JMLR: Workshop and Conference Proceedings 44 (2015) 72-89 NIPS 2015

The 1st International Workshop "Feature Extraction: Modern Questions and Challenges"

Kernel Extraction via Voted Risk Minimization

Corinna Cortes CORINNA@GOOGLE.COM
Google Research
111 8th Avenue, New York, NY 10011

Prasoon Goyal PGOYAL@NYU.EDU
Courant Institute of Mathematical Sciences
251 Mercer Street, New York, NY 10012

Vitaly Kuznetsov VITALY@CIMS.NYU.EDU
Courant Institute of Mathematical Sciences
251 Mercer Street, New York, NY 10012

Mehryar Mohri MOHRI@CS.NYU.EDU
Courant Institute and Google Research
251 Mercer Street, New York, NY 10012

Editor: Sanjiv Kumar

Abstract

This paper studies a new framework for learning a predictor in the presence of multiple kernel functions where the learner selects or extracts *several* kernel functions from potentially complex families and finds an accurate predictor defined in terms of these functions. We present an algorithm, *Voted Kernel Regularization*, that provides the flexibility of using very complex kernel functions such as predictors based on high-degree polynomial kernels or narrow Gaussian kernels, while benefitting from strong learning guarantees. We show that our algorithm benefits from strong learning guarantees suggesting a new regularization penalty depending on the Rademacher complexities of the families of kernel functions used. Our algorithm admits several other favorable properties: its optimization problem is convex, it allows for learning with non-PDS kernels, and the solutions are highly sparse, resulting in improved classification speed and memory requirements. We report the results of some preliminary experiments comparing the performance of our algorithm to several baselines.

Keywords: feature extraction, kernel methods, learning theory, Rademacher complexity

1. Introduction

Feature extraction is key to the success of machine learning. With a poor choice of features, learning can become arbitrarily difficult, while, a favorable choice can help even an unsophisticated algorithm succeed. In recent years, a number of methods have been proposed to reduce the requirement from the user to select features by seeking instead to automate the feature extraction process. These include unsupervised dimensionality reduction techniques (Roweis and Saul, 2000; Tenenbaum et al., 2000; Belkin and Niyogi, 2003), supervised embedding techniques (Hyvärinen and Oja, 2000; Mika et al., 1999; Mohri et al., 2015), and metric learning methods (Weinberger et al., 2006; Bar-Hillel et al., 2003; Goldberger et al., 2004).

For kernel-based algorithms, the problem of feature selection is substituted with that of selecting appropriate kernels. In the multiple kernel learning (MKL) framework, the learning algorithm is presented with a labeled sample and a family of kernel functions, typically a convex combination of base kernels, and the problem consists of using the sample to both extract the relevant kernel weights and learn a predictor based on that kernel (Lanckriet et al., 2004; Argyriou et al., 2005, 2006; Srebro and Ben-David, 2006; Lewis et al., 2006; Zien and Ong, 2007; Micchelli and Pontil, 2005; Jebara, 2004; Bach, 2008; Ong et al., 2005; Cortes et al., 2010, 2013).

This paper studies an alternative framework for learning a predictor in the presence of multiple kernel functions where the learner selects or extracts *several* kernel functions and finds an accurate predictor defined in terms of these functions. There are some key differences between the set-up we consider and that of MKL. For the particular case of a family of convex combinations $\sum_{k=1}^{p} \mu_k K_k$ based on p base kernels $K_1, \ldots, K_p$, $\mu_k \geq 0$, in MKL, the general form of the predictor solution f based a training sample $(x_1, \ldots, x_m)$ is $f = \sum_{i=1}^{m} \alpha_i (\sum_{k=1}^{p} \mu_k K_k(x_i, \cdot)) = \sum_{i=1}^{m} \sum_{k=1}^{p} \alpha_i \mu_k K_k(x_i, \cdot)$, with $\alpha_i \in \mathbb{R}$. In contrast, the predictors we consider have the more general form $f = \sum_{i=1}^{m} \sum_{k=1}^{p} \alpha_{i,k} K_k(x_i, \cdot)$, with $\alpha_{i,k} \in \mathbb{R}$. Furthermore, we allow kernels to be selected from possibly very complex families, thanks to the use of capacity-conscient regularization. An approach similar to ours is that of Cortes et al. (2011), where for each base kernel a different predictor is used and where the predictors are then combined to define a single predictor, these two tasks being performed in a single stage or in two subsequent stages. The algorithm where the task is performed in a single stage bears the most resemblance with ours. However the regularization is different and, most importantly, not capacity-dependent. We will further emphasize these key points in Section 3 where our Voted Kernel Regularization algorithm is further discussed.

The hypothesis returned by learning algorithms such as SVMs (Cortes and Vapnik, 1995) and other algorithms for which the representer theorem holds is a linear combination of kernel feature functions $K(x, \cdot)$, where K is the kernel function used and x is a training sample. The generalization guarantees for SVMs depend on the sample size and the margin, but also on the complexity of the kernel function K used, measured by its trace (Koltchinskii and Panchenko, 2002). These guarantees suggest that, for a moderate margin, learning with very complex kernels, such as sums of polynomial kernels of degree up to some large d may lead to overfitting, which frequently is observed empirically. Thus, in practice, simpler kernels are typically used, that is small ds for sums of polynomial kernels. On the other hand, to achieve a sufficiently high performance in challenging learning tasks, it may be necessary to augment a linear combination of such functions $K(x, \cdot)$ with a function $K'(x, \cdot)$, where K' is possibly a substantially more complex kernel, such as a polynomial kernel of degree $d' \gg d$. This flexibility is not available when using SVMs or other learning algorithms such as kernel Perceptron (Aizerman et al., 1964; Rosenblatt, 1958) with the same solution form: either a complex kernel function K' is used and then there is a risk of overfitting, or a potentially too simple kernel K is used, thereby limiting the performance achievable in some tasks.

This paper presents an algorithm, Voted Kernel Regularization (VKR), that precisely provides the flexibility of using potentially very complex kernel functions such as predictors based on much higher-degree polynomial kernels, while benefitting from strong learning guarantees. VKR simultaneously selects (multiple) base kernel functions and learns a discriminative model based on these functions. We show that our algorithm benefits from strong data-dependent learning bounds that are expressed in terms of the Rademacher complexities of the reproducing kernel Hilbert spaces (RHKS) of the kernel functions used. These results are based on the framework of *Voted Risk Minimization* originally introduced by Cortes et al. (2014) for ensemble methods. We further ex-

tend their results, using a local Rademacher complexity analysis, to show that faster convergence rates are possible when the spectrum of the kernel matrix is controlled. The regularization terms of our algorithm is directly based on the Rademacher complexities of the families already mentioned and therefore benefits from the data-dependent properties of these quantities. We give an extensive analysis of these complexity penalties in the case of kernel families commonly used.

Besides the theoretical guarantees, VKR admits a number of additional favorable properties. Our formulation leads to a convex optimization problem that can be solved either via Linear Programming or Coordinate Descent. VKR does not require the kernel functions to be positive-definite or even symmetric. This enables the use of much richer families of kernel functions. In particular, some standard distances known not to be PSD such as the edit-distance can be used with VKR.

Yet another advantage of our algorithm is that it learns highly sparse feature representations providing greater efficiency and less memory needs. In that respect, VKR is similar to so-called *norm-1 SVM* (Vapnik, 1998; Zhu et al., 2003) and *Any-Norm-SVM* (Dekel and Singer, 2007) which all use a norm-penalty to reduce the number of support vectors. However, to the best of our knowledge these regularization terms on their own has not led to performance improvement over regular SVMs (Zhu et al., 2003; Dekel and Singer, 2007). In contrast, our preliminary experimental results show that VKR can outperform both regular SVM and norm-1 SVM, and at the same time significantly reduce the number of support vectors. In some other work, hybrid regularization schemes are combined to obtain a performance improvement (Zou, 2007). Possibly this technique could be applied to our VKR algorithm as well resulting in additional performance improvements.

The rest of the paper is organized as follows. Some preliminary definitions and notation are introduced in Section 2. The VKR algorithm is presented in Section 3. In Section 4, we show that it benefits from strong data-dependent learning guarantees, including when using highly complex kernel families. In Section 4, we also prove local complexity bounds that detail how faster convergence rates are possible provided that the spectrum of the kernel matrix is controlled. Section 5 discusses the implementation of the VKR algorithm, including optimization procedures and a novel theoretical analysis of the Rademacher complexities of relevant kernel families. We conclude with some early experimental results in Section 7 and Appendix D, which we hope to complete in the future with a more extensive analysis, including large-scale experiments.

