = \{\boldsymbol{x}_i, y_i\}_{i=1}^l$ be a training data set, where $\boldsymbol{x} \in \mathbb{R}^d$ and d is the dimension of features. Without loss of generality let $y \in \{+1, -1\}$ indicate class labels, with negative class contains hidden subgroups. Consider a feature mapping $\phi : \mathbb{R}^d \to \mathcal{H}$, in the new Hilbert space a hyperplane can be written as $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle_{\mathcal{H}} + b$. A decision rule for data $\boldsymbol{x'}$ is given by the sign of $f(\boldsymbol{x'})$. Formalizing the idea of using intersections of M base kernel mappings for positive class and the union of their complement for negative class, the composed classifier $g(\cdot)$ is such that

$$\begin{aligned}\{g(\boldsymbol{x}) > 0\} &= \{f_1(\boldsymbol{x}) > 0\} \cap \cdots \cap \{f_M(\boldsymbol{x}) > 0\} \\ &= \left\{ \min_{1,\cdots,M} \{f_1(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})\} > 0 \right\}\end{aligned}$$

On the other hand, the acceptance region for negative class is $\{\min_{1,\cdots,M}\{f_1(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})\} \leq 0\}$. For short notation, let us denote $\langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle_{\mathcal{H}} = \boldsymbol{w} \cdot \phi(\boldsymbol{x})$ as the inner product and $||\cdot||$ as the corresponding norm in $\mathcal{H}$. Then the combined classifier is simply

$$g(\boldsymbol{x}) = \min\{\boldsymbol{w}_1 \cdot \phi(\boldsymbol{x}) + b_1, \cdots, \boldsymbol{w}_M \cdot \phi(\boldsymbol{x}) + b_M\}$$

Before proceeding to any method to learn this classifier, we conduct complexity analysis in MKL framework for generalization bound and model selection purpose. Let the function class of g be denoted as $\mathcal{G}$, and that of f_j be denoted as $\mathcal{F}_j$. As a classic measure of richness, the *Empirical Rademacher Complexity* for a function class $\mathcal{F}$ is defined as $\widehat{\mathcal{R}}(\mathcal{F}(\boldsymbol{x}_1^l)) \triangleq E_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \left| \frac{2}{l} \sum_{i=1}^l \sigma_i f(\boldsymbol{x}_i) \right| \right]$ where $\sigma_1, \cdots, \sigma_l$ are i.i.d. Rademacher variables such that $\mathbb{P}(\sigma_i = +1) = \mathbb{P}(\sigma_i = -1) = 1/2$. The definition measures the complexity/richness of function class $\mathcal{F}$ in terms of its ability to "match" Rademancher variables. With Talagrand's Lemma and an induction argument, we show that

Theorem 1. *The function class $\mathcal{G}$ of VCMKL has*

$$\widehat{\mathcal{R}}(\mathcal{G}\left(\boldsymbol{x}_1^l\right)) \leq 2 \sum_{j=1}^M \widehat{\mathcal{R}}\left(\mathcal{F}_j(\boldsymbol{x}_1^l)\right)$$

Further assume $\mathcal{F}_j$ forms a bounded function class with kernel $\kappa_j(\cdot,\cdot)$ and kernel matrix $\boldsymbol{K}_j$ such that $\mathcal{F}_j = \left\{ \boldsymbol{x} \mapsto \sum_{i=1}^l \alpha_i \kappa_j(\boldsymbol{x}_i, \boldsymbol{x}) \mid \boldsymbol{\alpha}^T \boldsymbol{K}_j \boldsymbol{\alpha} \leq B_j \right\}$ then $\widehat{\mathcal{R}}\left(\mathcal{G}(\boldsymbol{x}_1^l)\right) \leq \frac{4}{l} \sum_{j=1}^M B_j \sqrt{tr(\boldsymbol{K}_j)}$.

Note that in general it is hard to tighten the additive nature of the complexity[1]. With the above results at hand, the generalization guarantee of the MKVCL can be obtained immediately from the classic results in the PAC learning framework. Let $L(g) = E_{\mathcal{S}}[\mathbf{1}_{\text{sgn}(g(\boldsymbol{x}))\neq y}]$ be the generalization error of the MKVC classifier g, and let $\widehat{L}_\rho(g) \triangleq \frac{1}{l} \sum_{i=1}^l \Psi_\rho\left(y_i g(\boldsymbol{x}_i)\right)$ be the empirical $\rho-margin$ loss with

[1] In fact in the case of all linear kernels, it is shown in (Zhou, Li, and Spanos 2015) that the VC dimension has additive lower bound.

$\Psi_\rho(t) = [\min\{1, 1 - t/\rho\}]_+$. Then we have with probability at least $1 - \delta$

$$L(g) \leq \widehat{L}_\rho(g) + \frac{8}{\rho}\sum_{j=1}^{M}\frac{B_j\sqrt{\mathrm{tr}(\boldsymbol{K}_j)}}{l} + 3\sqrt{\frac{\log(2/\delta)}{2l}}$$

3.2 Large Margin Learning Objective

To learn the multi-kernels classifier from data, we adopt a learning objective that maximize the total margin defined in (Kantchelian et al. 2014) while minimize the hinge loss of misclassifications. Inspired by the advantages of νSVM (Scholkopf B 2000), the following large margin multi-ν learning formulation is proposed:

$$\begin{aligned}\min_{\boldsymbol{w}_m, b_m, \rho_m} \frac{1}{2}\sum_{m=1}^{M}||\boldsymbol{w}_m||^2 - \sum_{m=1}^{M}\nu_m\rho_m \\ + \frac{\gamma}{l}\sum_{i\in I^+}\max_m\{[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+\} \\ + \frac{1-\gamma}{l}\sum_{i\in I^-}\min_m\{[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+\}\end{aligned}$$