2. Preliminaries

Let $\mathcal{X}$ denote the input space. We consider the familiar supervised learning scenario. We assume that training and test points are drawn i.i.d. according to some distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, +1\}$ and denote by $S = ((x_1, y_1), \ldots, (x_m, y_m))$ a training sample of size m drawn according to $\mathcal{D}^m$.

Let $\rho > 0$. For a function f taking values in $\mathbb{R}$, we denote by $R(f)$ its binary classification error, by $\widehat{R}_S(f)$ its empirical error, and by $\widehat{R}_{S,\rho}(f)$ its empirical margin error for the sample S:

$$R(f) = \mathop{\mathrm{E}}_{(x,y)\sim\mathcal{D}}[1_{yf(x)\leq 0}], \quad \widehat{R}_S(f) = \mathop{\mathrm{E}}_{(x,y)\sim S}[1_{yf(x)\leq 0}], \text{ and } \widehat{R}_\rho(f) = \mathop{\mathrm{E}}_{(x,y)\sim S}[1_{yf(x)\leq \rho}],$$

where the notation $(x, y) \sim S$ indicates that (x, y) is drawn according to the empirical distribution defined by S. We will denote by $\widehat{\mathfrak{R}}_S(H)$ the empirical Rademacher complexity of a hypothesis set H on the set S of functions mapping $\mathcal{X}$ to $\mathbb{R}$, and by $\mathfrak{R}_m(H)$ the Rademacher complexity

(Koltchinskii and Panchenko, 2002; Bartlett and Mendelson, 2002):

$$\widehat{\mathfrak{R}}_S(H) = \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(x_i) \right] \qquad \mathfrak{R}_m(H) = \mathop{\mathrm{E}}_{S \sim D^m} \left[\widehat{\mathfrak{R}}_S(H) \right],$$

where the random variables σ_i are independent and uniformly distributed over $\{-1, +1\}$.

3. Voted Kernel Regularization Algorithm

In this section, we present the VKR algorithm. Let $K_1, \ldots, K_p$ be p positive semi-definite (PSD) kernel functions with $\kappa_k = \sup_{x \in \mathcal{X}} \sqrt{K_k(x,x)}$ for all $k \in [1,p]$. We consider p corresponding families of functions mapping from $\mathcal{X}$ to $\mathbb{R}$, $H_1, \ldots, H_p$, defined by $H_k = \{x \mapsto \pm K_k(x,x') \colon x' \in \mathcal{X}\}$, where the sign accounts for two possible ways of classifying a point $x' \in \mathcal{X}$. The general form of a hypothesis f returned by the algorithm is the following:

$$f = \sum_{j=1}^{m} \sum_{k=1}^{p} \alpha_{k,j} K_k(\cdot, x_j),$$

where $\alpha_{k,j} \in \mathbb{R}$ for all j and k. Thus, f is a linear combination of hypotheses in H_ks. This form with many αs per point is distinct from that of MKL solutions which admit only one α per point. Since the families H_k are symmetric, this linear combination can be made a non-negative combination. Our algorithm consists of minimizing the Hinge loss on the training sample, as with SVMs, but with a different regularization term that tends to penalize hypotheses drawn from more complex H_ks more than those selected from simpler ones and to minimize the norm-1 of the coefficients $\alpha_{k,j}$. Let r_k denote the empirical Rademacher complexity of H_k: $r_k = \widehat{\mathfrak{R}}_S(H_k)$. Then, the following is the objective function of VKR:

$$F(\boldsymbol{\alpha}) = \frac{1}{m} \sum_{i=1}^{m} \max\left(0, 1 - y_i \sum_{j=1}^{m} \sum_{k=1}^{p} \alpha_{k,j} y_j K_k(x_i, x_j)\right) + \sum_{j=1}^{m} \sum_{k=1}^{p} (\lambda r_k + \beta) |\alpha_{k,j}|, \qquad (1)$$

where $\lambda \geq 0$ and $\beta \geq 0$ are parameters of the algorithm. We will adopt the notation $\Lambda_k = \lambda r_k + \beta$ to simplify the presentation in what follows.

Note that the objective function F is convex: the Hinge loss is convex, thus, its composition with an affine function is also convex, which shows that the first term is convex; the second term is convex as the absolute value terms with non-negative coefficients; and F is convex as the sum of these two convex terms. Thus, the optimization problem admits a global minimum. VKR returns the function f defined by (3) with coefficients $\boldsymbol{\alpha} = (\alpha_{k,j})_{k,j}$ minimizing F.

This formulation admits several benefits. First, it enables us to learn with very complex hypothesis sets and yet, as we will see later, benefit from strong learning guarantees, thanks to the Rademacher complexity-based penalties assigned to coefficients associated to different H_ks. Notice further that the penalties assigned are data-dependent, which is a key feature of the algorithm. Second, observe that the objective function (7) does not require the kernels K_k to be positive-definite or even symmetric. Function F is convex regardless of the kernel properties. This is a significant benefit of the algorithm which enables us to extend its use beyond what algorithms such as SVMs require. In particular, some standard distances known not to be PSD such as the edit-distance and many others could be used with this algorithm. Another advantage of this algorithm compared to

standard SVM and other ℓ_2-regularized methods is that ℓ_1-norm regularization used for VKR leads to sparse solutions. The solution $\boldsymbol{\alpha}$ is typically sparse, which significantly reduces prediction time and the memory needs.

Note that hypotheses $h \in H_k$ are defined by $h(x) = K_k(x, x')$ where x' is an arbitrary element of the input space $\mathcal{X}$. However, our objective only includes those x_j that belong to the observed sample. For PDS kernels, this does not cause any loss of generality. Indeed, observe that for $x' \in \mathcal{X}$ we can write $\Phi_k(x') = \mathbf{w} + \mathbf{w}^\perp$, where Φ_k is a feature map associated with the kernel K_k and where $\mathbf{w}$ lies in the span of $\Phi_k(x_1), \ldots, \Phi_k(x_m)$ and $\mathbf{w}^\perp$ is in orthogonal compliment of this subspace. Therefore, for any sample point x_i

$$\begin{aligned} K_k(x_i, x') &= \langle \Phi_k(x_i), \Phi_k(x') \rangle_{\mathcal{H}_k} = \langle \Phi_k(x_i), \mathbf{w} \rangle_{\mathcal{H}_k} + \langle \Phi_k(x_i), \mathbf{w}^\perp \rangle_{\mathcal{H}_k} \\ &= \sum_{j=1}^{m} \beta_j \langle \Phi_k(x_i), \Phi_k(x_j) \rangle_{\mathcal{H}_k} = \sum_{j=1}^{m} \beta_j K_k(x_i, x_j), \end{aligned}$$

which leads to objective (1). Note that selecting $-K_k(\cdot, x_j)$ with weight $\alpha_{k,j}$ is equivalent to selecting $K_k(\cdot, x_j)$ with $(-\alpha_{k,j})$, which accounts for the absolute value on the $\alpha_{k,j}$s.

The VKR algorithm has some connections with other algorithms previously described in the literature. In the absence of any regularization, that is $\lambda = 0$ and $\beta = 0$, it reduces to the minimization of the Hinge loss and is therefore close to the SVM algorithm (Cortes and Vapnik, 1995). For $\lambda = 0$, that is when discarding our regularization based on the different complexity of the hypothesis sets, the algorithm coincides with an algorithm originally described by Vapnik (1998)[pp. 426-427], later by several other authors starting with (Zhu et al., 2003), and often referred to as the norm-1 SVM.

4. Learning Guarantees

In this section, we provide strong data-dependent learning guarantees for the VKR algorithm.

Let $\mathcal{F}$ denote $\operatorname{conv}(\bigcup_{k=1}^{p} H_k)$, that is the family of functions f of the form $f = \sum_{t=1}^{T} \alpha_t h_t$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_T)$ is in the simplex Δ and where, for each $t \in [1, T]$, H_{k_t} denotes the hypothesis set containing h_t, for some $k_t \in [1, p]$. Then, the following learning guarantee holds.

Theorem 1 ((Cortes et al., 2014)) *Assume $p > 1$. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m drawn i.i.d. according to $\mathcal{D}^m$, the following inequality holds for all $f = \sum_{t=1}^{T} \alpha_t h_t \in \mathcal{F}$:*

$$R(f) \leq \widehat{R}_{S,\rho}(f) + \frac{4}{\rho} \sum_{t=1}^{T} \alpha_t \mathfrak{R}_m(H_{k_t}) + \frac{2}{\rho} \sqrt{\frac{\log p}{m}} + \sqrt{\left\lceil \frac{4}{\rho^2} \log \left\lceil \frac{\rho^2 m}{\log p} \right\rceil \right\rceil \frac{\log p}{m} + \frac{\log \frac{2}{\delta}}{2m}}.$$

Theorem 1 can be used to derive the VKR objective. We provide the full details of that derivation in Appendix B. Theorem 1 can be further improved using a local Rademacher complexity analysis showing that faster rates of convergence are possible.