(**OPT1**)

where I^+ and I^- are index sets for positive and negative class, respectively. The hyperparameters $\nu_1, \cdots, \nu_M \in [0, 1]$ weight the margins, and two types of losses are treated differently by introducing $\gamma \in [0, 1]$. The multi-ν formulation still reflects the Veto-Consensus intuition: the loss for positive class is the maximum over all decision boundaries, while for negative class only the one with minimum loss is counted. The first three terms in the above minimization problem are convex, however the last term is non-convex as the minimum of M truncated hyperplanes. We handle this term by introducing new variables. Consider a weighted version of the M losses over a simplex:

$$L_{avg}(\boldsymbol{w}, \boldsymbol{x}_i, \boldsymbol{\lambda}_i) = \sum_{m=1}^{M}\lambda_{im}[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+ \quad (1)$$

with $\boldsymbol{\lambda}_i \in \left\{\boldsymbol{\lambda}_i : \sum_{m=1}^{M}\lambda_{im} = 1, \lambda_{im} \geq 0\right\} \triangleq \mathbb{S}^M$, a row vector in the $|I^-| \times M$ matrix $\boldsymbol{\lambda}$ containing the loss weighting parameters of negative samples. Denote $L_{min}(\boldsymbol{w}, \boldsymbol{x}_i) = \min_m\{[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+\}$ as the original loss for $\boldsymbol{x}_i$, it is straightforward that

$$L_{min}(\boldsymbol{w}, \boldsymbol{x}_i) = \min_{\boldsymbol{\lambda}_i\in\mathbb{S}^M} L_{avg}(\boldsymbol{w}, \boldsymbol{x}_i, \boldsymbol{\lambda}_i) \quad (2)$$

With this trick we reformulate the learning objective as

Proposition 1. ***(OPT1)** is equivalent to*

$$\begin{aligned}\min_{\boldsymbol{\lambda}_i\in\mathbb{S}^M}\min_{\substack{\boldsymbol{w}_m, b_m, \\ \rho_m\geq 0}} \frac{1}{2}\sum_{m=1}^{M}||\boldsymbol{w}_m||^2 - \sum_{m=1}^{M}\nu_m\rho_m \\ + \frac{\gamma}{l}\sum_{i\in I^+}\max_m\{[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+\} \\ + \frac{1-\gamma}{l}\sum_{i\in I^-}\sum_{m=1}^{M}\lambda_{im}[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+\end{aligned}$$

(**Primal**)

The newly introduced variables $\boldsymbol{\lambda}$ can be viewed as hidden subgroup indicators, hence VCMKL can indeed be thought of as a multi-kernel extension of learning with latent variables. Considering the form of OPT1 and Primal, it is tempting to apply CCCP and alternating heuristics. Yet in this work a rigorous optimization algorithm will be developed to approach global optimum. But before that let's look into the relation between training error and the hyperparameters ν_m, γ in the learning formulation. Replacing the inner optimization of the Primal with its dual, we obtain that the Primal is equivalent to

$$\min_{\boldsymbol{\lambda}_i\in\mathbb{S}^M}\mathcal{J}_d(\boldsymbol{\lambda}) \qquad \text{where}$$

$$\mathcal{J}_d(\boldsymbol{\lambda}) = \begin{cases} \min_{\boldsymbol{\alpha}}\frac{1}{2}\sum_{m=1}^{M}\sum_{i,j=1}^{l}\alpha_{im}y_i\kappa_m(\boldsymbol{x}_i, \boldsymbol{x}_j)y_j\alpha_{jm} \\ subject\ to \\ \alpha_{im} \geq 0 \quad \forall i,\ \forall m \\ \alpha_{im} \leq \frac{1-\gamma}{l}\lambda_{im} \quad \forall i \in I^-,\ \forall m \\ \sum_{m=1}^{M}\alpha_{im} \leq \frac{\gamma}{l} \quad \forall i \in I^+ \\ \sum_{i=1}^{l}\alpha_{im} \geq \nu_m \quad \forall m \\ \sum_{i=1}^{l}\alpha_{im}y_i = 0 \quad \forall m \end{cases}$$

(**Dual**)

To see the effect of hyperparameters, it is useful to define partition of samples similar to the classical SVM:

Definition 1. *Partition of Samples : Based on the value of α_{im} at optimal, the i^{th} sample is called*

- *positive support vector if $i \in I^+$ and $\sum_m \alpha_{im} > 0$.*
- *positive bounded support vector if $i \in I^+$ and $\sum_m \alpha_{im} = \frac{\gamma}{l}$.*
- *negative support vector of class m if $i \in I^-$ and $\alpha_{im} > 0$.*
- *negative bounded support vector of class m if $i \in I^-$ and $\alpha_{im} = \frac{1-\gamma}{l}$.*

All the other samples are called non-support vectors. The following proposition relates the choice of hyperparameters to the training error tolerance.

Proposition 2. *Define $\nu^+ = \frac{l\sum_m \nu_m}{2\gamma|I^+|}$ and $\nu_m^- = \frac{l\nu_m}{2(1-\gamma)|I^-|}$, and denote $N^{sv+}, N_m^{sv-}, N^{bsv+}, N_m^{bsv-}$ as the number of all postive/negative support vectors, positive/negative bounded support vectors, respectively, then*

$$\frac{N^{bsv+}}{|I^+|} \leq \nu^+ \leq \frac{N^{sv+}}{|I^+|} \quad (3a)$$

$$\frac{N_m^{bsv-}}{|I^-|} \leq \nu_m^- \leq \frac{N_m^{sv-}}{|I^-|} \quad (3b)$$

Form the right hand side, ν^+ and ν_m^- give a lower bound on the fraction of positive support vectors and negative support vectors of class m, respectively. The left hand side upper bound is more interesting: by definition, $N^{bsv+}/|I^+|$ and $N_m^{bsv-}/|I^-|$ are respectively the training false negative error and false positive error of class m. Hence the bound implies that one can impose smaller training error of different types by decreasing corresponding ν. The role of γ is also significant: it can incorporate an uneven consideration of training errors committed in two classes, which

can be harnessed to handle imbalanced availability of positive/negative samples. In short, the advantage of the multi-ν formulation is that the training result can be controlled simply by tuning bounds as a function of hyperparameters.

4 PDDP Learning Method

We introduce the new global optimization method PDDP in this section with VCMKL as a special case. The first step involves writing the optimal solution $\boldsymbol{\alpha}^*$ and the optimal value $\mathcal{J}_d^*$ as a locally explicit function of $\boldsymbol{\lambda}$. In the terminology of mathematical programming, this non-trivial task is referred to as parametric program, or sensitivity analysis. For the ease of notation, we write $\mathcal{J}_d(\boldsymbol{\lambda})$ in a more compact matrix form

$$\min_{\boldsymbol{\alpha}_1,\cdots,\boldsymbol{\alpha}_M} \mathcal{J}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{m=1}^{M} \boldsymbol{\alpha}_m^T \boldsymbol{Q}_m \boldsymbol{\alpha}_m \qquad \textbf{(OPT2)}$$

subject to

$$\text{Group I:} \quad \boldsymbol{C}^{\alpha}\boldsymbol{\alpha}_m \le \boldsymbol{C}^{\lambda}\boldsymbol{\lambda}_m + \boldsymbol{C}^m \quad \forall m \tag{4a}$$

$$\text{Group II:} \quad [1,\cdots,1]\boldsymbol{\alpha}_i^T \le \frac{\gamma}{l} \quad \forall i \in I^+ \tag{4b}$$

$$\text{OHP:} \quad \boldsymbol{y}^T\boldsymbol{\alpha}_m = 0 \quad \forall m \tag{4c}$$

where $\boldsymbol{Q}_m = \boldsymbol{K}_m \circ yy^T$ with $\boldsymbol{K}_m$ the Gram matrix for mth base kernel. The inequalities addressing each subgroup m are contained in Group I, and constant matrices $\boldsymbol{C}^{\alpha}, \boldsymbol{C}^{\lambda}, \boldsymbol{C}^m$ encapsulate the coefficients. Group II includes the third inequality in the dual. The equality constraint (4c) is called orthogonal hyperplane property due to its geometric interpretation. With the solution $\boldsymbol{\alpha}^*(\boldsymbol{\lambda})$, the i^{th} row of the m^{th} constraint (4a) is active if $\boldsymbol{C}_i^{\alpha}\boldsymbol{\alpha}_m^*(\boldsymbol{\lambda}) = \boldsymbol{C}_i^{\lambda}\boldsymbol{\lambda}_m + \boldsymbol{C}_i^m$, and inactive if $\boldsymbol{C}_i^{\alpha}\boldsymbol{\alpha}_m^*(\boldsymbol{\lambda}) < \boldsymbol{C}_i^{\lambda}\boldsymbol{\lambda}_m + \boldsymbol{C}_i^m$. We have used column vector convention for $\boldsymbol{\alpha}_m, \boldsymbol{\lambda}_m$, and row/column selection of a matrix with subscripts.

In order to determine the relation between the optimum of OPT2 and its "parameters" $\boldsymbol{\lambda}$, we develop a series of new techniques for parametric program. Interested readers are referred to supplementary material for detailed proofs of the main theorems in this section. In particular, we find that the existence of local explicit expression is guaranteed if the following sample partition property is satisfied.

Definition 2. ***(Non-degeneracy)*** *A solution of the dual is called Non-degenerate if there exist at least one positive unbounded support vector and one negative unbounded support vector for each class m.*

Since the unbounded support vectors are essentially the sample points that lie on the margin boundaries, except for deliberately designed cases, in practice with large sample size the non-degeneracy is expected to be satisfied. With that we provide the local explicitness theorem.

Theorem 2. *Assume that the solution of the VCMKL inner optimization is non-degenerate, then*

1. *For all m, the optimal solution $\boldsymbol{\alpha}_m^*$ is a continuous piecewise affine (**PWA**) function of $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_M]$*
2. *The optimal objective $\mathcal{J}(\boldsymbol{\alpha}^*)$ is a continuous piece-wise quadratic (**PWQ**) function of $\boldsymbol{\lambda}$.*
3. *At optimal, let the index set of group I active constraints be denoted as $\mathcal{A}_m,\ \forall\ m = \{1,\cdots,M\}$, and the index set of group II active constraints be denoted as $\mathcal{B}$. Then in the critical region defined by inequalities (5) – (8), the optimal solution $\boldsymbol{\alpha}^*$ admits a **closed form***

$$\begin{aligned}\boldsymbol{\alpha}_m^* =& \boldsymbol{F}_m(\boldsymbol{C}_{\mathcal{A}_m}^{\alpha})^T\boldsymbol{G}_m^{-1}\left(\boldsymbol{C}_{\mathcal{A}_m}^{\lambda}\boldsymbol{\lambda}_m^* + \boldsymbol{C}_{\mathcal{A}_m}^m\right) \\ &- \tilde{\boldsymbol{R}}_m\boldsymbol{U}^{-1}\left(\sum_{m=1}^{M}\boldsymbol{H}_{\mathcal{B}m}\left(\boldsymbol{C}_{\mathcal{A}_m}^{\lambda}\boldsymbol{\lambda}_m^* + \boldsymbol{C}_{\mathcal{A}_m}^m\right) - \frac{\gamma}{l}\boldsymbol{1}_{\mathcal{B}}\right)\end{aligned}$$

where the involved matrices are computed by

$$\boldsymbol{F}_m \triangleq \frac{\boldsymbol{Q}_m^{-1}\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{Q}_m^{-1}}{\boldsymbol{y}^T\boldsymbol{Q}_m^{-1}\boldsymbol{y}} - \boldsymbol{Q}_m^{-1};\ \boldsymbol{G}_m \triangleq \boldsymbol{C}_{\mathcal{A}_m}^{\alpha}\boldsymbol{F}_m(\boldsymbol{C}_{\mathcal{A}_m}^{\alpha})^T;$$
$$\boldsymbol{H}_m \triangleq \boldsymbol{F}_m(\boldsymbol{C}_{\mathcal{A}_m}^{\alpha})^T\boldsymbol{G}_m^{-1};\ \boldsymbol{R}_m \triangleq \boldsymbol{F}_m(\boldsymbol{C}_{\mathcal{A}_m}^{\alpha})^T\boldsymbol{G}_m^{-1}\boldsymbol{C}_{\mathcal{A}_m}^{\alpha}\boldsymbol{F}_m - \boldsymbol{F}_m;$$
$$\boldsymbol{U} \triangleq \sum_{m=1}^{M}[\boldsymbol{R}_m]_{\mathcal{B}\times\mathcal{B}};\quad \tilde{\boldsymbol{R}}_m \triangleq [\boldsymbol{R}_{m0}]_{\bullet\mathcal{B}};$$

and $\boldsymbol{R}_{m0}$ is obtained from matrix $\boldsymbol{R}_m$ by setting rows that are not in $\mathcal{B}$ to $\boldsymbol{0}$

In essence the theorem indicates that each time the inner optimization OPT2 of the dual is solved, full information (closed form solution) in a well-defined neighborhood (critical region) can be retrieved as a function of outer optimization variable $\boldsymbol{\lambda}$. Besides this local explicitness result, we characterize the overall geometric structure of the optimality, and show in the next theorem that globally the optimal objective is convex in $\boldsymbol{\lambda}$, which serves as the underpinning for PDDP to be a global optimization algorithm.