Theorem 2 *Assume $p > 1$. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m drawn i.i.d. according to $\mathcal{D}^m$, the following inequality holds for all*

$f = \sum_{t=1}^{T} \alpha_t h_t \in \mathcal{F}$ and for any $K > 1$:

$$R(f) - \frac{K}{K-1}\widehat{R}_{S,\rho}(f) \leq 6K\frac{1}{\rho}\sum_{t=1}^{T}\alpha_t \mathfrak{R}_m(H_{k_t}) + 40\frac{K}{\rho^2}\frac{\log p}{m} + 5K\frac{\log\frac{2}{\delta}}{m} + 5K\left\lceil \frac{8}{\rho^2}\log\frac{\rho^2(1+\frac{K}{K-1})m}{40K\log p}\right\rceil\frac{\log p}{m}.$$

The proof of this result is given in Appendix A. Note that $O(\log m/\sqrt{m})$ in Theorem 1 is replaced with $O(\log m/m)$ in Theorem 2. For full hypothesis classes H_ks, $\mathfrak{R}_m(H_k)$ may be in $O(1/\sqrt{m})$ and thus dominate the convergence of the bound. However, if we use localized classes $H_k(r) = \{h \in H_k \colon \mathrm{E}[h^2] < r\}$, then, for certain values of r^*, the local Rademacher complexities $\mathfrak{R}_m(H_k(r^*))$ are in $O(1/m)$ which yields even stronger learning guarantees. Furthermore, this result leads to an extension of VKR objective:

$$F(\boldsymbol{\alpha}) = \frac{1}{m}\sum_{i=1}^{m}\max\left(0, 1 - \sum_{j=1}^{m}\sum_{k=1}^{p}\alpha_{k,j}y_i y_j K_k(x_i, x_j)\right) + \sum_{j=1}^{m}\sum_{k=1}^{p}(\lambda\mathfrak{R}_m(H_k(s)) + \beta)|\alpha_{k,j}|, \quad (2)$$

which is optimized over $\boldsymbol{\alpha}$ with the parameter s selected via cross-validation. In Section 6, we provide an explicit expression for the local Rademacher complexities of PDS kernel functions.

5. Optimization Solutions

We have derived and implemented two different algorithmic solutions for solving the optimization problem (1): a linear programming (LP) that we will briefly describe here and a coordinate descent (CD) approach described in Appendix C which enables us to learn with a very large number of base hypotheses.

Observe that by introducing slack variables ξ_i the optimization problem (1) can be equivalently written as follows:

$$\min_{\boldsymbol{\alpha},\xi} \quad \frac{1}{m}\sum_{i=1}^{m}\xi_i + \sum_{j=1}^{m}\sum_{k=1}^{p}\Lambda_k|\alpha_{k,j}| \quad \text{s.t.} \quad \xi_i \geq 1 - \sum_{j=1}^{m}\sum_{k=1}^{p}\alpha_{k,j}y_i y_j K_k(x_i, x_j), \forall i \in [1, m].$$

Next, we introduce new variables $\alpha^+_{k,j} \geq 0$ and $\alpha^-_{k,j} \geq 0$ such that $\alpha_{k,j} = \alpha^+_{k,j} - \alpha^-_{k,j}$. Then, for any k and j, $|\alpha_{k,j}|$ can be rewritten as $\alpha^+_{k,j} + \alpha^-_{k,j}$. The optimization problem is equivalent to the following:

$$\begin{aligned}\min_{\boldsymbol{\alpha}^+\geq 0, \boldsymbol{\alpha}^-\geq 0, \xi} \quad & \frac{1}{m}\sum_{i=1}^{m}\xi_i + \sum_{j=1}^{m}\sum_{k=1}^{p}\Lambda_k(\alpha^+_{k,j} + \alpha^-_{k,j}) \\ \text{s.t.} \quad & \xi_i \geq 1 - \sum_{j=1}^{m}\sum_{k=1}^{p}(\alpha^+_{k,j} - \alpha^-_{k,j})y_i y_j K_k(x_i, x_j), \forall i \in [1, m],\end{aligned}$$

since, conversely, a solution with $\alpha_{k,j} = \alpha^+_{k,j} - \alpha^-_{k,j}$ verifies the condition $\alpha^+_{k,j} = 0$ or $\alpha^-_{k,j} = 0$ for any k and j, thus $\alpha_{k,j} = \alpha^+_{k,j}$ when $\alpha_{k,j} \geq 0$ and $\alpha_{k,j} = \alpha^-_{k,j}$ when $\alpha_{k,j} \leq 0$. This is because if

$\delta = \min(\alpha^+_{k,j}, \alpha^-_{k,j}) > 0$, then replacing $\alpha^+_{k,j}$ with $\alpha^+_{k,j} - \delta$ and $\alpha^-_{k,j}$ with $\alpha^-_{k,j} - \delta$ would not affect $\alpha^+_{k,j} - \alpha^-_{k,j}$ but would reduce $\alpha^+_{k,j} + \alpha^-_{k,j}$.

Note that the resulting optimization problem is an LP problem since the objective function is linear in both ξ_is and $\boldsymbol{\alpha}^+$, $\boldsymbol{\alpha}^-$, and since the constraints are affine. There is a battery of well-established methods to solve this LP problem including interior-point methods and the simplex algorithm. An additional advantage of this formulation of the VKR algorithm is that there is a large number of generic software packages for solving LPs making the VKR algorithm easier to implement.

6. Complexity penalties

An additional benefit of the learning bounds presented in Section 4 is that they are data-dependent. They are based on the Rademacher complexities r_ks of the base hypothesis sets H_k, which can be accurately estimated from the training sample. Our formulation directly inherits this advantage. However, in some cases, computing these estimates may be very costly. In this section, we derive instead several upper bounds on these complexities that be can readily used in an efficient implementation of the VKR algorithm.

Note that the hypothesis set $H_k = \{x \mapsto \pm K_k(x, x') \colon x' \in \mathcal{X}\}$ is of course distinct from the RKHS $\mathcal{H}_k$ of the kernel K_k. Thus, we cannot use known upper bounds on $\widehat{\mathfrak{R}}_S(\mathcal{H}_k)$ to bound $\widehat{\mathfrak{R}}_S(H_k)$. Observe that $\widehat{\mathfrak{R}}_S(\mathcal{H}_k)$ can be expressed as follows:

$$\widehat{\mathfrak{R}}_S(H_k) = \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{x' \in \mathcal{X}, s \in \{-1,+1\}} \sum_{i=1}^m \sigma_i s K_k(x_i, x') \right] = \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{x' \in \mathcal{X}} \left| \sum_{i=1}^m \sigma_i K_k(x_i, x') \right| \right] \tag{3}$$

The following lemma gives an upper bound depending on the trace of the kernel matrix $\mathbf{K}_k$.

Lemma 3 (Trace bound) *Let $\mathbf{K}_k$ be the kernel matrix of the PDS kernel function K_k for the sample S and let $\kappa_k = \sup_{x \in \mathcal{X}} \sqrt{K_k(x, x)}$. Then, the following inequality holds: $\widehat{\mathfrak{R}}_S(H_k) \leq \frac{\kappa_k \sqrt{\mathrm{Tr}[\mathbf{K}_k]}}{m}$.*

Proof By (3) and the Cauchy-Schwarz inequality, we can write

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(H_k) &= \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{x' \in \mathcal{X}} \left| \sum_{i=1}^m \sigma_i K_k(x_i, x') \right| \right] = \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{x' \in \mathcal{X}} \left| \sum_{i=1}^m \sigma_i \Phi_k(x_i) \cdot \Phi_k(x') \right| \right] \\
&\leq \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{x' \in \mathcal{X}} \|\Phi_k(x')\|_{\mathcal{H}_k} \left\| \sum_{i=1}^m \sigma_i \Phi_k(x_i) \right\|_{\mathcal{H}_k} \right] = \frac{\kappa_k}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^m \sigma_i \Phi_k(x_i) \right\|_{\mathcal{H}_k} \right].
\end{aligned}$$