Theorem 3. *Assuming non-degeneracy, then*

1. *The dual optimization has finite number of polyhedron critical regions $CR_1,\cdots,CR_{N_r}$ which constitute a **partition** of the feasible set of $\boldsymbol{\lambda}$, i.e. each feasible $\boldsymbol{\lambda}$ belongs to one and only one critical region.*
2. *The optimal objective $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}))$ is a **convex** PWQ function of $\boldsymbol{\lambda}$, and is almost everywhere differentiable.*

Now the learning problem is reduced to minimizing a non-smooth but convex function $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}))$. Projected subgradient based method is a natural choice. At each step, OPT2 is solved with current $\boldsymbol{\lambda}$, and with Theorem 2 one can directly compute the critical region boundaries (5)–(8), as well as the gradients with respect to $\boldsymbol{\lambda}_m$:

$$\nabla_{\boldsymbol{\lambda}_m}\mathcal{J}(\boldsymbol{\alpha}^*) = \boldsymbol{D}_m^T\boldsymbol{Q}_m\boldsymbol{\alpha}_{\cdot m}^* - \boldsymbol{E}_m^T\sum_{m'\neq m}\tilde{\boldsymbol{R}}_{m'}^T\boldsymbol{Q}_{m'}\boldsymbol{\alpha}_{\cdot m'}^*$$

$$\boldsymbol{E}_m = \boldsymbol{U}^{-1}\boldsymbol{H}_{\mathcal{B}m}\boldsymbol{C}_{\mathcal{A}m}^{\lambda};\ \boldsymbol{D}_m = \boldsymbol{F}_m(\boldsymbol{C}_{\mathcal{A}_m}^{\alpha})^T\boldsymbol{G}_m^{-1}\boldsymbol{C}_{\mathcal{A}_m}^{\lambda} - \tilde{\boldsymbol{R}}_m\boldsymbol{E}_m$$

The function is not differentiable only on the boundary of critical regions. In this case the subgradient set is the convex hull of the gradients of adjacent regions, as the dual can be viewed as point-wisely optimizing $\boldsymbol{\alpha}$ with each $\boldsymbol{\lambda}$ (Stephen Boyd 2004). Thus for both cases one can simply use the gradient induced by current optimal objective and proceed to update $\boldsymbol{\lambda}$. Since each $\boldsymbol{\lambda}_i$ is a M dimensional simplex, the updated value is projected row by row onto this space. By theorem 2, if $\boldsymbol{\lambda}^t$ is in the critical regions that have been explored before, all information could be retrieved in

$$\boldsymbol{G}_m^{-1}\left(\boldsymbol{C}^{\lambda}_{\mathcal{A}_m}\boldsymbol{\lambda}^*_m+\boldsymbol{C}^{m}_{\mathcal{A}_m}\right)-[\boldsymbol{G}_m^{-1}\boldsymbol{C}^{\alpha}_{\mathcal{A}_m}\boldsymbol{F}_m]_{\bullet\mathcal{B}}\boldsymbol{U}^{-1}\left(\sum_{m=1}^{M}\boldsymbol{H}_{\mathcal{B}m}\left(\boldsymbol{C}^{\lambda}_{\mathcal{A}_m}\boldsymbol{\lambda}^*_m+\boldsymbol{C}^{m}_{\mathcal{A}_m}\right)-\tfrac{\gamma}{l}\mathbf{1}_{\mathcal{B}}\right)\geq 0 \tag{5}$$

$$\boldsymbol{C}^{\alpha}_{\mathcal{A}^c_m}\boldsymbol{F}_m(\boldsymbol{C}^{\alpha}_{\mathcal{A}_m})^T\boldsymbol{G}_m^{-1}\left(\boldsymbol{C}^{\lambda}_{\mathcal{A}_m}\boldsymbol{\lambda}^*_m+\boldsymbol{C}^{m}_{\mathcal{A}_m}\right)-\boldsymbol{C}^{\alpha}_{\mathcal{A}^c_m}\tilde{\boldsymbol{R}}_m\boldsymbol{U}^{-1}\left(\sum_{m'=1}^{M}\boldsymbol{H}_{\mathcal{B}m'}\left(\boldsymbol{C}^{\lambda}_{\mathcal{A}_{m'}}\boldsymbol{\lambda}^*_{m'}+\boldsymbol{C}^{m'}_{\mathcal{A}'_m}\right)-\tfrac{\gamma}{l}\mathbf{1}_{\mathcal{B}}\right)-\boldsymbol{C}^{\lambda}_{\mathcal{A}^c_m}\boldsymbol{\lambda}^*_m+\boldsymbol{C}^{m}_{\mathcal{A}^c_m}\leq 0 \tag{6}$$

$$\boldsymbol{U}^{-1}\left(\sum_{m=1}^{M}\boldsymbol{H}_{\mathcal{B}m}\left(\boldsymbol{C}^{\lambda}_{\mathcal{A}_m}\boldsymbol{\lambda}^*_m+\boldsymbol{C}^{m}_{\mathcal{A}_m}\right)-\tfrac{\gamma}{l}\mathbf{1}_{\mathcal{B}}\right)\geq 0 \tag{7}$$

$$\sum_{m=1}^{M}\boldsymbol{F}_{\mathcal{B}m}(\boldsymbol{C}^{\alpha}_{\mathcal{A}_m})^T\boldsymbol{G}_m^{-1}\left(\boldsymbol{C}^{\lambda}_{\mathcal{A}_m}\boldsymbol{\lambda}^*_m+\boldsymbol{C}^{m}_{\mathcal{A}_m}\right)-\sum_{m=1}^{M}\tilde{\boldsymbol{R}}_{\mathcal{B}m}\boldsymbol{U}^{-1}\left(\sum_{m'=1}^{M}\boldsymbol{H}_{\mathcal{B}m'}\left(\boldsymbol{C}^{\lambda}_{\mathcal{A}_{m'}}\boldsymbol{\lambda}^*_{m'}+\boldsymbol{C}^{m'}_{\mathcal{A}'_m}\right)-\tfrac{\gamma}{l}\mathbf{1}_{\mathcal{B}}\right)-\tfrac{\gamma}{l}\mathbf{1}_{\mathcal{B}}\leq 0 \tag{8}$$

Algorithm 1 PDDP for VCMKL

Input $\mathcal{S},\nu,\gamma,\mathcal{K}=\{\kappa_1,\cdots,\kappa\}$
$\boldsymbol{\lambda}^1\leftarrow$ initLambda($\mathcal{S}$); $\{\mathcal{Q},\mathcal{C}\}\leftarrow$ calMatrix($\mathcal{S},\nu,\gamma,\mathcal{K}$)
$CR_{explored}\leftarrow\emptyset$, $n\leftarrow itermax$, $\tau\leftarrow stepsize$, $t\leftarrow 1$
while $Improved$ & $t\leq n$ **do**
  if $\boldsymbol{\lambda}^t\in\mathcal{R}$ **then**
    $\{\boldsymbol{\alpha}^*,\mathcal{A}_1^M,\mathcal{B}\}\leftarrow$ solOPT2($\mathcal{Q},\mathcal{C}$)
    $CR_{new}\leftarrow$ getRegion($\mathcal{Q},\mathcal{C},\mathcal{A}_1^M,\mathcal{B}$) % (5)-(8)
    $CR_{explored}\leftarrow CR_{explored}\cup CR_{new}$
  else
    $\boldsymbol{\alpha}^*\leftarrow$ alphaInsideR($\boldsymbol{\lambda}^t,\mathcal{A}_1^M,\mathcal{B}$) % theorem 2
  end if
  $\boldsymbol{g}\leftarrow$ getGrad($\boldsymbol{\alpha}^*,\mathcal{A}_1^M,\mathcal{B}$) % (9)
  $\boldsymbol{\lambda}^{t+1}\leftarrow$ Proj($\boldsymbol{\lambda}^t-\tau\boldsymbol{g}$); $t\leftarrow t+1$
end while
return $\boldsymbol{\alpha}^*$,$\boldsymbol{\lambda}$

an explicit form and there is no need to solve OPT2 again. However, when the variable goes to a new critical region, a QP solver for OPT2 has to be invoked again for optimal solution and sample partition. The overall PDDP for VCMKL is summarized in Algorithm 1. The following convergence result to the global optimal is a consequence of theorem 3:

Theorem 4. *Convergence Guarantee*
Let $\sup_{\boldsymbol{\lambda}}||\boldsymbol{\lambda}^1-\boldsymbol{\lambda}||=B$, *and the Lipschitz constant of* $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}))$ *be* G, *then algorithm I with iteration* n *and optimal step size* $\tau_i=B/G\sqrt{n}\ \ \forall i$ *converges to global optimum within* $O\left(1/\sqrt{n}\right)$. *To be specific, let* $\mathcal{O}^*$ *be the global optimum of the learning objective (Dual form), then*

$$\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}^n_{best}))-\mathcal{O}^*\leq\frac{BG}{\sqrt{n}},\qquad \textit{where}$$

$$\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}^n_{best}))\triangleq\min\left\{\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}^1)),\cdots,\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}^n))\right\}$$

Hence in order to get $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}^n_{best}))-\mathcal{O}^*\leq\epsilon$, the algorithm needs $O(1/\epsilon^2)$ iterations. B is bounded because the feasible set of $\boldsymbol{\lambda}$ is simplexes. Also as $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\lambda}))$ is globally convex (hence continuous) and locally quadratic (hence has bounded gradient), G must be bounded as well. The constant step size is optimal in the sense that it minimizes the gap upper bound. Other choices, such as a diminishing step size that is square summable but not summable, could also be used if faster convergence is a concern.

The most computational expensive step is the inner QP solver. However note that since OPT2 shares a similar structure as the classic SVM dual, variety of existing methods could be reused for acceleration (Chang and Lin 2011). In the functions getRegion(), alphaInsiderRegion(), and getGrad(), the computational overhead is mostly matrix inversions of $\boldsymbol{Q},\boldsymbol{G},\boldsymbol{U}$. Fortunately, from the proof of the theorem 2 they are either symmetric positive definite or symmetric negative definite matrices and various of decomposition methods are thusly available for efficient inversion.

Table 1: Comparison of algorithms

Method	PDDP	AM	SGD	CCCP
Objective (10^{-8})	**6.247**	10.21	13.55	8.784
Testing Accuracy (%)	**78.91**	70.17	67.54	71.38
# of Iterations	79	**17**	237	31
Elapsed Time (s)	417	166	**34**	226

5 Experiment

5.1 PDDP Optimization Performance

Firstly we compare the performance of the proposed PDDP with existing optimization methods, including alternating minimization (Dundar et al. 2008), stochastic gradient descent (Kantchelian et al. 2014), and concave-convex procedure (Yu and Joachims 2009). The public UCI (Lichman 2013) pima data set is used in this experiment, and the learning problem OPT1 is solved with different algorithms. Since the other methods can only be applied to the primal, for comparison purpose we restrict to all linear kernels. Hyperparameters are set with $M=5$, $\nu_m=0.02\ \forall m$ and $\gamma=|I^-|/l$ [2]. For PDDP initialization a simple K-mean is applied to the negative class and $\boldsymbol{\lambda}^1$ is assigned according to cluster labels. The final value of the objective function, the corresponding testing accuracy, number of iterations and the time consumed are shown in Table 1, for which $|\mathcal{J}_{t+1}-\mathcal{J}_t|/\mathcal{J}_t\leq 10^{-4}$ is chosen as the stopping criteria. We observer that PDDP achieves a much better objective value, about 28.9% lower than the runner-up CCCP. The improved training also leads to 7.53% increase in testing accuracy. As a global optimization algorithm, it is expected that PDDP consumes more time than algorithms that only converge to local minimums, as is shown in the last row of the table. One possible acceleration is the design of approximate "larger region" that can reduce the number of invocations of the quadratic solver.