By Jensen's inequality, the following inequality holds:

$$\mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^m \sigma_i \Phi_k(x_i) \right\|_{\mathcal{H}_k} \right] \leq \sqrt{\mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sum_{i,j=1}^m \sigma_i \sigma_j \Phi_k(x_i) \cdot \Phi_k(x_j) \right]} = \sqrt{\mathrm{Tr}[\mathbf{K}_k]},$$

which concludes the proof. ■

The expression given by the lemma can be precomputed and used as the parameter r_k of the optimization procedure. However, the upper bound just derived is not fine enough to distinguish

between different normalized kernels since for any normalized kernel K_k, $\kappa = 1$ and $\mathrm{Tr}[\mathbf{K}_k] = m$. In that case, finer bounds in terms of localized complexities can be used. In particular, the local Rademacher complexity of a set of functions H is defined as $\mathfrak{R}_m^{\mathrm{loc}}(H, r) = \mathfrak{R}_m(\{h \in H \colon \mathrm{E}[h^2] \leq r\})$. If $(\lambda_i)_{i=1}^{\infty}$ is a sequence of eigenvalues associated with the kernel K_k then one can show (Mendelson, 2003; Bartlett et al., 2005) that for every $r > 0$, the following inequality holds: $\mathfrak{R}_m^{\mathrm{loc}}(H, r) \leq \sqrt{\frac{2}{m} \min_{\theta \geq 0} \left(\theta r + \sum_{j > \theta} \lambda_j\right)} = \sqrt{\frac{2}{m} \sum_{j=1}^{\infty} \min(r, \lambda_j)}$. Furthermore, there is an absolute constant c such that if $\lambda_1 \geq \frac{1}{m}$, then for every $r \geq \frac{1}{m}$, $\frac{c}{\sqrt{m}} \sum_{j=1}^{\infty} \min(r, \lambda_j) \leq \mathfrak{R}_m^{\mathrm{loc}}(H, r)$. Note that the choice $r = \infty$ recovers the earlier bound $\mathfrak{R}_m(H_k) \leq \sqrt{\mathrm{Tr}[\mathbf{K}_k]/m}$. On the other hand, one can show that for instance in the case of Gaussian kernels $\mathfrak{R}_m^{\mathrm{loc}}(H, r) = O(\sqrt{\frac{r}{m} \log(1/r)})$ and using the fixed point of this function leads to $\mathfrak{R}_m^{\mathrm{loc}}(H, r) = O(\frac{\log m}{m})$. These results can be used in conjunction with the local Rademacher complexity extension of VKR discussed in Section 4.

If all of the kernels belong to the same family such as, for example, polynomial or Gaussian kernels it may be desirable to use measures of complexity that account for specific properties of the given family of kernels such a the polynomial degree or the bandwidth of a Gaussian kernel. Below we present alternative upper bounds that precisely address these questions.

For instance, if K_k is a polynomial kernel of degree k, then we can use an upper bound on the Rademacher complexity of H_k in terms of the square-root of its pseudo-dimension $\mathrm{Pdim}(H_k)$, which coincides with the dimension d_k of the feature space corresponding to a polynomial kernel of degree k, which is given by

$$d_k = \binom{N+k}{k} \leq \frac{(N+k)^k}{k!} \leq \left(\frac{(N+k)e}{k}\right)^k. \tag{4}$$

Lemma 4 (Polynomial kernels) *Let K_k be a polynomial kernel of degree k. Then, the empirical Rademacher complexity of H_k can be upper bounded as $\widehat{\mathfrak{R}}_S(H_k) \leq 12 \kappa_k^2 \sqrt{\frac{\pi d_k}{m}}$.*

Proof By the proof of Lemma 3, we can write

$$\widehat{\mathfrak{R}}_S(H_k) \leq \frac{\kappa_k}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \Phi_k(x_i) \right\|_{\mathcal{H}_k} \right] = 2\kappa_k^2 \, \widehat{\mathfrak{R}}_S(H_k^1),$$

where H_k^1 is the family of linear functions $H_k^1 = \{\mathbf{w} \mapsto \mathbf{w} \cdot \Phi_k(x) \colon \|\mathbf{w}\|_{\mathcal{H}_k} \leq \frac{1}{2\kappa_k}\}$. By Dudley's formula (Dudley, 1989), we can write

$$\widehat{\mathfrak{R}}_S(H_k^1) \leq 12 \int_0^{\infty} \sqrt{\frac{\log \mathcal{N}(\epsilon, H_k^1, L_2(\widehat{\mathcal{D}}))}{m}} d\epsilon,$$

where $\widehat{\mathcal{D}}$ is the empirical distribution. Since H_k^1 can be viewed as a subset of a d_k-dimensional linear space and since $|\mathbf{w} \cdot \Phi_k(x)| \leq \frac{1}{2}$ for all $x \in \mathcal{X}$ and $w \in H_k^1$, we have $\log \mathcal{N}(\epsilon, H_k^1, L_2(\widehat{\mathcal{D}})) \leq \log \left[(\frac{1}{\epsilon})^{d_k}\right]$. Thus, we can write

$$\widehat{\mathfrak{R}}_S(H_k^1) \leq 12 \int_0^1 \sqrt{\frac{d_k \log \frac{1}{\epsilon}}{m}} d\epsilon = 12 \sqrt{\frac{d_k}{m}} \int_0^1 \sqrt{\log \frac{1}{\epsilon}} d\epsilon = 12 \sqrt{\frac{d_k}{m}} \frac{\sqrt{\pi}}{2},$$

which completes the proof. ■

Thus, in view of the lemma, we can use $r_k = \kappa_k^2\sqrt{d_k}$ as a complexity penalty in the formulation of the VKR algorithm with polynomial kernels, with d_k given by (4).

Another family of kernels that is commonly used in applications is Gaussian kernels $H_\gamma = \{x \mapsto \pm\exp(-\gamma\|x - x'\|_2^2) \colon x' \in \mathcal{X}\}$. Our next result provides a bound on the Rademacher complexity of this family of kernels in terms of the parameter γ.

Lemma 5 (Gaussian kernels) *The empirical Rademacher complexity of H_γ can be bounded as follows: $\widehat{\mathfrak{R}}_S(H_\gamma) \leq \gamma\widehat{\mathfrak{R}}_S(\{x \mapsto \|x - x'\|_2^2\})$.*

Proof Observe that the function $z \mapsto \exp(-\gamma z)$ is γ-Lipschitz for $z \geq 0$ since the absolute value of its derivative, $|-\gamma\exp(-\gamma z)|$ is bounded by γ. Thus, by (3) and Talagrand's contraction principle (Ledoux and Talagrand, 1991), the following holds:

$$\widehat{\mathfrak{R}}_S(H_\gamma) = \frac{1}{m}\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\sup_{x' \in \mathcal{X}}\left|\sum_{i=1}^m \sigma_i e^{-\gamma\|x_i - x'\|_2^2}\right|\right] \leq \frac{\gamma}{m}\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\sup_{x' \in \mathcal{X}}\left|\sum_{i=1}^m \sigma_i\|x_i - x'\|_2^2\right|\right].$$

which concludes the proof. ■

Note that while we could in fact bound $\mathfrak{R}_S(\{x \mapsto \|x - x'\|_2^2\})$, we do not need to find its expression explicitly since it does not vary with γ. Thus, in view of the lemma, we can use $r_k = \gamma_k$ as a complexity penalty in the formulation of the VKR algorithm with Gaussian kernels defined by parameters $\gamma_1, \ldots, \gamma_p$. Talagrand's contraction principle helps us derive similar bounds for other families of kernels including those that are not PDS. In particular, a similar proof using the Lipschitzness of $\tanh$ shows the following result for sigmoid kernels.

Lemma 6 (Sigmoid kernels) *Let $H_{a,b} = \{x \mapsto \pm\tanh(ax \cdot x' + b) \colon x' \in \mathcal{X}\}$ with $a, b \in \mathbb{R}$. Then, the following bound holds: $\widehat{\mathfrak{R}}_S(H_{a,b}) \leq 4|a|\widehat{\mathfrak{R}}_S(\{x \mapsto x \cdot x'\})$.*

7. Experiments

Here, we report the results of some preliminary experiments with several benchmark datasets from the UCI repository: `breastcancer`, `climate`, `diabetes`, `german(numeric)`, `ionosphere`, `musk`, `ocr49`, `phishing`, `retinopathy`, `vertebral` and `waveform01`. The notation `ocr49` refers to the subset of the OCR dataset with classes 4 and 9, and similarly `waveform01` refers to the subset of `waveform` dataset with classes 0 and 1. See Appendix E for details.