Figure 2 shows the corresponding objective value, gradient norm, $\boldsymbol{\lambda}$ value, and testing accuracy of PDDP in each iteration. Note that for clear presentation only a subset of $\boldsymbol{\lambda}$ with same initial value is shown. We see that the iterative

[2] Here the choice of these hyperparameters is not a concern, as the primary goal is to compare optimization algorithms.

Table 2: Testing Cost = $c_1 \cdot FP \cdot P(y=-1) + c_2 \cdot FN \cdot P(y=+1)$ and their rankings

Data Set	c_2:c_1	l-VCML	rbf-VCML	csrbf-SVM	2LMKL	Lasso-LR	NN	AdaBoost
Robot	1:1	0.197 (5)	**0.144** (1)	0.176 (3)	0.169 (2)	0.210 (7)	0.189 (4)	0.206 (6)
	1:4	0.289 (2)	**0.245** (1)	0.315 (4)	0.314 (3)	0.337 (5)	0.366 (7)	0.352 (6)
Music	1:1	0.277 (4)	**0.229** (1)	0.250 (2)	0.264 (3)	0.317 (7)	0.297 (5)	0.315 (6)
	1:4	0.443 (4)	**0.330** (1)	0.410 (3)	0.398 (2)	0.514 (5)	0.519 (6)	0.556 (7)
Vetebral	1:1	0.131 (2)	**0.121** (1)	0.140 (4)	0.133 (3)	0.168 (7)	0.152 (5)	0.157 (6)
	1:4	0.214 (4)	**0.156** (1)	0.218 (5)	0.211 (3)	0.239 (6)	0.192 (2)	0.332 (7)
Vowel	1:1	0.132 (5)	**0.025** (1)	0.113 (3)	0.122 (4)	0.159 (7)	0.111 (2)	0.142 (6)
	1:4	0.184 (3)	**0.051** (1)	0.181 (2)	0.199 (4)	0.276 (6)	0.228 (5)	0.293 (7)

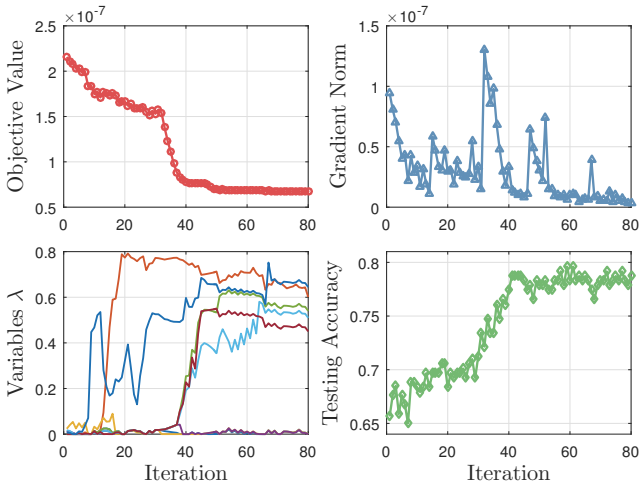

Figure 2: PDDP Convergence Results

result is similar to the general subgradient descent for non-smooth convex functions (Alber, Iusem, and Solodov 1998): The learning objective is in general decreasing, with some fluctuations due to non-smoothness between two critical regions. The pattern of gradient norm and $\boldsymbol{\lambda}$ evolution also reflects this character of insider region "exploitation" and beyond region "exploration". Note that although the algorithm takes 79 steps to converge, only 38 calls of the quadratic solver is involved, as in the critical region that has been explored before, explicit computation can be done with theorem 2. The testing accuracy in each iteration is shown at the bottom right of Figure 2, which increases correspondingly with the decrease of learning objective.

5.2 VCMKL Classification Performance

We test VCMKL with the PDDP solver on various data set. To begin with, two data sets (i.e. the UCI robot execution and vowel) are used to demonstrate the effect of number of kernels M on classification performance. Note that labels of the raw data are transformed into a binary case, e.g. for the robot data we set $y = +1$ if the label is NM, and all the other samples with label LP1-LP4 are pooled together as $y = -1$. Similar for the vowel data, $y = +1$ if the label is 'hOd' or 'hod', and $y = -1$ for the rest of 8 classes. Figure 3 shows the testing accuracy versus the number of kernels with 3 commonly used kernel families (linear, polynomial with different orders and RBF with various σ). Interestingly, in all cases it is seen that the testing accuracy is improved in the first few steps as M increases, however, further combining more kernels does not help, or even lead to a degraded performance due to overfitting (for linear and nonlinear kernels, respectively). In particular, the saturated point for linear kernels is just the number of hidden subgroups, i.e. $M = 4$ for robot data and $M = 8$ for vowel data, while for nonlinear kernels the optimal M is smaller as in the transformed space the subgroups may merge. This observation not only justifies the veto-consensus intuition to describe hidden subgroups, but also is consistent with the implications of theorem 1, which provides additive upper bound and does not encourage the use of many kernels.

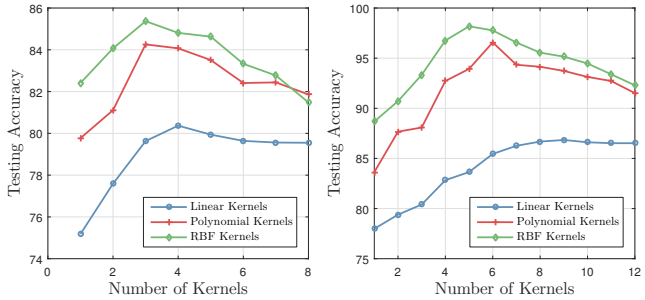

Figure 3: Testing Accuracy vs. M number of kernels. Left:Robot, Right:Vowel

Finally, VCMKL with linear and RBF kernels is compared to other methods, including cost sensitive SVM with RBF kernel, Two-layer MKL (Zhuang, Tsang, and Hoi 2011), Lasso Logistic Regression, multi-layer Neural Network, and AdaBoost. We compare a cost sensitive loss with c_1 and c_2 the cost rate for false positive (FP) and false negative (FN), respectively. Accordingly we set $\gamma = c_1|I^-|/(c_1|I^-| + c_2|I^+|)$. As is suggested by preceding discussion, an incremental CV is applied for selecting M. The other hyperparameters for all methods are chosen with 10 folds CV. Due to page limits, only the results for 4 data sets with hidden subgroups are shown in Table 2. Looking at the testing error ($c_1 = c_2 = 1$), it is seen that rbf-VCMKL outperforms all the other methods. Especially on the vowel data set the improvement is quite significant (testing accuracy from 88.9% to 97.5%). The performance of linear VCMKL is somewhere among those existing methods (average rank $\simeq$ 4). When the cost of false positive is higher

$(c_1 = 4, c_2 = 1)$, the performance improvement in using VCMKL is even more significant, as can be seen in the second row for each experiment. This justifies our intuition that the veto-consensus construction is sensitive to false positive while is robust to false negative, making it a promising tool for cost sensitive domain description problems.

6 Conclusion and Discussion

We study the veto-consensus combination of multiple kernels for classification. Our contributions are three folds (1) An upper bound on the Rademacher complexity of the veto-consensus multi-kernel classifier is provided. Previous bound obtained for polyhedron classifiers (Zhou, Jin, and Spanos 2015) can be viewed as a special case with only linear kernels. (2) A large margin multi-ν learning objective is formulated, which has provable training error bound in terms of hyperparameters. (3) A new global optimization algorithm, namly PDDP, is established. Sufficient conditions and new techniques for parametric program have been derived along the way. Although PDDP is developed in the context of VCMKL, it is also adaptable for a class of learning problems with similar structures.

Experimental results demonstrate the effectiveness of the proposed approach. Future work consists of (1) Apply PDDP to other learning problems such as hidden structured SVM, robust SVM, and piecewise linear function learning. (2) Obtain data dependent convergence bound for PDDP, in particular detailed bound for the Lipschtiz constant. (3) Design accelerated algorithm that uses enlarged approximate critical regions to reduce invocations of quadratic solver. The key issue is the trade off between region approximation and convexity of the dual optimality.

7 Appendix

The proof of Theorem I is given here. The proofs for PDDP could be found in supplementary material.

Proof. For the first part, we need the following lemma

Lemma 1. *Talagrand's Lemma*
Let $\Phi : \mathbb{R} \mapsto \mathbb{R}$ be $\eta-$ Lipschitz, and $\Upsilon : \mathbb{R} \mapsto \mathbb{R}$ be convex and nondecreasing. Then for any function class $\mathcal{F}$ of real-valued functions, the following inequality holds:

$$\hat{\mathcal{R}}(\Upsilon \circ \Phi \circ \mathcal{F}(\boldsymbol{x}_1^l)) \leq \eta \hat{\mathcal{R}}(\Upsilon \circ \mathcal{F}(\boldsymbol{x}_1^l))$$

Now for the main theorem, consider the case $M = 2$. The following inequality is straightforward:

$$\begin{aligned}&\sup_{g\in\mathcal{G}} \left| \frac{1}{l} \sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right| \\ &\leq \left[\sup_{g\in\mathcal{G}} \frac{1}{l} \sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right]_+ + \left[\sup_{g\in\mathcal{G}} \frac{1}{l} \sum_{i=1}^{l} -\sigma_i g(\boldsymbol{x}_i) \right]_+\end{aligned}$$

Noticing that $-\sigma_1, \cdots, -\sigma_l$ has the same distribution as $\sigma_1, \cdots, \sigma_l$, we get

$$\begin{aligned}\hat{\mathcal{R}}\left(\mathcal{G}(\boldsymbol{x}_1^l)\right) &= E_{\boldsymbol{\sigma}} \left[\sup_{g\in\mathcal{G}} \left| \frac{1}{l} \sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right| \right] \\ &\leq 2E_{\boldsymbol{\sigma}} \left[\sup_{g\in\mathcal{G}} \frac{1}{l} \sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right]_+\end{aligned}$$

Writing $g = \min\{f_1, f_2\} = \frac{1}{2}(f_1 + f_2) - \frac{1}{2}|f_1 - f_2|$, the last term yields

$$\begin{aligned}&\left[\sup_{g\in\mathcal{G}} \frac{1}{l} \sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right]_+ \\ &\overset{(a)}{\leq} \left[\sup_{\mathcal{F}_1,\mathcal{F}_1} \frac{1}{l} \sum_{i=1}^{l} \sigma_i \frac{1}{2}(f_1(\boldsymbol{x}_i) + f_2(\boldsymbol{x}_i)) \right]_+ \\ &\quad + \left[\sup_{\mathcal{F}_1,\mathcal{F}_1} \frac{1}{l} \sum_{i=1}^{l} -\sigma_i \frac{1}{2}|f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)| \right]_+ \\ &\overset{(b)}{\leq} \frac{1}{2} \left[\sup_{\mathcal{F}_1} \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i) \right]_+ + \frac{1}{2} \left[\sup_{\mathcal{F}_2} \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i) \right]_+ \\ &\quad + \left[\sup_{\mathcal{F}_1,\mathcal{F}_1} \frac{1}{l} \sum_{i=1}^{l} -\sigma_i \frac{1}{2}|f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)| \right]_+\end{aligned}$$

where (a) and (b) are due to the upper additive property of sup and $[\cdot] = max\{0, \cdot\}$ function. Taking expectations for this upper bound and applying Talagrand's Lemma with $\Upsilon = [\cdot]$ and $\Phi = |\cdot|$ yields

$$\begin{aligned}&E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_1,\mathcal{F}_1} \frac{1}{l} \sum_{i=1}^{l} -\sigma_i \frac{1}{2}|f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)| \right]_+ \\ &\leq \frac{1}{2} E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_1} \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i) \right]_+ + \frac{1}{2} E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_2} \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i) \right]_+\end{aligned}$$