Our experiments compared VKR to regular SVM, that we refer to as L_2-SVM, and to norm-1 SVM, called L_1-SVM. In all of our experiments, we used `lp_solve`, an off-the-shelf LP solver, to solve the VKR and L_1-SVM optimization problems. For L_2-SVM, we used `LibSVM`.

In each of the experiments, we used standard 5-fold cross-validation for performance evaluation and model selection. In particular, each dataset was randomly partitioned into 5 folds, and each algorithm was run 5 times, with a different assignment of folds to the training set, validation set and test set for each run. Specifically, for each $i \in \{0, \ldots, 4\}$, fold i was used for testing, fold $i + 1 \pmod 5$ was used for validation, and the remaining folds were used for training. For each setting of the parameters, we computed the average validation error across the 5 folds, and selected the parameter setting with minimum average validation error. The average test error across the 5 folds was then computed for this particular parameter setting.

	Error (%)								Number of support vectors							
Dataset	L2 SVM		L1 SVM		VKRT		VKRD		L2 SVM		L1 SVM		VKRT		VKRD	
	Mean	(Stdev)	Mean	(Stdev)	Mean	(Stdev)	Mean	(Stdev)	Mean	(Stdev)	Mean	(Stdev)	Mean	(Stdev)	Mean	(Stdev)
ocr49	5.05	(0.65)	3.50	(0.85)	**2.70**	(0.97)	**3.50**	(0.85)	449.8	(3.6)	140.0	(3.6)	6.8	(1.3)	164.6	(9.5)
phishing	4.64	(1.38)	4.11	(0.71)	***3.62***	(0.44)	***3.87***	(0.80)	221.4	(15.1)	188.8	(7.5)	73.0	(3.2)	251.8	(4.0)
waveform01	8.38	(0.63)	8.47	(0.52)	8.41	(0.97)	8.57	(0.58)	415.6	(8.1)	13.6	(1.3)	18.4	(1.5)	14.6	(2.3)
breastcancer	11.45	(0.74)	12.60	(2.88)	11.73	(2.73)	11.30	(1.31)	83.8	(10.9)	46.4	(2.4)	66.6	(3.9)	29.4	(1.9)
german	23.00	(3.00)	22.40	(2.58)	24.10	(2.99)	24.20	(2.61)	357.2	(16.7)	34.4	(2.2)	25.0	(1.4)	30.2	(2.3)
ionosphere	6.54	(3.07)	7.12	(3.18)	***4.27***	(2.00)	***3.99***	(2.12)	152.0	(5.5)	73.8	(4.9)	43.6	(2.9)	30.6	(1.8)
pima	31.90	(1.17)	30.85	(1.54)	31.77	(2.68)	***30.73***	(1.46)	330.0	(6.6)	26.4	(0.6)	33.8	(3.6)	40.6	(1.1)
musk	15.34	(2.23)	11.55	(1.49)	**10.71**	(1.13)	**9.03**	(1.39)	251.8	(12.4)	115.4	(4.5)	125.6	(8.0)	108.0	(5.2)
retinopathy	24.58	(2.28)	24.85	(2.65)	25.46	(2.08)	24.06	(2.43)	648.2	(21.3)	42.6	(3.7)	43.6	(4.0)	48.0	(3.1)
climate	5.19	(2.41)	5.93	(2.83)	5.56	(2.85)	6.30	(2.89)	66.0	(4.6)	19.0	(0.0)	51.0	(6.7)	18.6	(0.9)
vertebral	17.74	(6.35)	18.06	(5.51)	17.10	(7.27)	17.10	(6.99)	75.4	(4.0)	4.4	(0.6)	9.6	(1.1)	8.2	(1.3)

Table 1: Experimental results with VKR and polynomial kernels. VKRT and VKRD refer to the algorithms obtained by using for the complexity terms the trace bound (Lemma 3) or the polynomial degree bound (Lemma 4) respectively. Boldfaced results are statistically significant at a 5% confidence level, boldfaced and in italics are better at a 10% level, both in comparison to L_2-SVM.

In the first set of experiments we used polynomial kernels of the form $K_k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^k$. We report the results in Table 1. For VKR, we optimized over $\lambda \in \{10^{-i} \colon i = 0, \ldots, 8\}$ and $\beta \in \{10^{-i} \colon i = 0, \ldots, 8\}$. The family of kernel functions H_k for $k \in [1, 10]$ was chosen to be the set of polynomial kernels of degree k. In our experiments we compared the bounds of both Lemma 3 and Lemma 4 used as an estimate of the Rademacher complexity. For L_1-SVM, we cross-validated over degrees in range 1 through 10 and β in the same range as for VKR. Cross-validation for L_2-SVM was also done over the degree and regularization parameter $C \in \{10^i \colon i = -4, \ldots, 7\}$.

On 5 out of 11 datasets VKR outperformed L_2-SVM with a considerable improvement on 2 data sets. On the rest of the datasets, there was no statistical difference between these algorithms. Similar improvements are seen over L_1-SVM. Observe that the solutions obtained by VKR are often up to 10 times sparser than those of L_2-SVM. In other words, VKR admits the benefit of sparse solutions and often an improved performance, which provides empirical evidence in support of our formulation. The second set of experiments with Gaussian kernels is presented in Appendix D, where we report a significant improvement for 3 datasets over a different baseline, L2-SVM with uniform Gaussian kernel.

8. Conclusion

We presented a new support vector algorithm, Voted Kernel Regularization, that simultaneously selects (multiple) base kernel functions and learns an accurate hypothesis based on these functions. Our algorithm benefits from strong data-dependent guarantees for learning with complex kernels. We further improved these learning guarantees using a local complexity analysis leading to an extension of VKR algorithm. The key ingredient of our algorithm is a new regularization term that makes use of the Rademacher complexities of different families of kernel functions used by the VKR algorithm. We gave a thorough analysis of several alternatives that can be used for this approximation. We also described two practical implementations of our algorithm based on linear programming and coordinate descent. Finally, we reported the results of preliminary experiments showing that our algorithm always finds highly sparse solutions and that it can outperform other formulations.

ACKNOWLEDGMENTS

This work was partly funded by NSF IIS-1117591 and CCF-1535987, and the NSERC PGS D3.

References

Mark A. Aizerman, E. M. Braverman, and Lev I. Rozonoèr. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

Andreas Argyriou, Charles Micchelli, and Massimiliano Pontil. Learning convex combinations of continuously parameterized basic kernels. In *COLT*, 2005.

Andreas Argyriou, Raphael Hauser, Charles Micchelli, and Massimiliano Pontil. A DC-programming algorithm for kernel selection. In *ICML*, 2006.

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. *NIPS 2009*, 2008.

Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *Proceedings of the ICML*, pages 11–18, 2003.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *JMLR*, 3, 2002.

Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Localized rademacher complexities. In *COLT*, 2002.

Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Local Rademacher complexities. *Ann. Stat.*, 33(4):1497–1537, 2005.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, June 2003.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Generalization bounds for learning kernels. In *ICML*, 2010.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Ensembles of kernel predictors. In *UAI*, 2011.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13:795–828, 2012.

Corinna Cortes, Marius Kloft, and Mehryar Mohri. Learning kernels using local rademacher complexity. In *In NIPS*, pages 2760–2768, 2013.

Corinna Cortes, Mehryar Mohri, and Umar Syed. Deep boosting. In *ICML*, pages 1179 – 1187, 2014.

Ofer Dekel and Yoram Singer. Support vector machines on a budget. In *NIPS*, 2007.

R. M. Dudley. *Real Analysis and Probability*. Wadsworth, Belmont, CA, 1989.

Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, 2004.

A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Netw.*, 13(4-5):411–430, May 2000.

Tony Jebara. Multi-task feature and kernel selection for SVMs. In *ICML*, 2004.

Vladmir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30, 2002.

Gert Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5, 2004.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, 1991.

Darrin P. Lewis, Tony Jebara, and William Stafford Noble. Nonstationary kernel combination. In *ICML*, 2006.

Shahar Mendelson. On the performance of kernel classes. *J. Mach. Learn. Res.*, 4:759–771, 2003.

Charles Micchelli and Massimiliano Pontil. Learning the kernel function via regularization. *JMLR*, 6, 2005.

Sebastian Mika, Bernhard Schlkopf, Alex Smola, Klaus-Robert Mller, Matthias Scholz, and Gunnar Rtsch. Kernel pca and de-noising in feature spaces. In *NIPS 11*, pages 536–542. MIT Press, 1999.

Mehryar Mohri, Afshin Rostamizadeh, and Dmitry Storcheus. Foundations of coupled nonlinear dimensionality reduction. *arXiv preprint arXiv:1509.08880v2*, 2015.