Putting two inequalities together we have

$$\begin{aligned}&E_{\boldsymbol{\sigma}} \left[\sup_{g\in\mathcal{G}} \frac{1}{l} \sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right]_+ \\ &\leq \frac{1}{2} E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_1} \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i) \right]_+ + \frac{1}{2} E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_2} \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i) \right]_+ \\ &\quad + \frac{1}{2} E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_1} \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i) \right]_+ + \frac{1}{2} E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_2} \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i) \right]_+ \\ &\leq E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_1} \left| \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i) \right| \right] + E_{\boldsymbol{\sigma}} \left[\sup_{\mathcal{F}_2} \left| \frac{1}{l} \sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i) \right| \right] \\ &= \hat{\mathcal{R}}\left(\mathcal{F}_1(\boldsymbol{x}_1^l)\right) + \hat{\mathcal{R}}\left(\mathcal{F}_2(\boldsymbol{x}_1^l)\right)\end{aligned}$$

hence finally,

$$\hat{\mathcal{R}}\left(\mathcal{G}(\boldsymbol{x}_1^l)\right) \leq 2\left[\hat{\mathcal{R}}\left(\mathcal{F}_1(\boldsymbol{x}_1^l)\right) + \hat{\mathcal{R}}\left(\mathcal{F}_2(\boldsymbol{x}_1^l)\right)\right]$$

Also it is straightforward to generalize the above argument to $M > 2$ with simple induction. Finally we get

$$\hat{\mathcal{R}}(\mathcal{G}\left(\boldsymbol{x}_1^l\right)) \leq 2 \sum_{j=1}^{M} \hat{\mathcal{R}}\left(\mathcal{F}_j(\boldsymbol{x}_1^l)\right)$$

The second part of the theorem can be obtained with a standard approach in bounding empirical Rademacher complexity of Kernels. □

8 Acknowledgments

This research is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for

the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

References

Alber, Y. I.; Iusem, A. N.; and Solodov, M. V. 1998. On the projected subgradient method for nonsmooth convex optimization in a hilbert space. *Mathematical Programming* 81(1):23–35.

Bach, F. 2009. High-dimensional non-linear variable selection through hierarchical kernel learning. *arXiv preprint arXiv:0909.0844*.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2009. Learning non-linear combinations of kernels. In *Advances in neural information processing systems*, 396–404.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2010. Generalization bounds for learning kernels. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 247–254.

Dundar, M. M.; Wolf, M.; Lakare, S.; Salganicoff, M.; and Raykar, V. C. 2008. Polyhedral classifier for target detection: a case study: colorectal cancer. In *Proceedings of the 25th international conference on Machine learning*, 288–295. ACM.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32(9):1627–1645.

Floudas, C. A. 1995. Nonlinear and mixed-integer optimization: Fundamentals and applications. Oxford University Press.

Hussain, Z., and Shawe-Taylor, J. 2011. Improved loss bounds for multiple kernel learning. In *International Conference on Artificial Intelligence and Statistics*, 370–377.

Jagarlapudi, S. N.; Dinesh, G.; Raman, S.; Bhattacharyya, C.; Ben-Tal, A.; and Kr, R. 2009. On the algorithmics and applications of a mixed-norm based kernel learning formulation. In *Advances in neural information processing systems*, 844–852.

Kantchelian, A.; Tschantz, M. C.; Huang, L.; Bartlett, P. L.; Joseph, A. D.; and Tygar, J. 2014. Large-margin convex polytope machine. In *Advances in Neural Information Processing Systems*, 3248–3256.

Kloft, M.; Brefeld, U.; Sonnenburg, S.; and Zien, A. 2011. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research* 12:953–997.

Lanckriet, G. R.; Cristianini, N.; Bartlett, P.; Ghaoui, L. E.; and Jordan, M. I. 2004. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research* 5:27–72.

Lichman, M. 2013. UCI machine learning repository.

Liu, X.; Wang, L.; Zhang, J.; and Yin, J. 2014. Sample-adaptive multiple kernel learning. *Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14)*.

Liu, X.; Wang, L.; Yin, J.; Dou, Y.; and Zhang, J. 2015. Absent multiple kernel learning. *Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Manwani, N., and Sastry, P. 2010. Learning polyhedral classifiers using logistic function. In *ACML*, 17–30.

Manwani, N., and Sastry, P. 2011. Polyceptron: A polyhedral learning algorithm. *arXiv preprint arXiv:1107.1564*.

Ong, C. S.; Williamson, R. C.; and Smola, A. J. 2005. Learning the kernel with hyperkernels. In *Journal of Machine Learning Research*, 1043–1071.

Rakotomamonjy, A.; Bach, F.; Canu, S.; and Grandvalet, Y. 2008. Simplemkl. *Journal of Machine Learning Research* 9:2491–2521.

Scholkopf B, Smola AJ, W. R. B. P. 2000. New support vector algorithms. *Advances In Neural Information Processing Systems (NIPS)*.

Stephen Boyd, L. V. 2004. Convex optimization. ISBN:0521833787. Cambridge University Press New York.

Xu, L.; Crammer, K.; and Schuurmans, D. 2006. Robust support vector machine training via convex outlier ablation. In *AAAI*, volume 6, 536–542.

Ying, Y., and Campbell, C. 2009. Generalization bounds for learning the kernel. In *22nd Annual Conference on Learning Theory (COLT 2009)*.

Yu, C.-N. J., and Joachims, T. 2009. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 1169–1176. ACM.

Zhou, Y.; Jin, B.; and Spanos, C. 2015. Learning convex piecewise linear machine for data-driven optimal control. In *International conference on Machine Learning and Applications (ICMLA)*. IEEE.

Zhou, Y.; Li, D.; and Spanos, C. 2015. Learning optimization friendly comfort model for hvac model predictive control. In *International conference on Data Mining (ICDMW 2015)*. IEEE.

Zhuang, J.; Tsang, I. W.; and Hoi, S. 2011. Two-layer multiple kernel learning. In *International Conference on Artificial Intelligence and Statistics*, 909–917.