Cheng Soon Ong, Alex Smola, and Robert Williamson. Learning the kernel with hyperkernels. *JMLR*, 6, 2005.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

Nathan Srebro and Shai Ben-David. Learning bounds for support vector machines with learned kernels. In *COLT*, 2006.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.

Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

K.Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS 18*. MIT Press, Cambridge, MA, 2006.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Robert Tibshirani. 1-norm support vector machines. In *NIPS*, pages 49–56, 2003.

Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In *ICML 2007*, 2007.

Hui Zou. An improved 1-norm svm for simultaneous classification and variable selection. In *AISTATS*, volume 2, pages 675–681. Citeseer, 2007.

Appendix A. Proof of Theorem 2

Proof For a fixed $\mathbf{h} = (h_1, \ldots, h_T)$, any $\boldsymbol{\alpha} \in \Delta$ defines a distribution over $\{h_1, \ldots, h_T\}$. Sampling from $\{h_1, \ldots, h_T\}$ according to $\boldsymbol{\alpha}$ and averaging leads to functions g of the form $g = \frac{1}{n}\sum_{i=1}^{T} n_t h_t$ for some $\mathbf{n} = (n_1, \ldots, n_T)$, with $\sum_{t=1}^{T} n_t = n$, and $h_t \in H_{k_t}$.

For any $\mathbf{N} = (N_1, \ldots, N_p)$ with $|\mathbf{N}| = n$, we consider the family of functions

$$G_{\mathcal{F},\mathbf{N}} = \left\{ \frac{1}{n} \sum_{k=1}^{p} \sum_{j=1}^{N_k} h_{k,j} \,|\, \forall (k,j) \in [p] \times [N_k], h_{k,j} \in H_k \right\},$$

and the union of all such families $G_{\mathcal{F},n} = \bigcup_{|\mathbf{N}|=n} G_{\mathcal{F},\mathbf{N}}$. Fix $\rho > 0$. We define a class $\Phi \circ G_{\mathcal{F},\mathbf{N}} = \{\Phi_\rho(g) \colon g \in G_{\mathcal{F},\mathbf{N}}\}$ and $\mathcal{G}_r = \mathcal{G}_{\Phi,\mathcal{F},\mathbf{N},r} = \{r\ell_g / \max(r, \mathrm{E}[\ell_g]) \colon \ell_g \in \Phi \circ G_{\mathcal{F},\mathbf{N}}\}$ for r to be chosen later. Observe that for $v_g \in \mathcal{G}_{\Phi,\mathcal{F},\mathbf{N},r}$ $\mathrm{Var}[v_g] \leq r$. Indeed, if $r > \mathrm{E}[\ell_g]$ then $v_g = \ell_g$. Otherwise, $\mathrm{Var}[v_g] = r^2 \mathrm{Var}[\ell_g]/(\mathrm{E}[\ell_g])^2 \leq r(\mathrm{E}[\ell_g^2])/\mathrm{E}[\ell_g] \leq r$.

By Theorem 2.1 in Bartlett et al. (2005), for any $\delta > 0$ with probability at least $1 - \delta$, for any $0 < \beta < 1$,

$$V \leq 2(1+\beta)\mathfrak{R}_m(\mathcal{G}_{\Phi,\mathcal{F},\mathbf{N},r}) + \sqrt{\frac{2r \log \frac{1}{\delta}}{m}} + \left(\frac{1}{3} + \frac{1}{\beta}\right)\frac{\log \frac{1}{\delta}}{m},$$

where $V = \sup_{v \in \mathcal{G}_r} (\mathrm{E}[v] - \mathrm{E}_n[v])$ and β is a free parameter. Next we observe that $\mathfrak{R}_m(\mathcal{G}_{\Phi,\mathcal{F},\mathbf{N},r}) \leq \mathfrak{R}_m(\{\alpha\ell_g \colon g \in \Phi \circ G_{\mathcal{F},\mathbf{N}}, \alpha \in [0,1]\}) = \mathfrak{R}_m(\Phi \circ G_{\mathcal{F},\mathbf{N}})$. Therefore, using Talagrand's contraction lemma and convexity we have that $\mathfrak{R}_m(\mathcal{G}_{\Phi,\mathcal{F},\mathbf{N},r}) \leq \frac{1}{\rho}\sum_{k=1}^{p} \frac{N_k}{n}\mathfrak{R}_m(H_k)$. It follows that, for any $\delta > 0$, with probability at least $1 - \delta$, for all $0 < \beta < 1$, the following holds:

$$V \leq 2(1+\beta)\frac{1}{\rho}\sum_{k=1}^{p} \frac{N_k}{n}\mathfrak{R}_m(H_k) + \sqrt{\frac{2r \log \frac{1}{\delta}}{m}} + \left(\frac{1}{3} + \frac{1}{\beta}\right)\frac{\log \frac{1}{\delta}}{m}.$$

Since there are at most p^n possible p-tuples $\mathbf{N}$ with $|\mathbf{N}| = n$, by the union bound, for any $\delta > 0$, with probability at least $1 - \delta$,

$$V \leq 2(1+\beta)\frac{1}{\rho}\sum_{k=1}^{p} \frac{N_k}{n}\mathfrak{R}_m(H_k) + \sqrt{\frac{r \log \frac{p^n}{\delta}}{m}} + \left(\frac{1}{3} + \frac{1}{\beta}\right)\frac{\log \frac{p^n}{\delta}}{m}.$$

Thus, with probability at least $1-\delta$, for all functions $g = \frac{1}{n}\sum_{i=1}^{T} n_t h_t$ with $h_t \in H_{k_t}$, the following inequality holds

$$V \leq 2(1+\beta)\frac{1}{\rho}\sum_{t=1}^{T} \frac{n_t}{n}\mathfrak{R}_m(H_{k_t}) + \sqrt{\frac{r \log \frac{p^n}{\delta}}{m}} + \left(\frac{1}{3} + \frac{1}{\beta}\right)\frac{\log \frac{p^n}{\delta}}{m}.$$

Taking the expectation with respect to $\boldsymbol{\alpha}$ and using $\mathrm{E}_{\boldsymbol{\alpha}}[n_t/n] = \alpha_t$, we obtain that for any $\delta > 0$, with probability at least $1 - \delta$, for all $\mathbf{h}$, we can write

$$\mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[V] \leq 2(1+\beta)\frac{1}{\rho}\sum_{t=1}^{T}\alpha_t \mathfrak{R}_m(H_{k_t}) + \sqrt{\frac{r\log\frac{p^n}{\delta}}{m}} + \Big(\frac{1}{3}+\frac{1}{\beta}\Big)\frac{\log\frac{p^n}{\delta}}{m}.$$

We now show that r can be chosen in such a way that $\mathrm{E}_{\boldsymbol{\alpha}}[V] \leq r/K$. The right hand side of the above bound is of the form $A\sqrt{r} + B$. Note that solution of $r/K = C + A\sqrt{r}$ is bounded by $K^2A^2 + 2KC$ and hence by Lemma 5 in (Bartlett et al., 2002) the following bound holds

$$\mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[R_{\rho/2}(g) - \frac{K}{K-1}\widehat{R}_{S,\rho}(g)] \leq 4K(1+\beta)\frac{1}{\rho}\sum_{t=1}^{T}\alpha_t \mathfrak{R}_m(H_{k_t}) + \Big(2K^2 + 2K\Big(\frac{1}{3}+\frac{1}{\beta}\Big)\Big)\frac{\log\frac{1}{\delta}}{m}.$$

Set $\beta = 1/2$, then we have that

$$\mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[R_{\rho/2}(g) - \frac{K}{K-1}\widehat{R}_{S,\rho}(g)] \leq 6K\frac{1}{\rho}\sum_{t=1}^{T}\alpha_t \mathfrak{R}_m(H_{k_t}) + 5K\frac{\log\frac{1}{\delta}}{m}.$$

Then, for any $\delta_n > 0$, with probability at least $1 - \delta_n$,

$$\mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[R_{\rho/2}(g) - \frac{K}{K-1}\widehat{R}_{S,\rho}(g)] \leq 6K\frac{1}{\rho}\sum_{t=1}^{T}\alpha_t \mathfrak{R}_m(H_{k_t}) + 5K\frac{\log\frac{p^n}{\delta_n}}{m}.$$

Choose $\delta_n = \frac{\delta}{2p^{n-1}}$ for some $\delta > 0$, then for $p \geq 2$, $\sum_{n\geq 1}\delta_n = \frac{\delta}{2(1-1/p)} \leq \delta$. Thus, for any $\delta > 0$ and any $n \geq 1$, with probability at least $1 - \delta$, the following holds for all $\mathbf{h}$:

$$\mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[R_{\rho/2}(g) - \frac{K}{K-1}\widehat{R}_{S,\rho}(g)] \leq 6K\frac{1}{\rho}\sum_{t=1}^{T}\alpha_t \mathfrak{R}_m(H_{k_t}) + 5K\frac{\log\frac{2p^{2n-1}}{\delta}}{m}. \tag{5}$$

Now, for any $f = \sum_{t=1}^{T}\alpha_t h_t \in \mathcal{F}$ and any $g = \frac{1}{n}\sum_{i=1}^{T} n_t h_t$, we can upper bound $R(f) = \Pr_{(x,y)\sim\mathcal{D}}[yf(x) \leq 0]$, the generalization error of f, as follows:

$$\begin{aligned} R(f) = \Pr_{(x,y)\sim\mathcal{D}}[yf(x) - yg(x) + yg(x) \leq 0] &\leq \Pr[yf(x) - yg(x) < -\rho/2] + \Pr[yg(x) \leq \rho/2] \\ &= \Pr[yf(x) - yg(x) < -\rho/2] + R_{\rho/2}(g). \end{aligned}$$

We can also write

$$\widehat{R}_\rho(g) = \widehat{R}_{S,\rho}(g - f + f) \leq \widehat{\Pr}[yg(x) - yf(x) < -\rho/2] + \widehat{R}_{S,3\rho/2}(f).$$

Combining these inequalities yields

$$\begin{aligned} &\Pr_{(x,y)\sim\mathcal{D}}[yf(x) \leq 0] - \frac{K}{K-1}\widehat{R}_{S,3\rho/2}(f) \leq \Pr[yf(x) - yg(x) < -\rho/2] \\ &+ \frac{K}{K-1}\widehat{\Pr}[yg(x) - yf(x) < -\rho/2] + R_{\rho/2}(g) - \frac{K}{K-1}\widehat{R}_{S,\rho}(g). \end{aligned}$$

Taking the expectation with respect to $\boldsymbol{\alpha}$ yields

$$R(f) - \widehat{R}_{S,3\rho/2}(f) \leq \mathop{\mathrm{E}}_{x\sim\mathcal{D},\boldsymbol{\alpha}}[1_{yf(x)-yg(x)<-\rho/2}] + \frac{K}{K-1}\mathop{\mathrm{E}}_{x\sim\mathcal{D},\boldsymbol{\alpha}}[1_{yg(x)-yf(x)<-\rho/2}] + \mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[R_{\rho/2}(g) - \frac{K}{K-1}\widehat{R}_{S,\rho}(g)].$$

Since $f = \mathrm{E}_{\boldsymbol{\alpha}}[g]$, by Hoeffding's inequality, for any x,

$$\mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[1_{yf(x)-yg(x)<-\rho/2}] = \Pr_{\boldsymbol{\alpha}}[yf(x)-yg(x)<-\rho/2] \leq e^{-\frac{n\rho^2}{8}}$$
$$\mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[1_{yg(x)-yf(x)<-\rho/2}] = \Pr_{\boldsymbol{\alpha}}[yg(x)-yf(x)<-\rho/2] \leq e^{-\frac{n\rho^2}{8}}.$$

Thus, for any fixed $f \in \mathcal{F}$, we can write

$$R(f) - \widehat{R}_{S,3\rho/2}(f) \leq \Big(1 + \frac{K}{K-1}\Big)e^{-n\rho^2/8} + \mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[R_{\rho/2}(g) - \frac{K}{K-1}\widehat{R}_{S,\rho}(g)].$$

Thus, the following inequality holds:

$$\sup_{f\in\mathcal{F}}\Big(R(f) - \frac{K}{K-1}\widehat{R}_{S,\rho}(f)\Big) \leq \Big(1 + \frac{K}{K-1}\Big)e^{-n\rho^2/8} + \sup_{\mathbf{h}}\mathop{\mathrm{E}}_{\boldsymbol{\alpha}}[R_{\rho/2}(g) - \frac{K}{K-1}\widehat{R}_{S,\rho/2}(g)].$$

Therefore, in view of (5), for any $\delta > 0$ and any $n \geq 1$, with probability at least $1-\delta$, the following holds for all $f \in \mathcal{F}$:

$$R(f) - \frac{K}{K-1}\widehat{R}_{S,\rho}(f) \leq \Big(1 + \frac{K}{K-1}\Big)e^{-n\rho^2/8} + 6K\frac{1}{\rho}\sum_{t=1}^{T}\alpha_t\mathfrak{R}_m(H_{k_t}) + 5K\frac{\log\frac{2p^{2n-1}}{\delta}}{m}.$$

To conclude the proof, we optimize over n, $f\colon n \mapsto v_1e^{-nu}+v_2n$, which leads to $n = (1/u)\log(uv_2/v_1)$. Therefore, we set

$$n = \left\lceil \frac{8}{\rho^2}\log\frac{\rho^2(1+\frac{K}{K-1})m}{40K\log p}\right\rceil$$

to obtain the following bound:

$$R(f) - \frac{K}{K-1}\widehat{R}_{S,\rho}(f) \leq 6K\frac{1}{\rho}\sum_{t=1}^{T}\alpha_t\mathfrak{R}_m(H_{k_t}) + 40\frac{K}{\rho^2}\frac{\log p}{m} + 5K\frac{\log\frac{2}{\delta}}{m} + 5K\left\lceil\frac{8}{\rho^2}\log\frac{\rho^2(1+\frac{K}{K-1})m}{40K\log p}\right\rceil\frac{\log p}{m}.$$

Thus, taking $K = 2$, simply yields

$$R(f) \leq 2\widehat{R}_{S,\rho}(f) + \frac{12}{\rho}\sum_{t=1}^{T}\alpha_t\mathfrak{R}_m(H_{k_t}) + O\left(\frac{\log p}{\rho^2 m}\log\Big(\frac{\rho m}{\log p}\Big) + \frac{\log\frac{1}{\delta}}{m}\right)$$

and the proof is complete. ■

Appendix B. Optimization Problem

This section provides the derivation for the VKR optimization problem. We will assume that $H_1, \ldots, H_p$ are p families of functions with increasing Rademacher complexities $\mathfrak{R}_m(H_k)$, $k \in [1, p]$, and, for any hypothesis $h \in \cup_{k=1}^p H_k$, denote by $d(h)$ the index of the hypothesis set it belongs to, that is $h \in H_{d(h)}$. The bound of Theorem 1 holds uniformly for all $\rho > 0$ and functions $f \in \operatorname{conv}(\bigcup_{k=1}^p H_k)$ at the price of an additional term that is in $O\Big(\sqrt{\frac{\log\log \frac{2}{\rho}}{m}}\Big)$. The condition $\sum_{t=1}^T \alpha_t = 1$ of Theorem 1 can be relaxed to $\sum_{t=1}^T \alpha_t \leq 1$. To see this, use for example a null hypothesis ($h_t = 0$ for some t). Since the last term of the bound does not depend on $\boldsymbol{\alpha}$, it suggests selecting $\boldsymbol{\alpha}$ to minimize

$$G(\boldsymbol{\alpha}) = \frac{1}{m}\sum_{i=1}^m 1_{y_i \sum_{t=1}^T \alpha_t h_t(x_i) \leq \rho} + \frac{4}{\rho}\sum_{t=1}^T \alpha_t r_t,$$

where $r_t = \mathfrak{R}_m(H_{d(h_t)})$. Since for any $\rho > 0$, f and f/ρ admit the same generalization error, we can instead search for $\alpha \geq 0$ with $\sum_{t=1}^T \alpha_t \leq 1/\rho$ which leads to

$$\min_{\boldsymbol{\alpha} \geq 0} \frac{1}{m}\sum_{i=1}^m 1_{y_i \sum_{t=1}^T \alpha_t h_t(x_i) \leq 1} + 4\sum_{t=1}^T \alpha_t r_t \quad \text{s.t.} \quad \sum_{t=1}^T \alpha_t \leq \frac{1}{\rho}.$$

The first term of the objective is not a convex function of $\boldsymbol{\alpha}$ and its minimization is known to be computationally hard. Thus, we will consider instead a convex upper bound based on the Hinge loss: let $\Phi(-u) = \max(0, 1-u)$, then $1_{-u} \leq \Phi(-u)$. Using this upper bound yields the following convex optimization problem:

$$\min_{\boldsymbol{\alpha} \geq 0} \frac{1}{m}\sum_{i=1}^m \Phi\Big(1 - y_i \sum_{t=1}^T \alpha_t h_t(x_i)\Big) + \lambda \sum_{t=1}^T \alpha_t r_t \quad \text{s.t.} \quad \sum_{t=1}^T \alpha_t \leq \frac{1}{\rho}, \tag{6}$$

where we introduced a parameter $\lambda \geq 0$ controlling the balance between the magnitude of the values taken by function Φ and the second term. Introducing a Lagrange variable $\beta \geq 0$ associated to the constraint in (6), the problem can be equivalently written as

$$\min_{\boldsymbol{\alpha} \geq 0} \frac{1}{m}\sum_{i=1}^m \Phi\Big(1 - y_i \sum_{t=1}^T \alpha_t h_t(x_i)\Big) + \sum_{t=1}^T (\lambda r_t + \beta)\alpha_t.$$

Here, β is a parameter that can be freely selected by the algorithm since any choice of its value is equivalent to a choice of ρ in (6). Let $(h_{k,j})_{k,j}$ be the set of distinct base functions $x \mapsto K_k(\cdot, x_j)$. Then, the problem can be rewritten as F be the objective function based on that collection:

$$\min_{\boldsymbol{\alpha} \geq 0} \frac{1}{m}\sum_{i=1}^m \Phi\Big(1 - y_i \sum_{j=1}^N \alpha_j h_j(x_i)\Big) + \sum_{t=1}^N \Lambda_j \alpha_j, \tag{7}$$

with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^N$ and $\Lambda_j = \lambda r_j + \beta$, for all $j \in [1, N]$. This coincides precisely with the optimization problem $\min_{\boldsymbol{\alpha} \geq 0} F(\boldsymbol{\alpha})$ defining VKR. Since the problem was derived by minimizing a Hinge loss upper bound on the generalization bound, this shows that the solution returned by VKR benefits from the strong data-dependent learning guarantees of Theorem 1.

Appendix C. Coordinate Descent (CD) Formulation

An alternative approach for solving the VKR optimization problem (1) consists of using a coordinate descent method. A key advantage of this formulation over the LP formulation is that there is no need to explicitly store the whole vector of $\boldsymbol{\alpha}$s but rather only non-zero entries. This enables learning with a very large number of base hypotheses, including scenarios in which the number of base hypotheses is infinite.

A coordinate descent method proceeds in rounds. At each round, it maintains a parameter vector $\boldsymbol{\alpha}$. Let $\boldsymbol{\alpha}_t = (\alpha_{t,k,j})_{k,j}^\top$ denote the vector obtained after $t \geq 1$ iterations and let $\boldsymbol{\alpha}_0 = \mathbf{0}$. Let $\mathbf{e}_{k,j}$ denote the unit vector in direction (k,j) in $\mathbb{R}^{p \times m}$. Then, the direction $\mathbf{e}_{k,j}$ and the step η selected at the tth round are those minimizing $F(\boldsymbol{\alpha}_{t-1} + \eta \mathbf{e}_{k,j})$, that is

$$F(\boldsymbol{\alpha}) = \frac{1}{m} \sum_{i=1}^{m} \max\Big(0, 1 - y_i f_{t-1} - y_i y_j \eta K_k(x_i, x_j)\Big) + \sum_{j=1}^{m} \sum_{k=1}^{p} \Lambda_k |\alpha_{t-1,j,k}| + \Lambda_k |\eta + \alpha_{t-1,k,j}|,$$

where $f_{t-1} = \sum_{j=1}^{m} \sum_{k=1}^{p} \alpha_{t-1,j,k} y_j K_k(\cdot, x_j)$. To find the best descent direction, a coordinate descent method computes the sub-gradient in the direction (k,j) for each $(k,j) \in [1,p] \times [1,m]$. The sub-gradient is given by

$$\delta F(\alpha_{t-1}, \mathbf{e}_j) = \begin{cases} \frac{1}{m} \sum_{i=1}^{m} \phi_{t,j,k,i} + \operatorname{sgn}(\alpha_{t-1,k,j}) \Lambda_k & \text{if } \alpha_{t-1,k,j} \neq 0 \\ 0 & \text{else if} \left| \frac{1}{m} \sum_{i=1}^{m} \phi_{t,j,k,i} \right| \leq \Lambda_k \\ \frac{1}{m} \sum_{i=1}^{m} \phi_{t,j,k,i} - \operatorname{sgn}\left(\frac{1}{m} \sum_{i=1}^{m} \phi_{t,j,k,i} \right) \Lambda_k & \text{otherwise} , \end{cases}$$

where $\phi_{t,j,k,i} = -y_i K_k(x_i, x_j)$ if $\sum_{k=1}^{p} \sum_{j=1}^{m} \alpha_{t-1,k,j} y_i y_j K(x_i, x_j) < 1$ and 0 otherwise. Once the optimal direction $\mathbf{e}_{k,j}$ is determined, the step size η_t can be found using a line search or other numerical methods.

Appendix D. Additional Experiments

This section presents additional experiments with our VKR algorithm.

In these experiments, we used families of Gaussian kernels based on distinct values of the parameter γ. We used the bound of Lemma 5 as an estimate of the Rademacher complexity and we refer to the resulting algorithm as VKRG. We compare VKRG to a different baseline, L_2-SVM with the uniform kernel combination, which is referred to as L_2-SVM-uniform. It has been observed empirically that L_2-SVM-uniform often outperforms most existing MKL algorithms (Cortes et al., 2012).

In our experiments, both VKRG and L_2-SVM-uniform are given a fixed set of base kernels with $\gamma \in \{10^{i/2} \colon i = -4, \ldots, 4\}$. For L_2-SVM-uniform, the range of the regularization parameter was $C \in \{10^i \colon i = -5, \ldots, 7\}$. We used the same range for the λ and β parameters for VKRG as in our experiments with polynomial kernels presented in Section 7.

The results of our experiments are comparable to the results with polynomial kernels and are summarized in Table 2. Voted Kernel Regularization outperforms L_2-SVM-uniform on 9 out of 13 datasets with considerable improvement on 3 datasets, including two additional large-scale datasets (`ml-prove` and `white`). On the rest of the datasets, there was no statistical difference between these algorithms. Observe that solutions obtained by VKRG are often up to 10 times sparser than

Dataset	**Error(%)**				**Number of support vectors**			
	L_2-SVM-uniform		VKRG		L_2-SVM-uniform		VKRG	
	Mean	(Std)	Mean	(Std)	Mean	(Std)	Mean	(Std)
ocr49	2.85	(1.26)	3.45	(1.05)	710.2	(8.2)	160.4	(8.0)
phishing	3.50	(1.17)	3.50	(1.09)	506.0	(10.2)	172.8	(10.3)
waveform01	8.63	(0.56)	8.90	(0.87)	662.4	(8.7)	24.2	(1.9)
breastcancer	9.30	(1.04)	**8.44**	(1.30)	217.0	(4.9)	124.2	(5.9)
german	24.2	(2.77)	24.6	(3.42)	418.2	(12.5)	32.6	(2.3)
ionosphere	4.27	(1.73)	3.98	(2.31)	150.6	(3.7)	53.2	(2.3)
pima	32.43	(3.00)	31.39	(2.95)	326.8	(10.7)	34.8	(1.6)
musk	10.72	(2.30)	9.46	(2.24)	271.0	(2.6)	105.6	(4.2)
retinopathy	27.98	(1.16)	**25.54**	(2.10)	471.0	(6.6)	29.0	(2.6)
climate	7.04	(2.50)	6.11	(3.44)	158.8	(13.1)	41.4	(9.3)
vertebral	19.03	(4.17)	16.13	(4.70)	85.0	(2.8)	11.6	(1.5)
white	31.42	(4.21)	29.83	(5.19)	1284.2	(106.9)	123.2	(8.0)
ml-prove	25.37	(5.67)	**19.91**	(2.12)	1001.8	(70.3)	1212.0	(65.1)

Table 2: Experimental results for VKRG. As in Table 1, boldfaced values represent statistically significant results at 5% level.

those of L_2-SVM-uniform. In other words, as in the case of polynomial kernels, VKRG has a benefit of sparse solutions and often an improved performance, which again provides empirical evidence in the support of our formulation.

Appendix E. Dataset Statistics

The dataset statistics are provided in Table 3.

Table 3: Dataset statistics.

Data set	Examples	Features
breastcancer	699	9
climate	540	18
diabetes	768	8
german	1000	24
ionosphere	351	34
ml-prove	6118	51
musk	476	166
ocr49	2000	196
phishing	2456	30
retinopathy	1151	19
vertebral	310	6
waveform01	3304	21
white	4894	11